UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Özgür L. Özçep

Ontology Change II

Lecture 11: Revision for Ontology Change
25 June 2020

Informationssysteme CS4130
(Summer 2020)

Recap of Lecture 9

- Considered postulates and concrete operators for change operators on belief-sets
  - Belief-Sets = logically closed sets over given language
  - change operators: expansion (just adding and closing), contraction (eliminating), revision (adding and consistency)
  - Different ways to construct operators: we considered partial-meet based operators

- Criticisms: discussed recovery, minimality, success

- Need for extensions and adaptations from ontology change perspective
  - Finiteness: (Finite) Belief bases instead of belief sets
  - Syntax sensitive revision
  - Continue today with semantic belief revision for belief bases

End of Recap

Semantical Belief-Base Revision

- Semantical belief-revision demands syntax insensitivity in both arguments: trigger and also the belief base

- In this scenario: belief bases = knowledge bases

Schema for semantical belief revision

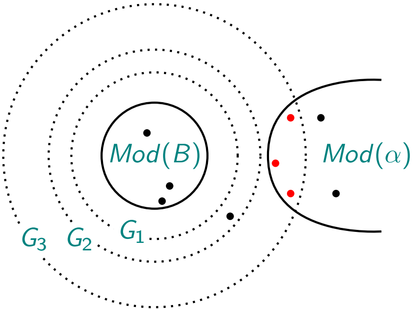
$$B * \alpha = \mathit{FinRep}(\mathit{Mod}(B) *_{sem} \mathit{Mod}(\alpha))$$

- $\mathit{Mod}(X)$ = Models of X
- $*_{sem}$ a semantical revision operator operating on pairs of sets of models
- $\mathit{FinRep}(M)$ = Formula or finite set of formulae that hold in all models in M

Approach 1 to Semantical Revision: Generalization

- Generalize (weaken) your belief base B' minimally s.t. enlarged set of models G_i intersects with Models of trigger

- Dalal's approach
  - Defined for propositional logic models • •
  - G_i = models with Hamming distance $\leq i$ to models in $Mod(B)$

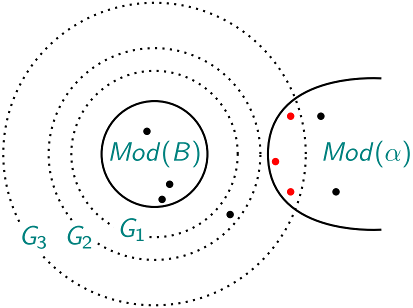


Lit: M. Dalal. Investigations into a theory of knowledge base revision: preliminary report. In AAAI-88, pages 475–479, 1988.

Approach 1 to Semantical Revision: Generalization

- Generalize (weaken) your belief base B' minimally s.t. enlarged set of models G_i intersects with Models of trigger

- Groves's approach: spheres
  - Defined on possible worlds • •
  - Possible world = maximally consistent set w.r.t. logic $(\mathcal{L}, Cn)$
  - G_i = sphere = set of possible worlds



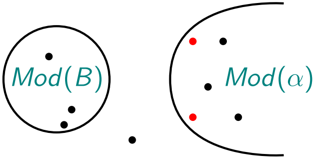
- Note: Maximal consistent sets correspond to models
- "Semantics" also possible in logics defined by $(\mathcal{L}, Cn)$

Lit: A. Grove. Two modellings for theory change. Journal of Philosophical Logic, 17:157–170, 1988.

Approach 2 to Semantical Revision: Minimal distance

- Dual but more general approach to generalization: minimality
- Find trigger models • with "minimal distance" to $Mod(B)$
  $B * \alpha = FinRep\big(Min_{\leq_{Mod(B)}}(Mod(\alpha))\big)$

- Various ways to specify minimal distance
  - incorporating order, cardinality, etc.



Lit: K. Satoh. Nonmonotonic reasoning by minimal belief revision. In FGCS-88, 455–462, 1988.

Lit: A. Borgida. Language features for flexible handling of exceptions in information systems. ACM Trans. Database Syst., 10(4):565–603, 1985.

Lit: A. Weber. Updating propositional formulas. In Expert Database Conf., pp. 487–500, 1986.

Lit: M. Winslett. Updating Logical Databases. Cambridge University Press, 1990.

Lit: K. D. Forbus. Introducing actions into qualitative simulation. In IJCAI-89, 1273–1278, 1988.

Complexity of Revision

- A main requirement in implementing BR operators: Feasibility of testing: $B * \alpha \models \beta$.
- Even for finite proposition B not really feasible
- Reason: Consistency testing is hard and you have potentially all subsets as culprit candidates
- Roughly the complexities are between NP and the second level of the polynomial hierarchy (so in $PSPACE$)

Lit: T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artif. Intell., 57:227–270, October 1992.

Complexity of Revision

- A main requirement in implementing BR operators: Feasibility of testing: $B * \alpha \models \beta$.
- Even for finite proposition B not really feasible
- Reason: Consistency testing is hard and you have potentially all subsets as culprit candidates
- Roughly the complexities are between NP and the second level of the polynomial hierarchy (so in $PSPACE$)

  Lit: T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artif. Intell., 57:227–270, October 1992.

- How to react to this?
  - Restrict logic to be used
  - Restrict the set of culprits: E.g., allow only culprits in ABox
  - Restrict other relevant parameters: treewidth, common variables

    Lit: A. Pfandler et al. On the parameterized complexity of belief revision. In IJCAI-15, pages 3149–3155, 2015.

Further Requirements

- Trigger is by itself a belief base: Multiple Belief Revision

Further Requirements

- Trigger is by itself a belief base: Multiple Belief Revision
- There is not a single trigger, but a whole sequence: Iterated revision

Further Requirements

- Trigger is by itself a belief base: Multiple Belief Revision
- There is not a single trigger, but a whole sequence: Iterated revision
- Learning ontologies: need non-amnestic (dynamic) iterated belief revision (connections to inductive learning)

Further Requirements

- Trigger is by itself a belief base: Multiple Belief Revision
- There is not a single trigger, but a whole sequence: Iterated revision
- Learning ontologies: need non-amnestic (dynamic) iterated belief revision (connections to inductive learning)
- Need different logics (not fulfilling, e.g., Deduction property): Revision for ontologies in DLs

Further Requirements

- Trigger is by itself a belief base: Multiple Belief Revision
- There is not a single trigger, but a whole sequence: Iterated revision
- Learning ontologies: need non-amnestic (dynamic) iterated belief revision (connections to inductive learning)
- Need different logics (not fulfilling, e.g., Deduction property): Revision for ontologies in DLs
- Need to revise other structures such as mappings

Ontology Change

Classification of Ontology Change

- Group 1 ("Overcome Heterogeneity")
  - Approaches where the main purpose is to resolve heterogeneity of ontologies by bridging between them
  - Ontologies are not changed (directly)
  - But mappings may change
  - Examples: ontology mapping, o. alignment, o. morphisms etc.

Lit: G. Flouris et al. Ontology change: classification and survey. The Knowledge Engineering Review, 23(2):117–152, 2008.

Classification of Ontology Change

- Group 1 ("Overcome Heterogeneity")
  - Approaches where the main purpose is to resolve heterogeneity of ontologies by bridging between them
  - Ontologies are not changed (directly)
  - But mappings may change
  - Examples: ontology mapping, o. alignment, o. morphisms etc.

Lit: G. Flouris et al. Ontology change: classification and survey. The Knowledge Engineering Review, 23(2):117–152, 2008.

Classification of Ontology Change

- Group 1 (“Overcome Heterogeneity”)
  - Approaches where the main purpose is to resolve heterogeneity of ontologies by bridging between them
  - Ontologies are not changed (directly)
  - But mappings may change
  - Examples: ontology mapping, o. alignment, o. morphisms etc.

- Group 2 (“Combine ontologies”)
  - Build new ontology based on input ontologies
  - Examples: ontology merge (input ontologies have same domain), ontology integration (input ontologies have similar domains)

Lit: G. Flouris et al. Ontology change: classification and survey. The Knowledge Engineering Review, 23(2):117–152, 2008.

Classification of Ontology Change

- Group 1 (“Overcome Heterogeneity”)
  - Approaches where the main purpose is to resolve heterogeneity of ontologies by bridging between them
  - Ontologies are not changed (directly)
  - But mappings may change
  - Examples: ontology mapping, o. alignment, o. morphisms etc.

- Group 2 (“Combine ontologies”)
  - Build new ontology based on input ontologies
  - Examples: ontology merge (input ontologies have same domain), ontology integration (input ontologies have similar domains)

- Group 3 (“Modify ontologies”)
  - Change ontologies (not necessarily caused by other ontologies)
  - Examples: ontology debugging, ontology repair, ontology evolution

Lit: G. Flouris et al. Ontology change: classification and survey. The Knowledge Engineering Review, 23(2):117–152, 2008.

Requirements due to Ontology Merge (and others)

Ontology Merge (Flouris et al. 08)

Purpose:	Fuse knowledge from ontologies over same domain
Input:	Two ontologies (from identical domains)
Output:	An ontology
Properties:	Fuse knowledge to describe domain more accurately

Requirements for OC operators

- Trigger by itself is a belief base: multiple revision

Requirements due to Ontology Merge (and others)

Ontology Merge (Flouris et al. 08)

Purpose:	Fuse knowledge from ontologies over same domain
Input:	Two ontologies (from identical domains)
Output:	An ontology
Properties:	Fuse knowledge to describe domain more accurately

Requirements for OC operators

- Trigger by itself is a belief base: multiple revision
- Belief base formulated in non-FOL (such as DLs)
  - Notion of AGM compliant contraction/revision
    Lit: G. Flouris, D. Plexousakis, and G. Antoniou. Generalizing the AGM postulates: preliminary results and applications. NMR-04, pp. 171–179, 2004.
  - Different postulates (to capture e.g. minimality):
    Lit: M. M. Ribeiro and R. Wassermann. Minimal change in AGM revision for non-classical logics. In KR-14, 2014.

AGM-Compliance

- Remember the additional properties on Cn required by AGM
  - Language expressivity: Language $\mathcal{L}$ should contain all propositional connectors
  - Supra-classicality: If α can be derived from X by propositional logic, then $\alpha \in Cn(X)$
  - Compactness: If $\alpha \in Cn(X)$, then $\alpha \in Cn(X')$ for some finite $X' \subseteq X$.
  - Deduction: $\beta \in Cn(X \cup \{\alpha\})$ iff $(\alpha \rightarrow \beta) \in Cn(X)$
  - Disjunction in premisses: If $\gamma \in Cn(X \cup \{\alpha\}) \cap Cn(X \cup \{\beta\})$, then $\gamma \in Cn(X \cup \{\alpha \vee \beta\})$.

AGM-Compliance

- Remember the additional properties on Cn required by AGM
  - Language expressivity: Language $\mathcal{L}$ should contain all propositional connectors
  - Supra-classicality: If α can be derived from X by propositional logic, then $\alpha \in Cn(X)$
  - Compactness: If $\alpha \in Cn(X)$, then $\alpha \in Cn(X')$ for some finite $X' \subseteq X$.
  - Deduction: $\beta \in Cn(X \cup \{\alpha\})$ iff $(\alpha \rightarrow \beta) \in Cn(X)$
  - Disjunction in premisses: If $\gamma \in Cn(X \cup \{\alpha\}) \cap Cn(X \cup \{\beta\})$, then $\gamma \in Cn(X \cup \{\alpha \vee \beta\})$.

- Are these really necessary in order to define a contraction operator fulfilling all six basic postulates?

(C1)	$K \div \alpha \in \mathcal{BS}_{\mathcal{L}}$	(Closure)
(C2)	$K \div \alpha \subseteq K$	(Inclusion)
(C3)	If $\alpha \notin K$, then $K = K \div \alpha$	(Vacuity)
(C4)	If $\alpha \notin Cn(\emptyset)$, then $\alpha \notin K \div \alpha$.	(Success)
(C5)	If $\alpha \in K$, then $K \subseteq (K \div \alpha) + \alpha$.	(Recovery)
(C6)	If $\alpha \leftrightarrow \beta \in Cn(\emptyset)$, then $K \div \alpha = K \div \beta$.	((Right) Extensionality)

AGM-Compliance

- Remember the additional properties on Cn required by AGM
  - Language expressivity: Language $\mathcal{L}$ should contain all propositional connectors
  - Supra-classicality: If α can be derived from X by propositional logic, then $\alpha \in Cn(X)$
  - Compactness: If $\alpha \in Cn(X)$, then $\alpha \in Cn(X')$ for some finite $X' \subseteq X$.
  - Deduction: $\beta \in Cn(X \cup \{\alpha\})$ iff $(\alpha \to \beta) \in Cn(X)$
  - Disjunction in premisses: If $\gamma \in Cn(X \cup \{\alpha\}) \cap Cn(X \cup \{\beta\})$, then $\gamma \in Cn(X \cup \{\alpha \vee \beta\})$.

- Are these really necessary in order to define a contraction operator fulfilling all six basic postulates?

  (C1) $K \div \alpha \in \mathcal{BS}_{\mathcal{L}}$ (Closure)
  (C2) $K \div \alpha \subseteq K$ (Inclusion)
  (C3) If $\alpha \notin K$, then $K = K \div \alpha$ (Vacuity)
  (C4) If $\alpha \notin Cn(\emptyset)$, then $\alpha \notin K \div \alpha$. (Success)
  (C5) If $\alpha \in K$, then $K \subseteq (K \div \alpha) + \alpha$. (Recovery)
  (C6) If $\alpha \leftrightarrow \beta \in Cn(\emptyset)$, then $K \div \alpha = K \div \beta$. ((Right) Extensionality)

- AGM Compliance

AGM compliance and Main Theorem

Definition

$(\mathcal{L}, Cn)$ is called AGM-compliant iff there is a contraction operator $\div$ fulfilling all six basic AGM contraction postulates (C1)–(C6).

AGM compliance and Main Theorem

Definition

$(\mathcal{L}, Cn)$ is called AGM-compliant iff there is a contraction operator $\div$ fulfilling all six basic AGM contraction postulates (C1)–(C6).

Definition

Let X, K be sets of formulae s.t.

- $K = Cn(K)$ and
- $Cn(\emptyset) \subsetneq Cn(X) \subsetneq K$

Define

$$K^{-}(X) = \{K' \mid Cn(K') \subsetneq Cn(K) \text{ and } Cn(K' \cup X) = Cn(K)\}$$

$(\mathcal{L}, Cn)$ is called decomposable iff for any X, K the set $K^{-}(X)$ is not empty.

Theorem (Flouris et al. 16)

A logic is AGM compliant iff it is decomposable

Theorem (Flouris et al. 16)

A logic is AGM compliant iff it is decomposable

- So we have a simple criterion (not many such as deduction, supraclassicality etc.) to test for AGM-compliance.

Theorem (Flouris et al. 16)

A logic is AGM compliant iff it is decomposable

- So we have a simple criterion (not many such as deduction, supraclassicality etc.) to test for AGM-compliance.
- Observation: Most DLs are not AGM compliant
- Hence: Cannot transfer AGM results directly to DLs
- This is hot research topic.
  - Contraction/revision for expressive DLs:
    Lit: M. M. Ribeiro and R. Wassermann. Base revision for ontology debugging. Journal of Logic and Computation. Advanced Access, published September 5, 2008, 2008.
  - Contraction/revision for lightweight DLs
    Lit: Z. Zhuang, Z. Wang, K. Wang, and G. Qi. Dl-lite contraction and revision. J. Artif. Intell. Res., 56:329–378, 2016.

Requirements due to Ontology Merge (and others)

Ontology Merge (Flouris et al. 08)

Purpose:	Fuse knowledge from ontologies over same domain
Input:	Two ontologies (from identical domains)
Output:	An ontology
Properties:	Fuse knowledge to describe domain more accurately

Requirements for OC operators

- Belief base formulated in non-FOL (such as DLs)
  - Need to consider generalizations of consistency such as coherence or even arbitrary integrity constraints

Requirements due to Ontology Mapping

Ontology Mapping (Flouris et al. 08)

Purpose:	Heterogeneity resolution, interoperability of ontologies
Input:	Two (heterogeneous) ontologies
Output:	A mapping between the ontologies' vocabularies
Properties:	The output identifies related vocabulary entities

Requirements for OC operators

- Mappings should not lead to inconsistencies
- Change of mappings in design time or due to change in ontologies
- **Lit:** C. Meilicke and H. Stuckenschmidt. Reasoning support for mapping revision. Journal of Logic and Computation, 2009.
- **Lit:** G. Qi, Q. Ji, and P. Haase. A conflict-based operator for mapping revision. In DL-09, volume 477 of CEUR Workshop Proceedings, 2009.

Mappings for Ontologies

- Data exchange provided mappings between schemata
- Here: Mappings between mappable "elements" of an ontology
- No unique representation format for ontology mappings

Definition (Mappings according to (Meilicke et al. 09))

$$(e_1 , e_2 , c , deg)$$

- $e_1 \in$ mappable elements of first ontology $\mathcal{O}_1$
  (e.g. concept symbols of $\mathcal{O}_1$)
- $e_2 \in$ mappable elements of second ontology $\mathcal{O}_2$
- c: type of mapping
  (e.g. c is equivalence or subsumption if e_i concepts)
- deg : degree of trust in the mapping

Example (Incompatible ontologies)

$\mathcal{O}_A$

- A1 $Article_A \equiv \exists publ_A.Journal_A$
- A2 $Journal_A \sqsubseteq \neg Proceedings_A$
- A3 $(func\ publ_A)$

$\mathcal{O}_B$

- B1 $Article_B \equiv \exists publ_B.Journal_B \sqcup Proceedings_B$
- B2 $publish_B(ab, procXY)$
- B3 $Proceedings_B(procXY)$

- Following set of mappings $\mathcal{M}_1$ is not consistent with $\mathcal{O}_A \cup \mathcal{O}_B$
  - $(Article_A, Article_B, \equiv, 1)$
  - $(Journal_A, Journal_B, \equiv, 1)$
  - $(Proceedings_A, Proceedings_B, \equiv, 1)$
  - $(publ_A, publ_B, \equiv, 1)$

Example (Incompatible ontologies)

$\mathcal{O}_A$

A1 $Article_A \equiv \exists publ_A.Journal_A$

A2 $Journal_A \sqsubseteq \neg Proceedings_A$

A3 $(func\ publ_A)$

$\mathcal{O}_B$

B1 $Article_B \equiv \exists publ_B.Journal_B \sqcup Proceedings_B$

B2 $publish_B(ab, procXY)$

B3 $Proceedings_B(procXY)$

- Following set of mappings $\mathcal{M}_2$ is consistent with $\mathcal{O}_A \cup \mathcal{O}_B$
  - $(Article_A, Article_B, \subseteq, 1)$
  - $(Journal_A, Journal_B, \equiv, 1)$
  - $(Proceedings_A, Proceedings_B, \equiv, 1)$
  - $(publ_A, publ_B, \equiv, 1)$

Example (Incompatible ontologies)

$\mathcal{O}_A$

- A1 $Article_A \equiv \exists publ_A.Journal_A$
- A2 $Journal_A \sqsubseteq \neg Proceedings_A$
- A3 $(func\ publ_A)$

$\mathcal{O}_B$

- B1 $Article_B \equiv \exists publ_B.Journal_B \sqcup Proceedings_B$
- B2 $publish_B(ab, procXY)$
- B3 $Proceedings_B(procXY)$

- Following set of mappings $\mathcal{M}_2$ is consistent with $\mathcal{O}_A \cup \mathcal{O}_B$
  - $(Article_A, Article_B, \subseteq, 1)$
  - $(Journal_A, Journal_B, \equiv, 1)$
  - $(Proceedings_A, Proceedings_B, \equiv, 1)$
  - $(publ_A, publ_B, \equiv, 1)$

$\Longrightarrow$ Can use revision on mappings to get from $\mathcal{M}_1$ to $\mathcal{M}_2$.

Requirements due to Ontology Evolution

Ontology Evolution (Flouris et al. 08)

Purpose: Respond to a change in the domain or its conceptualization

Input: An ontology and a (set of) change operation(s)

Output: An ontology

Properties: Implements a (set of) change(s) to the source ontology

Requirements for OC operators

- Change in domain due to change in environment: update vs. revision
- Evolution calls for iterative revision

Requirements due to Ontology Learning

Ontology Learning (my addition)

Purpose:	Respond to new bits of information from sender
Input:	A start ontology and a potentially infinite sequence of information
Output:	An ontology (sequence)
Properties:	Learns an ontology from a sequence

- Related to evolution: but emphasis on change of informedness and potential infinity
- Requirements for OC operators
  - Informed iterated revision on potentially infinite sequences
  - Notion of learning success (e.g. stabilization, reliability)

    Lit: D. Zhang and N. Y. Foo. Convergency of learning process. In AI-02, vol 2667 of LNCS, pp. 547?556, 2002.

    Lit: K. T. Kelly. Iterated belief revision, reliability, and inductive amnesia. Erkenntnis, 50:11–58, 1998.

Update vs. Revision

- Early CS work related to BR in Database Theory

  Lit: A. M. Keller and M. Winslett. On the use of an extended relational model to handle changing incomplete information. IEEE Transactions on Software Engineering, 11(7):620–633, 1985.
- Problem: Preserve integrity constraints when DB is updated
- Two main differences to BR
  - In DB: Not a theory to update but a structure
  - Update operators $\diamond$ fulfill different postulates
- Reason is: different conflict diagnostics
  - Revision: Conflict caused by false information
  - Update: Conflict caused by outdated information
  - In ontology change even a third diagnostics is possible: different terminology

Lit: H. Katsuno and A. Mendelzon. On the difference between updating a knowledge base and revising it. In KR-91, pages 387–394,1991.

Example (Winslett 1988)

- Input belief set: There is either a book on the table or a magazine $Cn(\alpha \leftrightarrow \neg\beta)$
- Trigger information: A book is put on the table α

Lit: M. Winslett. Reasoning about action using a possible models approach. In Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88), pp. 89–93, 1988.

Example (Winslett 1988)

- Input belief set: There is either a book on the table or a magazine $Cn(\alpha \leftrightarrow \neg\beta)$
- Trigger information: A book is put on the table α
- Apply revision operator fulfilling Postulates (R3) and (R4)

  (R3): $K * \alpha \subseteq K + \alpha$

  (R4): If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$. (Vacuity)

Lit: M. Winslett. Reasoning about action using a possible models approach. In Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88), pp. 89–93, 1988.

Example (Winslett 1988)

- Input belief set: There is either a book on the table or a magazine
  $$Cn(\alpha \leftrightarrow \neg\beta)$$
- Trigger information: A book is put on the table α
- Apply revision operator fulfilling Postulates (R3) and (R4)

  (R3): $K * \alpha \subseteq K + \alpha$

  (R4): If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$. (Vacuity)

- Output belief set: There is a book on the table and no magazine.
  $$Cn(\{\alpha \leftrightarrow \neg\beta\} \cup \{\alpha\}) = Cn(\alpha \wedge \neg\beta)$$

Lit: M. Winslett. Reasoning about action using a possible models approach. In Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88), pp. 89–93, 1988.

Example (Winslett 1988)

- Input belief set: There is either a book on the table or a magazine

$$Cn(\alpha \leftrightarrow \neg\beta)$$

- Trigger information: A book is put on the table α
- Apply revision operator fulfilling Postulates (R3) and (R4)

  (R3): $K * \alpha \subseteq K + \alpha$

  (R4): If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$. (Vacuity)

- Output belief set: There is a book on the table and no magazine.

$$Cn(\{\alpha \leftrightarrow \neg\beta\} \cup \{\alpha\}) = Cn(\alpha \wedge \neg\beta)$$

- Alternative postulate instead of vacuity

  If $\alpha \in K$, then $K \diamond \alpha = K$

Lit: M. Winslett. Reasoning about action using a possible models approach. In Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88), pp. 89–93, 1988.

Iterated Belief Revision

Iterating

- Aim: Apply change operators on sequence of triggers $\alpha_1, \alpha_2, \ldots$
- Static approach: same operator in every step on revision result

$$(\ldots((B * \alpha_1) * \alpha_2) * \ldots,) * \alpha_n)$$

- Dynamic Approach
  - operator my change depending on history

$$(\ldots((B *_1 \alpha_1) *_2 \alpha_2) *_3 \ldots,) *_n \alpha_n)$$

  - Belief base may encode history

Iterated AGM Revision

- AGM BR not tailored towards iteration:
  Considers only postulates for arbitrary but fixed belief set
- Only one interesting result for iterated AGM revision:

Proposition

If $*$ fulfills all AGM revision postulates (R1)–(R8), then it fulfills

$$\textit{If } \neg\beta \notin K * \alpha, \ \textit{then } (K * \alpha) * \beta = K * (\alpha \wedge \beta)$$

- In words: If second trigger compatible with revision result with first trigger, then revising with both triggers is the same as revising with conjunction

Need for Iteration Postulates

- Systematic study of iterated revision started in 1994

  Lit: A. Darwiche and J. Pearl. On the logic of iterated belief revision. In TARK-94, 5–23, 1994.

Example (Darwiche, Pearl 94)

- Agent hears an animal X barking like a dog
- So he thinks X is not a bird and cannot fly.

$$K \equiv \neg Bird \wedge \neg Flies$$

Need for Iteration Postulates

- Systematic study of iterated revision started in 1994

  Lit: A. Darwiche and J. Pearl. On the logic of iterated belief revision. In TARK-94, 5–23, 1994.

Example (Darwiche, Pearl 94)

- Agent hears an animal X barking like a dog
- So he thinks X is not a bird and cannot fly.

$$K \equiv \neg Bird \wedge \neg Flies$$

- But if he were told that X is a bird, he would assume that it flies.

$$K * Bird \equiv Bird \wedge Flies$$

Need for Iteration Postulates

- Systematic study of iterated revision started in 1994

  Lit: A. Darwiche and J. Pearl. On the logic of iterated belief revision. In TARK-94, 5–23, 1994.

Example (Darwiche, Pearl 94)

- Agent hears an animal X barking like a dog
- So he thinks X is not a bird and cannot fly.

$$K \equiv \neg Bird \land \neg Flies$$

- But if he were told that X is a bird, he would assume that it flies.

$$K * Bird \equiv Bird \land Flies$$

- If agent were to come to know that X can fly, then he should still believe: If X were a bird, then X would fly. Formally: $(K * Flies) * Bird \models Bird \land Flies$.

Need for Iteration Postulates

- Systematic study of iterated revision started in 1994

  Lit: A. Darwiche and J. Pearl. On the logic of iterated belief revision. In TARK-94, 5–23, 1994.

Example (Darwiche, Pearl 94)

- Agent hears an animal X barking like a dog
- So he thinks X is not a bird and cannot fly.

$$K \equiv \neg Bird \wedge \neg Flies$$

- But if he were told that X is a bird, he would assume that it flies.

$$K * Bird \equiv Bird \wedge Flies$$

- If agent were to come to know that X can fly, then he should still believe: If X were a bird, then X would fly. Formally: $(K * Flies) * Bird \models Bird \wedge Flies$.
- But one can construct AGM-conform revision $*$ (say amnesic revision) s.t.:

$$(K * Flies) * Bird \equiv Bird$$

Iteration Postulates (First Try)

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"If second trigger stronger than first, then second trigger overrides effects of first".

Iteration Postulates (First Try)

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"If second trigger stronger than first, then second trigger overrides effects of first".

DP2 If $\neg\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"For incompatible triggers the second one overrides the first one's effects"

Iteration Postulates (First Try)

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"If second trigger stronger than first, then second trigger overrides effects of first".

DP2 If $\neg\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"For incompatible triggers the second one overrides the first one's effects"

DP3 If $\alpha \in K * \beta$, then $\alpha \in (K * \alpha) * \beta$.
"If revision only by second trigger entails first trigger, then the sequential revision with both triggers does too."

Iteration Postulates (First Try)

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"If second trigger stronger than first, then second trigger overrides effects of first".

DP2 If $\neg\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.
"For incompatible triggers the second one overrides the first one's effects"

DP3 If $\alpha \in K * \beta$, then $\alpha \in (K * \alpha) * \beta$.
"If revision only by second trigger entails first trigger, then the sequential revision with both triggers does too."

DP4 If $\neg\alpha \notin K * \beta$, then $\neg\alpha \notin (K * \alpha) * \beta$.
"If revision only by second trigger is compatible with first trigger, then sequential revision with both triggers is too."

Wake-Up-Question

Which one of the DP Postulates rules out the bird example?

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.

DP2 If $\neg\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.

DP3 If $\alpha \in K * \beta$, then $\alpha \in (K * \alpha) * \beta$.

DP4 If $\neg\alpha \notin K * \beta$, then $\neg\alpha \notin (K * \alpha) * \beta$.

Example (Darwiche, Pearl 94)

- $K \equiv \neg Bird \wedge \neg Flies$
- $K * Bird \equiv Bird \wedge Flies$
- $(K * Flies) * Bird \equiv Bird$

Wake-Up-Question

Which one of the DP Postulates rules out the bird example?

DP1 If $\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.

DP2 If $\neg\alpha \in Cn(\beta)$, then $(K * \alpha) * \beta = K * \beta$.

DP3 If $\alpha \in K * \beta$, then $\alpha \in (K * \alpha) * \beta$.

DP4 If $\neg\alpha \notin K * \beta$, then $\neg\alpha \notin (K * \alpha) * \beta$.

Example (Darwiche, Pearl 94)

- $K \equiv \neg Bird \wedge \neg Flies$
- $K * Bird \equiv Bird \wedge Flies$
- $(K * Flies) * Bird \equiv Bird \wedge Flies$

Need More Information

- (DP2) cannot be fulfilled by any AGM revision operator for belief sets [Freund/Lehmann, 02]
- Reason is mainly: AGM allows for inconsistent belief sets

Lit: M. Freund and D. J. Lehmann. Belief revision and rational inference. Computing Research Repository (CoRR), cs.AI/0204032, 2002.

Need More Information

- (DP2) cannot be fulfilled by any AGM revision operator for belief sets [Freund/Lehmann, 02]
- Reason is mainly: AGM allows for inconsistent belief sets

- Reaction by [Darwiche/Pearl 97]: consider postulates with epistemic states Ψ instead of belief sets
- Allows dynamic (state-based) iteration: history encoded in state Ψ and not captured by logic
  - Every state Ψ induces belief set $BS(\Psi)$
  - But revision depends on state Ψ not induced belief set $BS(\Psi)$
  - In particular: $\Psi_1 * \alpha \neq \Psi_2 * \alpha$ possible even if $BS(\Psi_1) = BS(\Psi_2)$.

Lit: M. Freund and D. J. Lehmann. Belief revision and rational inference. Computing Research Repository (CoRR), cs.AI/0204032, 2002.

Lit: A. Darwiche and J. Pearl. On the logic of iterated belief revision. Artificial intelligence, 89:1–29, 1997.

Epistemic States

- Epistemic states are described in the postulates as abstract entities
  - Situation is the same as, say, in modal (temporal) logic or finite automata etc.
- But in order to construct concrete operators one has to construct epistemic states.

- There is a very popular approach based on ranking functions developed by W. Spohn in a series of papers and in a book.
- Ranking function κ: Assigns ordinal numbers to possible worlds (e.g., truth assignments in propositional logic)
- Does not give ranking only but also specifies plausibility distances.

Lit: W. Spohn. The Laws of Belief: Ranking Theory and Its Philosophical Applications. Oxford University Press, 2012.

Dynamic Operators

- Other approaches stick to belief sets (or belief bases) but allow dynamic revision operators.
- **Lit:** D. J. Lehmann. Belief revision, revised. In IJCAI-95, 1534–1540, 1995.
- **Lit:** A. C. Nayak, M. Pagnucco, and A. Sattar. Changing conditional beliefs unconditionally. In TARK-96, 119–135, 1996.

Infinite Iteration

Formal Learning Theory for Infinite Revision

- Iterable revision operators applied to potentially infinite sequence of triggers
- Define principles (postulates) that describe adequate behaviour

Formal Learning Theory for Infinite Revision

- Iterable revision operators applied to potentially infinite sequence of triggers
- Define principles (postulates) that describe adequate behaviour
- Minimality ideas and other principles of BR are not sufficient
- Hence, instead: Let you guide by principles of inductive learning and formal learning theory
  - Compare PAC (Probably Approximately Correct) framework
  - Compare FOIL (First-Order Inductive Learning) framework

The Scientist-Nature-Scenario

- Class of possible worlds (one of them the real world = nature)
- Scientist has to answer queries regarding the real world
- He gets stream of data compatible with the real world
- Conjectures according to some strategy at every new arrival of trigger a hypothesis on the correct answer
- Success: Sequence of answers stabilizes to a correct hypothesis.
  - Various stabilization criteria

Lit: E. Martin and D. Osherson: Elements of Scientific Inquiry. The MIT Press, 1998

Lit: K. T. Kelly. The Logic of Reliable Inquiry. Oxford University Press, 1995.

Example: The Scientist-Nature-Scenario for Orders

- Class of possible worlds

Example (Components of Order Example)

Strict$(\mathbb{N})$ = Strict total orders $<$ of $\mathbb{N}$

- 0,1,2,3, ... (isomorphic to $\omega = \{0, 1, 2, 3, \dots\}$ with natural ordering)
- 1,0,2,3, ... (isomorphic to ω)
- ...3,2,1,0 (isomorphic to $\omega^* = \{\dots 3, 2, 1, 0\}$ with inverse natural ordering)
- 0,2,4,6, ..., 1,3,5,7, ... (isomorphic to $\omega\omega$)

Example: The Scientist-Nature-Scenario for Orders

- He gets stream of data compatible with the real world

Example (Components of Order Example)

Stream of data made up by facts (called environments e)

- R(2,3), R(1,2), R(0,2), R(1,4) . . .
  (for world: 0,1,2,3, . . .)
- R(4,3), R(5,2), . . .
  (for world: . . . 3,2,1,0)

Example: The Scientist-Nature-Scenario for Orders

- Scientist answers query regarding the real world (problem)

Example (Components of Order Example)

Problem set: orders that are isomorphic ($\sim$) to ω or to ω^*

- 0,1,2,3, ... is isomorphic to ω
- ... 3,2,1,0 is isomorphic to ω^*.
- Problem query: Has order a least element (i.e., is it isomorphic to ω)?

Example: The Scientist-Nature-Scenario for Orders

- Success: Sequence of answers stabilizes to a correct hypothesis.

Example (Components of Order Example)

Scientist solves problem P iff for every order $< \in P$ and every environment e:

- If $<$ has least element, then cofinitely often scientist says yes on $e(n)$ (= n-prefix of environment e)
- If $<$ has no least element, then for cofinitely many n scientist says no on $e(n)$

Example: The Scientist-Nature-Scenario for Orders

- Conjectures—according to some strategy—at every new arrival of trigger a hypothesis on the correct answer
- Success: Sequence of answers stabilizes to a correct hypothesis.

Example (Components of Order Example)

$P = \{< \in \mathit{Strict}(\mathbb{N}) \mid <$ is isomorphic to ω or to $\omega^*\}$ solvable

- L-score: For any finite prefix of any environment smallest number not occurring in right argument of R
- G-score: smallest number not occurring in left argument of R
- Scientist: If L-score lower than G-score on given prefix, say yes, otherwise no.

Example (Proof of solvability)

- L-score: smallest number not occurring in right argument of R
- G-score: smallest number not occurring in left argument of R
- Scientist: If L-score lower than G-score on given prefix, say yes, otherwise no.

- Proof of solvability:
  - Intuitively: The L-score (G-score) is the best candidate for the least (greatest) element of $<$ (if there is one)
  - Suppose $< \sim \omega$. Then least element of $<$ appears somewhere as left but never as right element. Hence: L-scores of $e[n]$ is bounded. Every number appears as first argument. Hence: The G-scores of $e[n]$ are unbounded.
  - Suppose $< \sim \omega^*$. Situation reversed.
  - Moreover: scores are monotonic w.r.t. increasing prefix.
  - Hence: If $< \sim \omega$, then coinfinitely often L-score is smaller than G score
  - If $< \sim \omega^*$, then coinfinitely often G-score is smaller than L-score

Learning Aims of Scientist-Nature-Scenario

- Above scenario generalized to arbitrary FOL structures in (Martin/Osherson 1998)
- Also (Martin/Osherson 1998) consider revision operators for guessing the true world (see next slides)

- Similar principles as in PAC learning from machine learning
- But two main differences
  - Approach of (Martin/Osherson 1998) has not a pre-determined finite set of data items (as is the case for most scientific inquiry situations)
  - Exact prediction of the real world (not approximate prediction within some tolerance interval as in PAC)

Lit: E. Martin and D. Osherson: Elements of Scientific Inquiry. 1998, The MIT Press

Choosing Revision as Strategy

- Kelly investigates learning based on various revision operators defined for epistemic states
- Hypotheses = sentences in the belief sets
- Main (negative) result in (Kelly 98)

Theorem

Revision operators implementing a minimal (one-step) revision suffer from inductive amnesia: If and only if some of the past is forgotten, stabilization is guaranteed.

Lit: K. T. Kelly. Iterated belief revision, reliability, and inductive amnesia. Erkenntnis, 50:11–58, 1998.

Choosing Revision as Strategy

- Martin/Osherson investigate learning based revision operators defined for finite sequences
- So their revision operators have always the whole history of triggers (they do not have to store the history)
- This leads to positive results

Theorem

Revision operators provide ideal learning strategies: There is a revision operator a scientist can use to solve every (solvable) problem.

Lit: E. Martin and D. Osherson. Scientific discovery based on belief revision. Journal of Symbolic Logic, 62(4):1352–1370, 1997.

Stabilization for Ontology Learning

Example (Book Shopping Agent)

$$
\begin{array}{lcll}
O_{rec} & \models & cheap \equiv costs < 5\$, & \neg costs < 5\$('Faust') \\
O_{send} & \models & cheap \equiv costs < 6\$, & costs < 6\$('Faust')
\end{array}
$$

Lit: Eschenbach and Ö. Ontology revision based on reinterpretation. Logic Journal of the IGPL, 18(4):579–616, 2010.

Stabilization for Ontology Learning

Example (Book Shopping Agent)

$$O_{rec} \models cheap \equiv costs < 5\$, \quad \neg costs < 5\$(\text{'}Faust'\text{)}$$
$$O_{send} \models cheap \equiv costs < 6\$, \quad costs < 6\$(\text{'}Faust'\text{)}$$

- Receiver: "List all cheap books by Goethe"
- Sender stream: $\alpha_1 = cheap(\text{'}Faust')\lightning, \alpha_2, \alpha_3, \ldots$

Lit: Eschenbach and Ö. Ontology revision based on reinterpretation. Logic Journal of the IGPL, 18(4):579–616, 2010.

Stabilization for Ontology Learning

Example (Book Shopping Agent)

$$O_{rec} \models cheap \equiv costs < 5\$, \quad \neg costs < 5\$(\text{'}Faust\text{'})$$
$$O_{send} \models cheap \equiv costs < 6\$, \quad costs < 6\$(\text{'}Faust\text{'})$$

- Receiver: "List all cheap books by Goethe"
- Sender stream: $\alpha_1 = cheap(\text{'}Faust\text{'})$ ⚡ $, \alpha_2, \alpha_3, \ldots$
- Integrating stream elements by revision operator $\circ$ gives Output stream $(O^i_{rec})_{i \in \mathbb{N}}$:

$$(O_{rec},\ O_{rec} \circ \alpha_1,\ (O_{rec} \circ \alpha_1) \circ \alpha_2,\ \ldots)$$

Lit: Eschenbach and Ö. Ontology revision based on reinterpretation. Logic Journal of the IGPL, 18(4):579–616, 2010.

Stabilization for Ontology Learning

Example (Book Shopping Agent)

$$O_{rec} \models cheap \equiv costs < 5\$, \quad \neg costs < 5\$(\text{'}Faust') $$
$$O_{send} \models cheap \equiv costs < 6\$, \quad costs < 6\$(\text{'}Faust') $$

- Receiver: "List all cheap books by Goethe"
- Sender stream: $\alpha_1 = cheap(\text{'}Faust')$↯$, \alpha_2, \alpha_3, \ldots$
- Integrating stream elements by revision operator $\circ$ gives Output stream $(O^i_{rec})_{i \in \mathbb{N}}$:

$$(O_{rec},\ O_{rec} \circ \alpha_1,\ (O_{rec} \circ \alpha_1) \circ \alpha_2,\ \ldots)$$

- For which operators stabilization?

Lit: Eschenbach and Ö. Ontology revision based on reinterpretation. Logic Journal of the IGPL, 18(4):579–616, 2010.