Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Theoretische Informatik

Einführung in die autoepistemische Logik

Bachelorarbeit

Gia Luat Thieu
Matrikelnummer: 10004996

Erstprüfer: Prof. Dr. Heribert Vollmer
Zweitprüfer: Dr. Arne Meier
Betreuer: Fabian Müller

25. April 2020

Inhaltsverzeichnis

1. Einleitung

In der Informatik lernt man am Anfang oftmals die klassische Aussagenlogik kennen. Die Aussagenlogik ist *monoton*. Das bedeutet, dass gültige Aussagen gültig bleiben, selbst wenn neue Prämissen hinzugefügt werden. Diese Monotonie-Eigenschaft ist eine Idealisierung, denn in der Realität erhalten wir immer neue Informationen, wodurch vorher gültigen Aussagen falsch werden können. Aus diesem Grund existieren die *nicht-monotonen Logiken*, die dieses Problem beheben.

Diese Arbeit gibt eine Einführung in die autoepistemische Logik von Robert C. Moore [1]. Er nannte sie die Logik des Glaubens.

Zunächst wird erklärt, was im Zusammenhang von Logiken die Monotonie bedeutet. Dabei stellen wir einen Bereich der nichtklassischen Logiken vor: die nicht-monotonen Logiken. Im zweiten Kapitel werden die Grundlagen der autoepistemischen Logik untersucht. Mithilfe eines neuen Operators wird versucht, den Glauben möglichst genau zu beschreiben und die Frage zu beantworten, wann ein Glauben einen optimalen Zustand besitzt. In den weiteren Kapiteln wird gezeigt, wie die autoepistemische Logik in der Modallogik umgesetzt werden kann. Die Auswirkungen werden untersucht und es wird ein Algorithmus zur Berechnung der optimalen Glaubenszustände vorgestellt. Dieser Algorithmus wird Probleme, die angesprochen werden, mit sich bringen. Zuletzt wird die Anwendung der autoepistemischem Logik in der logischen Programmierung gezeigt. Dafür wird das zugehörige Programmierparadigma erklärt und die logische Programmiersprache Prolog vorgestellt.

2. Nicht-monotone Logiken

In der klassischen Aussagenlogik sind inkonsistente Prämissenmengen erlaubt. Das bedeutet, dass sich in der Prämissenmenge die Prämissen widersprechen können. Dies liegt an der *Monotonie*-Eigenschaft.

Definition 1. *Seien* $\Sigma = \{\varphi_1, ..., \varphi_n\}, \Sigma' = \{\psi, ..., \psi\}$ *potentiell inkonsistente Formelmengen und* A *eine Annahme. Dann ist die Ableitungsrelation* $\vdash$ *monoton genau dann, wenn gilt: Wenn* $\Sigma \vdash A$ *gilt, dann gilt auch* $\Sigma \cup \Sigma' \vdash A$.

Die Monotonie-Eigenschaft legt fest, dass Folgerungen bei Hinzufügen von weiteren Formeln beziehungsweise Prämissen beständig bleiben. In anderen Worten: wenn A eine Folgerung von Σ ist, dann ist A ebenfalls eine Folgerung aller Mengen, die Σ als Untermenge beinhaltet. Was beispielsweise in der Aussagenlogik einmal bewiesen wurde, bleibt trotz späterem Hinzufügen von Informationen weiterhin gültig.

Beispiel 1.

Haben wir in der Aussagenlogik nun folgende Prämissen:

- *Max ist krank.*
- *Wenn Max krank ist, isst er eine Suppe.*

Eine korrekte Folgerung ist:

- *Max isst eine Suppe.*

Wir fügen nun eine Prämisse hinzu, die der Folgerung oder einer bisherigen Prämisse widerspricht:

Beispiel 2.

Haben wir in der Aussagenlogik nun folgende Prämissen:

- *Max ist krank.*
- *Wenn Max krank ist, isst er eine Suppe.*
- *Max isst keine Suppe.*

Die folgende Folgerung ist weiterhin korrekt:

- *Max isst eine Suppe.*

Durch die Monotonie-Eigentschaft bleibt die Folgerung korrekt, obwohl sie einer Prämisse widerspricht. Es entsteht das Problem, dass widersprüchliche Aussagen oder Folgerungen nicht zurückgezogen werden können. *Nicht-monotone* Logiken umgehen dieses Problem. Eine Logik heißt nicht-monoton, wenn sie die Monotonie-Eigenschaft verletzt.

Folgerungen im Alltag sind oft nicht-monoton, weil die Menschen Informationen nicht objektiv sondern subjektiv aufnehmen. Es ist möglich, dass nicht vollständige Informationen erhalten werden und falsche Schlussfolgerungen entstehen, die vorerst als richtig gelten. Dadurch kann der Fall auftreten, dass wir Folgerungen basierend auf unseren Annahmen aufgrund von neuen Informationen zurückziehen müssen.

Beispiel 3. *Tux ist das offizielle Maskottchen des freien Linux-Kernels. Tux ist ein Vogel. Durch das Hintergrundwissen im Alltag, Vögel können üblicherweise fliegen, folgt, dass auch Tux die Fähigkeit zu fliegen hat. Offenbart sich die Information, dass Tux ein Pinguin ist, wäre es sinnvoll, diese Folgerung zurückzuziehen. Pinguine sind nämlich Vögel, die nicht fliegen können.*

Die Monotonie-Eigenschaft wird benötigt, um sicherzustellen, dass durch Hinzufügen neuer Prämissen keine Informationen beziehungsweise Folgerungen geschwächt werden oder verloren gehen. Nun stellt sich die Frage, welche neuen Eigenschaften einer nicht-monotonen Logik hinzugefügt werden müssen. Wir werden später erfahren, welche Eigenschaften in der autoepistemischen Logik genutzt werden, um die Monotonie zu ersetzen.

Im Alltag oder bei wissenschaftlichen Argumentationen werden oftmals vorläufig Schlüsse gezogen, die streng logisch nicht gültig sind oder später zurückgezogen werden müssen. Nicht-monotone Logiken finden Anwendung bei der Abduktion, in der zu einem Sachverhalt und dazugehörigen Regeln eine Hypothese zur Erklärung dieses Sachverhaltes aufgestellt wird. Eine weitere Anwendung ist die Default-Logik. Dabei wird angenommen, dass bestimmte standardmäßige Aussagen (z.B. alle Vögel, die keine Pinguine oder Strauße sind, können fliegen) wahr sind. Die autoepistemische Logik von Moore [1] bezieht sich auf den Glauben eines Individuums, welches berücksichtigt, dass vorher geglaubte Annahmen durch neue Informationen reflektiert und ersetzt werden können. In der autoepistemischen Logik sind die Fixpunkte die *stabilen Expansionen*. Wir werden diesen Begriff im nächsten Kapitel kennenlernen.

3. Grundlagen der autoepistemischen Logik

Die autoepistemische Logik modelliert den Glauben (im Englischen "Belief", wobei nicht der religiöse Glauben gemeint ist) eines rational denkenden Individuums, der seinen eigenen Glauben reflektieren und ändern kann. Es kann Aussagen aufstellen wie "Wenn ich nicht an P glaube, dann ist Q wahr.", an denen es sich hält.

Beispiel 4. *Das Individuum schaut im Sommer aus seinem Fenster heraus und betrachtet die Straße. Zurzeit regnet es nicht. Er bemerkt, dass die Straßen trocken sind. Mithilfe seines Hintergrundwissens, den er im Alltag aufgebaut hat, glaubt er, dass eine trockene Straße keinen Regen impliziert. Dadurch hat er den Glauben: „Wenn die Straßen nicht trocken sind, hat es geregnet.“*

Das Individuum nimmt an, dass durch einen Regen die Straßen nass wurden. Wenn die Straßen doch trocken sind, ist das Individuum nicht mehr dazu verpflichtet, zu glauben, dass es geregnet hat, solange keine zusätzliche Begründung für einen Regen vorliegt.

Die autoepistemische Logik ist wie die Aussagenlogik jedoch mit einem zusätzlichen Glaubensoperator B aufgebaut. Wir interpretieren B(P) als „Es wird an P geglaubt.“, wobei P eine beliebige Formel ist. Sei P die Aussage: „Es hat geregnet.“. Die Aussage

„Wenn es geregnet hat, würde ich glauben, dass es geregnet hat.“ stellen wir nun dar durch die Formel:

$$P \supset B(P).$$

Hierbei behandeln wir die Relation ” $\supset$ ” wie eine Implikation, die nur im Kontext mit Formeln der Form B(P) angewendet wird. Die autoepistemische Logik folgt dem Prinzip, dass wenn P nicht der Fall ist, das Invididuum dann nicht an P glaubt. Auf unser Beispiel bezogen wäre die Aussage „Wenn ich nicht glaube, dass es geregnet hat, hat es auch nicht geregnet.“ oder formal:

$$\neg B(P) \supset \neg P. \tag{3.1}$$

Das Individuum hat keine zusätzlichen Informationen über einen Regen erhalten. Wir nehmen dadurch an, dass Individuum keinen Grund hat, an einen Regen zu glauben. „Ich glaube nicht, dass es geregnet hat.“

$$\text{„Ich glaube nicht, dass es geregnet hat.“ bzw. } \neg B(P). \tag{3.2}$$

Das Individuum kann Aussagen reflektieren, an denen es nicht glaubt. Wendet es den Modus Ponens bei (3.1) und (3.2) an, folgt es:

$$\text{„Es hat nicht geregnet.“ bzw. } \neg P.$$

Im weiteren Verlauf dieses Kapitels werden wir grundlegende Formalitäten und Methoden für die autoepistemische Logik kennenlernen. Am Ende werden wir in Form einer *stabilen Expansion* klären, was ein günstiger „Zustand“für das Individuum ist.

Wir möchten den Glauben des Individuums analysieren können. Der Glauben wird durch Formeln interpretiert

Definition 2. *[2] Eine* autoepistemische Theorie $T = \{\varphi_1, ..., \varphi_n\}$ *ist eine Menge aus Formeln.*

Wir schreiben nun Theorie anstatt autoepistemischer Theorie, solange dies nicht gesondert erläutert wird.

Eine Theorie hat einen autoepistemischen Teil, der die Formeln B(P) enthält und einen Teil, der aussagenlogische Formeln enthält. Wir nennen diese Formeln *objektiv.*

Definition 3. *[2] Sei* $T = \{\varphi_1, ..., \varphi_n\}$ *eine Theorie. Die Menge der* objektiven Formeln *von T sind alle Formeln von T, die nicht von der Form B(P) sind.*

Der Wahrheitswert des Glaubens des Individuums ist bestimmt durch den aussagenlogischen Wahrheitswert in der Umwelt mithilfe von Aussagenlogik und die Berücksichtigung, welche Aussagen vom Individuum geglaubt werden.

Definition 4. *[2] Eine* autoepistemische Interpretation $I = \{\mathfrak{I}(\varphi_1), \mathfrak{I}(\varphi_2), ..., \mathfrak{I}(\varphi_n)\}$ *einer Theorie* $T = \{\varphi_1, ..., \varphi_n\}$ *ist eine Menge von Belegungen der Formeln von T, die folgende Bedingungen erfüllt:*

1. *I verhält sich auf dem aussagenlogischen Teil der Formeln wie eine Belegung in der Aussagenlogik;*
2. *eine Formel B(P) ist wahr in I genau dann, wenn* $P \in T$.

Beispiel 5. *Sei* $T = \{P, \neg Q, P \rightarrow B(\neg Q)\}$. *Dann ist* $M = \{\mathfrak{I}(P), \mathfrak{I}(Q), \mathfrak{I}(\mathfrak{B}(\neg\mathfrak{Q}))\}$ *mit* $\mathfrak{I}(P) = 0, \mathfrak{I}(Q) = 0, \mathfrak{I}(B(\neg Q)) = 1$ *eine Interpretation von T.*

Unsere Formeln sind widerspruchsfrei. Für den Operator B bedeutet dies: eine Formel ist widerspruchsfrei, wenn ihre Negation nicht geglaubt wird.

Wir schreiben Interpretation statt autoepistemische Interpretation, solange etwas anderes nicht explizit erwähnt wird. Wenn nur Bedingung 1 erfüllt ist und der Wahrheitswert von Formeln der Form B(P) und aussagenlogische Konstanten beliebig zugewiesen sind, nennen wir die Interpretation eine aussagenlogische Interpretation. Mit dem Begriff „aussagenlogische Konstante “ im Kontext einer Belegung meinen wir eine Belegung einer Formel oder einer Variable mit dem Wert 0 oder 1.

Ein *autoepistemisches Modell* drückt aus, dass der Glauben des Individuums im folgenden Sinne richtig ist.

Definition 5. *[2] Ein* autoepistemisches Modell *für eine Theorie* $T = \{\varphi_1, ..., \varphi_n\}$ *ist eine Interpretation I, in der alle Formeln von T erfüllt werden.*

Wir schreiben $I \models T$ für „I heißt Modell für T“ und Modell statt autoepistemisches Modell, solange etwas anderes nicht explizit erwähnt wird. Eine aussagenlogische Interpretation, in der alle Formeln von T erfüllt werden, nennen wir aussagenlogisches Modell.

Beispiel 6. *Sei T wie in Beispiel 5 gewählt. Dann heißt* $M = \{\mathfrak{I}(P), \mathfrak{I}(Q), \mathfrak{I}(B(\neg Q)\}$ *mit* $\mathfrak{I}(P) = 1,\ \mathfrak{I}(Q) = 0,\ \mathfrak{I}(B(\neg Q)) = 1$ *ein Modell für T beziehungsweise* $M \models T$.

Im Folgenden werden die Begriffe *Vollständigkeit* und *Korrektheit* für die autoepistemische Theorie definiert.

Definition 6. *[2] Eine Theorie* $T = \{\varphi_1, ..., \varphi_n\}$ *ist* vollständig *genau dann, wenn T jede Formel enthält, die in allen Modellen für T wahr ist.*

Definition 7. *[2] Eine Theorie $T = \{\varphi_1, ..., \varphi_n\}$ ist korrekt in Bezug auf eine Prämissenmenge A genau dann, wenn für jede Interpretation I von T gilt :*
$I \models A \rightarrow I \models T$.

Um herauszufinden, ob eine Theorie korrekt in Bezug auf eine Prämissenmenge oder vollständig ist, müssten wir jede Interpretation oder jedes Modell für eine Theorie aufstellen. Dieses Verfahren könnte aufwändig werden, wenn eine Theorie viele Interpretationen und Formeln hat. Im Laufe dieses Kapitels werden wir andere Methoden zur Bestimmung von Korrektheit und semantischer Vollständigkeit untersuchen.

In der klassischen Logik stellt die Monotonie sicher: wenn eine Schlussfolgerung getroffen wurde, dann bleibt sie trotz Hinzufügen von neuen Prämissen dennoch gültig. Dies führt zur Zuverlässigkeit von Schlussfolgerungen. Es ist schwierig, in einer nicht-monotonen Logik eine zuverlässige Schlussfolgerung zu treffen, da diese später durch andere Schlussfolgerungen ungültig gemacht werden können. Deswegen muss ein Zustand festgestellt werden, in dem keine weiteren Schlussfolgerungen vom Individuum erschlossen werden können. Dieser Zustand wird bezeichnet als *Stabilität*. Stalnaker [3] hat Bedingungen für diesen Zustand allgemein für nicht-monotone Logiken definiert. Wir wenden diese auf die autoepistemische Logik an.

Definition 8. *[2] Sei $T = \{\varphi_1, ..., \varphi_n\}$ eine Theorie. T ist stabil genau dann, wenn die folgenden Bedingungen erfüllt sind:*

1. *Wenn $P_1, ..., P_n \in T$ und $P_1, ..., P_n \vdash Q$, dann $Q \in T$. ($\vdash$ heißt tautologische Folgerung)*
2. *Wenn $P \in T$ gilt, dann gilt $B(P) \in T$.*
3. *Wenn $P \notin T$ gilt, dann gilt $\neg B(P) \in T$.*

Zusätzlich werden für mehrere Beweise widerspruchsfreie stabile Theorien benötigt. Wenn eine stabile Theorie widerspruchsfrei ist, dann gelten zwei zusätzliche Bedingungen:

4. *Wenn $B(P) \in T$, dann $P \in T$,*
5. *Wenn $\neg B(P) \in T$, dann $P \notin T$.*

Wenn $B(P) \in T$ und $P \notin T$, dann würde nach Bedingung 3 $\neg B(P) \in T$ gelten, sodass T inkonsistent wäre. Deswegen muss Bedingung 4 gelten. Die Bedingung 5 gilt. Wenn andernfalls $\neg B(P) \in T$ und $P \in T$ gälten, dann gälte nach Bedingung 2 $B(P) \in T$ und somit T inkonsistent wäre.

Wir können zeigen, dass der Wahrheitswert jeder Formel einer stabilen Theorie nur abhängig vom Wahrheitswert seiner objektiven Formeln (siehe Definition 3) ist.

Satz 1. *[2] Sei $T = \{\varphi_1, ..., \varphi_n\}$ eine stabile Theorie. Dann ist jede autoepistemische Interpretation von T, die ein aussagenlogisches Modell der objektiven Formeln von T ist, ein autoepistemisches Modell für T.*

Beweis. *[2] Angenommen, T sei eine stabile Theorie und I eine Interpretation von T, sodass I in Bezug auf die objektiven Formeln von T ein aussagenlogisches Modell von T ist. Das bedeutet, dass alle objektiven Formeln von T wahr in I sind. T ist widerspruchsfrei, weil eine inkonsistente stabile Theorie alle Formeln der Sprache der autoepistemischen Logik enthalten würde, sodass es viele objektive Formeln gäbe, die nicht wahr wären in I. Sei P eine beliebig gewählte Formel in T. Ausgehend von der Aussagenlogik wissen wir, dass P äquivalent zur Konjunktion einer Menge von Formeln $P_1, ..., P_k$ ist, in der für jedes i von 1 bis k ein n und m existiert, sodass P_i von der Form*

$$P_{i,1} \vee B(P_{i,2}) \vee ... \vee B(P_{i,n}) \vee \neg B(P_{i,n+1}) \vee ... \vee \neg B(P_{i,m})$$

ist. Hierbei ist $P_{i,1}$ eine objektive Formel. Betrachten wir nun zwei Fälle:

1. *Angenommen, mindestens eine der Formeln $B(P_{i,2}), ..., B(P_{i,m})$, $\neg B(P_{i,n+1}), ..., \neg B(P_{i,m})$ sei in T. Nach den Bedingungen 4 und 5 gilt: Falls solch eine Formel in T wäre, dann müsste diese Formel wahr in I sein, weil T widerspruchsfrei und I eine Interpretation von T ist. Wenn eine der Formeln von P_i wahr in I ist, muss P_i auch wahr in I sein.*

2. *Angenommen, Fall 1 gelte nicht. Dann wird für jede Formel P laut Bedingungen 2 und 3 für stabile Theorien (siehe Definition 8) entweder $B(P)$ oder $\neg B(P)$ in T enthalten sein. Wenn T nicht die Formeln $B(P_{i,2}), ..., B(P_{i,n})$ enthält, dann sind alle Formeln $\neg B(P_{i,n+1}), ..., \neg B(P_{i,m})$ in T. Jedoch wird $P_{i,1}$ von P_i und den Formeln $B(P_{i,2}), ..., B(P_{i,m}), \neg B(P_{i,n+1}), ..., \neg B(P_{i,m})$ abgeleitet, sodass $P_{i,1}$ in T sein muss. Wir wissen aber auch, dass $P_{i,1}$ eine objektive Formel ist, weswegen $P_{i,1}$ ebenfalls in I wahr sein muss. Aber wenn $P_{i,1}$ wahr in I ist, dann muss P_i auch wahr in I sein.*

In beiden Fällen wird P_i wahr in I sein. P ist die Vereinigung aller P_i. Im Zuge dessen muss P wahr in I sein. Wir haben unsere Formel P beliebig gewählt, wodurch jede Formel von T wahr in I sein muss. Wenn jede Formel von T wahr in I ist, ergibt I ein Modell von T. □

Der Satz 1 besagt also: wenn alle objektiven Formeln einer stabilen Theorie wahr sind, dann sind alle Formeln der Theorie wahr. Die objektiven Formeln einer stabilen Theorie sind mächtig. Sie können sogar alle Formeln einer Theorie bestimmen.

Satz 2. *[2] Wenn zwei stabile Theorien die gleichen objektiven Formeln enthalten, dann enthalten sie insgesamt die gleichen Formeln.*

Beweis. *[2] Seien T_1 und T_2 stabile Theorien, die die gleichen objektiven Formeln beinhalten. Zusätzlich enthält T_1 die Formel P. Wir zeigen mithilfe von Induktion, dass T_2 dann P enthalten muss. Wir nennen die Tiefe der Verschachtelung des Operators B in P die „B-Tiefe“ von P.*
Induktionsanfang: Wenn die B-Tiefe von P 0 ist, dann ist P eine objektive Formel, wodurch dieser Satz wahr ist.
Induktionsschritt: Angenommen, die B-Tiefe von P sei die natürliche Zahl d, die größer als 0 ist. Außerdem nehmen wir an, dass wenn zwei stabile Theorien die gleichen objektiven Formeln enthalten, sie dann insgesamt die gleichen Formeln mit der B-Tiefe kleiner d enthalten. Wir nennen diese Annahme „$(\star)$“.

Ausgehend von der Aussagenlogik wissen wir, dass P äquivalent zur Konjunktion einer Menge von Formeln $P_1, ..., P_k$ ist, in der für jedes i von 1 bis k ein n und m existiert, sodass P_i von der Form

$$P_{i,1} \vee B(P_{i,2}) \vee ... \vee B(P_{i,n}) \vee \neg B(P_{i,n+1}) \vee ... \vee \neg B(P_{i,m})$$

ist. Hierbei ist $P_{i,1}$ eine objektive Formel. Wir können die B-Tiefe in der Aussagenlogik nicht verändern, weil alle Formeln der Form $B(P)$ als aussagenlogische Konstanten behandelt werden. In der Aussagenlogik kann fortlaufend die B-Tiefe nicht größer als d sein.
Wir nehmen ebenfalls an, dass T_1 und T_2 widerspruchsfrei sind. Wenn eine dieser Theorien inkonsistent wäre, dann würden alle Formeln der Sprache in dieser Theorie enthalten sein. T_1 und T_2 enthalten aber die gleichen objektiven Formeln, wodurch die andere Theorie ebenfalls alle Formeln der Sprache enthalten muss.
Für jede Formel P_i betrachten wir drei Fälle:

1. *Sei j eine natürliche Zahl mit $2 < j < n$. T_1 enthält $B(P_{i,j})$ für die Zahl j. Laut Bedingung 4 (siehe Definition 8) für widerspruchsfreie stabile Theorien muss T_1 eine Formel $P_{i,j}$ enthalten. Da die B-Tiefe von $P_{i,j}$ gleich der B-Tiefe von $B(P_{i,j})$ minus eins ist, ist die B-Tiefe von $P_{i,j}$ durch Transitivität kleiner als d. Nach $(\star)$ muss T_2 $P_{i,j}$ enthalten. Nach Bedingung 2 für stabile Theorien muss T_2 auch $B(P_{i,j})$ enthalten. Außerdem wird P_i von $B(P_{i,j})$ abgeleitet, sodass P_i in T_2 ist.*

2. *Sei j eine natürliche Zahl mit $n + 1 < j < m$. T_1 enthält $\neg B(P_{i,j})$ für die Zahl j. Laut Bedingung 5 für widerspruchsfreie stabile Theorien kann T_1 die Formel $P_{i,j}$ nicht enthalten. Da die B-Tiefe von $P_{i,j}$ gleich die B-Tiefe von $\neg B(P_{i,j})$ minus eins ist, ist die B-Tiefe von $P_{i,j}$ durch Transitivität kleiner als d. Nach $(\star)$ darf T_2 nicht $P_{i,j}$ enthalten. Nach Bedingung 3 für stabile Theorien muss T_2 auch $\neg B(P_{i,j})$ enthalten. Aber P_i wird von $\neg B(P_{i,j})$ abgeleitet, sodass P_i in T_2 ist.*

3. *Angenommen, keiner der beiden vorherigen Fälle liegt vor. Dann wird für jede Formel P laut Bedingungen 2 und 3 entweder $B(P)$ oder $\neg B(P)$ in T enthalten*

sein. Wenn T_1 nicht die Formeln $B(P_{i,2}), ..., B(P_{i,n})$ enthält, dann sind alle Formeln $\neg B(P_{i,n+1}), ..., \neg B(P_{i,m})$ in T_1. Wir wissen auch, dass $P_{i,1}$ von P_i und den Formeln $B(P_{i,2}), ..., B(P_{i,m}), \neg B(P_{i,n+1}), ..., \neg B(P_{i,m})$ abgeleitet wird, wodurch $P_{i,1}$ in T_1 sein muss. $P_{i,1}$ ist auch in T_2, weil diese Formel objektiv ist. Folglich ist P_i in T_2, weil P_i von $P_{i,1}$ abgeleitet wird.

Also sind alle Formeln $P_1, ..., P_k$ in T_2. Diese Formeln leiten P ab. P ist also ebenfalls in T_2. Da P beliebig gewählt wurde, ist jede Formel von T_1 auch in T_2 und umgekehrt. □

Wir möchten, dass das Individiuum idealerweise rational ist, sodass sein Glauben vollständig ist. Aus diesem Grund soll sein Glauben alles enthalten, das auch semantisch begründet ist, sodass der Agent durch Ableitung seines Glaubens und seines Wissens davon überzeugt ist, dass dieser Glauben von ihm stammt. Wir benutzen die bekannten Sätze, um den folgenden Satz zu beweisen.

Satz 3. *[2] Eine Theorie $T = \{\varphi_1, ..., \varphi_n\}$ ist vollständig genau dann, wenn sie stabil ist.*

Beweis. *[2] „⇐“-Richtung: Wir zeigen: wenn T eine stabile Theorie ist, dann enthält T jede Formel, die wahr ist, in jedem Modell von T. Sei T eine stabile Theorie und sei P eine beliebig gewählte Formel, die nicht in T ist. Wir zeigen, dass ein Modell von T existiert, in dem P falsch ist.*
Ausgehend von der Aussagenlogik wissen wir, dass P äquivalent zur Konjunktion einer Menge von Formeln $P_1, ..., P_k$ ist, in der für jedes i von 1 bis k ein n und m existiert, sodass P_i von der Form

$$P_{i,1} \vee B(P_{i,2}) \vee ... \vee B(P_{i,n}) \vee \neg B(P_{i,n+1}) \vee ... \vee \neg B(P_{i,m}).$$

Hierbei ist $P_{i,1}$ eine objektive Formel. Da T stabil ist und P von $P_1, ..., P_k$ abgeleitet wird, wissen wir durch die Bedingung 1 für stabile Theorien, dass wenn P nicht in T ist, dann darf mindestens eine Formel aus $P_1, ..., P_k$ nicht in T sein. Sei P_i eine solche Formel. Dann wird P_i von jeder dieser Disjunktionen abgeleitet, sodass keine dieser Formeln in T sein kann. Wir zeigen nun, dass es ein Modell von T existiert, in der all diese Disjunktionen falsch sind.

$P_{i,1}$ ist nicht in T. Folglich darf $P_{i,1}$ nicht aus der Menge der objektiven Formeln von T abgeleitet werden. Betrachten wir nun zusätzlich, dass $P_{i,1}$ eine objektive Formel ist, dann muss es zu T laut Vollständigkeitsatzes aus der Aussagenlogik eine Belegung geben, in der $P_{i,1}$ falsch ist und alle objektiven Formeln von T erfüllt werden. Erweitern wir nun diese Belegung zu einer Interpretation I von T, wobei $P_{i,1}$ falsch in I ist. Aussagenlogisch betrachtet ist I ein Modell der objektiven Formeln von T, sodass nach Satz 1 I ein autoepistemisches Modell von T ist, in dem P falsch ist.

Betrachten wir nun alle anderen Disjunktionen von P_i. Nach Bedingungen 2 und

3 enthält eine stabile Theorie alle Formeln von der Form $B(P)$ oder $\neg B(P)$, die in der Interpretation der Theorie erfüllt werden. Die Disjunktionen $B(P_{i,2}) \vee ... \vee B(P_{i,n}) \vee \neg B(P_{i,n+1}) \vee ... \vee \neg B(P_{i,m})$ sind in keiner Interpretation von T enthalten, weil sie ebenfalls nicht in T enthalten sind. Folglich ist I ein Modell von T, in der P_i falsch ist, weil alle Disjunktionen von P_i in I falsch sind.

$\Rightarrow$-Richtung: Wir zeigen, wenn T vollständig ist, dann ist T stabil beziehungsweise die Bedingungen 1 - 3 (siehe Definition 10) gelten. Angenommen, T sei vollständig. Dann gilt für jede Formel P: wenn P in jedem Modell von T wahr ist, dann ist P in T. Sei M ein beliebiges gewähltes Modell von T. Wenn P in M wahr ist, dann ist P in T, weil P durch die beliebige Wahl von M in jedem Modell von T wahr ist. Wir zeigen die drei Bedingungen für stabile Theorien: Sei T eine Theorie und Q eine Formel.

1. *Angenommen, die Formeln $P_1, ..., P_n$ sind in T und $P_1, ..., P_n \vdash Q$ gälten. $P_1, ..., P_n$ ist wahr in M, weil M ein Modell für T ist. Q ist wahr in M, weil Q von $P_1, ..., P_n$ abgeleitet wird und $P_1, ..., P_n$ wahr in M ist. Also ist Q in T.*
2. *Angenommen, P sei in T. Dann ist B(P) wahr in M, weil M ein Modell von T ist. Deswegen ist B(P) in T.*
3. *Angenommen, P sei nicht in T. Dann ist $\neg B(P)$ wahr in M, da I ein Modell in T ist. Also ist $\neg B(P)$ in T.*

Da die drei Bedingungen für stabile Theorien erfüllt sind, ist T stabil. □

Die Stabilität stellt also die semantische Vollständigkeit einer autoepistemischen Theorie sicher. Sie trifft jedoch keine Aussage darüber, was das Individuum nicht glauben soll. Deswegen ist die Korrektheit in Bezug auf den Prämissen des Individuums nicht sicher gestellt. Es ist möglich, dass das Individuum Aussagen glaubt, die nicht an seinen Prämissen gebunden sind. Dies soll verhindert werden. Folglich darf das Individuum nur an Aussagen glauben, die seinen Prämissen entsprechen oder für die Stabilität benötigt sind.

Definition 9. *[2] Eine Theorie $T = \{\varphi_1, ..., \varphi_n\}$ ist* gebunden *an eine Menge aus Prämissen A genau dann, wenn jede Formel von T in der Menge*

$$A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$$

enthalten ist.

T ist dementsprechend gebunden an A, wenn T keine verschiedene Formel zur der Menge $A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$ hat. Wir brauchen die Korrekheit, um sicherzustellen, dass der Glauben des Individuums wahr ist, wenn all seine Prämissen erfüllt werden.

Satz 4. *[2] Eine Theorie* $T = \{\varphi_1, ..., \varphi_n\}$ *ist* korrekt *zu einer Menge A aus Prämissen genau dann, wenn T gebunden an A ist.*

Beweis. *[2] „⇐“-Richtung: Angenommen, T ist gebunden an A. Dann enthält die Menge von tautologischen Folgerungen* $A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$ *alle Formeln von T. Wir zeigen, T ist korrekt zu A bzw., dass jede Interpretation von T, in der alle Formeln von A erfüllt werden, ein Modell von T ist.*
Sei I eine Interpretation von T, in der alle Formeln von A erfüllt werden. Wir zeigen, dass I ein Modell von T ist. Wenn P in A ist, dann ist P trivialerweise wahr in I. Wenn P in der Form B(Q) und Q in T ist, oder wenn P in der Form $\neg B(Q)$ *und Q nicht in T ist, dann ist P wahr in I, weil I eine Interpretation von T ist. Da alle Formeln in der Menge* $A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$ *wahr in I sind, sind auch all diese tautologischen Folgerungen wahr in I. I ist ein Modell von T, weil alle Formeln von T in der Menge enthalten sind. Man kann eine beliebige Interpretation I, in der alle Formeln von A erfüllt werden, von T wählen, sodass jede Interpretation, in der alle Formeln von A erfüllt werden, ein Modell von T ist.*

„⇒“-Richtung: Angenommen, T sei korrekt zu einer Menge A. Dann ist jede Interpretation, in der alle Formeln von A erfüllt werden, von T ein Modell von T. Wir zeigen, T ist gebunden an A, beziehungsweise dass jede Formel von T in der Menge $A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$ *ist.*
Sei $F = A \cup \{B(P) \mid P \in T\} \cup \{\neg B(P) \mid P \notin T\}$*. Für alle P gilt: Wenn P in T ist, dann ist B(P) in F, sodass B(P) in allen aussagenlogischen Modellen von F erfüllt wird. Wenn P nicht in T ist, dann ist* $\neg B(P)$ *in F und B(P) falsch in jedem aussagenlogischen Modell von F. Also gilt für jedes aussenlogische Modell von F: B(P) ist wahr genau dann, wenn P in T ist, sodass jedes aussagenlogische Modell von F eine Interpretation von T ist. Da jede Interpretation von T, in der alle Formeln von A erfüllt werden, ein Modell von T ist, ist jedes aussagenlogische Modell von F ein Modell von T.*
Dadurch, dass jede Formel von T wahr in allen Modellen von T ist, ist jede Formel von T wahr in jedem aussagenlogischen Modell von F. Laut des Vollständigkeitssatzes für die Aussagenlogik ist jede Formel von T eine tautologische Folgerung von F. Also ist T gebunden an A. □

Definition 10. *[2]* **??** *Eine Theorie* $T = \{\varphi_1, ..., \varphi_n\}$ *ist eine stabile Expansion einer Menge aus Prämissen A, wenn T eine Obermenge von A ist, und T ist stabil und gebunden an A.*

Durch stabile Expansionen können wir nun mögliche stabile Theorien T, die gebunden zu einer Prämissenmenge A sind, finden. Dabei können mehrere stabile Expansionen existieren.

Beispiel 7. *Sei A eine Prämissenmenge mit $A = \{\neg B(P) \supset Q, \neg B(Q) \supset P\}$ und T eine stabile Theorie. Wenn $P \notin T$ gilt, dann gilt $Q \in T$ und umgekehrt. Wenn T gebunden an A ist, dann ist entweder P oder Q in T.*

Aus diesem Grund gibt es mehrere stabile Expansionen. Nun zeigen wir ein Beispiel, in der keine stabile Expansion existiert.

Beispiel 8. *Sei A eine Prämissenmenge mit $A = \{\neg B(P) \supset P\}$. Eine stabile autoepistemische Theorie T muss folglich $\neg B(P) \supset P$ und P enthalten. Dadurch ist T nicht gebunden an A, sodass keine stabile Expansion existieren kann.*

Das Individuum versucht bestenfalls stabile autoepistemische Theorien zu haben, die an Prämissen gebunden sind, um zuverlässige Schlussfolgerungen zu schließen. Für das Individuum stellt sich die Frage, wie er seinen Glauben interpretieren soll, wenn mehrere oder gar keine stabilen Expansionen existieren.
Wenn es genau eine stabile Expansion existiert, dann ist dem Individuum klar, wie er diese interpretieren soll. Das Fehlen einer stabilen Expansion führt zur Unzuverlässigkeit seiner Schlussfolgerungen. In der klassischen Logik tritt dieses Problem nicht auf, weil das Individuum keinen Zugriff auf eine „objektive“ Wahrheit hat, sondern nach seinem Glauben rational handelt.

4. Anwendung in der Modallogik

Unsere Belegungen, welche Formeln unser Individuum glaubt oder nicht glaubt, für autoepistemische Logiken wurden bisher beliebig festgelegt. Zudem können unsere Theorien unendlich viele Formeln enthalten, wodurch wir keine Interpretationen und Modelle analysieren können.
Um dieses Problem zu lösen, geben wir dem Individuum ein Fundament, worauf es seinen Glauben bauen kann. Dieses Fundament soll in unserem Fall die Kripke-Struktur aus der Modallogik sein. Wir möchten zielführend stabile autoepistemische Theorien durch Kripke-Strukturen in der Modallogik darstellen.
Am Ende dieses Kapitels möchten wir zeigen, dass für jede Interpretation beziehungsweise jedes Modell einer stabilen Theorie eine entsprechenden Interpretation beziehungsweise ein modallogisches Modell existiert.

Wir erinnern uns zurück an die Definition einer Kripke-Struktur und seinen Operatoren.

Definition 11. *[4] Eine Kripke-Struktur ist ein Tripel $M = (W, R, V)$ mit*

- *W ist eine nichtleere Menge (die Menge der Welten) und*

- *R ist eine binäre Relation auf W*
- *$V : Var \to \mathcal{P}(W)$ ist eine Abbildung, die jeder aussagenlogischen Variablen p eine Menge V(P) von Welten zuordnet, die Welten, in denen p wahr ist. (P(W) ist die Potenzmenge von W.)*

Zusätzlich zur Sprache der autoepistemischen Logik und der Aussagenlogik enthält die modale Sprache den unären Konnektor $\Box$ und den Konnektor $\Diamond$, den wir als Abkürzung von $\neg\Box\neg$ definieren. Sei φ eine Formel. Dann definieren wir:

- $\Box\varphi \approx$ „φ ist notwendigerweise wahr.“
- $\Diamond\varphi \approx$ „φ ist möglicherweise wahr.“

Die Wahrheit einer Formel P in einer möglichen Welt W_1 sei dann wie folgt definiert:

- $\Box\varphi$ ist wahr in W_1, wenn für alle von W_1 aus erreichbaren Welten W_2 gilt, dass φ in W_2 wahr ist. Sonst ist $\Box\varphi$ falsch in W_1.
- $\Diamond\varphi$ ist wahr in W_1, wenn es eine von W_1 aus erreichbare Welt W_2 gibt und φ in W_2 wahr ist. Sonst ist $\Diamond\varphi$ falsch in W_1.

Das rationale Individuum hat die Fähigkeit zu wissen, woran es glaubt und woran es nicht glaubt. Es glaubt zum Beispiel an eine Aussage P, wenn in allen von seiner Welt aus erreichenbaren möglichen Welten P wahr ist beziehungsweise $\Box P \to B(P)$. Vereinfacht ausgedrückt, fragt sich das Individuum, wie der Wahrheitswert von P ist, wenn alle erreichbaren möglichen Welten, in denen P wahr ist, real wären. In einer Kripke-Struktur beschränken wir uns also nicht auf die Menge der möglichen Welten, in denen P wahr ist, sondern auf die Menge der möglichen Welten, in denen das Individuum an P glaubt. Um diesen Glauben zu modellieren brauchen wir die Äquivalenzrelation. Mithilfe der Äquivalenzrelation können wir die Menge der möglichen Welten, die mit dem Glauben übereinstimmen, herausfinden. Diese Äquivanlenzrelation wird bei *S5-Strukturen* angewandt.

Definition 12. *[5] Eine Kripke-Struktur M = (W, R, V) ist eine* S5-Struktur*, wenn die Relation R eine Äquivalenzrelation ist, beziehungsweise folgende Axiomen gelten:*

1. $\Diamond\Diamond\phi \to \Diamond\phi$ *(Transitivität)*
2. $\phi \to \Diamond\phi$ *(Reflexivität)*
3. $\phi \to \Box\Diamond\phi$ *(Symmetrie)*

Der Wahrheitswert einer Formel wird nun durch die Welten W festgelegt. Eine Formel der Form B(P) ist für ein Modell $M = (W, R, V)$ in einer Welt $W_i \in W$ wahr genau dann, wenn P in jeder Welt, die von W_i aus erreichbar ist, wahr ist.

Wir zeichnen S5-Strukturen wie einen Graphen aus der Graphentheorie. Dabei werden reflexive Kanten weglassen, da die Relation reflexiv ist. Die restlichen Kanten haben keine Pfeile, um die Symmetrieeigentschaft zu verdeutlichen. Zusätzlich können wir Kanten weglassen, die nur durch die Ableitung von Transititiväten redundant werden. Zur Übersichtlichkeit werden in den Abbildungen transitive Relationen behalten. In unseren Beispielen sind die Modelle S5-Strukturen.

Beispiel 9. *Sei Modell* $M = (\{w_1, w_2, w_3\}, R, V)$ *mit Relationen* $R = \{\{w_1, w_2\}, \{w_4, w_5\}\}$ *und Belegungen V:* $V(P) = \{w_1, w_2, w_3, w_5\}$, $V(Q) = \{w_1, w_4, w_5\}$ *gegeben.*
Dann gilt beispielsweise:

$$M, w1 \models B(P)$$
$$M, w3 \models B(P)$$
$$M, w5 \not\models B(Q)$$

Die folgende Abbildung 1 stellt M dar.

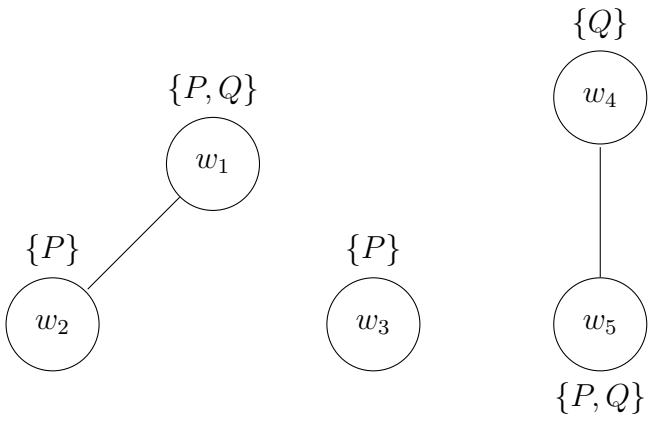

Abbildung 1: Wahrheitswerte einer Formel in S5

Anhand des folgenden Beispiels werden wir sehen, wie sich der neue Operator B zusammen mit den bereits existierenden Operatoren $\Diamond$ und $\Box$ verhält.

Beispiel 10. *Sei das folgende Modell* $M = (\{w_1, w_2, w_3), R, V)$ *mit Relationen* $R = \{\{w_1, w_2\}, \{w_1, w_3\}, \{w_2, w_3\}\}$ *und Belegungen V:* $V(P) = \{w_1, w_2, w_3\}$, $V(Q) = \{w_3\}$, $V(U) = \emptyset$ *gegeben. Dann gilt beispielsweise:*

$$M, w_1 \models \Box B(P)$$
$$M, w_3 \models \Diamond\Box((P \wedge Q) \wedge B(P))$$
$$M, w_3 \not\models \Box(Q \wedge \neg r)$$

Die folgende Abbildung 2 stellt M dar.

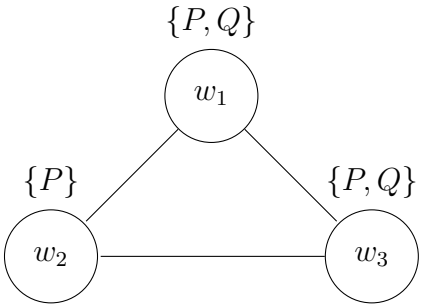

Abbildung 2: $\Box, \Diamond$ *und* B

Wir wollen zeigen, dass eine Menge von Formeln eine stabile Theorie ist genau dann, wenn diese Formeln wahr in jeder Welt einer S5-Struktur ist. Dabei müssen in der S5-Struktur alle Welten miteinander verbunden sein.

Definition 13. *[5] Eine S5-Struktur M = (W, R, V) ist* vollständig *genau dann, wenn jede Welt von jeder Welt erreichbar ist.*

Beispiel 11. *Das Modell* $M = (\{w_1, w_2, w_3, w_4, w_5\}, R, V)$ *mit Relationen* $R = \{\{w_1, w_2\}, \{w_1, w_3\}, \{w_1, w_4\}, \{w_2, w_3\}, \{w_2, w_4\}, \{w_3, w_4\}\}$ *und Belegungen V:* $V(P) = \{w_1, w_2, w_3, w_4\}$, $V(Q) = \{w_1, w_5\}$ *stellt eine vollständige S5-Struktur dar.*
Wir zeichnen eine vollständige S5-Struktur wie einen vollständigen Graphen aus der Graphentheorie. Das Diagramm in Abbildung 3 stellt M dar.

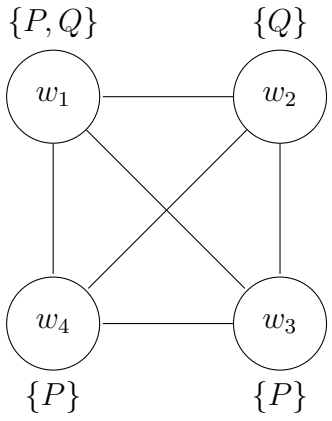

Abbildung 3: Eine vollständige S5-Struktur

Satz 5. *[5] Eine Menge von Formeln $T = \{\phi_1, ..., \phi_n\}$ ist wahr in allen Welten einer vollständigen S5-Struktur M = (W, R, V) genau dann, wenn T eine stabile Theorie ist.*

Beweis. *Sei $T = \{\phi_1, ..., \phi_n\}$ die Menge der Formeln, die in allen Welten einer vollständigen S5-Struktur wahr sind. Wir wissen, dass in der autoepistemischen Logik für Formeln der Form B(P) gilt: Wenn von einer Welt $W_i \in M$ aus für alle zugänglichen Welten P gilt, dann gilt für W_i auch B(P). Also gilt auch: $B(P) \in T$ genau dann, wenn $P \in T$. Wenn in einer Welt P nicht gilt, dann ist B(P) auch falsch für die Menge T beziehungsweise: $B(P) \notin T$ genau dann, wenn $P \notin T$. Nach der Korrektheit in der Aussagenlogik ist T unter tautologischer Konsequenz abgeschlossen. Da T eine Theorie ist und die drei Bedingungen für stabile Theorien (siehe Definition 8) erfüllt sind, ist T stabil.*

Angenommen, T ist stabil. Sei F die Menge der objektiven Formeln von T. Dann ist F ebenfalls unter tautologischer Konsequenz abgeschlossen und es existiert eine vollständige S5-Struktur M, in der alle Modelle für F jeweils eine mögliche Welt erzeugen. Dadurch enthält F alle objektiven Formeln, die wahr in jeder Welt dieses Modells sind. Bilden wir nun aus dieser S5-Struktur die stabile Theorie T', die die objektiven Formeln F enthält. Nach Satz 1 muss T' = T sein, da T' und T die gleichen objektiven Formeln enthalten. Deswegen ist T die Menge der Formeln, die wahr in jeder Welt einer vollständigen S5-Struktur sind. □

Beispiel 12. *Gegeben sei folgendes Modell $M = (\{w_1, w_2, w_3\}, R, V)$ und die Theorie $T = \{\neg B(P) \supset Q, \neg B(Q) \supset P\}$. R und V sind aus Abbildung 4 ersichtlich.*
Da alle Formeln von T wahr in allen Welten der vollständigen S5 Struktur M ist, ist T eine stabile Theorie.

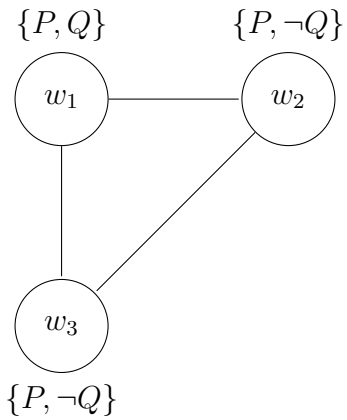

Abbildung 4: Stabile Theorien bei S5-Strukturen

Der Übersichtlichkeit halber schreiben wir für vollständige S5-Strukturen nur noch die Menge der Belegungen auf, weil die Relationen trivialerweise klar sind. Kennen wir die Anzahl der Welten, dann kennen wir auch die Relationen. Wir benennen die Welten für Diagramme w_i mit $1 \leq i \leq k$. Für Abbildung 3 wäre $M = (\{P, Q\}, \{P, \neg Q\}, \{P, \neg Q\})$.

Die Anwendung von vollständigen S5-Strukturen erlaubt uns die Begriffe „Interpretation“ und „Modell“ für stabile Theorien zu definieren. Hierfür brauchen wir ein Tupel bestehend aus einer vollständigen S5-Struktur und eine aussagenlogischen Wahrheitszuweisung. Die vollständige S5-Struktur stellt den Glauben des Individuums dar und die aussagenlogische Belegung weist den Welten Wahrheitswerte zu.

Definition 14. *[5] Sei M = (W, R, V) eine vollständige S5-Struktur, S eine Belegung und beide definiert auf den aussagenlogischen Konstanten einer Theorie* $T = \{\phi_1, ..., \phi_n\}$. *Dann ist das Tupel (M, S) eine modallogische Interpretation von T genau dann, wenn T aus allen Formeln besteht, die wahr in jeder Welt in M sind.*

Wir behandeln die Belegung S als eine potentiell tatsächliche Welt. Wenn der Glauben des Individuums falsch ist, wird es nicht mit der tatsächlichen Welt übereinstimmen. Dadurch realisiert es, dass sein Glauben falsch ist. Das Modell M beinhaltet also nicht die Belegung S.

Definition 15. *[5] Ein Tupel (M, S) ist ein modallogisches Modell für T einer Theorie* $T = \{\phi_1, ..., \phi_n\}$ *genau dann, wenn (M, S) eine modallogische Interpretation von T und jede Formel von T wahr in (M, S) ist.*

Im Kontext für unser Individuum beschreiben modallogische Modelle, dass sein Glauben wahr ist, wenn die potentiell tatsächliche Welt, die durch S beschrieben wird, mit seinem Glauben übereinstimmt. Der folgende Satz beschreibt diesen Sachverhalt.

Satz 6. *[5] Wenn (M, S) eine autoepistemische Interpretation von T ist, dann ist (M, S) ein autoepistemisches Modell für T genau dann, wenn die Belegung S mit der Belegung V in einer der möglichen Welten übereinstimmt.*

Beweis. *Wenn S mit einer der Welten von M übereinstimmt, dann ist jede Formel, die wahr in allen Welten in M ist, auch wahr in S. Anders gesagt: Jede Formel, die wahr in allen Welten in M ist, ist auch wahr in (M, S). Laut Definition 15 ist (M, S) ein modallogisches Modell für T.*

Angenommen, S stimmt keiner Welt in M überein. Dann existiert für jede Welt $W_i \in M$ *eine Formel* φ, *wodurch W und V nicht übereinstimmen. Bilden wir nun die Disjunktion der Formel, oder ihrer Negation, und allen entsprechenden Formeln für alle andere Welten in M. Dann ist diese Disjunktion wahr in allen Welten in*

M. Also ist sie eine Formel von T und gleichzeitig falsch in S, wodurch (M, S) kein modallogisches Modell für T ist. □

Die Sätze 5 und 6 ergeben also folgendes: es existiert für jede autoepistemische Interpretation beziehungsweise für jedes autoepistemische Modell eine modallogische Interpretation beziehungsweise ein modallogisches Modell.

Beispiel 13. *Sei* $M = \{\{P, Q\}, \{P, \neg Q\}\}$ *eine vollständige S5-Struktur. Das entsprechende Diagramm sehe wie folgt aus (siehe Abbildung 5). Es existiere eine stabile Theorie, dessen Formeln wahr in allen Welten von M sind.*
Eine autoepistemische Interpretation von T ist das Tupel (M, S) *mit Belegung* $S_1 = \{P, Q\} \vee S_2 = \{P, \neg Q\}$. *Da* $V_{w_1} = S_1$ *beziehungsweise* $V_{w_2} = S_2$, *ist* (M, S) *sogar ein modallogisches Modell von T.*

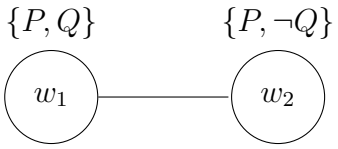

Abbildung 5: modallogische Interpretation und Modell

Wir haben gezeigt, dass für jede Interpretation beziehungsweise jedes Modell einer stabilen Theorie eine entsprechende Interpretation beziehungsweise ein modallogisches Modell existiert.

In Kapitel 2 haben wir bei Beispiel 8 eher argumentativ beschrieben, wieso eine stabile Expansion vorliegt, ohne überhaupt die Theorie T zu kennen. Mithilfe von vollständigen S5-Strukturen können wir nun exakte Aussagen treffen. Wir können also mithilfe von modallogischen Interpretationen und Modellen stabile Expansionen von nicht-trivialen Mengen bestimmen.

Beispiel 14. *Seien M und S wie in Beispiel 13 gewählt. Sei* $A = \{\neg B(P) \supset Q, \neg B(Q) \supset P\}$ *eine Prämissenmenge. Wegen der vollständigen S5-Struktur M können wir annehmen, dass eine stabile Theorie T existiert.*
Es ergeben sich zwei Modelle für T:

- $(M, S_1) = (\{\{P, Q\}\}, \{P, \neg Q\}, \{P, Q\})$
- $(M, S_2) = (\{\{P, Q\}\}, \{P, \neg Q\}, \{P, \neg Q\})$

Laut Definition 7 ist T korrekt in Bezug auf A, weil jedes Modell für T auch ein Modell für A ist. Also ist T auch gebunden an A nach Satz 4. Da T stabil ist und A beinhaltet, ist T laut Definition **??** *eine stabile Expansion von A.*

5. Rechnen mit autoepistemischer Logik

Wir zeigen, wie das Individuum seinen Glauben reflektieren kann. Der folgende Algorithmus entscheidet für eine Prämissenmenge A, welche Formeln in jeder stabilen Expansion von A immer enthalten sind. Diese Formeln bilden die möglichen Schlussfolgerungen des Individuums.

Algorithmus 1 (ineffizienter Algorithmus)**.** *[1] Sei $A = \{\phi_1, ..., \phi_n\}$ eine Prämissenmenge. Dann kann folgender Algorithmus entscheiden, ob eine Formel P von A in jeder stabilen Expansion beinhaltet wird.*

1. *Erstelle alle möglichen Mengen von Belegungen zu den aussagenlogischen Konstanten in A. Erstelle hieraus vollständige S5-Strukturen.*
2. *Überprüfe für jede dieser S5-Strukturen $M = (W, R, V)$: jede Formel P von A ist in jeder möglichen Welt $w \in W$ wahr. Das bedeutet, wir betrachten jede Formel P und bestimmen den Wahrheitswert in jeder Welt in M. Wir ignorieren alle S5-Strukturen, die diese Bedingung nicht erfüllen.*
3. *Wähle nun eine S5-Struktur M aus Schritt 2 aus. Erstelle ein Tupel (M, S) für jede Wahrheitszuweisung S zu den aussagenlogischen Konstanten, die in A vorkommen.*
4. *Überprüfe für jedes Tupel (M, S): Jede Formel von A ist wahr in (M, S) genau dann, wenn $S \in M$ (S ist in einer der Welten von M enthalten). Wenn dies der Fall ist, dann ist die Theorie, die M beschreibt, eine stabile Expansion von A.*
5. *Überprüfe für jede Formel P, ob sie wahr in jeder Welt $w \in W$ ist. Wenn dies der Fall ist, wird P in jeder stabilen Expansion von A enthalten sein.*

Wenn A endlich ist, ist die Anzahl der Belegungen in einer Menge, die in Schritt 1 erstellt wird, endlich. Die Anzahl der Mengen von Belegungen ist ebenfalls endlich, wenn A endlich ist.

Um diesen Algorithmus zu verdeutlichen, wenden wir diesen an unserem Regenbeispiel aus Kapitel 2 an.

Beispiel 15. *Sei P die Aussage: „Es hat geregnet.“ und sei $A = \{P \supset B(P)\}$ eine Prämissenmenge. Wir wollen zeigen, dass $\neg P$ („Es hat nicht geregnet.“) in jeder stabilen Expansion von A ist.*

1. *Wir haben in unserer Prämissenmenge nur die Konstante P. Deswegen können wir nur vier Mengen von Belegungen zu den Konstanten in A erstellen:*

$$\{\emptyset\}, \{\{P\}\}, \{\{\neg P\}\}, \{\{P\}, \{\neg P\}\}.$$

Hieraus ergeben sich vier Kripke-Strukturen mit: keiner Welt, einer Welt mit Belegung P, einer Welt mit Belegung $\neg P$ und zwei Welten, in der in einer P wahr ist und in der anderen P falsch ist.

2. *In den ersten drei Kripke-Strukturen ist $P \supset B(P)$ wahr in jeder Welt, weil P in allen erreichbaren Welten wahr ist. Wir schreiben S5-Strukturen wie in Kapitel 3 abgekürzt:*
$$\{\emptyset\}, \{\{P\}\}, \{\{\neg P\}\}.$$

3. *Wir generieren nun die Tupel (M,S) und fügen nur die Tupel hinzu, für die $\{P \supset B(P)\}$ wahr ist. Daraus ergeben sich vier Tupel:*
$$(\{\emptyset\}, \{\neg P\}), (\{\{P\}\}, \{\neg P\}), (\{\{\neg P\}\}, \{\neg P\}), (\{\{\neg P\}\}, \{P\}). \tag{5.1}$$

4. *Dabei gilt $S \in M$ nur für ein Tupel:*
$$(\{\{\neg P\}\}, \{\neg P\}).$$

5. *$\neg P$ ist wahr in jeder Welt der Kripke-Struktur $(\{\{\neg P\}\}, \{\neg P\})$. Also wird $\neg P$ in jeder stabilen Expansion der Menge $\{P \supset B(P)\}$ enthalten sein.*

Der Algorithmus arbeitet konsistent. Wenn das Individiuum vorher die Information erlangt hat, dass es nicht geregnet hat, dann erwarten wir, es könne nicht mehr folgern, dass es geregnet hat. Das bedeutet, dass $A = \{P, P \supset B(P)\}$, sodass $\neg P$ in keiner stabilen Expansion von A enthalten sein kann.

Beispiel 16. *Sei $A = \{P, P \supset B(P)\}$. Wir kürzen den Algorithmus ab.*
Die S5-Strukturen, für die in jeder Welt alle Formeln von A wahr sind, sind nun:
$$\emptyset, \{P\}$$
Da $\{P\} \notin \emptyset$, betrachten wir nur die S5-Struktur $\{P\}$ mit (M,S) Tupeln:
$$(\{\{P\}\}, \{P\}), (\{\{P\}\}, \{\neg P\}).$$
$A = \{P, P \supset B(P)\}$ ist wahr für beide Tupel. Jedoch ist P nur für $(\{\{P\}\}, \{P\})$ wahr. Also wird P in jeder stabilen Expansion von $\{P, P \supset B(P)\}$ enthalten sein.

Wir haben nun einen Algorithmus, der die Reflektion des Glaubens eines Individuums darstellt. Das Individuum kann mithilfe des Algorithmus' selbstständig „denken“.

Dieser Algorithmus ist unpraktisch anzuwenden. Dadurch, dass alle S5-Strukturen herausgefunden werden müssen, indem alle möglichen Mengen von Belegungen aufgestellt werden, arbeitet der Algorithmus in hyperexponentieller Zeit. Darüber hinaus ist das Problem nicht entscheidbar, wenn die Menge A unendlich groß ist. Mithilfe der Peano-Axiome können wir zeigen, dass keine stabile Expansion einer Prämissenmenge rekursiv aufzählbar zu ermitteln ist, es sei denn, die Menge der ordinären Ableitungen der Prämissen entscheidbar ist. Eine Formel P heißt ordinär, wenn in P der Operator B nicht vorkommt. Es wird nicht näher auf Peano-Axiome eingegangen, da diese nicht Bestandteil der Arbeit sind.

Wir müssen die Sprache unserer Logik abschwächen, um sie entscheidbar zu machen. Wir beschränken die Sprache unserer Logik auf die Prädikatenlogik und lassen zusätzlich noch für den Operator B *geschlossene Formeln* zu. Das bedeutet, dass Quantoren für Formeln mit dem Operator B nur noch innerhalb des Operators angewendet werden dürfen. Beispielweise sei dann $B(\exists x P(x))$ zugelassen und $\exists x B(P(x))$ nicht zugelassen. Danach nehmen wir die Axiome einer entscheidbaren Prädikatenlogik als Prämissen und erweitern diese mit einer endlichen Anzahl von autoepistemischen Formeln als zusätzliche Prämissen. Die stabilen Expansionen der resultierenden Menge sind entscheidbar und können bestimmt werden.

Um zu zeigen, wie dieser Plan praktisch umsetzbar ist, brauchen wir zunächst *ordinäre Formeln*. Wir nehmen an, dass alle zusätzlichen Prämissen ordinär sind. Die Formel, an der wir einen Entscheidungsalgorithmus anwenden wollen, ist eine ordinäre Formel. Daraus ergibt sich folgender Satz, der in dieser Arbeit nicht bewiesen wird.

Satz 7. *[1] Sei $A = \{\phi_1, ..., \phi_n\}$ eine Menge von Prämissen. Dann gilt: wenn A nur ordinäre Formeln beinhaltet, dann hat A eine eindeutige stabile Expansion E, dessen ordinäre Formeln ableitbar von A sind.*

In anderen Worten: wenn wir zusätzliche Prämissen $P_1, ..., P_n$ haben und wir wissen wollen, ob eine ordinäre Formel Q in der stabilen Expansion E der erweiterten Prämissenmenge enthalten ist, dann müssen wir einen Entscheidungsalgorithmus auf $P_1 \wedge ... \wedge P_n \supset Q$ anwenden. Wir definieren einen Algorithmus zur Transformation der Prämissen, sodass für jede stabile Expansion einer beliebigen autoepistemischen Prämissenmenge eine ordinäre Prämissenmenge generiert wird. Diese ordinäre Prämissenmenge hat die gleiche stabile Expansion wie die autoepistemische Prämissenmenge. Wir nennen diesen Algorithmus das *Transformationsverfahren.*

Algorithmus 2. *[1] Sei $T = \{\varphi_1, ..., \varphi_n\}$ eine entscheidbare prädikatenlogische Theorie und $A = \{\psi_1, ..., \psi_n\}$ eine zusätzliche Prämissenmenge in der Sprache von T, die die Anwendung des Operators B in Verbindung mit geschlossenen Formeln zulässt. Dann kann die Transformation auf folgende Weise durchgeführt werden.*

1. *Erstelle alle möglichen Belegungen für die Formeln der Form B(P), die in A auftauchen.*
2. *Für jede Belegung S gilt: Reduziere A, indem alle Formeln der Form B(P), die in A auftauchen, entfernt werden. Die reduzierte Menge sei A'.*
3. *Für jede Formel P, für die auch Formeln der Form B(P) in A vorkommen, gilt: Sei P' die reduzierte Menge von P in Bezug zu S. Wende den Entscheidungsalgorithmus für T an, um zu bestimmen, ob P' von $T \cup A'$ abgeleitet wird.*
4. *Wenn für alle Formeln der Form B(P), die in A vorkommen, B(P) wahr ist und P ableitbar von $T \cup A'$ ist, dann ist die eindeutige stabile Expansion von $T \cup A'$ ebenfalls eine stabile Expansion von $T \cup A$.*

Algorithmus 3 (effizienter Algorithmus)**.** *[1] Sei P eine autoepistemische Formel, $T = \{\varphi_1, ..., \varphi_n\}$ eine entscheidbare prädikatenlogische Theorie und A' die reduzierte Prämissenmenge, die bei Durchführen des Transformationsverfahrens entstanden ist. Dann entscheidet folgender Algorithmus, ob P in der stabilen Expansion von $T \cup A'$ ist.*

1. *Wenn P keine Subformeln der Form $B(P_i)$ enthält, dann sei $P' = P$.*
2. *Wenn P Subformeln der Form $B(P_i)$ enthält, dann:*
   Rufe dieses Verfahren rekursiv für jedes P_i auf, sodass $B(P_i)$ in keiner solchen Subformel einbezogen wird. Wenn P_i in der stabilen Expansion von $T \cup A'$, dann weise $B(P_i)$ den Wert „wahr“ zu, sonst weise $B(P_i)$ den Wert „falsch“ zu.
   Sei P' das Resultat, das durch die Reduzierung von P nach diesen Belegungen entstanden ist.
3. *Wende den Entscheidungsalgorithmus für T an, um zu bestimmen, ob P' von $T \cup A'$ ableitbar ist. Wenn dies der Fall ist, dann ist P eine stabile Expansion von $T \cup A$, sonst ist P keine stabile Expansion von $T \cup A$.*

Wenden wir diesen Algorithmus wieder an unserem Regenbeispiel an.

Beispiel 17. *Sei $A = \{P \supset B(P)\}$ die Aussage: "Wenn es geregnet hat, dann glaube ich, dass es geregnet hat."Da B(P) die einzige Formel mit Operator B ist, weisen wir B(P) Wahrheitswerte zu. Wenn B(P) wahr ist, dann wäre die reduzierte Menge $A' = \{\emptyset\}$. Da A' P nicht enthält, existiert keine stabile Expansion, sodass B(P) wahr wird.*
Wenn B(P) falsch ist, dann wäre die reduzierte Menge $A' = \{\neg P\}$. Hiermit wäre die stabile Expansion von A' ebenfalls eine stabile Expansion von $\{P \supset B(P)\}$. Die letztere Expansion enthält $\neg P$, aber nicht P, was wir in Beispiel 16 erfahren haben.

Dieser Algorithmus arbeitet ebenfalls konsistent. Im folgenden Beispiel erhält das Individuum erneut die Information, dass es bereits geregnet hat. Wir erwarten wieder, dass das gleiche Ergebnis wie in Beispiel 17 auftritt.

Beispiel 18. *Sei $A = \{P, P \supset B(P)\}$ die Aussage: „Es hat geregnet. Wenn es geregnet hat, dann glaube ich, dass es geregnet hat.“*
Da $B(P)$ die einzige Formel mit Operator B ist, weisen wir $B(P)$ Wahrheitswerte zu: Wenn $B(P)$ falsch ist, dann wäre die reduzierte Menge $A' = \{P, \neg P\}$. Aus dieser Menge kann P abgeleitet werden. Also existiert keine stabile Expansion von A, die nicht P enthält. Wenn $B(P)$ wahr ist, dann wäre die reduzierte Menge $A' = \{\neg P\}$, sodass eine stabile Expansion von A existiert, die P enthält und nicht $\neg P$ enthält.

Dieses Verfahren arbeitet im Vergleich zum ineffizienten Algorithmus effizienter. Die Bestimmung der Belegungen für die stabilen Expansionen müssen wir nur noch für eine Prämissenmenge ausführen, wodurch das Verfahren „nur“ in Exponentialzeit arbeitet. Gleichzeitig ist das Verfahren zur Bestimmung, ob eine Formel in einer gewählten stabilen Expansion ist, im schlimmsten Fall um einen linearen Faktor schlimmer als das Entscheidungsverfahren selber. Das Entscheidungsverfahren wird nämlich für jede Anwendung des Operators B in der Formel aufgerufen.

6. Autoepistemische Logik in der logischen Programmierung

Die Prinzipien, die wir aus der autoepistemischen Logik erlernt haben, finden ihre Anwendung in Systemen, die sich bewusst sind, dass ihr zurzeitiger Wissensstand unvollständig sein könnte. In diesem Kapitel stellen wir die Anwendung der autoepistemischen Logik in der logischen Programmiersprache Prolog vor. Die folgende Einführung der grundlegenden Konzepte von Prolog basiert auf dem Werk von William F. Clocksin and Christopher S. Mellish [6].

Die logische Programmierung ist ein Programmierparadigma. Sie wurde um die 1960er Jahre erfunden, um automatische Beweise zu bestimmten Aussagen zu konstruieren. Ein logisches Programm besteht aus der Datenbasis *Fakten* und *Regeln*, die sich nach gewählten Logiksystemen verhalten. Wenn diesem Programm eine *Frage* gestellt wird, so untersucht das Programm mithilfe der logischen Schlüsse aus der Datenbasis, ob die Frage ableitbar ist. Die Eingabe ist also die Frage und das logische Programm gibt dann *ja*, *nein* oder bei Fragen mit Variablen die *Variablenbelegungen* aus.

Beispiel 19. *Die folgende Datenbasis besteht aus 3 Fakten und 2 Regeln.*
Datenbasis: Ein Mensch ist ein Säugetier.

Tux ist ein Vogel.
Ein Mensch hat zwei Beine.
Ein Säugetier hat entweder vier Beine und keine Arme oder
zwei Beine und zwei Arme.
Ein Vogel kann fliegen.
Frage: Kann Tux fliegen? → Nein
Wie viele Arme hat ein Mensch? → 2

Wir können die Fakten, Regeln und Fragen als logische Ausdrücke schreiben. Logische Ausdrücke werden aufgebaut aus Prädikaten, also Funktionen die Wahrheitswerte *true* oder *false* liefern. Wir brauchen Konstanten und Variablen, die als Argumente der Prädikate liefern. Zur Verknüpfung der Prädikate benutzen wir die boolschen Funktionen *and*, *or*, *not* und die *Implikation* beziehungsweise. $\wedge$, $\vee$, $\neg$ und $\Rightarrow$. Wir gehen dabei von einer *closed world assumption aus*. Das heißt, dass die Prädikate für die Argumente den Wert true liefern und für nicht mithilfe der Regeln herleitbaren Regeln den Wert false liefern. Zusätzlich gibt es noch den Allquantor $\forall$ und den Existenzquantor $\exists$, die jeweils eine Variable einführen, die in einem Prädikat als Argument auftreten kann.

Beispiel 20. *Wir wandeln das Beispiel 19 in die Prädikat-Schreibweise um.*
Datenbasis: saeugetier(mensch).
vogel(tux).
beine(mensch,2).
$\forall X(vogel(X) \rightarrow fliegen(X))$
$\forall X(saeugetier(X) \rightarrow (beine(X,4) \wedge arme(X,0)) \vee$
$(beine(X,2) \wedge arme(X,2)))$
Frage: *fliegen(tux)?* → *Nein*
arme(mensch)? → 2

Wir möchten die Anwendung der autoepistemischen Logik in der logischen Programmiersprache *Prolog* zeigen. Prolog beschränkt sich auf die Horn-Klauseln. Seien P_i, die keine Quantoren und kein $\vee$ enthalten. Dann haben Hornformeln die Form:

$$P_1 \wedge P_2 \wedge ... \wedge P_n \Rightarrow Q$$

Hierbei heißen Q der Kopf und $P_1 \wedge P_2 \wedge ... \wedge P_n$ der Rumpf der Klausel. Jede Variable, die im Kopf auftritt, ist mit einem Allquantor verbunden. Jede *nur* im Rumpf auftretende Variable wird mit einem Existenzquantor verbunden.

In Prolog ersetzen wir das $\wedge$ durch ein Komma und das Implikationszeichen durch :- und schreiben die Formel in umgekehrter Reihenfolge:

$$Q \text{ :- } P_1, P_2, ..., P_n$$

Wir wandeln die Datenbasis des Beispiels 22 in die Prolog-Schreibweise um. Fakten haben einen leeren Rumpf, Variablen werden mit Großbuchstaben geschrieben und Prädikate und Symbole mit Kleinbuchstaben. Wir ersetzen die Regel für die Beine und Arme bei Säugetieren durch zwei Horn-Klauseln.

Beispiel 21. *Prolog-Schreibweise:*
saeugetier(mensch).
vogel(tux).
beine(mensch,2).
fliegen(X) *:-* vogel(X)
beine(X,4) *:-* saeugetier(X), arme(X,0)
beine(X,2)*:-* saeugetier(X), arme(X,2)

Um aus Fakten und Regeln, die durch Horn-Klauseln beschrieben werden, andere Aussagen herzuleiten, passen wir die Parameter an (*Unifikation*) und kombinieren diese Klauseln (*Resolution*). Wir können aus den Klauseln

$$b \text{ :- } a_1, a_2, ..., a_n. \text{ und } c \text{ :- } d_1, d_2, ..., d_k \; mit \; c = a_i$$

die folgende Klausel erzeugen:

$$b \text{ :- } a_1, ..., a_{i-1}, d_1, ..., d_k, a_{i+1}, ..., a_n$$

Falls a_i und c Parameter besitzen, dann wird festgestellt, ob man diese Parameter so an Werte binden, dass beide Teilausdrücke identisch werden (Unifikation).

Die prinzipielle Arbeitsweise funktioniert so, dass das Programm eine eingegebene Frage durch die Regeln in einfachere Teilziele zerlegt, um an die Fakten zu gelangen. Die Zerlegung geschieht in einer festen Reihenfolge. Zunächst wird die Datenbasis auf eine Horn-Klausel mit passendem Kopf durchsucht. Dann Variablen der Parameter durch Unifikation mit Werten belegt. Im Rumpf werden diese Werte nach den Regeln substituiert. Man versucht, die einzelnen Teile des Rumpfes beziehungsweise Teilziele auf die gleiche Weise zu bearbeiten. Wenn man nur noch Teilziele hat, die Fakten der Datenbasis sind beziehungsweise als wahr gelten, dann endet die Berechnung. Wenn keine Unifikation möglich ist, also man bildlich gesehen in eine Sackgasse gelangt, dann wird backtracking angewendet. Backtracking bedeutet, dass die Berechnung ähnlich wie bei der Tiefensuche soweit rückgängig gemacht wird, dass eine andere Alternative verfolgt werden kann.

Beispiel 22. *Sei die folgende Datenbasis gegeben:*
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
ancestor(X,X).
parent(alice,bob).

Bei der Frage ?- ancestor(A,bob). *gibt das Programm* A = alice *und* A = bob *aus.*

Die Abbildung 6 (siehe Anhang A) beschreibt die Arbeitsweise des Programms. Die Anpassung der Parameter wurde blau gekennzeichnet. Literale, die nicht mit Regeln der Datenbasis unifiziert werden können, sind rot gekennzeichnet. Literale, die als wahr erkannt wurden beziehungsweise mit den Fakten der Datenbasis unifiziert werden konnten, sind grün gekennzeichnet. Die Reihenfolge des Backtrackings wurde mit ① und ② nummeriert.

Es gibt in Prolog das Prädikat *cut* (Schreibweise „!“), das immer den Wert true hat. Ein cut stoppt die einmal gemacht Auswahl der Klausel ein. Das bedeutet: wenn beim Backtracking ein cut erreicht, dann werden weitere Alternativen ausgeschlossen. Bildlich betrachtet, wird der Baum aller möglichen Lösungswege beschnitten. Cut wird benutzt, um Endlosschleifen zu vermeiden oder um gezielt die Effizienz zu steigern.

Beispiel 23. *Sei die folgende Datenbasis gegeben:*
ancestor(X,Y) :- parent(X,Z), !, ancestor(Z,Y).
ancestor(X,X).
parent(alice,bob).

Bei der Frage ?- ancestor(A.bob)- *gibt das Programm* A = alice *aus. Die Abbildung 7 (siehe Anhang A) beschreibt die Arbeitsweise des Programms. Das Prädikat cut führt zur Beschneidung beim zweiten Backtracking.*

Das Prädikat fail wird zur Erzwingung von Backtracking und zur Ausgabe aller möglichen Ergebnisse angewendet. fail scheitert immer.

In der logischen Programmierung wird *negation as failure*[7] (Schreibweise \+) als Prädikat benutzt, um atomaren Aussagen, die durch die Regeln und Fakten gegebene Resolution nicht beweisen kann, vorerst den Wahrheitswert false zuzuweisen. In Prolog wird das Prädikat \+ im Rumpf eingesetzt und nimmt als Eingabe eine Frage A an. \+ A ist true genau dann, wenn A false ist. Negation as failure wird in Prolog definiert durch eine cut-fail-Kombination realisiert:

```
\+ A :- A, !, fail.
\+ A.
```

Aus ästhetischen Gründen schreiben wir not(A) statt \+ A für eine negation as failure. Wir benutzen den neu kennengelernten Operator im folgenden Beispiel.

Beispiel 24. *Bob isst generell sehr gerne Burger. Er mag aber Burgersorte „Chili-Cheese“ nicht. Wir realisieren dies in Prolog mithilfe des neuen Prädikats:*

Datenbasis: cheese(b_1).
chilicheese(b_2).
ham(b_3).
burger(X) :- cheese(X).
burger(X) :- chilicheese(X).
burger(X) :- ham(X).
moegen(bob,X) :- burger(X), not(chilicheese(X)).
Frage: ?- moegen(bob,b_1).
yes
?- moegen(bob,b_2).
no
?- moegen(bob,b_3).
yes

Negation as failure ist nicht äquivalent zur logischen Negation ¬.

Beispiel 25. *Wenn wir in unserer Burger-Datenbasis aus Beispiel 24 die Frage* ?- moegen(bob,X) *stellen, dann erhalten wir die Sequenz:*

X = b1 ;
X = b3 ;
no

Angenommen, der Operator not ist äquivalent zu ¬. Dann wäre $burger(X) \wedge not(chilicheese(X))$ äquivalent zu $not(chilicheese(X)) \wedge burger(X)$. Dies ist aber nicht der Fall.

Beispiel 26. *Ersetzen wir die Regel aus der Datenbasis aus Beispiel 24* moegen(bob,X) :- burger(X), not(chilicheese(X)) *mit* moegen(bob,X) :- not(chilicheese(X)), burger(X). *Bei der Frage „*?- moegen(bob,X)*“ erhalten wir das folgende Ergebnis:*

no.

In diesem Fall überprüft Prolog zuerst, ob not(chilicheese(X)) wahr ist. Dafür muss überprüft werden, ob chilicheese(X) falsch zurückgibt. Aber chilicheese(b2) ist wahr, sodass not(chilicheese(X)) falsch ist. Die cut-fail-Kombination führt zum Fehlschlag der Frage, wodurch die b1 und b3 nicht mehr betrachtet werden. Im vorherigen Fall

wurde die Variable X zuerst instanziiert und dann not(chilicheese(X)) abgefragt, sodass alle wahren Antworten betrachtet werden können. Der Operator not sollte demnach mit Vorsicht angewendet werden.

Gelfond [8] hat gezeigt, dass eine Verbindung zwischen der autoepistemischen Logik und der logischen Programmierung existiert. Man weist dem negation as failure Operator not(P) die Bedeutung „Es wird nicht geglaubt, dass P der Fall ist“ zu.

Beispiel 27. *Das Individuum weiß aus dem Alltag, dass die meisten Vögel fliegen können. Es glaubt, dass Pinguine die einzigen Vögel sind, die nicht fliegen können. Wir stellen diesen Sachverhalt im Prolog-Programm dar:*
fliegen(X) :- vogel(X), not(pinguin(X)).
vogel(X) :- pinguin(X)

Diese Idee wird in der *Stable Model Semantik*[9] (im Englischen „stable model semantics“) umgesetzt. Die Stable Model Semantik wird in der logischen Programmiersprachen angewendet, um mithilfe der negation as failure die Semantik eines Programms zu bestimmen.

7. Zusammenfassung und Ausblick

Am Anfang wurden die nicht-monotonen Logiken vorgestellt. Wir haben einen kleinen Überblick bekommen, welche Typen von nicht-monotonen Logiken existieren.

Aus den Grundlagen haben wir die Begriffe „Interpretation“ und „Modelle“ angewendet, um günstige Zustände beziehungsweise stabile Expansionen zu definieren. Stabile Expansionen stellen die Vollständigkeit und Korrektheit beziehungsweise Stabilität und Gebundenheit sicher. Dies sind essentielle Eigenschaften der autoepistemischen Logik.

Mithilfe der S5-Strukturen haben wir eine Methode gefunden, autoepistemische Theorien aus einem modallogischen Modell zu generieren und strukturierter darzustellen. Dabei wurden ähnlich wie in Kapitel 2 Interpretationen und Modelle benutzt, um stabile Expansionen zu bilden.

Wir haben einen Algorithmus definiert, der bestimmen kann, ob eine Formel einer Prämissenmenge in jeder stabilen Expansion enthalten sein wird. Dieser ist leider „ineffizient“, sodass wir die Sprache der Logik beschränken mussten. Wir haben ein Transformationsverfahren angewendet und haben mit den transformierten Mengen einen effizienteren Algorithmus definiert.

Die autoepistemische Logik ist in der Technik nicht weit verbreitet. Dies liegt zum Teil an der in Kapitel 5 angesprochenen Thematik, dass die Prämissenmenge zuerst transformiert werden muss, bevor man den effizienten Algorithmus anwendet. Es wurde von Moore vorgeschlagen, dass die autoepistemische Logik bei Systemen

angewendet wird, die sich bewusst sind, dass ihr Wissen über Daten unvollständig sein kann. Wir haben diesen Sachverhalt für die logische Programmiersprache Prolog gezeigt. Das Prädikat *negation as failure* kann so wie ein negierter Glaubensoperator interpretiert werden.

In anderen Bereichen hat sich die autoepistemische Logik nicht durchgesetzt, weil es wenig Systeme gibt, die unvollständige Informationen berücksichtigen müssen, und folglich ihr Verhalten anpassen.

Die Stable Model Semantik hat eine Basis für ein neues Programmierparadigma gelegt, das *Answer Set Programming*. Answer Set Programming wird für schwierige Suchprobleme angewendet. Dabei wird versucht, die Suchprobleme in sogenannte stabile Modelle zu reduzieren. Mehr dazu ist im Paper von Lifschitz[10] zu finden.

C. J. van Rijsbergen[11] schlug vor, im Information Retrieval die Anfragen (Queries) als eine Prämissenmenge zu interpretieren, und hat eine neues Bewertungskriterium zur Suche von Dokumenten abhängig einer Query vorgestellt. Dieses Bewertungskriterium wendet Ideen der autoepistemischen Logik an.

In Kapitel 5 haben wir unsere Logik auf die Prädikatenlogik mit der Anwendung des Glaubensoperators in geschlossenen Formeln beschränkt, um unsere Logik entscheidbar zu machen. Es wäre interessant zu sehen, ob andere Begrenzungen angewendet werden können, um die Logik entscheidbar zu machen.

Literatur

[1] Robert C. Moore. “Autoepistemic Logic”. In: *Non-Standard Logics for Automated Reasoning*. Hrsg. von Philippe Smets. Academic Press, 1988, S. 105–136.

[2] Robert C. Moore. “Semantical Considerations on Nonmonotonic Logic”. In: *Artif. Intell.* 25.1 (1985), S. 75–94. DOI: 10.1016/0004-3702(85)90042-6. URL: https://doi.org/10.1016/0004-3702(85)90042-6.

[3] Robert Stalnaker. “A Note on Non-Monotonic Modal Logic”. In: *Artif. Intell.* 64.2 (1993), S. 183–196. DOI: 10.1016/0004-3702(93)90104-J. URL: https://doi.org/10.1016/0004-3702(93)90104-J.

[4] Dirk van Dalen. *Logic and structure (3. ed.)* Universitext. Springer, 1994. ISBN: 978-3-540-57839-0.

[5] Robert C. Moore. “Possible-World Semantics for Autoepistemic Logic”. In: *Proceedings of the Non-Monotonic Reasoning Workshop, Mohonk Mountain House, New Paltz, NY 12561, USA, October 17-19, 1984*. American Association for Artificial Intelligence (AAAI), 1984, S. 344–354.

[6] William F. Clocksin und Christopher S. Mellish. *Programming in Prolog (4. ed.)* Springer, 1994. ISBN: 978-3-540-58350-9.

[7] Johan Bos Patrick Blackburn und Kristina Striegnitz. *Learn Prolog Now! (Vol.7)*. College Publications, 2006.

[8] Michael Gelfond. "On Stratified Autoepistemic Theories". In: *Proceedings of the 6th National Conference on Artificial Intelligence. Seattle, WA, USA, July 1987*. Hrsg. von Kenneth D. Forbus und Howard E. Shrobe. Morgan Kaufmann, 1987, S. 207–211. URL: http://www.aaai.org/Library/AAAI/1987/aaai87-037.php.

[9] Michael Gelfond und Vladimir Lifschitz. "The Stable Model Semantics for Logic Programming". In: *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*. Hrsg. von Robert A. Kowalski und Kenneth A. Bowen. MIT Press, 1988, S. 1070–1080.

[10] Vladimir Lifschitz. "What Is Answer Set Programming?" In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*. Hrsg. von Dieter Fox und Carla P. Gomes. AAAI Press, 2008, S. 1594–1597. URL: http://www.aaai.org/Library/AAAI/2008/aaai08-270.php.

[11] C. J. van Rijsbergen. "A Non-Classical Logic for Information Retrieval". In: *Comput. J.* 29.6 (1986), S. 481–485. DOI: 10.1093/comjnl/29.6.481. URL: https://doi.org/10.1093/comjnl/29.6.481.

Anhang

A. Zusätzliche Abbildungen

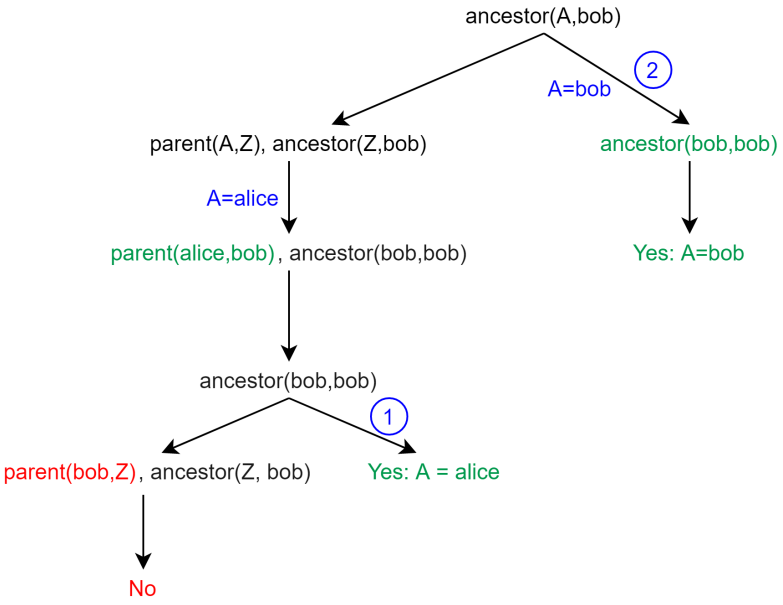

Abbildung 6: Berechnungsschritte in Prolog

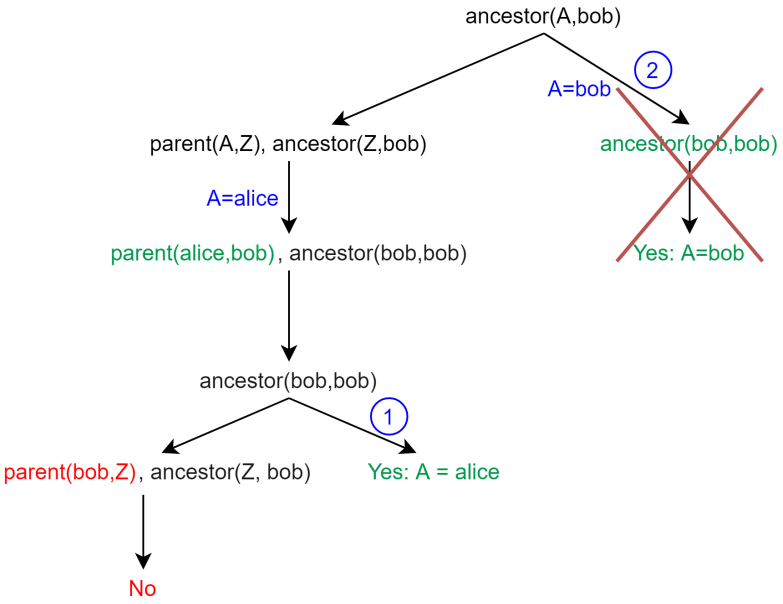

Abbildung 7: Verkleinerung des Suchraums durch cut

B. Erklärung selbstständiger Arbeit

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst habe, keine anderen Quellen und Hilfsmittel als angegeben verwendet habe, alle Ausführungen, die wörtlich oder sinngemäß aus anderen Quellen entnommen sind, als solche kenntlich gemacht habe und diese Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen habe.

Ort, Datum Unterschrift