UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Özgür L. Özçep

Ontology Change 2

Lecture 10: Revision for Ontology Change

24 January, 2017

Foundations of Ontologies and Databases for Information Systems

CS5130 (Winter 16/17)

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
  - Semantic belief revision

End of Recap

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.
- 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.

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.

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.

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
- Belief base formulated in non-FOL (such as DLs)
  - Notion of AGM compliant 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.

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 may be temporal change: **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
- 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, 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.

$$BSb \equiv \neg Bird \wedge \neg Flies$$

- 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 know beforehand that X can fly, then he should still believe: If X were a bird, then X would fly.
- But one can construct AGM-conform revision $*$ s.t.:

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

2-slide digression: Counterfactuals

- Counterfactual speak in example no coincidence
- $K \models \beta \mid \alpha$ iff $\beta \in K * \alpha$

  "K accepts β given α"

  "If α were the case, then β would hold."
- Can develop whole theory with this conditional $\mid$.
- Note: $\mid$ is a logical operator on the meta-level

2-slide digression

- If $|$ wanted in object language, an intuitive constraint is formulated in the Ramsey Test

Definition (Ramsey Test)

$$\alpha \mid \beta \in K \text{ iff } \beta \in K * \alpha$$

2-slide digression

- If $|$ wanted in object language, an intuitive constraint is formulated in the Ramsey Test

Definition (Ramsey Test)

$$\alpha \mid \beta \in K \text{ iff } \beta \in K * \alpha$$

Theorem

Only trivial AGM operators fulfill the Ramsey test.

- Non-trivial
  - There is K and three sentences not in K
- Theorem holds because Ramsey test entials monotonicity w.r.t. left argument.

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."

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."

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."

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)

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

Need More Information

- (DP2) cannot be fulfilled by any AGM revision operator for belief sets

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

- Reason is mainly: AGM allows for inconsistent belief sets

- Reaction by Darwiche and Pearl: consider postulates with epistemic states Ψ instead of belief sets

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

- 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 $BS(\Psi_1) = BS(\Psi_2)$.

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 specify 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
- Minimality ideas and other principles of BR are not sufficient
- Hence, instead: Let you guide by principles of inductive learning and formal learning theory
- Indeed, we need good principles for induction :)

  http://www.der-postillon.com/2015/10/autofahrer-entlarvt-geheimen.html http://www.der-postillon.com/2015/10/autofahrer-entlarvt-geheimen.html

Dienstag, 27. Oktober 2015

Autofahrer entlarvt geheimen Zahlentrick, mit dem sich jeder Blitzer überlisten lässt

Hannover (dpo) - Ein Autofahrer aus Hannover hat einen geheimen Trick entdeckt, mit dem sich Radarfallen und Blitzer auf Autobahnen zu 100 Prozent überlisten lassen. Erste Praxistests scheinen die Theorie des Hobbyphysikers zu bestätigen. Seine zahlenbasierte Methode soll für jeden Autofahrer innerhalb von Minuten erlernbar sein.

Seit 1990 ist Elektroingenieur Jörg Haffke jeden Tag auf deutschen Autobahnen unterwegs, was den Berufspendler oft teuer zu stehen kommt: "Früher verging kaum eine Woche, ohne dass ich ein Knöllchen im Briefkasten hatte, nur weil ich mal wieder in eine Radarfalle geraten war", klagt der 43-Jährige. Die verschiedensten Gegenmittel habe er schon ausprobiert: "Reflektierende CDs am Innenspiegel, abgedecktes Kennzeichen oder Blitzwarner-Apps – nichts hatte auf Dauer wirklich Erfolg", berichtet der Familienvater.

Haffke ist jedoch nicht nur ein cleverer Bastler, sondern auch ein guter Beobachter: "Irgendwann fiel mir auf, dass neben den Autobahnen immer wieder so merkwürdige Schilder mit Zahlen drauf versteckt sind."

Er hat den Code geknackt: Jörg Haffke

Zwei Jahre und 122 Excel-Tabellen später ist sich Haffke nun sicher, den geheimen Algorithmus hinter den kryptischen Blechtafeln endlich entlarvt zu haben: "Wenn man ein Tempo mit einem Tachowert fährt, der in etwa der Zahl auf dem letzten gesichteten Schild entspricht oder darunter liegt, verhindert man ein Auslösen des nächsten Blitzers! Man ist praktisch unsichtbar."

Freitag, 3. Januar 2014

Erfinder des Zwei-, Drei-, Vier-, Fünf- und Sechs-Klingen-Rasierers hat keine Ideen mehr

Stuttgart (dpo) - Der erfolgreiche Erfinder Erwin Merscheid ist "in ein kreatives Loch gefallen", wie er dem *Postillon* gegenüber sagte. Der 67-Jährige hatte in den letzten Jahrzehnten die Nassrasierer mit zwei, drei, vier, fünf und sechs Klingen jeweils entwickelt und an die Industrie verkauft. Nun jedoch habe der leidenschaftliche Tüftler keine Ideen mehr für weitere Erfindungen.

Merscheid, Sohn eines Bäckergesellen und einer Grundschullehrerin, litt bereits mit 13 Jahren unter starkem Bartwuchs. Mit den seinerzeit handelsüblichen Ein-Klingen-Rasierern konnte er sich jedoch nicht durch das Dickicht kämpfen. Von seinen Mitschülern gehänselt, machte er aus der Not eine Tugend, entwickelte den Zwei-Klingen-Rasierer und verdiente Millionen.

http://www.der-postillon.com/2014/01/erfinder-des-zwei-drei-vier-funf-und.html

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
- Scientist answers query regarding the real world (problem)
- 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.

Example (Component of Order Example)

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

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

Example: The Scientist-Nature-Scenario for Orders

- Class of possible worlds
- Scientist answers query regarding the real world (problem)
- 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.

Example (Component 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

- Class of possible worlds
- Scientist answers query regarding the real world (problem)
- 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.

Example (Component of Order Example)

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

- 0,1,2,3, $\ldots$ is isomorphic to ω
- $\ldots$ 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

- Class of possible worlds
- Scientist answers query regarding the real world (problem)
- 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.

Example (Component of Order Example)

Scientist solves problem P iff for every $< \in P$ 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

- Class of possible worlds
- Scientist answers query regarding the real world (problem)
- 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.

Example (Component of Order Example)

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

- L-score: For any finite sequence 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 $< \sim \omega$ coinfinitely often L-score is smaller than G score
  - If $< \sim \omega^*$ coinfinitely often G-score is smaller than L-score

The 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.

Stabilization for Ontology Learning

Example (Book Shopping Agent)

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

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

- Integrating stream elements by revision operator $\circ$ gives Output stream $(O^i_{rec})_{i \in \mathbb{N}}$:

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

Stabilization for (Amnesic) Ontology Learning

- Properties of $(O^i_{rec})_{i \in \mathbb{N}}$ depend on $\circ$
- Special case: $\circ =$ weak type-2 operator (forgets quite a lof of from "old ontology")
  - Prioritize incoming terminology
  - Simple mappings for disambiguation
    Example: $cheap_{rec} \sqsubseteq cheap_{send}$, $cheap \equiv cheap_{send}$

Theorem (Eschenbach & Ö., 2011)

For a (internally consistent) stream of atomic assertions the output streams of ontologies produced with weak type-2 operator stabilizes.

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

Stabilization for (Amnesic) Ontology Learning

- Properties of $(O^i_{rec})_{i \in \mathbb{N}}$ depend on $\circ$
- Special case: $\circ$ = weak type-2 operator (forgets quite a lof of from "old ontology")
  - Prioritize incoming terminology
  - Simple mappings for disambiguation
    Example: $cheap_{rec} \sqsubseteq cheap_{send}$, $cheap \equiv cheap_{send}$

Theorem (Eschenbach & Ö., 2011)

For a (internally consistent) stream of atomic assertions the output streams of ontologies produced with weak type-2 operator stabilizes.

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

Non-Stabilization for (Non-Amnesic) Ontology Learning

- Special case: $\circ$ = strong type-2 operator (remembers "old ontology")
  - Prioritize incoming terminology
  - Advanced mappings for disambiguation
    Example: $cheap_{rec} \sqsubseteq cheap_{send}$, $cheap_{send} \sqsubseteq cheap_{rec} \sqcup DifferConcept_{rec,send}$, $cheap \equiv cheap_{send}$

Theorem (Eschenbach & Ö., 2011)

*There is an ontology and a (internally consistent) stream of atomic assertions s.t. the output stream of ontologies produced with the strong type-2 operator does **not** stabilize.*

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

Non-Stabilization for (Non-Amnesic) Ontology Learning

- Special case: $\circ$ = strong type-2 operator (remembers "old ontology")
  - Prioritize incoming terminology
  - Advanced mappings for disambiguation
    Example: $cheap_{rec} \sqsubseteq cheap_{send}$,
    $cheap_{send} \sqsubseteq cheap_{rec} \sqcup DifferConcept_{rec,send}$, $cheap \equiv cheap_{send}$

Theorem (Eschenbach & Ö., 2011)

*There is an ontology and a (internally consistent) stream of atomic assertions s.t. the output stream of ontologies produced with the strong type-2 operator does **not** stabilize.*

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

Choosing Revision as Strategy

- Martin/Osherson investigate learning based revision operators defined for finite sequences
- So their revision operators have always the whole history within the trigger
- 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.

Solutions to Exercise 9 (10+10B Points)

Solution for Exercise 9.1 (2 Points)

Show that postulates (R1)–(R5) (and (E1)–(E5)) entail the following fact for consistent K:
If $\alpha \in K$, then $K * \alpha = K$.

Solution: Consistency of K means that $\neg\alpha \notin K$. Hence

- $K = Cn(K \cup \{\alpha\}) = K + \alpha \overset{(R4)}{\subseteq} K * \alpha$
- $K * \alpha \overset{(R3)}{\subseteq} K + \alpha = K$

Note that for inconsistent K the equality holds only for contradictory α.

Solution for Exercise 9.2 (2 Bonus Points)

Show that $*$ is not commutative, i.e., there are K, α, β such that:

$$(K * \alpha) * \beta \neq (K * \beta) * \alpha$$

Solution:

- Take $\beta = \neg\alpha$ (for contigent α, β) and assume K is consistent
- We have due to success: $\neg\alpha = \beta \in (K * \alpha) * \beta$
- Similarly $\alpha \in (K * \beta) * \alpha$
- But as $(K * \beta) * \alpha$ is consistent it mus be $\beta = \neg\alpha \notin (K * \beta) * \alpha$. Hence
- $(K * \alpha) * \beta \neq (K * \beta) * \alpha$

Solution for Exercise 9.3 (2 Bonus Points)

Show that Postulates (R1)–(R8) entail the following fact:
$K * \alpha = K * \beta$ iff $\alpha \in K * \beta$ and $\beta \in K * \alpha$

Solution: Direction $\leftarrow$:

$$\begin{aligned}
K * \alpha &= (K * \alpha) + \beta && \text{(Definition of } +\text{)} \\
&= K * (\alpha \wedge \beta) && \text{(due to (R7, R8))} \\
&= K * (\beta \wedge \alpha) && \text{(due to (R6))} \\
&= (K * \beta) + \alpha && \\
&= K * \beta &&
\end{aligned}$$

Direction $\rightarrow$: Follows by success postulate.

Solution for Exercise 9.4 (6 Bonus Points)

Show the following refined version of the theorem for the Levi-Identity:

If $*$ is defined by the Levi identity $K * \alpha = (K \div \neg\alpha) + \alpha$, then it fulfills Postulates (R*1)–(R*6) if $+$ fulfills Postulates (E1)–(E6) and $\div$ fulfills postulates (C1)–(C4) and (C6).

Solution:

- (R1) ($K * \alpha \in \mathcal{BS}_{\mathcal{L}}$): Clear due to (E1), (C1)
- (R2) ($\alpha \in K * \alpha$): Due to (E2)
- (R3) ($K * \alpha \subseteq K + \alpha$): $K * \alpha = (K \div \neg\alpha) + \alpha \overset{((C2),(E5)}{\subseteq} K + \alpha$
- (R4) (If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$)
  - Assume $\neg\alpha \notin K$.
  - Then $K = K \div \neg\alpha$ (due to (C3))
  - Now $K + \alpha = (K \div \neg\alpha) + \alpha = K * \alpha$.
- (R5) (If $\perp \in Cn(K * \alpha)$, then $\neg\alpha \in Cn(\emptyset)$.):
  If $\neg\alpha \notin Cn(\emptyset)$, then $\neg\alpha \notin K + \neg\alpha$ due to (C4). So $(K \div \neg\alpha) + \alpha$ is consistent.
- (R6) (If $\alpha \leftrightarrow \beta \in Cn(\emptyset)$, then $K * \alpha = K * \beta$): follows from (C6)

Solution for Exercise 9.5 (2 Points)

Show that the remainders for a belief set are by themselves belief sets.

Solution:

Let X be a belief set that is closed $Cn(X) = X$ and let Y be a remainder for some α. Assume Y is not closed, i.e., there is β s.t. $\beta \notin Y$, though $\beta \in Cn(Y)$. Consider $Y \cup \{\beta\}$. This is also a subset of X and does not entail α: Otherwise $Y \models \beta \wedge (\beta \rightarrow \alpha)$, i.e., $Y \models \alpha$, contradiction. But then $Y \cup \{\beta\}$ would be larger than Y—contradicting its maximality.

Solution Exercise 9.6 (2 Points)

Show that if $Y \in X \perp (\alpha \vee \beta)$, then $\alpha \notin Cn(Y)$.
Solution:
If α were in $Cn(Y)$, then $\alpha \vee \beta$ would be too, contradicting the fact that Y is a remainder w.r.t. $\alpha \vee \beta$.

Solution for Exercise 9.7 (4 Points)

Calculate the following remainder sets (**solutions** in red):

1. $\{p, q\} \perp p \wedge q = \{\{p\}, \{q\}\}$
2. $\{p, q, r\} \perp p \wedge q = \{\{p, r\}, \{q, r\}\}$
3. $\{q\} \perp p \wedge q = \{\{q\}\}$
4. $\emptyset \perp p \wedge q = \{\emptyset\}$