FernUniversität in Hagen
Faculty of Mathematics and Computer Science

AIG
Artificial Intelligence Group

Evaluating Revision Operators as Explanation for Human Reasoning over Two Variables

Master's Thesis

in partial fulfillment of the requirements for
the degree of Master of Science (M.Sc.)
in Praktische Informatik

submitted by
Nina Thorwart

First examiner: Dr. Kai Sauerwald
Artificial Intelligence Group

Advisor: Prof. Dr. Matthias Thimm
Artificial Intelligence Group

Statement

Ich erkläre, dass ich die Masterarbeit selbstständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht. Die Versicherung selbstständiger Arbeit gilt auch für enthaltene Zeichnungen, Skizzen oder graphische Darstellungen. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Arbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate geprüft werden kann und ausschließlich für Prüfungszwecke gespeichert wird.

	Yes	No
I agree to have this thesis published in the library.	☒	☐
I agree to have this thesis published on the webpage of the artificial intelligence group.	☒	☐
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	☒	☐
The source code is available under a GNU General Public License (GPLv3).	☐	☒
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	☐	☒

Würzburg, 03.06.2024 .... Nina Thorwart

(Place, Date) (Signature)

Zusammenfassung

In verschiedenen Experimenten, wie z. B. der Wason-Auswahlaufgabe [Was68], wurde nachgewiesen, dass das menschliche Schlussfolgern nicht dem klassischen logischen Schlussfolgern entspricht. In dieser Arbeit wird der entwickelte *sequenzielle Revisionsansatz* (*Sequential Revision Approach*) vorgestellt, der darauf abzielt, menschliche Schlussfolgerungsprozesse durch eine Kombination von verschiedenen Revisionsoperatoren und kognitiven Ansätzen, die von Menschen angewandt werden, zu simulieren. Der Ansatz wird anschließend anhand von Daten aus einem psychologischen Experiment zum menschlichen logischen Denken evaluiert und die Vorhersagekraft des Ansatzes mit der klassischen Aussagenlogik verglichen. Im Vergleich zur klassischen Aussagenlogik hat der entwickelte Ansatz eine höhere Genauigkeit als die klassische Logik und bildet daher das menschliche Denken besser ab als klassische logische Ansätze.

Abstract

In various experiments, such as the Wason Selection task [Was68], it has been proven that human reasoning does not conform to classical logical reasoning. In this thesis, the developed *Sequential Revision Approach* is presented, which aims to replicate human reasoning processes using a combination of different revision operators and cognitive approaches applied by humans. The approach is then evaluated using data from a psychological experiment on human propositional reasoning and the predictive performance of the approach is compared to classical propositional logic. In comparison to classical propositional reasoning, the developed approach has a higher accuracy and therefore replicates human reasoning better than classical logical approaches.

Contents

1 Introduction

Artificial Intelligence (AI) is advancing fast, with technologies such as generative AI (e.g. ChatGPT), recommendation systems (e.g. the Spotify algorithm), security systems (e.g. face recognition), and many more. In the years 2023 to 2030, it is predicted to have an annual growth rate of 37.3% [Haa23], which makes it one of the most important and promising technologies for the future.

Human-centered AI is a discipline within the field of AI that puts an emphasis on augmenting human abilities instead of suppressing them. Furthermore, it focuses on humans and AI systems cooperating [Gey+22]. To realise the successful cooperation between humans and AI systems, the AI needs to be able to predict the way humans might act in certain situations and therefore anticipate their actions [FPC23]. Furthermore, intelligent systems should be able to predict the errors humans tend to make, and subsequently issue warning so humans avoid making those mistakes.

It has been shown in various experiments, such as the Wason Selection Task [Was68], that human reasoning does not conform with classical logical reasoning, and therefore, classical propositional logic cannot be used as a descriptive language to replicate human reasoning. Therefore, the goal is to develop a so-called *cognitive logic*, which aims to bridge the gap between psychology, cognitive sciences and logic and to develop a formal concept how human reasoning works [Rag+20].

In this thesis, the developed so-called *Sequential Revision Approach* is presented. It is one attempt at replicating human reasoning, with a variety of different revision operators.

The thesis is structured as follows: in Chapter 2, the theoretical background necessary for the implemented *Sequential Revision Approach* is presented, that being the basics of logic, a short introduction to the concepts of plausibility orderings and epistemic states, afterwards the field of belief revision with different revision operators, and lastly, the cognitive background to human reasoning.

Afterwards, in Chapter 3, the realisation of the *Sequential Revision Approach* is explained, that being the approach itself and the implementation of the different mental approaches presented in the chapter before.

In Chapter 4, the implementation of the *Sequential Revision Approach* is outlined and the different revision operators and the incorporation of the different mental approaches are described in detail. Furthermore, the processing of the task given to the participants and the selection of an answer following the revision process is shown.

Chapter 5 contains the evaluation of the different operators implementing the *Sequential Revision Approach*. First, the experimental dataset used for the evaluation is presented. Afterwards, the different operators are compared and the most suitable operator is identified.

In Chapter 6, the results of the previous evaluation are discussed. Thereby, the results of selected operators are explained and debated. Furthermore, the *Sequential Revision Approach* is compared to the similar *Sequential Merging Approach* proposed by Ismail-Tsaous [Ism+23].

Lastly, Chapter 7 provides a summary of the thesis and offers an outlook into the future, how the implemented approach could be refined or extended.

The main contributions of this thesis are:

- An approach how humans process information and make conclusions based on them, the *Sequential Revision Approach*.
- Different revision operators used in the *Sequential Revision Approach*.
- Approaches for including human reasoning processes that include logical fallacies when deducing new information.
- An empirical evaluation of the different operators used in the implementation of the *Sequential Revision Approach*.
- Identification of the most suitable operator.
- Discussion of the predictive performance of the implemented *Sequential Revision Approach* and comparison to a similar approach.
- Perspective on how the implemented *Sequential Revision Approach* can be expanded.

2 Preliminaries

In this chapter, the main theoretical background for this thesis is presented. First, a brief background on logic is given. Second, plausibility orderings and epistemic states are introduced. Third, the concept of revision is defined and different revision operators are presented. Lastly, the cognitive background of human logical reasoning is outlined.

2.1 Background on Logic

In this thesis, only propositional logic is used. Propositional logic is a branch of classical logic which analyses the relationship between propositions and combinations thereof. Statements, also known as *propositions*, are abstract concepts, which are expressed in everyday language by sentences. In propositional logic, it is not the concrete content of the statements that is important, but only the decision as to whether a statement is true or false.

Propositions are often denoted by Latin letters and are represented in the propositional signature $\Sigma = \{a, b, c, p, q, ...\}$, which contains the non-logical symbols of the language.

The language $\mathcal{L}$ is a propositional language over Σ and is defined as follows:

Definition 1 (Propositional Language)**.** The language of propositional formulas $\mathcal{L}$ over Σ is given by:

- the propositional variables $a \in \Sigma$,
- a set of operators, called *logical connectives*,

and contains the set of well-formed formulae. Hereby, a *well-formed formula* is defined inductively:

- Each propositional variable $a \in \Sigma$ is a formula.
- If a is a formula, then $\neg a$ is a formula.
- If a and b are formulae, and $\circ$ is any binary connective, then $a \circ b$ is a formula.

The logical connectives used in propositional logic are presented in Table 1.

The set of propositional interpretations is denoted by Ω. Hereby, a propositional interpretation is defined as follows:

Definition 2 (Interpretation)**.** A propositional *Interpretation* I is a function $I : \Sigma \rightarrow \{0, 1\}$ which assigns exactly one truth value to each formula of $\mathcal{L}$.

We extend the domain of an interpretation I to formulae as follows:

- If $A = a$ for $a \in \Sigma$, then $I(A) = I(a)$.

Logical Connective	Meaning
$\neg a$, $\overline{a}$	Negation (NOT)
$a \wedge b$	Conjunction (AND)
$a \vee b$	Disjunction (OR)
$a \dot{\vee} b$	Exclusive Disjunction (XOR)
$a \Rightarrow b$	Implication (Conditional)
$a \Leftrightarrow b$	Equivalence (Biconditional)

Table 1: Logical Connectives in propositional logic

a	b	$\wedge$	$\vee$	$\dot{\vee}$	$\Rightarrow$	$\Leftrightarrow$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

Table 2: Truth table for $\wedge$, $\vee$, $\dot{\vee}$, $\Rightarrow$ and $\Leftrightarrow$

- If $A = \neg b$, then $I(A) = 1$ iff $I(b) = 0$.
- If $A = a \wedge b$, then $I(A) = 1$ iff $I(a) = 1$ and $I(b) = 1$.
- If $A = a \vee b$, then $I(A) = 1$ iff $I(a) = 1$ or $I(b) = 1$ or both.
- If $A = a \dot{\vee} b$, then $I(A) = 1$ iff either $I(a) = 1$ or $I(b) = 1$, but not both.
- If $A = a \Rightarrow b$, then $I(A) = 1$ iff $I(a) = 0$ or $I(b) = 1$.
- If $A = a \Leftrightarrow b$, then $I(A) = 1$ iff both $I(a) = 1$ and $I(b) = 1$ or both $I(a) = 0$ and $I(b) = 0$.

In propositional logic, formulae are mostly evaluated using so-called truth tables. A truth table shows the assignment of the propositions on the evaluation of the formulae. The truth tables for the logical connectives can be seen in Table 2.

Furthermore, $\models$ is the models relation, meaning $\omega \models a$ indicates that $\omega \in \Omega$ is a model of a, i.e., $I(a) = 1$. A model of a formula is an assignment of the proposition so that the evaluation of the formula yields *True*. $Mod(a)$ is the set of models of a.

2.2 Plausibility Orderings and Epistemic States

In this chapter, plausibility orders and epistemic states are introduced and explained and their relevance for belief revision is described.

In belief revision, the beliefs of an agent are often described by a belief set; a belief set is a deductively closed set of sentences of $\mathcal{L}$. However, when modelling human reasoning using revision, the state of mind of humans is represented using Epistemic States. Epistemic states are generalised belief sets, additionally containing information about the humans' reasoning process. Typically, an epistemic state is equipped with a *Total Preorder* $C_{\mathbb{E}}$, called *Plausibility Ordering*, which assigns as value to each valuation. Here we are using *Ordinal Conditional Functions (OCF)*, which are a generalisation of total preorders. Hereby, the lower the assigned number is, the more plausible the associated valuation is considered to be [BM06; DP97].

Spohn [Spo88] first characterised this kind of function under the name *Ordinal Conditional Functions* as a way to assign a degree of firmness to a possible world.

Definition 3 (Ordinal Conditional Function [Ker01; Spo88])**.** *Ordinal Conditional Functions* (OCFs, Ranking Functions) are functions $\kappa : \Omega \to \mathbb{N}_0$, that map worlds to ordinals such that some worlds are mapped to the minimal element 0.

If and only if an interpretation ω is believed with $0 = \kappa(\omega) < \kappa(\overline{\omega})$, an agent believes in an interpretation ω more than in its negation $\overline{\omega}$ (if $0 = \kappa(\omega) = \kappa(\overline{\omega})$, the agent is neutral about the interpretation ω). The function can therefore be seen as a way to express disbelief in a possible world. In addition, if $\kappa(\omega) < \kappa(\varphi)$ with $\varphi \in \Omega$, an agent considers ω to be more plausible than φ [Spo88].

In [Lib15], Liberatore defines total preorders as follows:

Definition 4 (Total Preorder)**.** A *Total Preorder* C is a partition of the models into a finite sequence of classes $[C(0), C(1), ..., C(n)]$ with $C(0) \neq \emptyset$.

Note that Definition 4 is just one way of present total preorders. Usually, total preorders are relations that are total, reflexive, transitive. Definition 4 is equivalent to that [Lib15]. Here, we use the representation by Libratore, because it is more convenient for the applications here. Because of Definition 4, total preorders can be visualised as a set of drawers (see Figure 3): in the bottom drawer, all maximally plausible models with rank 0 are stored. The further up the drawer is, the less plausible the stored models are. The models in the bottom drawer are then considered to be the right models, thus $C(0) = Mod(p)$. With each revision step, the drawers are shuffled around and other models can end up in the bottom drawer and are then considered to be the correct models [Lib15].

In the next section, the procedure of revision is illustrated and different revision operators are presented.

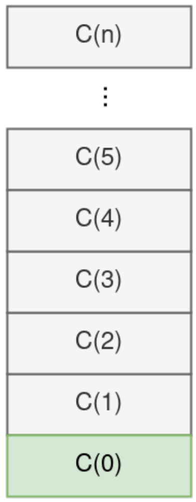

Figure 3: Total preorder (based on [Lib15]). The green box represents all models that are considered to be true in rank 0.

2.3 Belief Revision

Broadly speaking, belief revision is the process of incorporating new knowledge into existing beliefs, and adapting the existing beliefs to the new knowledge, if necessary [Ker01]. Revision is characterised by the following postulates, known as the AGM-framework, named after Alchourron, Gärdenfors and Makinson.

Let $\mathbb{B}$ be a propositional belief set, and let p be some proposition representing the newly acquired information which $\mathbb{B}$ is to be revised by. In the definition, $+$ stands for expansion, that is just incorporating new knowledge regardless whether it produces a conflict. Therefore, the new piece of information is just added to the existing knowledge base. Furthermore, $*$ describes the revision operator.

AGM postulates for revision [AGM16]
(AGM $*$1) $\mathbb{B} * p$ is a belief set.
(AGM $*$2) $p \in \mathbb{B} * p$.
(AGM $*$3) $\mathbb{B} * p \subseteq \mathbb{B} + p$.
(AGM $*$4) If $\neg p \notin \mathbb{B}$ then $\mathbb{B} + p \subseteq \mathbb{B} * p$.
(AGM $*$5) $\mathbb{B} * p$ is inconsistent iff p is contradictory.
(AGM $*$6) If p and q are logically equivalent, then $\mathbb{B} * p = \mathbb{B} * q$.
(AGM $*$7) $\mathbb{B} * p \wedge q \subseteq (\mathbb{B} * p) + q$.
(AGM $*$8) If $\neg q \notin \mathbb{B} * p$ then $(\mathbb{B} * p) + q \subseteq \mathbb{B} * p \wedge q$.

These postulates do not describe one fixed revision operator, but rather a class of possible operators. Later, Katsuno and Mendelzon [KM91] rephrased the postulates to be more precise for propositional logic.

(AGM' $*$1) $\mathbb{B} * p$ implies p.
(AGM' $*$2) If $\mathbb{B} \wedge p$ is satisfiable, then $\mathbb{B} * p \equiv \mathbb{B} \wedge p$.
(AGM' $*$3) If p is satisfiable, then $\mathbb{B} * p$ is also satisfiable.
(AGM' $*$4) If $\mathbb{B}_1 \equiv \mathbb{B}_2$ and $p_1 \equiv p_2$, then $\mathbb{B}_1 * p_1 \equiv \mathbb{B}_2 * p_2$.
(AGM' $*$5) $(\mathbb{B} * p) \wedge q$ implies $\mathbb{B} * (p \wedge q)$.
(AGM' $*$6) If $(\mathbb{B} * p) \wedge q$ is satisfiable, then $\mathbb{B} * (p \wedge q)$ implies $(\mathbb{B} * p) \wedge q$.

As can be seen in **(AGM $*$1)**, classical revision works with belief sets. As described in Section 2.2, iterated revision uses epistemic states instead of belief sets to represent the cognitive state of the human agent. Subsequently, revision operators have to be refined to incorporate working with epistemic states.

Definition 5 (Revision Operator for Ranking Functions)**.** A function $* : C \times \mathcal{L} \to C$ is called a *Revision Operator for Ranking Functions* if for each $a \in \mathcal{L}$ and each $\kappa \in C$ holds:

$$\{\omega \in \Omega \mid \kappa * a(\omega) = 0\} \subseteq Mod(a)$$

Formally, the AGM postulates have been expanded to work with epistemic states instead of belief sets, which allows the application to iterated belief revision in a dynamic system of belief change [Ker01].

AGM postulates for revising epistemic states [DP97]
Let Ψ, Ψ_1, Ψ_2 be epistemic states and $p_1, p_2, q \in \mathcal{L}$.
(AGM* $*$1) p is believed in $\Psi * p : Bel(\Psi * p) \models p$.
(AGM* $*$2) If $Bel(\Psi) \wedge p$ is satisfiable, then $Bel(\Psi * p) \equiv Bel(\Psi) \wedge p$.
(AGM* $*$3) If p is satisfiable, then $Bel(\Psi * p)$ is also satisfiable.
(AGM* $*$4) If $\Psi_1 = \Psi_2$ and $p_1 \equiv p_2$, then $Bel(\Psi_1 * p_1) \equiv Bel(\Psi_2 * p_2)$.
(AGM* $*$5) $Bel(\Psi * p) \wedge q$ implies $Bel(\Psi * (p \wedge q))$.
(AGM* $*$6) If $Bel(\Psi * p) \wedge q$ is satisfiable, then $Bel(\Psi * (p \wedge q))$ implies $Bel(\Psi * p) \wedge q$.

The most important addition is that only identical epistemic states yield the same revised epistemic states. If only the belief sets associated to the epistemic states are identical, the revision may yield a different outcome (see **(AGM $*$4)** and **(AGM* $*$4)**).

Formally, revision can be described in the form of revision sequences, which represent how the beliefs of the agent change over time.

Definition 6 (Revision Sequence [Lib15])**.** Let $C_0, ..., C_n$ be epistemic states and and $p_0, ..., p_n$ be propositions. A *Revision Sequence* is an odd sequence of consistent propositional formulae $[C_0, p_1, C_1, p_2, ..., p_n, C_n]$ over a finite set of variables.

In the following subsections, different revision operators for iterated revision are presented and their differences are elaborated.

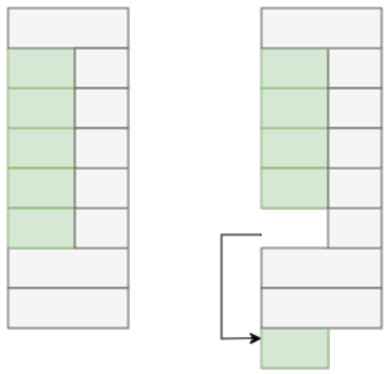

Figure 4: Natural Revision (based on [Lib15]). The green boxes represent models of the proposition p, the grey boxes represent interpretations that do not model p.

2.3.1 Natural Revision

The main characteristic of *Natural Revision* [Bou96] is the fact that the new preorder of the possible models is as close as possible to the one before the revision.

To get to that new preorder, only the minimal models of the formula p that the preorder is revised with are moved to rank 0 in the new preorder. The other possible models are all moved up one rank, so that only the minimal models of p remain in the lowest rank [Lib15].

Definition 7 (Natural Revision [Lib15])**.** Let C be the total preorder before the revision with the proposition p and C_p the preorder after the revision with p. Furthermore, let i be the minimal index so that $C(i) \cap Mod(p) \neq \emptyset$. The natural revision of the total preorder C by the proposition p is defined as:

$$C_p(j) = \begin{cases} C(i) \cap Mod(p) & \text{if } j = 0 \\ C(j-1) \backslash C_p(0) & \text{otherwise} \end{cases}$$

Graphically, Natural Revision can be seen as cutting out the lowest model of the formula and putting it in the bottom drawer (rank 0), while all other possible interpretations move up one drawer. Figure 4 shows how Natural Revision can be depicted.

2.3.2 Restrained Revision

Restrained Revision was first proposed by Booth and Meyer in [BM06]. Like in Natural Revision (see Chapter 2.3.1), the new total preorder after the revision with the proposition p features all models with minimal rank in rank 0.

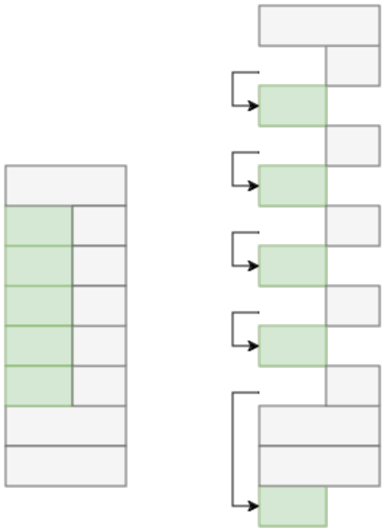

Figure 5: Restrained Revision (based on [Lib15]). The green boxes represent models of the proposition p, the grey boxes represent interpretations that do not model p.

Additionally, all other interpretations are separated based on whether they model the proposition p, and the classes they are in are split up in two classes, where the lower one contains models of p and the higher one contains all non-models. After the apportionment of the classes, Natural Revision is applied [Pap01].

Definition 8 (Restrained Revision [Lib15])**.** Let C be the total preorder before the revision with the proposition p and C_p the total preorder after the revision with p. Furthermore, let i be the minimal index so that $C(i) \cap Mod(p) \neq \emptyset$. Moreover, / denotes truncated integer division. The Restrained Revision of the total preorder C by the proposition p is defined as:

$$C_p(j) = \begin{cases} C(i) \cap Mod(p) & \text{if } j = 0 \\ (C((j-1)/2) \backslash C_p(0) \cap Mod(p) & \text{if } j > 0 \text{ odd} \\ (C((j-1)/2) \backslash C_p(0)) \backslash Mod(p) & \text{otherwise} \end{cases}$$

A graphical representation of the procedure of Restrained Revision can be seen in Figure 5.

2.3.3 Lexicographic Revision

The main idea behind Lexicographic Revision is that new knowledge is considered to be more plausible than older pieces of information. Therefore, all models of the proposition p, that the total preorder is revised with, are below all non-models of p [Lib15; Nay94; Spo88].

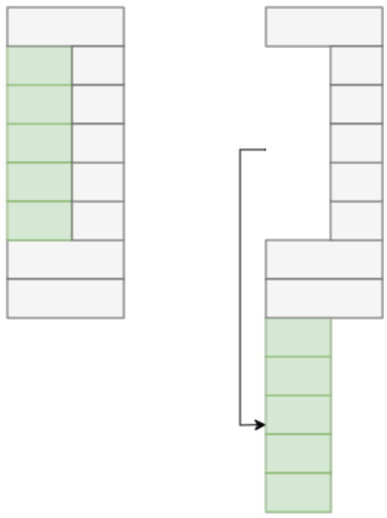

Figure 6: Lexicographic Revision (based on [Lib15]). The green boxes represent models of the proposition p, the grey boxes represent interpretations that do not model p.

Definition 9 (Lexicographic Revision [Lib15])**.** Let C be the total preorder before the revision with the proposition p and C_p the total preorder after the revision with p. Furthermore, let i be the minimal index so that $C(i) \cap Mod(p) \neq \emptyset$ and let j be the highest index so that $C(j) \cap Mod(p) \neq \emptyset$. The Lexicographic Revision of the total preorder C by the proposition p is defined by:

$$C_p(k) = \begin{cases} C(k+i) \cap Mod(p) & \text{if } k \leq j-i \\ C(k-j+i-1) \backslash Mod(p) & \text{otherwise} \end{cases}$$

In Figure 6, a graphical presentation of the procedure of Lexicographic Revision can be seen.

2.3.4 Reinforcement Revision

Contrary to all other revision operators that have been introduced before, the Reinforcement Revision operator not only receives a total preorder C and a proposition p to revise it with, it also receives a parameter m that expresses the belief in the proposition p, or, to be more precise, the disbelief in $\neg p$ [JT07].

Definition 10 (Reinforcement Revision [Lib15])**.** Let C be the total preorder before the revision with the formula p and C_p the preorder after the revision with p. Furthermore, let i be the minimal index so that $C(i) \cap Mod(p) \neq \emptyset$ and let j be the highest index so that $C(j) \cap Mod(p) \neq \emptyset$. Moreover, let $m > 0$ be the parameter of disbelief

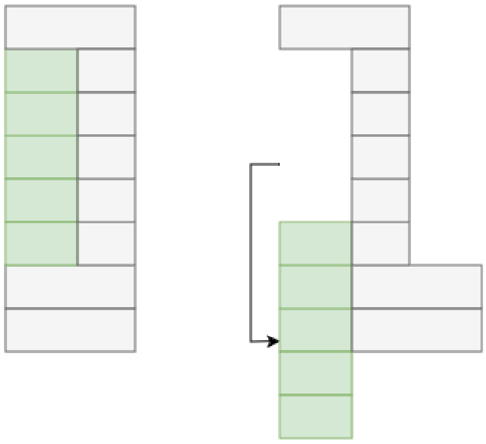

Figure 7: Reinforcement Revision (based on [Lib15]). In the graphic, the parameter $m = 2$ is used. The green boxes represent models of the proposition p, the grey boxes represent worlds that do not model p.

in $\neg p$. The Reinforcement Revision of the total preorder C by the proposition p is defined by:

$$C_p(j) = \begin{cases} C(i) \cap Mod(p) & \text{if } j = 0 \\ C(j-m) \backslash Mod(p) \cup C(j+i) \cap Mod(p) & \text{if } j > 0 \end{cases}$$

In Figure 7, a graphical interpretation of the Reinforcement Revision with the parameter $m = 2$ is visualised.

2.3.5 Revision Operator Proposed by Darwiche and Pearl

Based on the work of Spohn [Spo88], Darwiche and Pearl proposed a revision operator for iterated revision in [DP97], which will be called *DP-Revision* in this thesis.

Unlike the revision operators before, the definition of this operator and the following operators is not based on the classes of the possible models, but directly on the rank of the individual interpretations.

Definition 11 (Revision Operator by Darwiche and Pearl [DP97])**.** Let κ be a ranking function an and let $\omega \in \Omega$ be an interpretation. Furthermore, let p be the proposition that κ is revised with. Moreover, the rank $\kappa(p)$ of a proposition p is defined as $\kappa(p) = \min_{\omega \models p} \kappa(\omega)$. The DP-Revision of κ by p is defined as:

$$(\kappa * p)(\omega) = \begin{cases} \kappa(\omega) - \kappa(p) & \text{if } \omega \models p \\ \kappa + 1 & \text{if } \omega \models \neg p \end{cases}$$

Upon closer inspection, it can be seen that the DP-Revision is just a special case of reinforcement revision (see Chapter 2.3.4) with the parameter $m = 1$. That means that all models of p are moved downwards by the same number, so that the lowest model has rank 0 after the revision. Furthermore, all non-models are moved up one rank.

2.3.6 Conditional Revision

Conditional Revision is a revision operator especially for revising a ranking function with conditional information. This is especially necessary, because conditionals $p \Rightarrow q$ or $(q|p)$ partition worlds into three classes: first, worlds that fulfill $p \wedge q$ and therefore confirm the conditional; second, worlds that fulfil $p \wedge \neg q$ and thus contradict the conditional; and third, worlds that fulfill $\neg p$, so that the conditional cannot be applied [Ker99].

For this reason, conditionals can be seen as a generalised indicator function [Fin37]:

$$(q|p)(\omega) = \begin{cases} 1 & \text{if } \omega \models pq \\ 0 & \text{if } \omega \models p\overline{q} \\ \text{undefined} & \text{if } \omega \models \overline{p} \end{cases}$$

Based on the nature of conditionals, an additional set of postulates for conditional revision has been proposed in [Ker99]. For plain proposition, revision operators behave as AGM revision operators, since the new class of operators for revision by conditionals extends the operator for revision by propositions.

> **Postulates for conditional revision** [Ker99]
> Let Ψ be an epistemic state and $(q|p), (s|r)$ be conditionals. Furthermore, let $\Psi * (q|p)$ be the result of revising Ψ by $(q|p)$.
> **(CR1)** $\Psi * (q|p)$ is an epistemic state.
> **(CR2)** $\Psi * (q|p) \models (q|p)$.
> **(CR3)** $\Psi * (q|p) = \Psi$ iff $\Psi \models (q|p)$.
> **(CR4)** $\Psi * q = \Psi * (q|\top)$ induces a propositional AGM-revision operator.
> **(CR5)** $\Psi * (q|p) = \Psi * (s|r)$ whenever $(q|p) \equiv (s|r)$.
> **(CR6)** If $(s|r) \perp (q|p)$ then $\Psi \models (s|r)$ iff $\Psi * (q|p) \models (s|r)$.
> **(CR7)** If $(s|r) \sqsubseteq (q|p)$ and $\Psi \models (s|r)$ then $\Psi * (q|p) \models (s|r)$.
> **(CR8)** If $(s|r) \sqsubseteq (\overline{q}|p)$ and $\Psi * (q|p) \models (s|r)$ then $\Psi \models (s|r)$.

The main purpose of the postulates is to ensure that conditional beliefs, which are not in conflict with new pieces of information, are maintained and not removed from the knowledge set [Ker99].

Similarly to probability theory, $\kappa(q|p) = \kappa(pq) - \kappa(p)$ is defined [Ker99; GP96]. With the postulates, a revision operator is proposed as follows:

Definition 12 (Conditional Revision [Ker99])**.** Let $\kappa \in K$ be an OCF and $(q|p)$ be a conditional. The Conditional Revision of κ by $(q|p)$ is defined as:

$$(\kappa * (q|p))(\omega) = \begin{cases} \kappa(\omega) - \kappa(q|p) & \text{if } \omega \models pq \\ \kappa(\omega) + \alpha + 1 & \text{if } \omega \models p\overline{q} \\ \kappa(\omega) & \text{if } \omega \models \overline{p} \end{cases}$$

with

$$\alpha = \begin{cases} -1 & \text{if } \kappa(pq) < \kappa(p\overline{q}) \\ 0 & \text{else} \end{cases}$$

and $\kappa(r) = \min_{\omega \models r} \kappa(\omega)$.

2.3.7 Revision Operator Proposed by Häming and Peters

The revision operator proposed by Häming and Peters, which will be referenced by *HP-Revision* in this thesis, differentiates between propositional information and conditional information. Furthermore, like Reinforcement Revision (see Chapter 2.3.4), it features a strength parameter β, which is also used to differentiate between formulae: if a formula is already believed with strength β, then no changes need to be applied to the current ranking function [HP12]. It is an advancement of the Conditional Revision operator introduced in Chapter 2.3.6.

Definition 13 (Revision Operator for Propositional Information by Häming and Peters [HP12])**.** Let p be a proposition and $\omega \in \Omega$ be an interpretation. Furthermore, let $\beta > 0$ be a strength parameter. The revision of κ by the proposition p is defined as:

$$(\kappa * (p, \beta))(\omega) = \begin{cases} \kappa(\omega) & \text{if } \kappa(\overline{p}) \geq \beta \\ \kappa(\omega) - \kappa(p) & \text{if } \kappa(\overline{p}) < \beta \wedge \omega \models p \\ \kappa(\omega) + \beta - \kappa(\overline{p}) & \text{if } \kappa(\overline{p}) < \beta \wedge \omega \models \overline{p} \end{cases}$$

with $\kappa(r) = \min_{\omega \models r} \kappa(\omega)$.

For revision with conditional information $(q|p)$ with the antecedent p and the consequent q, another operator is used. The new operator takes into consideration that conditionals are already believed to be true when the minimal models fulfill $\overline{p}q$, and the ranks are not updated. This operator is an enhancement of Conditional Revision (see Chapter 2.3.6) and adds the new operator [HP11]

$$\kappa[p] = max\{\kappa(\omega) \,|\, \omega \models p\}$$

Based on the new operator, the revision with conditional knowledge is defined as follows.

Definition 14 (Revision Operator for Conditional Information by Häming and Peters [HP12]). Let $(q|p)$ be a conditional proposition and $\omega \in \Omega$ be and interpretation. Furthermore, let $\beta > 0$ be a strength parameter. The revision of the ranking function μ by the conditional $(q|p)$ is defined as:

$$(\kappa * (q|p, \beta))(\omega) =$$
$$\begin{cases} \kappa(\omega) & \text{if } D \geq \beta \\ \kappa(\omega) - \kappa(p \Rightarrow q) & \text{if } D > \beta \wedge \omega \models (p \Rightarrow q) \\ \kappa(\omega) + (\kappa[pq] - \kappa(p \Rightarrow q) + \beta) - \kappa(p\overline{q}) & \text{if } D < \beta \wedge \omega \models (p\overline{q}) \end{cases}$$

with $D = \kappa(p\overline{q}) - \kappa[pq]$ and $\kappa(r) = \min_{\omega \models r} \kappa(\omega)$.

Hereby, $p \Rightarrow q$ denotes the material implication $\neg p \vee q$.

2.4 Cognitive Background

In this chapter, the main characteristics of human logical reasoning applied in this thesis are outlined. Furthermore, the differences compared to classical logic are shown. First, the theory of Mental Models is explained and afterwards different frameworks for human logical reasoning are characterised.

The main goal of analysing human reasoning is to develop a *cognitive logic*, which recreates the human reasoning and inference process, for example for applications in human computer interactions, to make systems more adaptable to human interactions [Rag+20].

2.4.1 Mental Model Theory

The theory of Mental Models was introduced by Johnson-Laird and Byrne [JBS92] and describes how humans process linguistic information and translate it to a representation of the situation.

Mental Models work in the following way: an agent has a set of beliefs which is depicted in a set of Mental Models. In each model, one state that is deemed true is encoded. Hereby, one Mental Model corresponds to a row in a truth table, however, it does not necessarily contain all information in that row. Moreover, not all rows of the truth table need to be represented in a Mental Model. According to the *Principle of Truth*, only true clauses are represented. Furthermore, only clauses that are explicitly true in a possible model are depicted [JB02].

Mental Models are visualised in the following way: if the agent is given the task "If A is true, then B is true", the sentence refers to a set of possible models:

$$\begin{array}{c} A\,B \\ \ldots \end{array}$$

First, the conjunction of A and B, which is depicted in the first model AB, and second, the model ... refers to other possibilities with $\overline{A}$ [RKJ18]:

Inference Form	Major Premise	Minor Premise	Conclusion
Modus Ponens (MP)	$p \Rightarrow q$	p	q
Acceptance of the Consequent (AC)	$p \Rightarrow q$	q	p
Denial of the Antecedent (DA)	$p \Rightarrow q$	$\neg p$	$\neg q$
Modus Tollens (MT)	$p \Rightarrow q$	$\neg q$	$\neg p$

Table 8: Human inference forms [Obe06]

$$
\begin{array}{c}
A\,B \\
\overline{A}\,B \\
\overline{A}\,\overline{B}
\end{array}
$$

Depending on the age and mental capacity of the agent, a different number of possible models can be processed at the same time. As a result, simplified models are constructed. These can contain footnotes of other possible models, which may be forgotten or retained over time [BGL00].

Inferences are drawn in three steps: first, a set of mental models is formed, which fulfill both the major and the minor premise. Then, a conclusion which is true for all models is formulated, and last, the agent searches for possible counterexamples. If they cannot find any, the conclusion is considered true. If there are counterexamples, the agent infers that nothing can be followed [Obe06; BGL00].

In Table 8, four common reasoning schemes are presented.

Hereby it has to be noted that only *Modus Ponens (MP)* and *Modus Tollens (MT)* hold for classical propositional logic. Both *Acceptance of the Consequent (AC)* and *Denial of the Antecedent (DA)* do not hold for classical propositional logic; they are sometimes denoted as logical fallacies.

However, research shows that most humans apply MP correctly, and AC and DA are endorsed frequently. MT on the other hand is applied significantly less [Obe06; BGL00; RDJ19].

2.4.2 Fully Explicit Models

This approach to human reasoning assumes that human reasoning is logically correct. For this, all correct models are deduced and represented as *Fully Explicit Models (FEM)* [RKJ18]. Furthermore, only correct inference rules are applied and therefore the human reasoning is identical to deductive reasoning [Ism+23].

The human agent reasons in the following way: all possible interpretations are separated in two groups; one, where the interpretations are models of the proposition and one where they are not. Afterwards, only models are considered further and are therefore possible models after being presented with the next piece information.

Inference Form	Major Premise	Minor Premise	Conclusion
Modus Tollendo Ponens (MTP)	$p \vee q$	$\neg p$	q
Modus Ponendo Tollens (MPT)	$\overline{(p \wedge q)} \equiv \overline{p} \vee \overline{q}$	p	$\neg q$
Affirming a Disjunct (AD)	$p \vee q$	p	$\neg q$
Denying a Conjunct (DC)	$\overline{(p \wedge q)} \equiv \overline{p} \vee \overline{q}$	$\neg p$	q

Table 9: Human inference forms for OR [Ism+23]

2.4.3 Exclusive Disjunctions

Since in everyday use the connective *or* is mainly understood as exclusive, some humans tend to interpret the logical (inclusive) *or* as exclusive as well [NGC84; DC17].

With the interpretation of the inclusive OR ($\vee$) as exclusive OR (XOR / $\dot{\vee}$), the interpretations are evaluated using a different truth table (see Table 2).

Based on the exclusive interpretation of the logical OR, additional inference forms are applied by the agent (see Table 9).

Just like the inference forms applied by humans for conditionals, not all inference forms applied to disjunctions are valid in classical propositional logic: while *Modus Tollendo Ponens (MTP)* and *Modus Ponendo Tollens (MPT)* are logically correct, *Affirming a Disjunct (AD)* and *Denying a Conjunct (DC)* are logically wrong for inclusive disjunctions.

2.4.4 Biconditional Interpretation of Conditionals

It has also been observed that humans tend to interpreting conditionals $\Rightarrow$ as biconditionals $\Leftrightarrow$ [Wag07]. The corresponding truth table of the human interpretation can be seen in Table 2 in the corresponding columns.

For this phenomenon, three possible theories have been suggested:

Understanding conditionals as biconditionals Firstly, similarly to the *Exclusive Interpretation of Disjunctions* (see Section 2.4.3), humans tend to interpret conditional statements "if ... then ..." as biconditionals (meaning "if, and only if, ... then ..."). This is a result of the typical everyday use of conditional statements in human language as biconditionals [Ver+01]. For this reason, humans fail to construct all possible models and therefore draw conclusions that only hold in some, but not all, models [Joh13].

Human memory The second possible explanation is related to the concept of mental models as well: as can be seen in Table 2, the biconditional interpretation ($\Leftrightarrow$) has only two models

$$\begin{array}{c} A\,B \\ \overline{A}\,\overline{B} \end{array}$$

while the conditional interpretation ($\Rightarrow$) has three models

$$\begin{gathered} A\,B \\ \overline{A}\,B \\ \overline{A}\,\overline{B} \end{gathered}$$

with the additional model $\overline{A}B$. The higher number of models can lead to incorrect reasoning. On one hand, humans tend to prefer more intuitive models [RDJ19] (such as AB and $\overline{A}\,\overline{B}$) and on the other hand, models can be forgotten over time [BGL00].

Preference to give answers Lastly, humans want to draw conclusions rather than saying that nothing follows. For this reason, if possible answers and "nothing" as an answer are presented, humans tend to not choose "nothing" as an answer. This kind of reasoning is wrong if the antecedent of the conditional is wrong, since the consequent can be either true or false in that case for the conditional to be true. This phenomenon can be explained since humans tend to look for new information instead of confirming that no knowledge can be derived [Ver+01].

2.4.5 Preferred Interpretations

Similarly to the *Biconditional Interpretation of Conditionals* in the previous chapter, humans tend to interpret conditionals in another way that is logically incorrect. Research has shown that only the first model AB is apparent to most agents, while other models need more cognitive work to be constructed [BGL00; KBJ18].

Therefore, only the model AB is endorsed, while the models $\overline{A}B$ and $\overline{A}\,\overline{B}$ are omitted.

The comparison of the correct truth table for the conditional ($\Rightarrow$) and the preferred interpretation (AND / $\wedge$) can be seen in Table 2. This phenomenon is referred to as the *Principle of Preferred Interpretations (PoPI)* [Ism+23].

This also lines up with the explanation for the biconditional interpretation of conditionals: humans tend to prefer more apparent models over ones that are more difficult to construct [RDJ19].

3 Realisation of the Revision Approach

In this chapter, it will be presented how revision operators are implemented using the theoretical background presented in the previous chapter. In the first section, the developed *Sequential Revision Approach* is introduced and in the second section, the incorporation of the different mental approaches is explained.

3.1 Sequential Revision Approach

In Figure 10a, the pipeline of the *Sequential Revision Approach* is shown: As a first step, the initial ranking function κ_0 is set. This assigns the rank 0 to each possible interpretation, to visualise that the agent is not biased towards any particular interpretation. In the next step, the new piece of information p_1 is incorporated. This works as follows: the prior ranking function κ_0 is revised with the information p_1 and the new ranking function κ_1 is calculated.

This process can then be repeated for each new piece of information p_i, which is revised with κ_{i-1} to calculate the new ranking function κ_i. Lastly, the final ranking function κ_n is calculated by revising the final piece of information, p_n, with κ_{n-1} by calculating $R[*](\kappa_{n-1}, p_n)$.

Based on the final calculated ranks, the corresponding answer of the task can be selected. The procedure will be explained later in Chapter 4.2.

Hereby, the revision can be realised with different revision operators (see Chapter 2.3).

The different revision operators all work in a similar way: they are passed the ranks prior to the revision and the new piece of information. Hereby, the ranks are stored in a list and each index has to be linked to an interpretation. The operators then calculate the updated rank for each model, based on the used revision operator.

Furthermore, different mental approaches can be incorporated, which will be presented in the next section.

$$
\begin{tikzcd}
\kappa_0 \arrow[rr, "*"] & & \kappa_1 = R[*](\kappa_0, p_1) & \cdots & \kappa_{n-1} \arrow[rr, "*"] & & \kappa_n = R[*](\kappa_{n-1}, p_n) \\
& p_1 \arrow[u] & & \cdots & & p_n \arrow[u] &
\end{tikzcd}
$$

(a) *Fully Explicit Models*

$$
\begin{tikzcd}
\kappa_0 \arrow[rr, "*"] & & \kappa_1 = R[*](\kappa_0, p'_1) & \cdots & \kappa_{n-1} \arrow[rr, "*"] & & \kappa_n = R[*](\kappa_{n-1}, p'_n) \\
& p'_1 = f(p_1) \arrow[u] & p_1 \arrow[l] & \cdots & & p'_n = f(p_n) \arrow[u] & p_n \arrow[l]
\end{tikzcd}
$$

(b) *Biconditional Interpretation of Conditionals*, *Principle of Preferred Interpretations*, and *Exclusive Disjunctions*

$$
\begin{tikzcd}
\kappa_0 \arrow[rr, "*"] & & \kappa_1 = R[*](\kappa_0, [p_1]) & \cdots & \kappa_{n-1} \arrow[rr, "*"] & & \kappa_n = R[*](\kappa_{n-1}, [p_1, ..., p_n]) \\
& p_1 \arrow[u] & & \cdots & & p_n \arrow[u] &
\end{tikzcd}
$$

(c) *Mental Models*

Figure 10: Pipeline of the *Sequential Revision Approach* for the different mental approaches.

3.2 Mental Approaches

As presented in Chapter 2.4, humans often fail to draw correct conclusions from logical propositions. To incorporate the approaches *Principle of Preferred Interpretations* (see Chapter 2.4.5), *Biconditional Interpretation of Conditionals* (see Chapter 2.4.4), and *Exclusive Disjunctions* (see Chapter 2.4.3), an additional step is added to the revision.

As can be seen in Figure 10b, the new piece of information p_i is altered by the function f for the approaches *Biconditional Interpretation of Conditionals*, *Principle of Preferred Interpretations*, and *Exclusive Disjunctions*.

According to the mental approach used, the information is adapted in the following ways:

Biconditional Interpretation of Conditionals Since in the approach of *Biconditional Interpretation of Conditionals* humans interpret conditionals as biconditionals, the new piece of information is altered by replacing the implication $\Rightarrow$ with the biimplication $\Leftrightarrow$. Therefore, the task $a \Rightarrow b$ is reformulated to be $a \Leftrightarrow b$. This step is conducted by the function f.

Principle of Preferred Interpretations As explained in Chapter 2.4.5, humans tend to prefer the conjunction $a \wedge b$ for the proposition $a \Rightarrow b$, while other models are considered to be wrong. Therefore, the conditional proposition $a \Rightarrow b$ is rephrased to be $a \wedge b$. The function f conducts this reformulation.

Exclusive Disjunctions The human misconception that inclusive disjunctions $a \vee b$ are interpreted as exclusive disjunctions $a \dot{\vee} b$ is considered in this mental approach. To incorporate this approach in the sequential revision, the proposition $a \vee b$ is adapted to be $a \dot{\vee} b$ by the function f.

Unlike the three presented approaches, the *Mental Models* approach is not only based on the latest piece of information, but also on the information presented before. Therefore, strictly speaking, is not a classical revision.

Mental Models The adapted pipeline for the *Mental Models* approach is depicted in Figure 10c. The main difference is that in the revision itself, not only the current piece of information p_i is considered, but also all other propositions. Therefore, the revision operator is passed on all pieces of information $[p_1, ..., p_i]$.

This step is necessary since for AC, DA, and MP, the minor premise needs to be evaluated, and not just the conditional itself. It is a difference whether the antecedent or the consequent is known, if the conditional is applied by the agent. The corresponding rules can be seen in Table 8 in Chapter 2.4.1. Based on the given information, it is chosen whether an inference rule can be applied and afterwards, the corresponding rule is applied.

4 Implementation

In this chapter, the implemented revision approach is presented and the main features of the implementation are outlined. First, the used programming language and libraries are presented. Second, the processing of the tasks is described. Third, the developed *Sequential Revision Approach* is introduced and its functionality is explained. Fourth, the applied cognitive approaches and their implementation are described, and last, the implementation of the chosen revision operators is presented.

4.1 Programming Language and Libraries

The software for this thesis was developed using the programming language *Python 3*. Even though *Python* was mostly chosen out of personal preference, it also combines a number of benefits: First off, it is one of the most used programming languages, and can be used for a number of purposes, including Data Science, Machine Learning, and Scientific Computing. Additionally, it is open-source and can therefore be used by everyone for free. Furthermore, it has a number of standard libraries for all types of tasks [Sri17].

Two of these libraries have been used in this project: *sympy* and *pandas*. *sympy* is a library for symbolic mathematics and logic [Tea23] and is used in this thesis to evaluate logical expressions with different assignments. Furthermore, *pandas* is a tool for data analysis and manipulation [pan24] and is used later in the evaluation of the implemented software (see Chapter 5).

4.2 Processing of the Task

In this chapter, the processing of the single tasks is explained and the approach to make the task *Strings* able to be processed by the program. Afterwards, it is explained how an answer is chosen based on the calculated ranks.

Making *sympy*-expressions To evaluate propositions based on the assignment of the variables, the Python library *sympy* (see also Chapter 4.1) is used. Therefore, the expressions used in the task need to be converted into *sympy*-expressions from the Polish Notation [Wik24] of the tasks.

For the conversion, the function `makeExpression()` is used. It is a recursive formula which translates the tasks it is passed on into *sympy*-compatible expressions and works as follows: First, the expression is split at each `/` and the single parts of the task are stored in a list. In the next step, each part of the task is then split again at each `;` and stored as a list as well. Therefore, the task is split up into a list of lists, which each describe one part of the task.

Second, the actual translation of the tasks is performed, which is realised as a recursive function. The base cases are the atoms `A` and `B`: if the first item of a task is one of the base cases, the returned *sympy*-expression is just the atom itself. The recursive cases are unary operator NOT ($\neg$) and the binary operators OR ($\vee$), AND ($\wedge$), IF ($\Rightarrow$) and IFF ($\Leftrightarrow$) (see also Chapter 2.1).

The function `applyNOT()` for the application of the unary operator NOT works as follows: First, `expression` is set to `'Not('`, according to the rules of the *sympy*-expressions. Then, the first element (`Not` or `not`) is removed from the list of the task and depending on the new first element, the corresponding function is called recursively to translate the inner part of the NOT statement. Lastly, `expression` is finished by adding `')'`, so the expression is well-formed.

The application of the binary operators OR, AND, IF and IFF using the function `apply2V()` works similar to the application of NOT: first, `expression` is set to be the following, based on the first element of the list of the task:

`if`	`'Implies('`
`iff`	`'Equivalent('`
`or`	`'Or('`
`and`	`'And('`

Afterwards, the first part of the formula is translated by calling the respective function recursively and then adding `', '` to the expression and removing the first element of the list. Lastly, the second part of the formula is translated by calling the respective function again and finally adding `')'` to the expression.

This procedure is reiterated for each part of the task, and as a result, all parts of the task are converted into *sympy*-compatible expressions and can be used to evaluate the formulae.

Choosing the correct answer The second part of processing the task itself is choosing the answer(s) based on the calculated ranks of the different models. For this task, the function `chooseAnswer()` is used. It is passed on the final ranks of the different interpretations after the revision process and the possible answers. Before the actual answer is chosen, the available answers have to be adapted: the answer containing the negation of the minor premise has to be removed from the available answers. First, the possible interpretations are defined again and stored in a list. Second, the minimal rank is determined and saved in `minRank` and all interpretations with the rank of `minRank` are saved in a list. Third, the models are separated into their values for `A` and `B` and saved in `answerA` and `answerB` respectively.

As a last step, the answers for `A` and `B` are chosen. This is achieved by looking at the list of possible answers: if the possible answers are `[0, 0]` or `[0]`, the chosen answer is `not;A` or `not;B`, if the possible answers are `[1, 1]` or `[1]`, the chosen answer is `A` or `B`, and else, the chosen answer is `nothing`. The chosen answers are then saved in `answers`.

Next, the possible answers of the task are split at | and saved in `possibleAnswers`. The answers are then compared and the one that is in both the chosen and possible answers is returned as the correct answer.

4.3 Implementation of the Sequential Revision Approach

In this section, the developed *Sequential Revision Approach* is presented and the underlying mechanics are illustrated. In Figure 10 in Chapter 3.1, the pipeline of the algorithm is depicted.

The *Sequential Revision Approach* processes tasks in the following way: First, an initial ranking function κ_0 for all possible interpretations is set up. Since in the beginning all interpretations are viewed as equally probable, all interpretations are assigned the rank 0. Since in this thesis only reasoning over two variables is treated, the possible models are

$$\overline{A}\,\overline{B}$$
$$\overline{A}\,B$$
$$A\,\overline{B}$$
$$A\,B$$

and are represented as

$$[0,0]$$
$$[0,1]$$
$$[1,0]$$
$$[1,1]$$

Moreover, the initial ranks `[0, 0, 0, 0]` for the interpretations `[[0, 0], [0, 1], [1, 0], [1, 1]]` can be depicted as (see also Chapter 2.2)

1	
0	$A\,B, \overline{A}\,B, A\,\overline{B}, \overline{A}\,\overline{B}$

Second, a new piece of knowledge p_1 is introduced and the new ranking function κ_1 is calculated by revising the old ranking function κ_0 with p_1. The resulting ranking function is $\kappa_1 = R[*](\kappa_0, [p_1])$ and is calculated using the corresponding implemented revision operator.

Subsequently, each new ranking function is calculated by revising the current ranking function with the new piece of knowledge. The revision can be computed using different revision operators. Lastly, the final ranking function is calculated by revising the penultimate ranking function κ_{n-1} with the last piece of information p_n and is evaluated as $\kappa_n = R[*](\kappa_0, [p_1, p_2, ..., p_n])$.The ranking functions are calculated individually for each possible interpretation.

For calculating the revision, the new information or the application of propositional logic is adapted based on the cognitive approach used by the agent. The approaches are presented in Chapter 2.4 and their implementation will be described in detail in Chapter 4.5. Furthermore, the implementation of the different revision operators, which were first presented in Chapter 2.3, will be characterised in Chapter 4.4.

4.4 Revision Operators

In this chapter, the selected revision operators and their implementation are presented. First, the general mechanisms used for all revision operators are introduced. Then, the revision operators proposed in [Lib15] are presented. Those include Natural, Restrained, Lexicographic and Reinforcement Revision. Afterwards, the Revision Operator proposed by Darwiche and Pearl (DP-Revision) is presented and finally, two forms of conditional revision, the Conditional Revision operator and the operator proposed by Häming and Peters (HP-Revision), are described.

4.4.1 Structure of the Revision Operators

The general structure of the revision operators is similar for each of them. Therefore, it will be explained separately in this chapter. The following chapters then describe the different revision operators and the differences between them.

The general procedure is as follows: Before the actual revision is applied, the cognitive approach is taken into consideration: if the *Principle of Preferred Interpretations, Biconditional Interpretation of Conditionals* or the use of *Exclusive Disjunctions* is applied, the proposition has to be adapted. The applied functions for each approach can be seen in Table 11 and will be presented in Chapter 4.5.

Next, the initial ranks and the possible interpretations are defined as:

```
possibleModels = [[0, 0], [0, 1], [1, 0], [1, 1]]
ranks = [0, 0, 0, 0]
```

Afterwards, for each part of the task, the respective revision operator is applied and the ranks are updated. Lastly, the final ranks are returned and an answer is chosen (see Chapter 4.2).

The revision operator is passed the following parameters, that can be seen in Table 12.

As a default, `rankingFunction` is set to `'fem'`, but it can take on different values, one for each mental approach. The different abbreviations for the mental approaches can be seen in Table 13.

Principle of Preferred Interpretations	`makePreferredInterpretations()`
Biconditional Interpretation	`makeBiconditionalInterpretations()`
Exclusive Disjunctions	`makeExclusiveDisjunctions()`

Table 11: Functions for the different cognitive approaches

`task[i]`	Part of the task that is processed in this step.
`ranks`	Ranks before the revision.
`possibleModels`	All possible interpretations.
`task`	The entire task to be processed.
`rankingFunction`	The applied mental approach.

Table 12: Parameters passed on to the revision operator

Abbreviation	Mental Approach
`'fem'`	Fully Explicit Models
`'ed'`	Exclusive Disjunctions
`'bi'`	Biconditional Interpretation of Conditionals
`'pi'`	Principle of Preferred Interpretations
`'mm'`	Mental Models

Table 13: Abbreviations for the different mental approaches

4.4.2 Natural Revision Operator

The theoretical background behind the Natural Revision operator was introduced in Chapter 2.3.1. The main principle is that it only puts the model(s) with the lowest ranks in the new rank 0, while the ranks of all other interpretations (whether they are models of the formula or not) move up one rank.

Before the revision, the cognitive approach is applied, as described in the previous chapter. Then, as the first step of the actual revision process, the minimal rank of the models of the new piece of knowledge is calculated. To calculate said rank, all possible interpretations are evaluated on whether they model the new piece of information or not. Of the ranks of those that model the proposition, the lowest rank is stored in the variable `minRank`. Since the rank of formulae has to be calculated in all implemented revision operators, the function `calculateFormulaRank()` is written in a different script and imported into all revision operators.

In the next step, the new ranks are calculated as follows:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{minRank} & \text{if } i \models p \text{ and } \kappa_{old} = \texttt{minRank} \\ \kappa_{old}(i) + 1 & \text{else} \end{cases}$$

Whether or not an interpretation models the interpretation is evaluated using the developed `checkModel()` or `mentalModel()` function, depending on the applied cognitive approach. If any approach other than *Mental Models* is applied, `checkModel()` is used. The function `mentalModel()` will be explained later in the corresponding chapter (see Chapter 4.5.5).

The function `checkModel()` works as follows: first, a dictionary in the form of

$$\{A : True; B : False\}$$

is constructed and A and B are defined as *sympy* symbols. The example dictionary is for the model $A\overline{B}$ or $[1; 0]$. This dictionary is then passed to the *sympy* function `subs()` and applied to the formula in the new piece of information. `subs()` then returns whether the interpretation models the formula.

Just like `calculateFormulaRank()`, `checkModel()` is used in every implemented revision operator and therefore also stored in a single script and imported and used by each revision operator.

As described in Chapter 4.3, the process of revision for one task proceeds as follows: for each new information in a task, a new ranking function is calculated. To calculate the ranks, each possible interpretation is processed on its own one after the other.

4.4.3 Restrained Revision Operator

As introduced in Chapter 2.3.2, the principle of Restrained Revision is to partition all ranks into two ranks each, where the lower contains all models of the proposition, while the higher one contains all non-models. After the partitioning, Natural Revision is applied.

The procedure of the Restrained Revision Operator is therefore also similar to the application of Natural Revision. Before the revision, the cognitive approach is taken into consideration and the formula is adapted, if need be. Afterwards, the minimal rank of a model of the formula is calculated using the method `calculateFormulaRank()` and stored in the variable `minRank`.

In the next step, for each possible interpretation the new rank is calculated using the following formula:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{minRank} & \text{if } i \models p \text{ and } \kappa_{old} = \texttt{minRank} \\ \kappa_{old}(i) * 2 + 1 & \text{if } i \models p \text{ and } \kappa_{old} \neq \texttt{minRank} \\ \kappa_{old}(i) * 2 + 2 & \text{else} \end{cases}$$

To partition the interpretations into the corresponding categories, as a first step, it is checked using `checkModel()` whether an interpretation models the proposition. Out of those that do, the rank is retrieved and compared to `minRank`. The different models are then separated into three classes and the new rank is calculated accordingly for all interpretations.

The calculation of the updated ranks using $\kappa_{new}(i) = \kappa_{old}(i) * 2 + 1$ and $\kappa_{new}(i) = \kappa_{old}(i) * 2 + 2$ splits all old classes into two new classes, with the lower one being that of all models and the higher one that of all non-models. Calculating $\kappa_{new}(i) = \kappa_{old}(i) -$ `minRank` puts the model(s) with the lowest rank in the new rank 0.

4.4.4 Lexicographic Revision Operator

Lexicographic Revision was introduced in Chapter 2.3.3 and its key idea is as follows: all models of the proposition are placed in ranks below all non-models. However, unlike with Natural and Lexicographic Revision, the difference between the ranks of all models and non-models respectively remain the same.

As a first step, the formula is adapted according to the mental approach used.

To calculate the updated ranks, first, the maximum and minimum rank of models of the proposition are calculated and stored in `minRank` and `maxRank`. To calculate the minimal rank, `calculateFormulaRank()` is used.

For calculating the maximum rank, the formula `calculateMaxRank()` was implemented. Its functionality basically is the same as that of `calculateFormuaRank()`: first, the ranks of all models are stored in a list and then the maximum is returned.

The new ranks are then calculated using the following formula:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{minRank} & \text{if } i \models p \\ \kappa_{old}(i) - \texttt{minRank} + \texttt{maxRank} + 1 & \text{else} \end{cases}$$

In this formula, shifting the ranks of all models down by `minRank`, all models of the proposition are placed in the bottom ranks. Furthermore, all non-models are shifted up by `maxRank` $-$ `minRank` $+1$. Since all models and non-models respectively are treated in the same way, the relative ordering between them is preserved.

4.4.5 Reinforcement Revision Operator

The concept of Reinforcement Revision (see Chapter 2.3.4) is similar to that of Lexicographic Revision. However, not all models of the proposition are places below all non-models and instead, all non-models are shifted up by the so-called `beliefParameter`.

First, like with all revision operators, the proposition is adjusted according to the mental approach that is applied.

Second, the new ranks are calculated. Therefore, `minRank` is calculated using `calculateFormulaRank()`. Afterwards, the ranks are calculated as follows:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{minRank} & \text{if } i \models p \\ \kappa_{old}(i) + \texttt{beliefParameter} & \text{else} \end{cases}$$

The interpretations are partitioned into the two classes, models and non-models of the proposition, using the function `checkModel()` (see Chapter 4.4.2). The `beliefParameter` can be passed over to the function `reinforcementRevisionOperator()`, and is set to a default value of 2.

4.4.6 Darwiche-Pearl Revision Operator

The DP-Revision operator (see Chapter 2.3.5) is a particular case of the Reinforcement Revision operator presented in the last chapter.

Therefore, the procedure is similar to the one used in `reinforcementRevisionOperator()` with `beliefParameter` $= 1$. First, the proposition is adapted based on the mental approach. Then, the updated ranks are calculated using `minRank`, which is calculated using `calculateFormulaRank()`. The ranks are calculated using the following formula:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{minRank} & \text{if } i \models p \\ \kappa_{old}(i) + 1 & \text{else} \end{cases}$$

The interpretations are partitioned into the different classes, models and non-models, using the function `checkModel()` and the ranks are calculated accordingly.

4.4.7 Conditional Revision Operator

The Conditional Revision operator proposed by Kern-Isberner in [Ker99] is a revision operator specifically for conditionals. For this reason, the application of the *Principle of Preferred Interpretations* and the *Biconditional Interpretation of Conditionals* cannot be applied, since that would just induce another revision operator, namely the DP Revision Operator (see Chapter 4.4.6). Furthermore, the application of the Mental Model approach cannot be applied either, since it does not apply logic to conditionals correctly.

As a first step, the operator differentiates between conditional and propositional interpretation. To check for the nature of the information, the piece of information is transformed into a *String* and the substring `'Implies('` is searched for. If the information contains the substring, it is a conditional and the `conditionalRevisionOperator()` is applied. Otherwise, as follows from **(CR4)** in Chapter 2.3.6 in the *Postulates for Conditional Revision*, a propositional revision operator is induced. In this implementation, the DP-Revision operator is applied, however, any other operator presented in Chapters 4.4.2 to 4.4.6 could be used.

As can be seen in the definition of the revision operator (see Chapter 2.3.6), ranks of different combinations of the antecedent A and the consequent B have to be calculated for the correct calculation of the new ranks. The ranks that need to be calculated following from the definition are $\kappa(B|A) = \kappa(AB) - \kappa(A)$, and for the calculation of α $\kappa(AB)$ and $\kappa(A\overline{B})$. Furthermore, for partitioning the models into the corresponding classes, it has to checked whether an interpretation models AB, $A\overline{B}$, or $\overline{A}$.

To calculate the ranks for the different combinations of the antecedent A and the consequent B and to check whether an interpretation models AB, $A\overline{B}$, or $\overline{A}$, the conditional has to be disassembled into the antecedent and consequent.

For determining the antecedent and the consequent of the conditional, the function `findAntAndCons()` is used. It first converts the conditional into a *String* and removes the substrings `'Implies('` and `')'` using the builtin Python function `replace` on the conditional. Afterwards, the *String* is split at the `','` and the resulting propositions are converted back into a *sympy* expression using the `sympify()`

function and stored in `ant` and `cons`.

Afterwards, the required ranks are calculated and stored int he corresponding variables as can be seen in the following table:

$\kappa(AB)$	`rankAB`	`calculateFormulaRank(And(ant, cons))`
$\kappa(A)$	`rankA`	`calculateFormulaRank(ant)`
$\kappa(B\|A)$	`rankCond`	`rankAB - rankA`
$\kappa(A\overline{B})$	`rankANotB`	`calculateFormulaRank(And(ant, Not(cons)))`

The formula passed on to `calculateFormulaRank()` uses the *sympy* library and its built-in logic functions.

In the next step, α is determined by comparing `rankAB` and `rankANotB`. If `rankAB` > `rankA`, then `alpha` is set to be -1, and otherwise it is set to 0.

Finally, the updated ranks are calculated using

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) - \texttt{rankCond} & \text{if } i \models \texttt{And(ant, cons)} \\ \kappa_{old}(i) + \texttt{alpha} + 1 & \text{if } i \models \texttt{And(ant, Not(cons))} \\ \kappa_{old}(i) & \text{if } i \models \texttt{Not(ant)} \end{cases}$$

Inside the outer `if`-clause, the interpretations are at first separated into those that fulfill `ant` and those that do not using the function `checkModel()`. Those that do are then further separated using `checkModel()` into interpretations that fulfill `cons` and those that fulfill `Not(cons)`. Subsequently, the ranks are calculated using the according formula and returned.

4.4.8 Häming-Peters Revision Operator

Just like the Conditional Revision operator in the last chapter, the HP-Revision operator (see Chapter 2.3.7) distinguishes between conditional and propositional information. But unlike the Conditional Revision operator, it defines both the conditional and the propositional operator and does not rely on any other operators. For that reason, both the mental approaches *Principle of Preferred Interpretations* and *Biconditional Interpretation of Conditionals* and the *Mental Models* approach can be applied.

First, the piece of information is adapted according to the applied mental approach.

In the next step, it is evaluated whether the piece of information is a conditional that has to be processed using `calculateBeliefStrengthCond()` or a proposition that has to be processed using `calculateBeliefStrenghtProp()`. Pieces of information are separated into those that contain the substring `'Implies('`, which are conditionals, and those that do not, which are then processed as propositions.

Firstly, the propositional revision operator `calculateBeliefStrenghtProp()` will be presented. First the ranks of the proposition and the negation of the proposition are calculated and stored in corresponding variables as depicted in the following table:

$\kappa(P)$	`rankP`	`calculateFormulaRank(prop)`
$\kappa(\overline{P})$	`rankNotP`	`calculateFormulaRank(Not(prop))`

In the next step, the new ranks are calculated according to the formula below. For the calculation, a strength parameter, `strength`, is passed on to the function. As a default value, `strenght` is set to be 1.

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) & \text{if } \texttt{rankNotP} \leq \texttt{strength} \\ \kappa_{old}(i) - \texttt{rankNotP} & \text{elif } \texttt{checkModel(prop)} \\ \kappa_{old}(i) + \texttt{strength} - \texttt{rankNotP} & \text{elif } \texttt{checkModel(Not(prop))} \end{cases}$$

In the equation, *elif* stands for `rankNotP > strength` and the following condition. Afterwards, the updated ranks are returned and a new piece of information can be processed using the corresponding revision operator.

Secondly, the conditional revision operator `calculateBeliefStrengthCond()` is presented. First, the parameter `d` is calculated, which is used for the calculation of the new ranks. To calculate `d`, the conditional is passed on to the function `calculateD()`. D is defined as $\kappa(A\overline{B}) - \kappa[AB]$. Therefore, as the first step, the antecedent and the consequent are determined using `findAntAndCons()` (see Chapter 4.4.7), and `kAntNotCons` is calculated by finding the rank of the interpretation that fulfills `checkModel(And(ant, Not(cons)))`. Afterwards, `kAntCons` is calculated, which is defined as the maximum rank of the interpretations that fulfill `checkModel(And(ant, cons))`. To calculate `kAntCons`, the function `conditionalOperator()` is called. It works like `calculateMaxRank()` (see Chapter 2.3.3) and returns the corresponding rank. Finally, `d` is calculated as `kAntNotCons - kAntCons`.

In the next step, the conditional is again taken apart into `ant` and `cons` using `findAntAndCons()`. The new ranks are then calculated using the following formula:

$$\kappa_{new}(i) = \begin{cases} \kappa_{old}(i) & \text{if } \texttt{d} \geq \texttt{strength} \\ \kappa_{old}(i) - \texttt{rankAImplB} & \text{elif } \texttt{checkModel(cond)} \\ \kappa_{old}(i) + \texttt{(condAB - rankAImplB} & \\ \quad \texttt{+ strength) - rankANotB} & \text{elif } \texttt{checkModel(Not(cond))} \end{cases}$$

Again, like with the propositional revision operator, *elif* means that `d > strength` and additionally the other condition is true. The different variables are calculated as follows:

`rankAImplB`	$\kappa(A \Rightarrow B)$	`calculateFormulaRank(cond)`
`condAB`	$\kappa[AB]$	`conditionalOperator(And(ant, cons))`
`rankANotB`	$\kappa(A\overline{B})$	`calculateFormulaRank(And(A, Not(B)))`

The updated ranks are then returned and a new piece of information ca be processed using the corresponding revision operator.

4.5 Cognitive Approaches

In this section, the different implementation of the different cognitive approaches is outlined and explained. First, the implementation of *Fully Explicit Models* is presented. Then, *Biconditional Interpretation of Conditionals*, the *Principle of Preferred Interpretations*, and the *Exclusive Interpretation of Disjunctions* are outlined. And lastly, the approach using *Mental Models* is shown.

According to Figure 10, it can be seen that for the different mental approaches, the pipeline of the *Sequential Revision Approach* is slightly altered.

4.5.1 Fully Explicit Models

For the application of Fully Explicit Models, the task is not altered further. The propositions are evaluated using the function `checkModel()` (see Chapter 4.4.2). This approach processes the tasks in a logically correct way, therefore the calculated result is logically correct as well. The corresponding pipeline can be seen in Figure 10a.

4.5.2 Biconditional Interpretation of Conditionals

For the application of the *Biconditional Interpretation of Conditionals* (see Chapter 2.4.4), the proposition of the task is altered so that the conditional is replaced by a biconditional, or an equivalence. That is because humans tend to interpret conditionals as biconditionals. To replace a conditional with an equivalence, the task is passed to `makeBiconditionalInterpretations()`.

The applied pipeline can be seen in Figure 10b. The function `makeBiconditionalInterpretations()` then corresponds to the function f in the pipeline.

The function `makeBiconditionalInterpretations()` work as follows: First, the task is converted int a *String*. If the *String* contains the substring `'Implies'`, the substring is then replaced by `'Equivalent'`, using the builtin *Python* function `replace()` and returned.

4.5.3 Principle of Preferred Interpretations

As presented in Chapter 2.4.5, the *Principle of Preferred Interpretations* states that humans prefer the conjunction of the antecedent and the consequent over other possible models. For that reason, the conditional can be reformulated into the conjunction of the antecedent and the consequent.

To adapt the task, the function `makePreferredInterpretations()` is used and corresponds to the function f in Figure 10b. Its functionality is basically the same as the function `makeBiconditionalInterpretations()` in the last chapter. First, the task is converted into a *String* and searched for the substring `'Implies'`. If the *String* contains the substring, it is replaced by `'And'` using the builtin *Python* function `replace()` and then returned.

4.5.4 Exclusive Disjunctions

Similar to the *Principle of Preferred Interpretations*, the approach of the *Exclusive Disjunctions* represents the fact that humans tend to interpret inclusive disjunctions as exclusive.

To represent that fact, the task is adapted using the function `makeExclusiveDisjunction()`, which corresponds to the function f in Figure 10b. Like the two previous approaches, the function converts the task into a *String*. However, it then searches for the substring `'Or'` and replaces it with `'Xor'`, using the *Python* function `replace()`. Then, the updated task is returned.

4.5.5 Mental Models

Unlike the mental approaches before, the *Mental Model Approach* does not just alter the task, it is a completely different approach of reasoning. The details are presented in Chapter 2.4.1.

For this approach, three deduction rules need to be implemented: Modus Ponens (MP), Acceptance of the Consequent (AC), and Denial of the Antecedent (DA).

As a first step, it needs to be checked whether *Mental Models* can be applied. Therefore, it is checked whether the task contains the substring `'Implies'`. If it does not, the task is passed on to the function `checkModel()` and processed like using *Fully Explicit Models*. If it does, *Mental Models* can be applied and the function `mentalModel()` is used.

The function is passed the current part of the task, the interpretation that is supposed to be checked, and the entire task. That is necessary because for the *Mental Models* approach and the deduction rules, not only the current part of the task is important, but also the prior propositions for applying the rules.

This adjustment is reflected in the adapted *Sequential Revision Approach* in Figure 10c: not only p_i is passed on to the revision operator, but rather all prior parts of the task $[p_1, ..., p_i]$.

The rules are implemented in the following way:

Modus Tollens For all deduction rules, the conditional is first segmented into the antecedent and the consequent using `findAntAndCons()` (see Chapter 4.4.7) and stored in `ant` and `cons`. Furthermore, the minor premise (the part of the task that does only contain one atom) needs to be converted into a *sympy*-expression using `makeExpression()` (see Chapter 4.2) and is then stored in `minor`. For the application of MP, it is necessary that the minor premise `minor` is the same as the antecedent. If this condition is fulfilled, it needs to be checked whether the interpretation fulfills both the antecedent and the consequent. To verify this, the conjunction of the antecedent and the consequent `And(ant, cons)` and the current model are passed onto `checkModel()`. If both conditions are fulfilled, the model and the task can be solved by applying MP and the returned result is `True`, otherwise the returned result is `False`.

Acceptance of the Consequent For AC to be applied, the minor premise needs to correspond to the consequent of the conditional. If the two conform to each other, similarly to MP, it is then checked using `checkModel()` whether the current interpretation fulfills `And(ant, cons)`. If that condition is fulfilled as well, the reviewed model id verified by returning `True`, otherwise `False` is returned.

Denial of the Antecedent DA describes the reasoning process that if the antecedent of the conditional is wrong, the consequent has to be incorrect as well. To model this deduction process, it is first checked whether the minor premise is the negation of the consequent. This can be true in two cases: either, the negation of the consequent is the minor premise, or the negation of the minor premise corresponds to the consequent. If either of the conditions is fulfilled, DA can be applied. As a last step, it needs to be checked whether the interpretation currently viewed fulfills `And(Not(ant), Not(cons))`, that being both the antecedent and the consequent are false. If that condition is fulfilled as well, the returned answer for the considered interpretation is `True`, otherwise it is `False`.

For processing conditional tasks with more than two propositions, the approach must be adapted to incorporate the ranking function before the conditional information (and the answer derived from it) instead of just the presented minor premise.

In the next chapter, the implemented revision operators and mental approaches are evaluated and the most suitable combination of the both is identified.

5 Evaluation

In this section, the combinations of the implemented revision operators and mental approaches are evaluated. In Section 5.1, the experimental dataset is presented and the answers of the different participants are shown. Afterwards, in Section 5.2, the different operators are compared and the best option for replicating human reasoning is chosen.

5.1 Experimental Dataset

The dataset used to evaluate the different revision operators has also bee used in [Ism+23] for the evaluation of the *Sequential Merging Approach*. The experiment was conducted as follows: the participants were given two or three premises and answers in natural language. Afterwards, they were asked which of the given answers follows from the premises. Hereby, three of the response choices are propositions, while the fourth is `none`, which denotes that nothing follows from the premises. The tasks can be seen in Table 14 [Ism+23].

Formally, each recorded task R can be depicted as a tuple

$$R = \{[p_1, p_2], [\varphi_1, \varphi_2, \varphi_3], r\} \text{ with } r \in \{\varphi_1, \varphi_2, \varphi_3, \texttt{none}\}$$

where $[p_1, p_2]$ denote the premises, $[\varphi_1, \varphi_2, \varphi_3]$ denote the offered answers and r denotes the choice of the participant. In the case of the exclusive disjunction, the task has three premises $[p_1, p_2, p_3]$.

The experiment was conducted on Amazon Mechanical Turk with 35 participants who were not trained in classical logic. Each participant was posed each of the 16 tasks twice. The cleaned dataset contains 1097 records and can be accessed at `https://e.feu.de/ecsqaru2023data`.

In Table 15, the answers given by the participants can be seen. It can be observed that in most tasks, the most frequent answer was the correct one. The tasks with

Minor premise	**Major premise(s)**	**Response choices**
a	$a \Rightarrow b$	$\overline{a}, b, \overline{b}$, `none`
b	$a \Leftrightarrow b$	$a, \overline{a}, \overline{b}$, `none`
$\overline{a}$	$(a \vee b) \vee (a \wedge b)$	$a, b, \overline{b}$, `none`
$\overline{b}$	$(a \vee b); \overline{(a \wedge b)}$	$a, \overline{a}, b$, `none`

Table 14: Overview of the different tasks. Each task consists of a minor premise and one or two major premises. Each participant receives sixteen unique tasks, where each task is set twice. The sixteen tasks are a combination of a minor premise and a major premise and the corresponding response choice which does not contain the minor premise [Ism+23].

Minor Premise	Major Premise(s)	Frequency of Response				Sum	Correct
		$\overline{a}$	b	$\overline{b}$	`none`		
a	$a \Rightarrow b$	0	**68**	0	1	69	98.6%
	$a \Leftrightarrow b$	0	**69**	0	1	70	98.6%
	$(a \vee b) \vee (a \wedge b)$	1	15	2	**52**	70	74.3%
	$(a \vee b); \overline{(a \wedge b)}$	1	0	**65**	4	70	92.9%
		a	$\overline{a}$	$\overline{b}$	`none`		
b	$a \Rightarrow b$	$\underline{41}$	0	0	**26**	67	38.8%
	$a \Leftrightarrow b$	**62**	0	0	5	67	92.5%
	$(a \vee b) \vee (a \wedge b)$	18	6	0	**45**	69	65.2%
	$(a \vee b); \overline{(a \wedge b)}$	4	**62**	3	1	70	88.6%
		a	b	$\overline{b}$	`none`		
$\overline{a}$	$a \Rightarrow b$	0	1	$\underline{44}$	**24**	69	34.8%
	$a \Leftrightarrow b$	1	1	**65**	2	69	94.2%
	$(a \vee b) \vee (a \wedge b)$	1	**47**	3	15	66	71.2%
	$(a \vee b); \overline{(a \wedge b)}$	2	**58**	1	5	66	87.9%
		a	$\overline{a}$	b	`none`		
$\overline{b}$	$a \Rightarrow b$	1	**42**	0	27	70	60.0%
	$a \Leftrightarrow b$	0	**56**	1	12	69	81.2%
	$(a \vee b) \vee (a \wedge b)$	**43**	4	0	22	69	62.3%
	$(a \vee b); \overline{(a \wedge b)}$	**60**	1	1	5	67	89.6%

Table 15: Number of the answer choices of the participants. Numbers in bold represent the answer that complies with classical propositional logic, and underlined numbers indicate that the most frequent answer does not coincide with propositional logic.

the most incorrect answers were both conditionals, where the participants wrongly applied AC and DA (see Chapter 2.4.1).

5.2 Evaluation of the Operators

For the evaluation of the different revision operators and mental approaches, each of the seven revision operators was combined with each of the five mental approaches, yielding 32 different combinations (see Table 16).

As explained before in Chapter 4.4.7, the Conditional Revision operator cannot be combined with the mental approaches *Biconditional Interpretation*, *Principle of Preferred Interpretations*, and *Mental Models*, since that would just induce a different revision operator. Therefore, these combinations are not evaluated since they are

$R^{FEM}_{Natural}$	$R^{BI}_{Natural}$	$R^{PoPI}_{Natural}$	$R^{ED}_{Natural}$	$R^{MM}_{Natural}$
$R^{FEM}_{Restrained}$	$R^{BI}_{Restrained}$	$R^{PoPI}_{Restrained}$	$R^{ED}_{Restrained}$	$R^{MM}_{Restrained}$
$R^{FEM}_{Lexicographic}$	$R^{BI}_{Lexicographic}$	$R^{PoPI}_{Lexicographic}$	$R^{ED}_{Lexicographic}$	$R^{MM}_{Lexicographic}$
$R^{FEM}_{Reinforcement}$	$R^{BI}_{Reinforcement}$	$R^{PoPI}_{Reinforcement}$	$R^{ED}_{Reinforcement}$	$R^{MM}_{Reinforcement}$
R^{FEM}_{DP}	R^{BI}_{DP}	R^{PoPI}_{DP}	R^{ED}_{DP}	R^{MM}_{DP}
$R^{FEM}_{Conditional}$	-	-	$R^{ED}_{Conditional}$	-
R^{FEM}_{HP}	R^{BI}_{HP}	R^{PoPI}_{HP}	R^{ED}_{HP}	R^{MM}_{HP}

Table 16: All possible combinations of operators. Each operator consists of the revision operator and the mental approach used.

	Task															
Cognitive	$\overline{a}$				$\overline{b}$				a				b			
Approach	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$
FEM	b	b	$\overline{b}$	none	a	a	$\overline{a}$	$\overline{a}$	none	$\overline{b}$	b	b	none	$\overline{a}$	a	none
MM	b	b	$\overline{b}$	$\overline{b}$	a	a	$\overline{a}$	none	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
BI	b	b	$\overline{b}$	$\overline{b}$	a	a	$\overline{a}$	$\overline{a}$	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
PoPI	b	b	$\overline{b}$	none	a	a	$\overline{a}$	none	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
ED	b	b	$\overline{b}$	none	a	a	$\overline{a}$	$\overline{a}$	$\overline{b}$	$\overline{b}$	b	b	$\overline{a}$	$\overline{a}$	a	a

Table 17: Predicted responses for each task and cognitive approach. Each task consists of a minor premise in the first row and a major premise in the second row. Hereby, the symbols denote the following premises: $\vee$: $(a \vee b) \vee (a \wedge b)$; $\dot{\vee}$: $(a \vee b); \overline{(a \wedge b)}$; $\Leftrightarrow$: $a \Leftrightarrow b$; $\Rightarrow$: $a \Rightarrow b$. The answers of the *Fully Explicit Models* approach are also the logically correct ones.

identical to the revision operator using DP Revision and the corresponding mental approach.

Based on the mental approach applied, each task has a solution that might be different from the logically correct answer. The different expected responses can be seen in Table 17.

For the evaluation of the different operators, first, the performance of the individual operators will be evaluated. Afterwards, the aggregated performance of different operator groups will be discussed.

5.2.1 Individual Performance of the Operators

In Table 18, the predictive performance of each combination of a revision operator and a mental approach can be seen. It can be observed that the *Biconditional Interpretation of Conditionals* combined with each revision operator has the best predictive performance (80.1%) and the combination of the *Conditional Revision* operator and

	FEM	BI	PoPI	ED	MM
Natural	76.9%	80.1%	72.4%	68.8%	78.8%
Restrained	76.9%	80.1%	72.4%	68.8%	78.8%
Lexicographic	76.9%	80.1%	72.4%	68.8%	78.8%
Reinforcement	76.9%	80.1%	72.4%	68.8%	78.8%
Darwiche-Pearl	76.9%	80.1%	72.4%	68.8%	78.8%
Conditional	75.6%	-	-	67.5%	-
Häming-Peters	76.9%	80.1%	72.4%	68.8%	72.7%

Table 18: Predictive performance of the different revision operators in comparison with the answers of the participants

the *Exclusive Disjunctions* has the lowest predictive performance at 67.5%.

Based on the predictive performance of the different operators and mental approaches (see Table 18), the approaches can be split up into different groups that have the same predictive performance. It was verified that the matching performance is because the different operators return the same answer to each task. The groups can be seen in Table 19.

Hereby, **G-** denotes *General*, which indicates that most operators using the corresponding mental approach belong to this group. Furthermore, **C-** and **HP-** denote the Conditional and HP Revision Operators. Finally, the back part denotes the applied mental approach.

Label	Performance	Operators
G-FEM	76.9%	$R^{FEM}_{Natural}$, $R^{FEM}_{Restrained}$, $R^{FEM}_{Lexicographic}$, $R^{FEM}_{Reinforcement}$, R^{FEM}_{DP}, R^{FEM}_{HP}
G-BI	80.1%	$R^{BI}_{Natural}$, $R^{BI}_{Restrained}$, $R^{BI}_{Lexicographic}$, $R^{BI}_{Reinforcement}$, R^{BI}_{DP}, R^{BI}_{HP}
G-PoPI	72.4%	$R^{PoPI}_{Natural}$, $R^{PoPI}_{Restrained}$, $R^{PoPI}_{Lexicographic}$, $R^{PoPI}_{Reinforcement}$, R^{PoPI}_{DP}, R^{PoPI}_{HP}
G-ED	68.5%	$R^{ED}_{Natural}$, $R^{ED}_{Restrained}$, $R^{ED}_{Lexicographic}$, $R^{ED}_{Reinforcement}$, R^{ED}_{DP}, R^{ED}_{HP}
G-MM	78.8%	$R^{MM}_{Natural}$, $R^{MM}_{Restrained}$, $R^{MM}_{Lexicographic}$, $R^{MM}_{Reinforcement}$, R^{MM}_{DP}
C-FEM	75.6%	$R^{FEM}_{Conditional}$
C-ED	67.5%	$R^{ED}_{Conditional}$
HP-MM	72.2%	R^{MM}_{HP}

Table 19: Partition of the different combinations of revision operators and mental approaches.

5.2.2 Aggregated Performance of the Operator Groups

In Table 20, the performance of each operator group for each task group is calculated. It can be seen that the operators of the **G-BI** operator group perform the best in all task groups with an overall accuracy of 80.1%. The operator with the lowest overall accuracy is $R^{ED}_{Conditional}$ with an accuracy of 67.5%. For the Conditional tasks $\Rightarrow$, the operator group **G-PoPI** has the lowest accuracy of 40%, followed by R^{MM}_{HP} with an accuracy of 41.1%. For the Inclusive Disjunction $\vee$, it can be seen that all approaches not using *Exclusive Disjunctions* have an accuracy of 68.2%, while approaches using it have a much lower accuracy at 35.8%. Furthermore, all approaches have the same accuracy of 91.6% for the task group Biconditionals $\Leftrightarrow$ and 89.7% for the task group Exclusive Disjunctions $\dot{\vee}$.

In Table 21, the calculated answers for each task and each revision operator (group) are presented. The calculated answers which differ from the predicted ones are underlined.

It can be observed that the answers using the *Principle of Preferred Interpretations* do not conform with the predicted answers, if the minor premise is a negated proposition. If the calculated ranks for each step are analysed, it can be seen why the wrong answer is chosen. For example, the ranks for the task $[\bar{a}, a \Rightarrow b]$ are calculated as follows:

Operator Group	Number of Correct Predictions and Accuracy for Task Groups									
	All Tasks $n = 1097$		Conditional $n = 275$		Biconditional $n = 275$		Incl. Disjunct. $n = 274$		Excl. Disjunct. $n = 273$	
	CP	Accuracy	CP	Accuracy	CP	Accuracy	CP	Accuracy	CP	Accuracy
G-FEM	844	76.9%	160	58.2%	252	91.6%	**187**	**68.2%**	245	89.7%
G-BI	**879**	**80.1%**	**195**	**70.9%**	252	91.6%	**187**	**68.2%**	245	89.7%
G-PoPI	794	72.4%	<u>110</u>	<u>40.0%</u>	252	91.6%	**187**	**68.2%**	245	89.7%
G-ED	755	68.8%	160	58.2%	252	91.6%	<u>98</u>	<u>35.8%</u>	245	89.7%
G-MM	864	78.8%	180	65.5%	252	91.6%	**187**	**68.2%**	245	89.7%
C-FEM	829	75.6%	145	52.7%	252	91.6%	**187**	**68.2%**	245	89.7%
C-ED	<u>740</u>	<u>67.5%</u>	145	52.7%	252	91.6%	<u>98</u>	<u>35.8%</u>	245	89.7%
HP-MM	797	72.7%	113	41.1%	252	91.6%	**187**	**68.2%**	245	89.7%

Table 20: Number of correct predictions (CP) and accuracy of the operator groups (see Table 19). Numbers in bold denote the operator (group) with the highest accuracy and underlined numbers denote the operator (group) with the lowest accuracy for the task groups where different operator groups yield different performances.

	`[0, 0, 0, 0]`
`Not(A)`	`[0, 0, 1, 1]`
`Implies(A, B)`	`[1, 1, 2, 0]`

It can be seen that in the last step, because of the applied *Principle of Preferred Interpretations*, $A \Rightarrow B$ is interpreted to be $A \wedge B$. However, since $\neg A$ and $A \wedge B$ cannot be fulfilled at the same time, only the interpretation AB, which is in contradiction to $\neg A$ but fulfills $A \wedge B$, is a model of the last part of the task and therefore moved down to rank 0. Finally, the answer *b* is chosen, which is incorrect both logically and in comparison to the expected answer. This difference between the expected and calculated answer also additionally explains why the **G-PoPI** operator group has the lowest accuracy for the task group of conditionals. The same error also occurs for the task $[\neg B, A \Rightarrow B]$, where the calculated answer is a.

Furthermore, it can be observed that all further answers that differ from the expected ones occur at the task $[\neg B, A \Rightarrow B]$, for the operator (groups) **C-FEM**, **C-ED**, and **HP-MM**. For **C-FEM** and **C-ED**, Modus Tollens was not applied. In addition, for **HP-MM**, MP was wrongly applied, although it should not have been.

As a last part of the evaluation, the predictive performance of the operator groups is evaluated for each participant. In Table 22, the predictive performance of each operator group is depicted for each participant.

It can be observed, that for most participants, **G-BI** is the most accurate (19 out

	Task															
Operator	$\overline{a}$				$\overline{b}$				a				b			
Group	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$	$\vee$	$\dot{\vee}$	$\Leftrightarrow$	$\Rightarrow$
G-FEM	b	b	$\overline{b}$	none	a	a	$\overline{a}$	$\overline{a}$	none	$\overline{b}$	b	b	none	$\overline{a}$	a	none
G-BI	b	b	$\overline{b}$	$\overline{b}$	a	a	$\overline{a}$	$\overline{a}$	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
G-PoPI	b	b	$\overline{b}$	$\underline{b}$	a	a	$\overline{a}$	$\underline{a}$	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
G-ED	b	b	$\overline{b}$	none	a	a	$\overline{a}$	$\overline{a}$	$\overline{b}$	$\overline{b}$	b	b	$\overline{a}$	$\overline{a}$	a	none
G-MM	b	b	$\overline{b}$	$\overline{b}$	a	a	$\overline{a}$	none	none	$\overline{b}$	b	b	none	$\overline{a}$	a	a
C-FEM	b	b	$\overline{b}$	none	a	a	$\overline{a}$	<u>none</u>	none	$\overline{b}$	b	b	none	$\overline{a}$	a	none
C-ED	b	b	$\overline{b}$	none	a	a	$\overline{a}$	<u>none</u>	$\overline{b}$	$\overline{b}$	b	b	$\overline{a}$	$\overline{a}$	a	none
HP-MM	b	b	$\overline{b}$	$\overline{b}$	a	a	$\overline{a}$	$\underline{\overline{a}}$	none	$\overline{b}$	b	none	none	$\overline{a}$	a	none

Table 21: Answers calculated by the different revision operator (groups) for all tasks. Answers that differ from the expected ones (see Table 17) are underlined. Each task consists of a minor premise in the first row and a major premise in the second row. Hereby, the symbols denote the following premises: $\vee$: $(a \vee b) \vee (a \wedge b)$; $\dot{\vee}$: $(a \vee b); \overline{(a \wedge b)}$; $\Leftrightarrow$: $a \Leftrightarrow b$; $\Rightarrow$: $a \Rightarrow b$.

of 35 participants). Furthermore, **G-PoPI**, **C-ED**, and **HP-MM** are never the most accurate. **C-FEM** was the most accurate operator group for 11 participants, **G-MM** for 8 participants, **G-FEM** for 6 participants, and **G-ED** for only one.

It is also notable that for P_{17}, **C-FEM** had an accuracy of 100%.

Participant	**G-FEM**	**G-BI**	**G-PoPI**	**G-ED**	**G-MM**	**C-FEM**	**C-ED**	**HP-MM**
P_1	78.5%	**85.7%**	75.0%	67.9%	**85.7%**	78.6%	67.9%	78.6%
P_2	62.5%	**68.8%**	62.5%	56.3%	**68.8%**	62.5%	56.3%	56.3%
P_3	78.1%	**90.6%**	78.1%	68.8%	84.4%	71.9%	62.5%	78.1%
P_4	48.4%	51.6%	54.8%	45.2%	**58.1%**	54.8%	51.6%	38.7%
P_5	**84.4%**	71.9%	68.8%	75.0%	71.9%	**84.4%**	75.0%	71.9%
P_6	86.7%	**90.0%**	83.3%	73.3%	83.3%	80.0%	66.7%	76.7%
P_7	58.1%	**61.3%**	51.6%	58.1%	54.8%	51.6%	51.6%	51.6%
P_8	74.2%	**87.1%**	74.2%	74.2%	80.6%	67.7%	67.7%	74.2%
P_9	68.8%	68.8%	56.3%	**71.9%**	62.5%	62.5%	65.6%	68.8%
P_{10}	67.7%	80.6%	74.2%	54.8%	**87.1%**	74.2%	61.3%	67.7%
P_{11}	74.2%	80.6%	74.2%	61.3%	**87.1%**	80.6%	67.7%	74.2%
P_{12}	71.9%	**84.4%**	75.0%	68.8%	81.3%	68.8%	65.6%	71.9%
P_{13}	**93.8%**	81.3%	78.1%	81.3%	81.3%	**93.8%**	81.3%	87.5%
P_{14}	71.0%	80.6%	74.2%	58.1%	**87.1%**	77.4%	64.5%	71.0%
P_{15}	90.6%	78.1%	78.1%	78.1%	84.4%	**96.9%**	84.4%	78.1%
P_{16}	90.6%	**96.9%**	87.5%	78.1%	90.6%	84.4%	71.9%	84.4%
P_{17}	93.5%	80.6%	80.6%	80.6%	87.1%	**100%**	87.1%	80.6%
P_{18}	78.1%	71.9%	71.9%	71.9%	78.1%	**84.4%**	78.1%	65.6%
P_{19}	**96.7%**	83.3%	76.7%	83.3%	76.7%	90.0%	76.7%	83.3%
P_{20}	78.1%	**90.6%**	78.1%	75.0%	84.4%	71.9%	68.8%	78.1%
P_{21}	65.6%	**78.1%**	65.6%	62.5%	71.9%	59.4%	56.3%	65.6%
P_{22}	68.8%	68.8%	62.5%	59.4%	**75.0%**	**75.0%**	65.6%	68.8%
P_{23}	68.8%	**81.3%**	68.8%	68.8%	75.0%	62.5%	62.5%	68.8%
P_{24}	84.4%	78.1%	75.0%	71.9%	84.4%	**90.6%**	78.1%	78.1%
P_{25}	**75.9%**	62.1%	58.6%	69.0%	62.1%	**75.9%**	69.0%	62.1%
P_{26}	65.5%	**79.3%**	65.5%	72.4%	72.4%	58.6%	65.5%	69.0%
P_{27}	77.4%	**87.1%**	74.2%	77.4%	80.6%	71.0%	71.0%	77.4%
P_{28}	74.2%	**87.1%**	74.2%	64.5%	80.6%	67.7%	58.1%	74.2%
P_{29}	71.9%	**84.4%**	71.9%	59.4%	78.1%	65.6%	53.1%	71.9%
P_{30}	**87.5%**	81.3%	75.0%	75.0%	81.3%	**87.5%**	75.0%	81.3%
P_{31}	**68.8%**	**68.8%**	62.5%	56.3%	**68.8%**	**68.8%**	56.3%	62.5%
P_{32}	84.4%	**96.9%**	84.4%	71.9%	90.6%	78.1%	65.6%	84.4%
P_{33}	81.3%	**93.8%**	81.3%	68.8%	87.5%	75.0%	62.5%	81.3%
P_{34}	84.4%	**96.9%**	84.4%	75.0%	90.6%	78.1%	68.8%	84.4%
P_{35}	87.5%	75.0%	75.0%	75.0%	81.3%	**93.8%**	81.3%	75.0%

Table 22: Performance of the operator groups for each participant. The operator group with the highest accuracy for each participant is printed in bold.

5.3 Summary

To conclude, the following observations can be made: Firstly, humans tend to apply AC and DA, even though they are logical fallacies. Therefore, both the tasks $[\overline{a}, a \Rightarrow b]$ and $[b, a \Rightarrow b]$ are the tasks that have mostly not been answered correctly by the participants, with an accuracy of only 34.8% and 38.8% respectively. Furthermore, it can be observed that for each minor premise, the corresponding conditional had the lowest number of correct answers, compared to the other three tasks with the same minor premise. This phenomenon already led to the assumption that the most accurate operator could be using the mental approach *Biconditional Interpretation of Conditionals* or *Mental Models*, since in both approaches AC and DA are endorsed.

Secondly, the overall accuracy for all possible combinations of the operators was calculated and it could be deduced that operators with the same accuracy yield identical answers for each task. Therefore, operators with the same accuracy can be grouped and the groups can be compared to each other. Confirming the assumptions from before, operators using the mental approaches *Biconditional Interpretation of Conditionals* and *Mental Models* had the highest predictive performance with 80.1% and 78.8% respectively.

Thirdly, the different operator groups were compared to each other and it could be observed that for the task group of Conditionals $\Rightarrow$, approaches of the operator group **G-BB** had a higher predictive performance than those of the group **G-MM** with accuracies of 70.9% and 65.5% respectively. Hence, it can be concluded that using the *Biconditional Interpretation of Conditionals* is more accurate and therefore should be applied in the chosen operator for modeling human reasoning. Furthermore, it could be observed that for the task groups Biconditional $\Leftrightarrow$ and Exclusive Disjunctions $\dot{\vee}$, all operator groups had the same predictive performance. Furthermore, it can be seen that the approaches using *Exclusive Disjunctions* had a much lower predictive performance than approaches not using it, with accuracies of 68.2% and 35.8% respectively.

Fourthly, the calculated answers of the different operators were compared to the predicted answers. Hereby, it could be observed that for the *Principle of Preferred Interpretations*, the operator yielded an answer that is both logically incorrect and different from the predicted one, for the tasks $[\overline{a}, a \Rightarrow b]$ and $[\overline{b}, a \Rightarrow b]$. For this reason, operators using the *Principle of Preferred Interpretations* are not suitable as a way of replicating human reasoning. Furthermore, the operator groups **C-FEM** and **C-ED**, that both use the Conditional Revision operator, have deviating answers from the other operators using *Exclusive Disjunctions* and *Fully Explicit Models* and additionally have a lower predictive performance than the corresponding operators of the groups **G-FEM** and **G-ED**. Therefore, the Conditional Revision operator is not a suitable operator for replicating human reasoning, either. Lastly, the **HP-MM** revision operator, R_{HP}^{MM}, yields a answer different from the predicted answer for the task $[\overline{b}, a \Rightarrow b]$ as well. It here fails to not apply Modus Tollens and therefore does not fully fulfill the *Mental Models* approach. Furthermore, its predictive performance is lower than that of the other operators using *Mental Models*, with an accuracy of

72.7% compared to 78.8%. Therefore, it can be ruled out as the best revision operator for replicating human reasoning.

Fifthly, it was evaluated which operator group had the highest predictive performance for each participant. The operator group which yielded the highest accuracy for the most participants was the **G-BI** operator group, which was most accurate for 16 out of the 35 participants. The operator group with the highest accuracy for the second most people was **C-FEM**, with the most accurate prediction for 11 participants.

To summarise, it can be evaluated that the operator group that is most suitable for replicating human propositional reasoning is **G-BI**. It has the highest overall predictive performance and also yields the highest accuracy for 16 out of 35 participants. A second choice would be the **G-MM** operator group, since it has the second highest predictive performance and yields the most correct answers for 8 out of 35 participants. Furthermore, the approaches of *Principle of Preferred Interpretations* and *Exclusive Disjunctions* are not suitable.

In the next chapter, the implemented approaches will be further analysed in detail and it will be explained why the operator groups contain so many operators that calculate the same answers. Furthermore, the *Sequential Revision Approach* will be compared to the *Sequential Merging Approach* by Ismail-Tsaous in [Ism+23].

6 Discussion

In this chapter, the implemented operators and the answers they calculate will be further examined and it will be explained why many operators yield exactly the same responses. Afterwards, the main differences compared to the *Sequential Merging Approach* by Ismail-Tsaous [Ism+23] will be highlighted.

6.1 Explanation of the Ranks

In this section, it will be explained why a number of different revision operators calculate the same response for the same task. Furthermore, it is shown that the operators can yield different answers for other tasks.

In Chapter 5.2, we have seen that a number of revision operators calculated the same answer for all tasks.

In the following, an detailed analyse of this phenomenon will be given. Table 23 provides the revision results for all operators $R^{FEM}_{Natural}$, ... , R^{FEM}_{HP}. As one can easily see, all operators yield $\overline{A}\,\overline{B}$ on the lowest rank for the task `[Not(B), Implies(A, B)]`. As reasoning only depends on the lowest rank, all operators yield the same result. However, one can see that the ranks of all other interpretations (not in the lowest rank) is different. Thus, when performing an extra revision step by A, many operators yield different results.

As can be seen in the third row of Table 23, all revision operators from the operator group **G-FEM** yield the same interpretation $\overline{A}\,\overline{B}$, and therefore the answer `Not(A)`. However, it can also be observed that for $R^{FEM}_{Reinforcement}$ and R^{FEM}_{DP}, the order between the ranks is different from the other operators.

This leads to the phenomenon that when a proposition is added to the task, so that the propositions contradict each other, different answers are calculated. In Table 23, this is shown with an example: the task `[Not(B), Implies(A, B)]` is expanded by the proposition `A`, which is in direct contradiction with the original task. As a con-

Task	$R^{FEM}_{Natural}$	$R^{FEM}_{Restrained}$	$R^{FEM}_{Lexicographic}$	$R^{FEM}_{Reinforcement}$	R^{FEM}_{DP}	R^{FEM}_{HP}
	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$
`Not(B)`	$[0,1,0,1]$	$[0,2,0,2]$	$[0,1,0,1]$	$[0,2,0,2]$	$[0,1,0,1]$	$[0,1,0,1]$
`Implies(A, B)`	$[0,2,1,2]$	$[0,5,2,5]$	$[0,1,2,1]$	$[0,2,2,2]$	$[0,1,1,1]$	$[0,1,2,1]$
`A`	$[1,3,0,3]$	$[2,12,0,11]$	$[2,3,1,0]$	$[\mathbf{2},\mathbf{4},\mathbf{0},\mathbf{0}]$	$[\mathbf{1},\mathbf{2},\mathbf{0},\mathbf{0}]$	$[1,2,1,0]$
Answer	$a\overline{b}$	$a\overline{b}$	$a\overline{b}$	a	a	$a\overline{b}$

Table 23: Ranks of the different revision operators for the task $[\overline{a}, a \Rightarrow b]$ for the operator group **G-FEM**. In the second last row, the task is expanded by the proposition a. Ranks is bold denote operators that yield a different answer. In the last row, the deduced answer is displayed.

Task	$R^{FEM}_{Natural}$	$R^{FEM}_{Restrained}$	$R^{FEM}_{Lexicographic}$	$R^{FEM}_{Reinforcement}$	R^{FEM}_{DP}	R^{FEM}_{HP}
	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$	$[0,0,0,0]$
`B`	$[1,0,1,0]$	$[2,0,2,0]$	$[1,0,1,0]$	$[2,0,2,0]$	$[1,0,1,0]$	$[1,0,1,0]$
`Equivalent(A, B)`	$[2,1,2,0]$	$[5,2,6,0]$	$[1,2,3,0]$	$[2,2,4,0]$	$[1,1,2,0]$	$[1,1,2,0]$
`Not(A)`	$[3,0,3,1]$	$[11,0,14,2]$	$[0,1,5,2]$	$[0,0,6,2]$	$[0,0,3,1]$	$[0,0,3,1]$
Answer	$\overline{a}b$	$\overline{a}b$	$\overline{\overline{a}b}$	$\overline{a}$	$\overline{a}$	$\overline{a}$

Table 24: Ranks of the different revision operators for the task $[b, a \Leftrightarrow b]$ for the operator group **G-FEM**. In the second last row, the task is expanded by the proposition $\overline{a}$. In the last row, the deduced answer is displayed.

sequence, $R^{FEM}_{Reinforcement}$ and R^{FEM}_{DP} yield A as an answer, while the other operators return $A\overline{B}$. Therefore, the calculated answers are `A` and `[A, Not(B)]` respectively.

As a result, different beliefs are retained: while with $R^{FEM}_{Reinforcement}$ and R^{FEM}_{DP} both `A` and `Implies(A, B)` are believed, when using the other operators, `A` and `Not(B)` are believed.

One possible example from real life for this could be the following:

> You know that all animals that lay eggs are birds ($eggs \Rightarrow bird$). Furthermore, you see a platypus, which is not a bird ($\neg bird$). Therefore, you could follow that the animal should not lay eggs ($\neg eggs$).
>
> However, you then see that the platypus does indeed lay eggs ($eggs$).
>
> Do you still belief that the platypus is not a bird or that only birds lay eggs?

In the cases of $R^{FEM}_{Natural}$, $R^{FEM}_{Restrained}$, $R^{FEM}_{Lexicographic}$, and R^{FEM}_{HP}, the conditional $a \Rightarrow b$ is not believed anymore. This can be seen since the calculated answer $a\overline{b}$ contradicts the conditional $a \Rightarrow b$. Applied to the example, this means it is believed only that the platypus is not a bird and that it lays eggs, however the proposition that all animals that lay eggs is not believed anymore.

In the cases of $R^{FEM}_{Reinforcement}$ and R^{FEM}_{DP} however, only the last proposition a is believed, since both ab and $a\overline{b}$ are believed with the same strength [Ker01]. In regard to the example, this would mean that the agent only believes that the animal lays eggs, but not that it is not a bird or that all animals that lay eggs are birds.

Another example for this effect is shown in Table 24.

This leads to the conclusion that in cases that contain logical contradictions, different operators within the same operator group (see Chapter 5.2) may yield different answers for the same task. Therefore, when examining tasks with contradictions in

$$
\begin{tikzcd}
\kappa_0 \arrow[r, "\Delta"] & \kappa_1 = \Delta([\kappa_0, \kappa[p_1]]) & \cdots & \kappa_{n-1} \arrow[r, "\Delta"] & \kappa_n = \Delta([\kappa_{n-1}, \kappa[p_n]]) \\
\kappa[p_1] = C(p_1) \arrow[u] & p_1 \arrow[l] & \cdots & \kappa[p_n] = C(p_n) \arrow[u] & p_n \arrow[l]
\end{tikzcd}
$$

Figure 25: Pipeline of the *Sequential Merging Approach* proposed by Ismail-Tsaous in [Ism+23].

them, each operator needs to be evaluated separately, since they may return different answers. Furthermore, new operator groups may be identified.

6.2 Comparison with the Sequential Merging Approach

In this chapter, the implemented *Sequential Revision Approach* is compared to the *Sequential Merging Approach* by Ismail-Tsaous [Ism+23] and the main advantages and disadvantaged of each approach are discussed.

The pipeline of the *Sequential Merging Approach* by Ismail-Tsaous [Ism+23] can be seen in Figure 25. The approach itself relies on the same basic assumptions as the *Sequential Revision Approach*: first, the epistemic state of an agent can be represented by a ranking function κ, second, the information and beliefs of an agent are modelled by the underlying logic $\mathcal{L}$ and third, agents process new information sequentially [Ism+23].

In simple terms, the *Sequential Merging Approach* works as follows: First, all possible interpretations are assigned the rank 0, which results in the initial ranking function κ_0. For each new piece of information p_i, a so-called ranking construction function $C(p_i)$ is constructed and then merged with the prior ranking function using different merging constructions to calculate the new ranking function $\Delta([\kappa_{i-1}, \kappa[p_i]])$. [Ism+23]

The main difference to the presented *Sequential Revision Approach* is the following: for each new piece of information, a separate ranking construction function is calculated. Therefore, it is possible to assess each interpretation according to the new piece of information and to also grade the interpretations differently instead of just evaluating as true or false.

This leads to a more nuanced evaluation of the information. An example will be shown for the *Principle of Preferred Interpretations* for conditional statements [Ism+23]:

$$
C_{PoPI}(p)(\omega) = \begin{cases} \texttt{impl} & \text{if } \omega \not\models p \\ 2 & \text{if } \omega \models p \text{ and } \omega \models \neg a \wedge b \text{ and } p = a \Rightarrow b \\ 1 & \text{if } \omega \models p \text{ and } \omega \models \neg a \wedge \neg b \\ 0 & \text{otherwise} \end{cases}
$$

When $\omega \not\models p$ holds, all non-models of the proposition p are mapped to the rank `impl`, which means they are not in any case considered to be valid answers. In the second case, when $\omega \models p$, $\omega \models \neg a \wedge b$, and $p = a \Rightarrow b$ hold, the interpretation $\overline{a}b$ is mapped to the rank 2, which describe that agents usually consider this model to be the least plausible one. In the third case, when $\omega \models p$ and $\omega \models \neg a \wedge \neg b$ hold, $\overline{ab}$ is mapped to rank 1, since it is considered to be a more plausible model by the agent. In the last case, the last remaining model, ab, is the mapped to the rank 0 and therefore considered to be the most plausible one.

This then potentially makes a difference in the next merging step: an answer that is only seen to be less likely, but not completely wrong, has a higher potential of being considered the correct answer in the following steps of the task, whereas in the *Sequential Revision Approach*, a less plausible interpretation is just considered as false.

However, even though the *Sequential Merging Approach* seems to be more refined than the *Sequential Revision Approach*, the sequential merging operator with the highest predictive performance had an aggregated accuracy of 80.1% for the operator group *PoPI* [Ism+23], just like the *Sequential Revision Approach*. Furthermore, just like with the *Sequential Revision Approach*, some operators in the group with the most accurate operators made use of the *Biconditional Interpretation of Conditionals*, but also the *Principle of Preferred Interpretations*. Since the *Principle of Preferred Interpretations* yielded answers deviating from the predicted ones, it could not be chosen as the mot suitable operator.

Therefore, it can be concluded that in both cases, the mental approach of the *Biconditional Interpretation of Conditionals* is the most accurate approach to replicate human reasoning, as long as the propositions do not contradict each other. However, in the case of contradicting statements, further research needs to be conducted. Like evaluated for the *Sequential Revision Approach* in Chapter 5, calculated responses might differ for contradicting information and different operators might be more suitable.

7 Conclusion

In this thesis, the *Sequential Revision Approach* and its implementation were presented as a means of replicating the human reasoning process.

In Chapter 2, the necessary theoretical background was presented. Afterwards, in Chapter 3, the realisation of the *Sequential Revision Approach* was outlined and the incorporation of different mental approaches was described. Subsequently, in Chapter 4, the implementation of the approach was described in detail and the processing of given tasks was mapped out. In Chapter 5, the evaluation of operator groups was conducted and the results were discussed. In Chapter 6, the approach was examined further and compared to the *Sequential Merging Approach* presented in [Ism+23].

It was shown that with a predictive performance of 80.1% aggregated for all reasoners, the operator group **G-BI** contained the operators with the highest accuracy. Compared to an accuracy of only 76.9% for the logically correct answers, this also leads to the conclusion that human reasoning does indeed not follow classical logical rules, especially not for conditional statements. More precisely, humans tend to prefer the biconditional interpretation of conditionals over the logically correct interpretation.

As explained in Chapter 2.4.4, this can be the cause for one of three reasons: either, humans just interpret conditionals as biconditionals, as a lack of correct understanding of propositional logic, or they just tend to forget models of the conditional, or they just have a preference for giving a concrete answer instead of answering that nothing can be concluded from the given task.

Furthermore, it was found out that for pieces of information that contradict each other, the operators in the operator group **G-BI** did indeed yield different answers. Therefore, if the approach was to be extended to tasks with contradicting information, the operators from the different operator groups need to be evaluated individually and potentially separated into new operator groups.

Lastly, it was evaluated that the *Sequential Merging Approach*, even though it seems to be more sophisticated and nuanced than the *Sequential Revision Approach*, does not yield a better predictive performance, at least not for tasks with information that does not contradict each other. However, for contradicting information, both approaches would need to be reevaluated.

The *Sequential Revision Approach* presented in this thesis is only a first step on the way to replicating human reasoning. In the future, some possible options to expand the approach are the following:

Expansion for multiple conditional statements In an experiment conducted by Byrne [Byr89], it was found out that in the case of multiple conditionals $[a \Rightarrow b, b \Rightarrow c]$, participants failed to draw the conclusion c if only the premise a was given. However, if both a and b were given, participants succeeded in deducing the correct answer c. To incorporate this phenomenon in the approach, mechanisms would need to be put in place, that verify whether both, and not just one, premise is fulfilled.

Expansion for general knowledge Another phenomenon discovered by Byrne is that in the case of additional implicit information about the participants surroundings, such as promises or events, humans tend to make wrong conclusions [Byr89]. The same phenomenon is present in the case that general knowledge needs to be applied [VSd05]. To incorporate both of these additional sources of knowledge, general knowledge and specific knowledge about a situation, into the process of making decisions, there has to be introduced a way to store universal information or specific information about the current situations, and to take them into account when making inferences.

Extension for reasoning with possibilities This extension features three parts: first, humans tend to condense knowledge of two possibilities into a conjunction, such as *"A is possible and B is possible"* is condensed to *"A and B are possible"*, which might not necessarily be the case. Second, epistemic probabilities, which are synonymous to non-numerical probabilities, e.g. how likely an event is to occur, and lastly, that the statement *"A is possible"* is mostly equated to *"A is not true at the moment"*, until A is explicitly confirmed [RJ20]. To incorporate reasoning with possibilities, an additional measure would be necessary in the revision, which takes into account the possibilities.

Incorporation of indirect knowledge Similar to the concept of general knowledge, indirect knowledge is knowledge about the surroundings and inner workings about a specific person. This information can be applied to make an initial ranking function, which maps possible interpretations to different ranks [GP96]. Therefore, the initial ranking function would be different to the uniform ranks in the regular approach, to describe the bias of the agent. To incorporate this into the *Sequential Revision Approach*, a tool would be needed to depict this initial bias of the participant.

All in all, it can be recorded that the implemented *Sequential Revision Approach* is just the first step to replicate human reasoning, however, it represents a promising foundation that can be enhanced to incorporate more elements that might be important for human decision making.

References

[AGM16] Carlos E Alchourrón, Peter Gärdenfors, and David Makinson. „On the logic of theory change: partial meet contraction and revision functions". In: *Readings in Formal Epistemology: Sourcebook* (2016), pp. 195–217.

[BGL00] Pierre Barrouillet, Nelly Grosset, and Jean-François Lecas. „Conditional reasoning by mental models: Chronometric and developmental evidence". In: *Cognition* 75.3 (2000), pp. 237–266.

[BM06] R. Booth and T. Meyer. „Admissible and Restrained Revision". In: *Journal of Artificial Intelligence Research* 26 (2006), pp. 127–151. DOI: `10.1613/jair.1874`.

[Bou96] Craig Boutilier. „Iterated revision and minimal change of conditional beliefs". In: *Journal of Philosophical Logic* 25 (1996), pp. 263–305.

[Byr89] Ruth MJ Byrne. „Suppressing valid inferences with conditionals". In: *Cognition* 31.1 (1989), pp. 61–83.

[DC17] Paul Christian Dawkins and John Paul Cook. „Guiding reinvention of conventional tools of mathematical logic: Students' reasoning about mathematical disjunctions". In: *Educational Studies in Mathematics* 94 (2017), pp. 241–256.

[DP97] Adnan Darwiche and Judea Pearl. „On the logic of iterated belief revision". In: *Artificial intelligence* 89.1-2 (1997), pp. 1–29.

[Fin37] B. de Finetti. „La prévision, ses lois logiques et ses sources subjectives". In: *Ann. Inst. H. Poincaré*. Vol. 7. English translation in *Studies in Subjective Probability*, ed. H. Kyburg and H.E. Smokler, 1964, 93-158. New York: Wiley. 1937.

[FPC23] Andrew Fuchs, Andrea Passarella, and Marco Conti. „Modeling, replicating, and predicting human behavior: A survey". In: *ACM Transactions on Autonomous and Adaptive Systems* 18.2 (2023), pp. 1–47.

[Gey+22] Werner Geyer et al. „What is human-centered AI?" In: *IBM Research* (Aug. 2022).

[GP96] Moisés Goldszmidt and Judea Pearl. „Qualitative probabilities for default reasoning, belief revision, and causal modeling". In: *Artificial Intelligence* 84.1-2 (1996), pp. 57–112.

[Haa23] Katherine Haan. „24 Top AI Statistics And Trends In 2024". In: *Forbes Advisor* (Apr. 2023).

[HP11] Klaus Häming and Gabriele Peters. „Improved revision of ranking functions for the generalization of belief in the context of unobserved variables". In: *International Conference on Neural Computation Theory and Applications*. Vol. 2. SCITEPRESS. 2011, pp. 118–123.

[HP12] Klaus Haming and Gabriele Peters. „Iterated revisions in large and noisy state spaces using ranking functions“. In: *Engineering Intelligent Systems* 20.1-2 (2012), pp. 85–93.

[Ism+23] Eda Ismail-Tsaous et al. „On the Cognitive Logic of Human Propositional Reasoning: Merging Ranking Functions“. In: *European Conference on Symbolic and Quantitative Approaches with Uncertainty*. Springer. 2023, pp. 224–238.

[JB02] Philip N Johnson-Laird and Ruth MJ Byrne. „Conditionals: a theory of meaning, pragmatics, and inference.“ In: *Psychological review* 109.4 (2002), p. 646.

[JBS92] Philip N Johnson-Laird, Ruth M Byrne, and Walter Schaeken. „Propositional reasoning by model.“ In: *Psychological review* 99.3 (1992), p. 418.

[Joh13] Philip N Johnson-Laird. *Human and machine thinking*. Psychology Press, 2013.

[JT07] Yi Jin and Michael Thielscher. „Iterated belief revision, revised“. In: *Artificial Intelligence* 171.1 (2007), pp. 1–18.

[KBJ18] Sangeet S Khemlani, Ruth MJ Byrne, and Philip N Johnson-Laird. „Facts and possibilities: A model-based theory of sentential reasoning“. In: *Cognitive science* 42.6 (2018), pp. 1887–1924.

[Ker01] Gabriele Kern-Isberner. *Conditionals in nonmonotonic reasoning and belief revision: considering conditionals as agents*. Springer, 2001.

[Ker99] Gabriele Kern-Isberner. „Postulates for conditional belief revision“. In: *IJCAI*. Vol. 99. Citeseer. 1999, pp. 186–191.

[KM91] Hirofumi Katsuno and Alberto O. Mendelzon. „Propositional knowledge base revision and minimal change“. In: *Artificial Intelligence* 52.3 (1991). PII: 000437029190069V, pp. 263–294. ISSN: 00043702. DOI: 10.1016/0004-3702(91)90069-V.

[Lib15] Paolo Liberatore. „Revision by history“. In: *Journal of Artificial Intelligence Research* 52 (2015), pp. 287–329.

[Nay94] Abhaya C Nayak. „Iterated belief change based on epistemic entrenchment“. In: *Erkenntnis* 41.3 (1994), pp. 353–390.

[NGC84] Stephen E Newstead, Richard A Griggs, and Jay J Chrostowski. „Reasoning with realistic disjunctives“. In: *The Quarterly Journal of Experimental Psychology* 36.4 (1984), pp. 611–627.

[Obe06] Klaus Oberauer. „Reasoning with conditionals: A test of formal models of four theories“. In: *Cognitive Psychology* 53.3 (2006), pp. 238–283.

[pan24] pandas. *pandas - Python Data Analysis Library*. https://pandas.pydata.org. [Online; accessed 17. Apr. 2024]. Apr. 2024.

[Pap01] Odile Papini. „Iterated revision operations stemming from the history of an agent's observations“. In: *Frontiers in belief revision* (2001), pp. 279–301.

[Rag+20] Marco Ragni et al. „Cognitive logics–features, formalisms, and challenges“. In: *ECAI 2020*. IOS Press, 2020, pp. 2931–2932.

[RDJ19] Marco Ragni, Hannah Dames, and Philip N Johnson-Laird. „A meta-analysis of conditional reasoning“. In: *Proceedings of the 17th International Conference on Cognitive Modeling*. Applied Cognitive Science Lab, Penn State. 2019, pp. 151–156.

[RJ20] Marco Ragni and Philip N Johnson-Laird. „Reasoning about epistemic possibilities“. In: *Acta Psychologica* 208 (2020), p. 103081.

[RKJ18] Marco Ragni, Ilir Kola, and Philip N Johnson-Laird. „On selecting evidence to test hypotheses: A theory of selection tasks.“ In: *Psychological Bulletin* 144.8 (2018), p. 779.

[Spo88] Wolfgang Spohn. „Ordinal conditional functions: A dynamic theory of epistemic states“. In: *Causation in decision, belief change, and statistics: Proceedings of the Irvine Conference on Probability and Causation*. Springer. 1988, pp. 105–134.

[Sri17] KR Srinath. „Python–the fastest growing programming language“. In: *International Research Journal of Engineering and Technology* 4.12 (2017), pp. 354–357.

[Tea23] SymPy Development Team. *SymPy*. `https://www.sympy.org/en/index.html`. [Online; accessed 17. Apr. 2024]. Nov. 2023.

[Ver+01] Niki Verschueren et al. „Why do participants draw non-valid inferences in conditional reasoning?“ In: *Psychological Review* 98 (2001), pp. 182–203.

[VSd05] Niki Verschueren, Walter Schaeken, and Gery d'Ydewalle. „Everyday conditional reasoning: A working memory—dependent tradeoff between counterexample and likelihood use“. In: *Memory & Cognition* 33 (2005), pp. 107–119.

[Wag07] Pascal Wagner-Egger. „Conditional reasoning and the Wason selection task: Biconditional interpretation instead of reasoning bias“. In: *Thinking & Reasoning* 13.4 (2007), pp. 484–505.

[Was68] Peter C Wason. „Reasoning about a rule“. In: *Quarterly journal of experimental psychology* 20.3 (1968), pp. 273–281.

[Wik24] Contributors to Wikimedia projects. *Polish notation - Wikipedia*. `https://en.wikipedia.org/w/index.php?title=Polish_notation&oldid=1217638385`. [Online; accessed 6. May 2024]. Apr. 2024.