Einsatz von Wissensrevision für das robuste Verhalten eines navigierenden Agenten

Diplomarbeit am Arbeitsbereich
Wissens- und Sprachverarbeitung
des Departments Informatik
der Universität Hamburg

von

Jens Wächter,
Wolffsonweg 5,
22297 Hamburg

Betreuerin: Dr. Carola Eschenbach
Arbeitsbereich Wissens- und Sprachverarbeitung

Zweitbetreuer: Prof. Dr. Christopher Habel
Arbeitsbereich Wissens- und Sprachverarbeitung

19. März 2009

Inhaltsverzeichnis

Kapitel 1

Einleitung

Mobile Agenten, die über die Fähikgeit verfügen, auf Basis einer von außen an sie heran getragenen Routenbeschreibung (nachfolgend auch Routeninstruktion genannt) in einer unbekannten Umgebung zu navigieren, sind verschiedenen Problemen ausgesetzt. Zum einen müssen sie ein mentales Modell der Routeninstruktion aufbauen. Zusätzlich muss der Agent in der Lage sein, die Umwelt wahrzunehmen und sie in eine Form bringen, in der die Intergration der Perzeption mit der Instruktion möglich ist. Ein Agent, der diese Fähigkeiten besitzt, wird in [TSEHK 2003] vorgestellt. Der in [TSEHK 2003] vorgestellte Agent wird Geometrischer Agent genannt. Nachfolgend soll mit *der Agent*, wenn von ihm in einem eindeutigen Kontext gesprochen wird, der Geometrische Agent bezeichnet werden.

Der Geometrische Agent navigiert mithilfe einer im voraus gegebenen Routeninstruktion. Die Routeninstruktion wird dabei in räumliche und imperative Anteile zerlegt. Räumliche Anteile werden durch graphenartige Strukturen (Conceptual Route Instruction Language), Imperative durch einen Aktionsplan repräsentiert. Die Abarbeitung des Aktionsplans orientiert sich an den Schritten, die im Aktionsplan festgelegt sind. Aufgrund der Mehrdeutigkeit und Vahgeit von Routenbeschreibungen existieren bei der Integration von Instruktion und Perzeption an vielen Stellen verschiedene Wahlmöglichkeiten, die dazu führen, dass der Agent unterschiedliche Wege benutzen kann.

Im Rahmen der Navigation kann es vorkommen, dass an einer Stelle eine Integration von Perzeption und Instruktion nicht mehr möglich ist. In seinem aktuellen Zustand bricht der Agent die Verarbeitung der Routeninstruktion ab. Das Ziel dieser Diplomarbeit ist es, den Agenten zu ermöglichen, seine Instruktion auch in dem Fall weiter zu verfolgen, in dem eine Integration von Perzeption und Instruktion nicht mehr möglich ist.

Die Ursachen für die Unmöglichkeit der Integration von Instruktion und Perzeption können sehr vielfältig sein: Es kann die Instruktion falsch oder ungenügend sein, es kann das Verständnis des Agenten von der Instruktion falsch oder unzureichend sein. Es kann sein, dass die Perzeption unzureichend ist. Es kann aber auch sein, dass der Agent während der Navigation eine Fehlentscheidung getroffen hat, wodurch er, umgangsprachlich ausgedrückt, falsch abgebogen ist. Diese Vielzahl an möglichen Gründen der Unmöglichkeit der Integration von Perzeption und Instruktion werden innerhalb dieser Diplomarbeit untersucht. Das Ergebnis der Untersuchung ist, dass die Revision einer vorangegangenen Entscheidung, einen bestimmten Weg zu gehen, die naheliegendste Lösung des Problems ist.

Die Revision einer Entscheidung umfasst zwei Aspekte: zum einen muss die aus der Entscheidung resultierende Handlung rückgängig gemacht werden, zum anderen muss der Agent in der Lage sein, aufgrund der Revision sein mentales Modell zu änderen, um ein angemessenes Verhalten zu zeigen. Es ist beispielsweise nicht angemessen, dass der Agent, nachdem der falsch abgebogen ist, zu dem Punkt zurückkehrt, an dem er abgebogen ist, um im Anschluss an die Rückkehr wieder den gleichen Weg einzuschlagen. Diese Diplomarbeit ist auf Spezifikation der Revision des mentalen Modells beschränkt, was insbesondere bedeutet, dass die Revision der Handlung des Agenten *nicht* behandelt wird.

Um die Revision des mentalen Modells, was den Agenten dazu veranlasst hat, falsch abzubiegen, durchzuführen, ist es notwendig einen geeigneten Mechanismus zu finden, der die Revision des mentalen Modells durchführt. In der wissenschaftlichen Literatur wird das Thema als „Wissensrevision“ bezeichnet. Innerhalb der Wissensrevision werden Operatoren benutzt, die ein durch eine Integritätsbedingung eingeschränktes mentales Modell unter Berücksichtigung verschiedener Rationalitätskriterien in einen neuen Zustand, der wiederum die Integritätsbedingung erfüllt, überführen. Die Integritätsbedingung und die Rationalitätskriterien werden in Kapitel 3 vorgestellt. Dabei gibt es eine Vielzahl von möglichen Operatoren, die zwar die Rationalitätskriterien erfüllen, aber in verschiedenen Bereichen sich voneinander unterscheiden. Insbesondere sind die Operatoren mit verschiedenen Repräsentationsformen des mentalen Modells verknüpft. Um die Revision der Entscheidung des Agenten durchzuführen ist deshalb eine Untersuchung möglicher Wissensrevisionsoperatoren notwendig, vor allem, um einen Operator zu finden, der sich implementieren läßt. Die Implementationsfähikgeit hängt in erster Linie davon ab, ob der Operator für ein *endliches* mentales Modell definiert ist. Neben der Implementationsfähigkeit ist die Frage, ob das mentale Modell, auf dem der Revisionsoperator aufbaut, auch eine sinnvolle Repräsentation von Wissen oder Belief erlaubt.

Struktur der Diplomarbeit

In Kapitel 2 wird die Grundstruktur des Geometrischen Agenten sowie einige, in den nachfolgenden Kapiteln wichtige, Details ausgeführt. Die Darstellung ist dabei kurz gehalten. Für einen Einstieg in den Geometrischen Agenten ist es empfehlenswert, [TSEHK 2003] zu lesen. Die Diplomarbeit greift auch auf Details aus [H 2003] und [B 2005] zurück. Um ein komplettes Verständnis der von mir in Kapitel 2 zusammengefassten Details über bestimmte Aspekte des Agenten zu erhalten, kann es daher sinnvoll sein, auch den Inhalt der beiden genannten Diplomarbeiten zu kennen.

In Kapitel 3 wird eine Einführung in die Wissensrevision gegeben. Da Wissensrevisionsoperatoren eine logische Sprache verwenden, wird zunächst die Aussagenlogik, die auch in den folgenden Kapiteln Verwendung finden wird, formal eingeführt. Es folgt eine Darstellung der Grundlagen zur Wissensrevision. Auf Basis der Grundlagen der Wissensrevision werden in der Literatur beschriebene Wissensrevisionsoperatoren vorgestellt und in Hinblick auf ihre Verwendbarkeit zur Durchführung der Entscheidungsrevision bewertet. Ein Sonderfall ist dabei die Vorstellung der auch in Kapitel 4 verwendeten Finite Partial Entrenchment Rankings, da direkt an die Vorstellung der Finite Partial Entrenchment Rankings auch eine Optimierung eingeführt wird, die nicht aus der Literatur übernommen ist.

In Kapitel 4 wird zunächst die grundlegende Frage geklärt, welche der möglichen Ursachen für die Unmöglichkeit der Integration von Perzeption und Instruktion behandelt werden soll. Da die Wahl auf die Revision der bereits getroffenen Entscheidungen während des Navigierens fällt, wird im Rest von Kapitel 4 ein Revisionsmodul vorgestellt, was die Revision des mentalen Modells des Agenten durchführen kann.

Die Auswertung des Revisionsmoduls in Hinblick auf ihre sowohl qualitative wie auch quantitative Leistungsfähigkeit erfolgt in Kapitel 5. Die Auswertung soll insbesondere zeigen, dass die quantitative Leistungsfähigkeit (Performanz) auch in komplexen Beispielen angemessen ist.

Eine Zusammenfassung des in der Diplomarbeit vorgestellten Revisionsmoduls, seiner Leistungsfähigkeit und ein kurzer Ausblick auf mögliche ansetzende Forschung wird in Kapitel 6 gegeben.

Kapitel 2

Der geometrische Agent

2.1 Übersicht

Das Problem, in einer unbekannten Umgebung von einem Punkt ausgehend zu einem anderen zu gelangen, kann von Menschen unter Verwendung einer verbalen Routenbeschreibung eines Ortskundigen gelöst werden. Dass Erreichen des designierten Ziels hängt dabei davon ab, ob der sich bewegende Mensch die Routenbeschreibung versteht und unter Einbeziehung von räumlichem Wissen während der Navigation die Wegbeschreibung mit den wahrgenommenen Objekten in Übereinstimmung bringen kann. Prinzipiell kann ein Mensch die Wegbeschreibung im Vorauss erhalten oder während des Wegs zum Ziel. Der geometrische Agent (nachfolgend auch kurz als „GA“ oder „der Agent“ bezeichnet) ist nach [TSEHK 2003] eine Simulation eines nach einer im voraus gegebenen Routeninstruktion navigierenden Agenten in einer virtuellen Umgebung. Die Simulation umfasst dabei in der kognitiven Verarbeitung niedrig stehende Prozesse, wie z.B. das Erkennen von Objekten, die nicht detailliert durch den Agenten untersucht werden sollen. Die Simulation der niedrigen kognitiven Prozesse erlaubt es dem Agenten, sich auf die Simulation der in der kognitiven Hierarchie hochstehenden Prozesse zu beschränken. Der geometrische Agent ist als Plattform für die Untersuchung der höheren kognitiven Funktion konzipiert. Die Realisierung der Revision des mentalen Modells soll daher ebenfalls an en höheren kognitiven Prozessen orientiert sein und über die rein technische Lösung des Problems hinausgehen.

Kategorisierung des geometrischen Agenten anhand der Agententheorie

Als Agent kann nach [RN 2003], Kapitel 2, alles bezeichnet werden, was seine Umgebung durch Sensoren wahrnimmt und durch Aktuatoren verändert, weshalb der geometrische Agent auch zurecht als solcher bezeichnet wird. Nach der von Russell und Norvig vertretenen Agententheorie besitzen intelligente Agenten ein Performanzmaß. Die Maximierung dieses Performanzmaßes ist das Ziel eines rational handelnden Agenten. Als Performanzmaß für den geometrischen Agenten bietet sich der Vergleich des Verhaltens bei unterschiedlichen Navigationsdurchläufen an. Bittkowski skizziert in [B 2005] zwei verschiedene Szenarien, unter denen die Performanz des Agenten gemessen werden kann: zum einen die Verarbeitung unterschiedlicher Routeninstruktionen unter Verwendung der gleichen Methoden zur Verarbeitung der Routeninstruktion, zum anderen die Verwendung der gleichen Routeninstruktion unter Nutzung unterschiedlicher Methoden zur Verarbeitung. Da es Ziel dieser Diplomarbeit ist, die Methoden des Agenten zur Verarbeitung der Routeninstruktion zu verbessern ist es sinnvoll, die Performanz des Agenten nach dem zweiten Szenario zu messen. Allerdings soll das oben genanntem Szenario differenzierter gesehen werden, da mit den Methoden der Verarbeitung der Routeninstruktion auch eine bestimmte Repräsentationsform der Routeninstruktion und der perzipierten Umwelt verbunden ist, die ebenfalls Einfluss auf die Performanz hat. Somit ist es möglich, die gleichen Methoden zur Verarbeitung des Agenten zu verwenden, jedoch das Wissen über Instruktion und Perzeption unterschiedlich zu repräsentieren und durch die unterschiedliche Repräsentationsform eine unterschiedliche Performanz des Agenten mitverantworten können.

Zur Frage, welche Kriterien der Agent zur Beurteilung der Performanz verwenden sollte, beschreibt Bittkowski informell ein Performanzmaß, das zunächst das Erreichen des Zielobjekts, zu dessen Erreichung die Instruktion aufgestellt wurde, als Hauptfaktor gelten sollte. Für den Fall, dass der Agent das Ziel erreicht solle zusätzlich die zurückgelegte Strecke als Kriterium zur Beurteilung der Performanz hinzugezogen werden. Ich möchte mich der Wahl der oben genannten Kriterien anschließen, allerdings diese noch um das Kriterium der Dauer der Verarbeitung der Routeninstruktionen ergänzen. Insbesondere ist es kein rationales Verhalten, wenn der Agent bei der Verarbeitung der Instruktion lange Phasen des Überlegens einlegt und dadurch das Ziel später erreicht.

Die Umgebung eines Agenten wird nach [RN 2003] ab Seite 41 anhand verschiedener Kriterien kategorisiert. Eine Kategorisierung der Umgebung des Agenten nach den Kriterien aus [RN 2003] hat Bittkowski in [B 2005] vorgenommen. Die Untersuchung der Umgebung des Agenten wurde mit folgendem Ergebnis abgeschlossen:

- Die Umgebung des geometrische Agenten ist nur partiell beobachtbar, so dass er kein komplettes Wissen über seine Umgebung besitzt.
- Der Folgezustand der Umgebung des Agenten ist ausschließlich vom gegenwärtigem Zustand und den Aktionen des Agenten abhängig. Eine Umgebung mit diesen Eigenschaften wird deterministisch genannt. Aus der Determiniertheit der Umgebung folgt auch, dass die Umgebung statisch ist.
- Die Verarbeitung der Instruktion des Agenten erfolgt nach einem sequentiellem Muster: die Entscheidung für eine Instruktion hängt auch von der Perzeption in vorangegangen Zeitpunkten ab.
- Der Agent nimmt seine Umgebung nur zu besonderen Zeitpunkten wahr. die Wahrnehmung des Agenten ist damit diskret und nicht kontinuierlich.
- Der Agent ist der einzige Agent innerhalb seiner Umgebung.

Die simulierte Umgebung umfasst das Informatikum der Universität Hamburg. Die Umgebung und die Wahrnehmung des Agenten ist konfigurierbar. So ist in der momentanen Implementation wählbar, ob der Agent Bezeichnungen z.B. von Häusern sehen kann oder nicht. Auch kann zwischen Sommer und Winter gewählt werden, was Einfluss darauf hat, ob der Agent durch Bäume sehen kann oder nicht.

Einteilung in zwei Phasen

Der Agent erhält zunächst eine in natürlicher Sprache formulierte Routeninstruktion. Die natürlichsprachliche Routeninstruktion wird zunächst in eine interne Repräsentation überführt. Dieser Vorgang wird als Instruktionsphase bezeichnet. Der Instruktionsphase angeschlossen folgt die Navigationsphase, die den eigentlichen Navigationsvorgang nach Angaben der natürlichsprachigen Instruktion simuliert.

Die Instruktionsphase umfasst im wesentlichen die Umwandlung der natürlichsprachlichen Routeninstruktion in eine aus zwei Komponenten bestehende interne Repräsentationsform: zum einen wird ein Conceptual Route Instruction Language- Graph (nachfolgend kurz CRIL), der die in der Instruktion enthaltenen Beschreibung der Umgebung beinhaltet, zum anderen ein Aktionsplan bestehend aus imperativen Anweisungen zur Navigation auf höherer Ebene erzeugt. Bei der Umwandlung der natürlichsprachlichen Instruktion in das interne Modell findet dabei zunächst eine syntaktische und semantische Verarbeitung unter Verwendung eines Lexikons mit räumlichem Wissen statt (siehe Abbildung 2 in [TSEHK 2003]). Dabei wird deklaratives Wissen über die (räumliche) Beschreibung der zu erwartenden Szenerie, was die Basis für den CRIL-Graphen wird, von imperativen Anweisungen, die die Basis des Aktionsplanes werden, getrennt. Das Lexikon über räumliches Wissen, was bei der syntaktischen und semantischen Verarbeitung herangezogen wird, basiert nach [TSEHK 2003] auf linguistischen Untersuchungen zu räumlichen Ausdrücken. Die Semantik räumlicher Ausdrücke wird dabei mithilfe eines Inventars deskriptiver Operatoren erfasst. Die deskriptiven Operatoren gruppieren dabei bestimmte Verben nach semantischen Kategorieren, wie z.B. Verben der Bewegung nach ihrer semantischen Komponente GO(X,W), die aussagt, dass X den Weg W gehen soll.

Neben der semantischen Analyse von Verben werden mithilfe des Lexikons über räumliches Wissen auch „tracks“ sowie Bewegungspfade („paths of motion“) erkannt, die eine gerichtete Bewegung zwischen einem Start- und einem Endpunkt beschreiben. Bewegungspfade besitzen Endpunkte, die Entscheidungspunkte sein können. Entscheidungspunkte sind ein Bestandteil von Routeninstruktionen und markieren Punkte, an denen die instruierte Person zwischen unterschiedlichen folgenden Bewegungspfaden wählen kann.

Wenn die Instruktion innerhalb der Instruktionsphase verarbeitet ist, kann der Agent in der Navigationsphase anhand des Aktionsplanes und des CRIL-Graphen sowie der wahrgenommenen Umgebung, die ebenfalls in eine abstrakte, dem in der Instruktionsphase erstellten CRIL-Graphen ähnliche, Form umgewandelt wird, einen Weg zum Ziel finden. Ein wesentlicher kognitiver Prozess während der Navigationsphase ist dabei die Integration von Perzeption und Instruktion, in der versucht wird, die in der Instruktion enthaltene Beschreibung der Umgebung in der wahrgenommenen Umwelt wiederzufinden. Falls eine Übereinstimmung gefunden wird, so wird dies durch eine Koreferenz zwischen Knoten des Instruktions- und Knoten des Perzeptionsgraphs repräsentiert.

2.2 Repräsentiertes Wissen des geometrischen Agenten

2.2.1 Taxonomien

Taxonomien sind im allgemeinen Fall Einteilungen von Gegenständen in Kategorien. Nach [H 2003](S. 14ff.) sind Is-A-Taxonomien Vererbungshierarchien, durch die eine Zugehörigkeit eines Gegenstands zu einer allgemeineren Kategorie von Gegenständen ausgedrückt werden können. Durch die Hierarchisierung der in der Is-A-Taxonomie repräsentierten Gegenstände ist es möglich, Ähnlichkeiten zwischen den in den Taxonomien repräsentierten Gegenständen zu erfassen. Für den Agenten sind spezifischere Taxonomien notwendig, insbesondere die von räumlichen Konzepten. Der Begriff des Konzepts stammt aus der Repräsentationssprache Semantic Representation Language (SRL). Die SRL soll an dieser Stelle nicht formal eingeführt werden. Die SRL wird in [H 1986] im Rahmen von referentiellen Netzen vorgestellt. Habel gibt keine formale Definition von Konzepten. Er sieht allerdings, wie auch in [H 2003] zusammengefasst, Konzepte zum einen als strukturierte Gesamtheiten von Wissen und den Beziehungen zu anderen Konzepten zum anderen als kognitive Entitäten. In der Vererbungshierarchie einer Taxonomie wird nachfolgend das höchststehendste Konzept, von dem alle anderen Konzepte erben, mit $\top$ bezeichnet.

Werden räumliche Konzepte durch die Taxonomie beschrieben, so läßt sich auch die Ähnlichkeit der räumlichen Konzepte bestimmen. Die Möglichkeit, die Ähnlichkeit von räumlichen Konzepten zu bestimmen, ist eine der Grundlagen zur Integration von Instruktion und Perzeption, die Helwich in [H 2003] vorstellt. Da Taxonomien für den Agenten in erster Linie als Mittel zur Integration von Perzeption und Instruktion dienen, soll die formale Definition von Taxonomie aus [H 2003] (Definition 3.2.1) verwendet werden, ohne sie explizit in hier aufzunehmen. Eine wesentliche Eigenschaft der Taxonomie-Definition von Helwich ist die Antisymmetrie der Subsumptionsrelation, die verhindert, dass es Zyklen in der Subsumptionsbeziehung geben kann.

Da die Subsumptionsbeziehung alleine zur Berechnung von Ähnlichkeitsmaßen unzureichend ist, führt Helwich die Pfadlänge von Subsumptionsgraphen ein, die als Maß der Entfernung zwischen zwei Konzepten dient, da unter Verwendung von alleine der Subsumptionsbeziehung die Entfernung zwischen einem Konzept c und einem von c subsumierten Konzept identisch mit der Entfernung von Konzept c zu sich selbst ist. Die formale Definition der Entfernung $\text{entfernung}_\top$ zwischen zwei

Konzepten findet sich in Definition 3.2.5 (S. 44). Neben der Entfernung zwischen zwei Konzepten ist auch die Tiefe $\text{tiefe}_\top$ eines Konzepts c von Bedeutung, die Helwich in Definition 3.2.6 als die Anzahl der Elemente des kürzesten Pfads zwischen $\top$ und c definiert.

Mithilfe der oben genannten Definitionen stellt Helwich in Definition 4.2.6. ein Ähnlichkeitsmaß für zwei Konzepte vor, wobei die Ähnlichkeit 0 im Fall des gegenseitigem Ausschlusses beider Konzepte oder wenn der spezifistische gemeinsame Vater $\top$ ist. Ist beides nicht der Fall, so wird die Ähnlichkeit im wesentlichen durch die Tiefe des gemeinsamen Vaters berechnet.

2.2.2 CRIL-Graphen

Conceptual Route Instruction Language (CRIL)-Graphen sind eine Repräsentationsform von Routeninstruktionen. Sie sind strukturgleich zu referentiellen Netzen, die aber nicht Formal eingeführt werden sollen. Eine Einführung in referentielle Netze gibt es unter anderem in [H 1986]. Die wesentlichen Eigenschaften von CRIL-Graphen bestehen darin, dass CRIL-Knoten ein Referenzobjekt, wie z.B. ein in der Instruktion erwähntes oder ein perzipiertes Haus repräsentieren, CRIL-Kanten dagegen Relationen. Knoten können darüber hinaus mit einer Symbolmenge ergänzt werden, die Attribute, einstellige Prädikate oder den Namen des Referenzobjekts repräsentieren. In der formalen Definition von CRIL-Graphen in [H 2003] werden zwei Taxonomien vorausgesetzt: zum einen eine Taxonomie der Attribute, zum anderen eine Taxonomie der Konzepte. Diese Taxonomien sollen insbesondere den Prozess der Integration von Instruktion und Perzeption unterstützen, indem sie die Berechnung von Ähnlichkeiten der Konzepte aus Instruktion und Perzeption ermöglichen.

Helwich führt zwei CRIL-Graphen zur Unterscheidung zwischen Instruktion und Perzeption ein: den I-Graphen, der im wesentlichen das deklarative Wissen aus der Instruktion repräsentiert, und den P-Graphen, der das Wissen aus der Perzeption des Agenten repräsentiert. Der I-Graph kann Knoten der Perzeption enthalten, allerdings müssen alle Knoten der Perzeption über eine Koreferenzkante mit einem Knoten aus der Instruktion verbunden sein. In der in Kapitel 4 spezifizierten Wissensbasis werden in erster Linie Koreferenzen von Bedeutung sein. Da Koreferenzen nur zwischen Knoten der Instruktion und der Perzeption gezogen werden, soll an dieser Stelle nur die Menge der Knoten aus der Instruktion und die Menge der Knoten aus der Perzeption formal eingeführt werden.

Definition 1 (Menge der Knoten aus der Instruktion) *Die Menge aller Knoten aus dem Instruktionsgraph des geometrischen Agenten sei* $\mathbb{I}$.

Definition 2 (Menge der Knoten aus der Perzeption) *Die Menge aller Knoten aus dem Perzeptionsgraph des geometrischen Agenten sei* $\mathbb{P}$.

2.2.3 Aktionsplan

Der Aktionsplan umfasst die imperativen Anteile des Instruktionsmodells, die während der Instruktionsphase von den deklarativen Anteilen der Instruktion getrennt wurden. In [B 2005] führt Bittkowski in Tabelle 5.1 auf Seite 78 das Inventar der Anweisungen, die innerhalb des Instruktionsplans Verwendung finden können, auf:

1. Die Anweisung !GO(w) weist den Agenten an, sich auf dem durch w bezeichneten Pfad zu bewegen.

2. !CH_ORIENT(w) läßt den Agenten sich in Richtung des Pfades w drehen, so dass die Ausrichtung des Agenten mit der Richtung von w übereinstimmt.

3. !BE_AT(r) veranlasst den Agenten, sich zu versichern, innerhalb der Region r zu befinden. Falls der Agent sich nicht innerhalb von r befindet, so begibt er sich in die Region r.

4. !VIEW(o) stellt sicher, dass der Agent das Objekt o perzipiert.

Die in dem Aktionsplan aufgenommen Instruktionsanweisungen sind nicht als primitive Anweisungen, die direkt ausgeführt werden können, zu verstehen. Sie sind vielmehr eine Repräsentationsform der verbalen Instruktion, die allerdings unterspezifiziert ist. Aufgrund der Unterspezifiziertheit können die Anweisungen nicht direkt ausgeführt werden. Statt dessen wird während der Navigationsphase eine Folge primitiver Aktionen als Ergebnis der Interpretation der aus der verbalen Instruktion stammenden imperativen Anweisungen im Kontext der gegenwärtigen Situation des Agenten erstellt (siehe dazu Abschnitt 3.3.4 in [B 2005]). Diese Folge von Anweisungen wird „Lokale Aktionssequenz“ genannt.

Die 4 Anweisungen aus dem Inventar des Aktionsplans haben die Eigenschaft, dass sie jeweils einen Parameter erwarten, beispielsweise erwartet !GO einen Weg, dem der Agent folgen soll. Dieser Parameter soll nachfolgend als das „Ziel“ einer Aktion bezeichnet werden:

Definition 3 (Ziel einer Aktion) *Sei a eine Aktion* ($a \in \{$*!GO(x), !VIEW(x), !BE_AT(x), !CH_ORIENT(x)*$\}$)*. Dann sei das Ziel x der Aktion mit* $t(a)$ *bezeichnet.*

In Anhang D sind drei Aktionspläne als Beispiel aufgenommen.

2.3 Navigationsphase

Wenn die natürlichsprachliche Instruktion in der Instruktionsphase verarbeitet wurde, so liegen nun der CRIL-Graph, der das deklarative Wissen der Instruktion enthält, sowie der Aktionsplan mit dem imperativen Inhalten der Instruktion dem Agenten vor. Der Agent kann nun auf Basis von CRIL-Graph und Aktionsplan mit der Navigationsphase beginnen, während der er sich in seiner virtuellen Umgebung nach der Anleitung des Aktionsplans bewegt. Die virtuelle Umgebung des Agenten basiert nach [TSEHK 2003] auf einer planaren euklidischen Geometrie, die über ein absolutes Koordinatensystem verfügt. Objekte sind durch Punkte, Strahlen oder Polygone repräsentiert und besitzen zudem einige nicht geometrische Attribute, wie die Zuordnung eines Namens oder die Einordnung in eine Kategorie aus den Taxonomien. Innerhalb der virtuellen Umgebung besitzt der Agent eine Pose, die es erlaubt, das Sichtfeld des Agenten zu bestimmen und so die Perzeption zu simulieren. Die Perzeption des Agenten ist konfigurierbar, so dass es z.B. wählbar ist, ob der Agent die Namen von Häusern erkennen soll oder nicht. Die simulierte Perzeption erstellt den Perzeptionsgraphen des Agenten. Der Perzeptionsgraph weist dabei ein Artefakt auf: Wenn ein Objekt a durch ein anderes teilweise verdeckt wird, so wird innerhalb des Perzeptionsgraphen a durch mehrere Knoten repräsentiert, die auf dieser Ebene der Repräsentation identisch sind. Wenn beispielsweise das Haus C perzepiert wird, aber ein Baum sich direkt zwischen dem Agenten und Haus C befindet, dann werden innerhalb des Perzeptionsgraphen zwei Knoten existieren, die ein Haus mit zwei Stockwerken und der Aufschrift „Haus C“ repräsentieren.

Der Agent führt die Perzeption nicht kontinuierlich, sondern nur an bestimmten Orten, insbesondere Entscheidungspunkten durch. Ist die Perzeption durchgeführt worden, so verfügt der Agent über eine interne Repräsentation der virtuellen Umgebung. Auf Basis der wahrgenommenen Szene kann der Agent nun nach [TSEHK 2003] drei zentrale Aufgaben zur Navigationssteuerung ausführen:

1. Die Bildung von Koreferenzen zwischen den Perzeptions- und Instruktionsgraphen. Falls es keine sinnvollen Möglichkeiten zur Koreferenzbildung gibt, kann der Agent die nachfolgenden Aktionen nicht durchführen. Die Kriterien zur Bildung von Koreferenzen und werden in Abschnitt 2.4 detailliert ausgeführt.

2. Die Selbslokalisierung des Agenten in Hinblick auf den Fortschritt innerhalb des Modells der Umgebung. Die Selbstlokalisierung wechselt unter Einbeziehung der aktuell wahrgenommenen Umgebung den gegenwärtigen Knoten innerhalb des Instruktionsgraphen. Dieser Prozess verlangt, dass Koreferenzen im vorangegangenen Arbeitsschritt gebildet werden konnten.

3. Aufbau eines Aktionsplanes von einfachen Aktionen, die auf niedriger Navigationsebene ausgeführt werden können. Ein Grundproblem dabei ist, dass in natürlichsprachlichen Routenbeschreibungen nicht notwendigerweise alle Entscheidungspunkte oder der exakte Verlauf der zu

wählenden Straßenzügen beschrieben ist. Während des Aufbaus des Aktionsplans auf niedriger Ebene werden aus dieser Unterspezifiziertheit der Instruktion resultierende Probleme, wie z.B. die fehlende Sichtbarkeit des nächsten Entscheidungspunktes, miteinbezogen. Nach [B 2005] werden die Probleme der Unterspezifiziertheit der Instruktion unter anderem durch Folgeaktionen behoben. In Abschnitt 5.7 in [B 2005] werden Folgeaktionen ausgeführt, wenn die Vorbedingungen zur Ausführung der nachfolgenden Aktion nicht hergestellt werden können. Die Nichterfüllung der Vorbedingungen kann den Grund haben, dass das in den Vorbedingungen gesuchte Objekt, das ich als Ziel einer Aktion bezeichnet habe, entweder zuweit entfernt ist oder außerhalb des Sichtbarkeitsradius liegt. Bittkowski führt zur Herstellung der Vorbedingungen deshalb drei Typen von Folgeaktionen ein: Umherblicken, zufällige Exploration und Folgen der Fortsetzung eines Pfades. Das Folgen der Fortsetzung eines Pfades beruht darauf, dass der Agent bei der Planung der primitiven Aktionen im Fall der Wahlmöglichkeit von verschieden langen Pfaden zur Optimierung des Performanzmaßes „zurückgelegte Wegstrecke“ zunächst die kurzen Pfade präferiert. Falls es längere Strecken als die gewählte Strecke gibt und die längeren Strecken die gewählte kürzere Strecke beinhalten, so bezeichnet Bittkowski die langen Strecken als Fortsetzungen der kurzen Strecken (siehe Abschnitt 5.4.2 in [B 2005]).

Im Abschluss an diese drei Prozesse werden die nicht koreferenzierten Teile der Perzeption verworfen.

Im Rahmen des in dieser Diplomarbeit zu spezifizierenden Revisionsmoduls des Agenten kommt den Posen, den Zeitpunkten, den Entscheidungspunkten bzw. den Situationen, in denen der Agent die Umgebung perzipiert und die drei oben aufgeführten Prozesse durchläuft, eine besondere Bedeutung zu, da der in dieser Arbeit vorgestellte Revisionsmechanismus eine Fehlentscheidung des Agenten während der Navigationsphase korrigieren soll. Eine Fehlentscheidung des Agenten wird immer unter bestimmten Umständen getroffen, weshalb es wichtig ist, diese Umstände zu untersuchen und einen passenden Begriff zur Bezeichnung dieser Umstände zu finden. Um einen eindeutigen Sprachgebrauch zu gewährleisten muss deshalb geprüft werden, welcher der oben aufgeführten Begriffe als Beschreibung der Umstände, unter denen perzipiert wird, verwendet werden sollte. Es hat sich herausgestellt, dass der Begriff Situation am sinnvollsten ist, da es Gründe gegen die Verwendung der verbleibenden Begriffe gibt. Die Gründe gegen die Verwendung der restlichen Begriffe sollen nun untersucht werden. Dabei soll vorausgegriffen werden, dass die Reihenfolge der Situationen, an denen der Agent perzipiert, von Bedeutung ist, um es dem Agenten zu ermöglichen, wieder zurück in eine vorangegangene Situation zu gelangen, was notwendig ist, da die Perzeption nicht komplett gespeichert wird.

Es ist offensichtlich, dass der Agent die Perzeption zu bestimmten Zeitpunkten durchführt. Der zeitliche Aspekt ist von Bedeutung, da er es ermöglicht, die Reihenfolge der Durchführung der Perzeption zu bilden. Die Existenz dieser Reihenfolge ermöglicht es dem Agenten darüber zu räsonieren oder erinnern, was zu einem anderen Zeitpunkt für Umstände geherrscht haben. Der zeitliche Aspekt ist aber nicht ausreichend, um ein räsonieren über die Umstände zu vorangegangenen Zeitpunkten zu ermöglichen, da der Agent sich zu jedem Zeitpunkt in einer bestimmten Pose befindet, von der insbesondere die Perzeption zu diesem Zeitpunkt abhängt. Die Perzeption des Agenten ist in einer Pose aufgrund der statischen Umwelt des Agenten immer zu jedem Zeitpunkt identisch. Da der Agent in er Lage sein soll, mehr als nur über den zeitlichen Aspekt des Perzeptionsvorgangs zu räsonieren, so muss er deshalb seine Pose zu dem Zeitpunkt oder relevante Teile der Perzeption in dieser Pose erinnern. Das erinnern der exakten Pose ist für das Räsonieren prinzipiell nicht notwendig, da die der Perzeption nachfolgenden Prozesse, insbesondere das Matching, nicht die Pose sondern die Perzeption verwenden (siehe nachfolgenden Abschnitt). Allerdings soll der Agent in der Lage sein, zu dem Ort, an dem die Objekte perzipiert werden, zurückzukehren, weshalb die Erinnerung der Pose, in der perzipiert wurde, von Bedeutung ist. Für den Zweck des Zurückkehrens ist es ausreichend, dass der Agent wieder die gleiche Perzeption wahrnimmt, die er zu dem vorangegangenem Zeitpunkt besessen hat, weshalb es einen Toleranzbereich für die Pose gibt, da die Perzeption auch dann noch identisch ist, wenn der Agent sich in einer leicht veränderten Pose befindet. Da aber weder ein Zeitpunkt noch die Pose zu diesem Zeitpunkt alleine ausreichen, um die Umstände, unter denen perzipiert wird, zu beschreiben, sollten diese Umstände nicht nur als Pose oder Zeitpunkt bezeichnet werden.

Für den Matching-Algorithmus ist neben der Perzeption noch die Instruktion und der Aktionsplan von Bedeutung. Wenn die Umstände, unter denen der Agent perzipiert, Position, Instruktion, Aktionsplan umfassen, so läßt sich argumentieren, dass hier zumindest eine Teilmenge des Zustands, unter denen der Agent perzipiert, wichtig ist und deshalb von Zuständen, zu denen der Agent perzipiert, gesprochen werden sollte. Allerdings widerspricht die Anforderung, dass der Agent zu einem „Zustand“ zurückkehren können sollte, dieser Bezeichnung. Da der Agent deterministisch und die Umgebung statisch ist, würde eine „Rückkehr“ zu einem alten Zustand nichts am Verhalten des Agenten ändern. Da die Einbindung des Wissensrevisionsmoduls in den Agenten aber ein andereres, nämlich ein rationaleres Verhalten, bewirken soll, wäre es widersprüchlich, von einem Zustand zu sprechen, in den der Agent zurückkehrt und in dem der Agent dann ein rationaleres Verhalten zeigt.

Der Agent perzipiert immer dann, wenn er aufgrund der Instruktion einen Entscheidungspunkt erwartet. Daher ist es denkbar, die Entscheidungspunkte auch als Bezeichnung für die Orte, an denen perzipiert wird, zu verwenden. Allerdings besteht die Möglichkeit, dass der Agent sich nicht an einem Entscheidungspunkt befindet, wenn er perzipiert, da der natürlichsprachliche Instruktionsplan unzureichend war oder der Agent sich an einem Entscheidungspunkt befindet und perzipiert, der nicht im Instruktionsplan erwähnt wurde. In diesen Fällen soll das Modul zur Generierung des Aktionsplans auf niedriger Ebene dafür sorgen, dass mit Folgeaktionen der richtige Entscheidungspunkt gefunden wird. Damit kann nicht jeder Ort, an dem perzipiert wird, auch als Entscheidungspunkt angesehen werden, da der Agent bereits dann perzipiert, wenn er *erwartet* an einem solchen zu sein.

Der Begriff Situation ist kein strikt definierter Begriff. Man kann einer Situation unter anderem eine zeitliche Komponente zuschreiben, aber auch die sonstigen Umstände, die zu dieser Situation herrschen. Eine Situation soll im Kontext dieses Textes die folgenden Eigenschaften besitzen:

1. Eine Situation soll sich mit anderen Situationen in eine Reihenfolge bringen lassen (zeitliche Komponente).

2. Eine Situation umfasst Position und Perzeption des Agenten. Die Bickrichtung muss nicht zusätzlich aufgenommen werden, da die von Position und Blickwinkel abhängige Perzeption bereits in der Situation enthalten ist. Eine Folge der Aufnahme der Perzeption in eine Situation ist, dass sich der Agent selbst unter Beibehaltung der gleichen Pose zu unterschiedlichen Konfigurationen der Umwelt (z.B. Sommer im Vergleich zu Winter) in unterschiedlichen Situationen befindet. Der Agent soll sich auch dann in der gleiche Situation befinden, wenn er sich in leicht veränderten Posen befindet, solange die Perzeption identisch ist.

3. Eine Situation enthält den in der Instruktionsphase des Agenten aufgestellten Aktionsplan sowie den Instruktionsgraph, wodurch der Agent unter Verwendung unterschiedlicher Instruktionen sich auch automatisch in unterschiedlichen Situationen befindet. Wenn der Agent beispielsweise am Ausgang der Mensa positioniert ist, befindet sich der Agent bei Abarbeitung von Instruktion NBTest01 in einer anderen Situation als bei Abarbeitung von NBTest02 (NBTest01 und NBTest02 sind zwei unterschiedliche Instruktionspläne zur Beschreibung der Route von der Mensa nach Haus E).

4. Die Situation umfasst einen Teil des inneren Zustands des Agenten, insbesondere die Information, wo sich der Agent innerhalb des Aktionsplans befindet und welche Teile des Aktionsplans bereits ausgeführt wurden. Da der Agent einmal koreferenzierte Perzepte erinnert, sollen auch die in der Situation erinnerten Perzepte Teil der Situation sein. Darüber hinausgehende Zustandsinformationen, insbesondere die in Kapitel 4 eingeführt Wissensbasis, sollen *nicht* Teil der Situation des Agenten sein.

Eine formale Definition von Situation erfolgt in Kapitel 2.4.

2.4 Matching-Algorithmus

Der von Helwich in [H 2003] eingeführte Matching-Algorithmus hat die Aufgabe, die Perzeption, die während der Navigationsphase in diskreten Zeitabständen ermittelt wird, mit der aufgrund der Instruktion erwarteten Umwelt in Übereinstimmung zu bringen. Der Zeitpunkt und die damit verbundenen Umstände werden in dieser Arbeit als Situation bezeichnet. Der Matching-Algorithmus liefert die Information, welches Objekt der Instruktion mit welchem Objekt der Perzeption übereinstimmt, in Form einer Koreferenz zwischen einem Knoten aus dem Instruktionsgraphen und einem Knoten aus dem Perzeptionsgraphen. Die Koreferenz ist die explizite Repräsentation der Gleichheit zweier Knoten (siehe Kapitel 1.2.2 in [H 2003]) und kann in Graphen durch eine Kante oder in deklarativer Form durch eine Relation repräsentiert werden. Der in [H 2003] in Kapitel 6 eingeführte Matching-Algorithmus berechnet darüber hinaus auch ein Ähnlichkeitsmaß für die Koreferenz, was einen Wert im Intervall $[0, 1]$ haben kann. Wie das Ähnlichkeitsmaß berechnet wird, wird später ausgeführt werden. Es ist nun möglich, eine Koreferenz als ein Tripel bestehend aus dem koreferenzierten Knoten der Instruktion, dem koreferenzierten Knoten der Perzeption und dem Ähnlichkeitsmaß zu repräsentieren:

Definition 4 (Koreferenz) *Sei $c_I \in \mathbb{I}$, $c_P \in \mathbb{P}$, k eine rationale Zahl. Dann ist das Tripel (c_P, c_I, k) eine Koreferenz mit Konfidenzwert. Sei P der Perzeptionsgraph des geometrischen Agenten, I der Instruktionsgraph. Dann sei $\mathcal{K}(P, I)$ die Menge aller möglichen Koreferenzen zwischen P und I sowie $\mathcal{K}$ die Menge aller potentiell möglichen Koreferenzen insgesamt.*

Neben der Berechnung von Koreferenzen zwischen einzelnen Knoten erzeugt der Matching-Algorithmus auch Sammlungen von Koreferenzen für bestimmte Teile des Instruktionsgraphen, die zu der gegenwärtigen Situation passen, und versieht auch diese mit einem Ähnlichkeitsmaß. Welche Teile des Instruktionsgraphen dabei als Grundlage dienen ist für diese Arbeit nicht von Bedeutung. Die Auswahl des in einer Situation relevanten Teilgraphen durch den Matching-Algorithmus wird detailliert in [H 2003] erklärt. Von Bedeutung ist, dass man davon sprechen kann, dass es einen Teilgraphen des Instruktionsgraphen gibt, der für die Ausführung eines Befehls von Bedeutung ist.

Mithilfe der formalen Definition von Koreferenzen läßt sich eine formale Definition von Situation aufstellen. Wie in Abschnitt 2.3 bereits informell eingeführt, umfasst in dieser Arbeit eine Situation im wesentlichen die Umstände, unter denen der Agent perzipiert.

Definition 5 (Situation) *Eine Situation s ist ein Tupel $(\lambda, A, P, I, I^{'}, K)$, wobei λ die Position des geometrischen Agenten innerhalb seiner Umgebung, A der Aktionsplan aus der Instruktion, $P \subseteq \mathbb{P}$ der Perzeptionsgraph, $I \subseteq \mathbb{I}$ der Instruktionsgraph, $I^{'} \subseteq I$ der zur aktuellen Anweisung passende Teilgraph des Instruktionsgraphen und K die Menge der in den Situationen vor S berechneten Koreferenzen zwischen I und P ist. Die Menge aller Situation sei $\mathbb{S}$. Die Situationen sollen nach ihrem zeitlichen Eintreten geordnet sein, so dass es möglich ist, unter der Voraussetzung von identischer Konfiguration der Perzeption und Verwendung der gleichen Instruktion zwei Situationen in die zeitliche Beziehungen vor und nach zu setzen. Die Situation s_0 sei diejenige Situation, die zeitlich vor allen anderen Situationen liegt.*

Da der Matching-Algorithmus immer während einer bestimmten Situation die Koreferenzierung durchführt, soll das Ergebnis des Matching-Algorithmus in Abhängigkeit von einer Situation spezifiziert werden. Der Matching-Algorithmus berechnet für den Teilgraphen des Instruktionsgraphen, der zu der aktuellen Aktion gehört, eine Sammlung von Koreferenzen zu Knoten der Perzeption und berechnet einen Konfidenzwert basierend auf einem Ähnlichkeitsmaß. Eine solche Sammlung von Koreferenzen wird nachfolgend als Match bezeichnet. Es ist möglich, dass in einer Situation eine Vielzahl von möglichen Matches existiert.

Ein wesentliches Merkmal eines solchen Matches ist, dass ein Knoten der Instruktion maximal einem Knoten der Perzeption zugeordnet werden kann. Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation. Eine Menge match(s) von Koreferenzen mit match(s) $= \{(p_1, i_1, c_1), \ldots, (p_n, i_n, c_n)\}$ mit $p_1, \ldots, p_n \in$

$P, i_1, \ldots, i_n \in I^{'}$ und $c_1, \ldots, c_n \in [0,1]$ wird genau dann als ein zu der Situation s passendes Match bezeichnet, wenn

$$\forall_{(p_l,i_l,c_l),(p_j,i_j,c_j)\in \text{match(s)}} \quad l \neq j \supset i_l \neq i_j$$

Nachfolgend werde ich, um eine bessere Lesbarkeit zu gewährleisten, statt match(s) unter Umständen die Abkürzung $\mu(s)$ verwenden.

Berechnung der Matches einer Situation

Die Berechnung der Matches wird in [H 2003] in Algorithmus 6.1.1 dargestellt. Zu den wesentlichen Eigenschaften bei der Berechnung eines Matches gehört zum einen die Auswahl von Referenzobjekten aus dem zur aktuellen Aktion passenden Teilgraphen der Instruktion, für die zunächst auf Basis einer konzeptuellen Ähnlichkeit nach Entsprechungen im Perzeptionsgraphen gesucht wird. Damit verwendet der Matching-Algorithmus Taxonomien zur Berechnung des Matches. Es werden dabei Perzeptionsknoten, die mit den Referenzobjekten nur eine geringe Ähnlichkeit aufweisen, über ein Schwellenwertverfahren ausgeschlossen. Es soll an dieser Stelle darauf hingewiesen werden, dass es insbesondere durch das Artefakt der Perzeption, ein durch Hindernisse nicht komplett wahrgenommenes Objekt durch duplikate identische Knoten zu repräsentieren, zu einer Vielzahl von gleich bewerteten alternativen Koreferenzen der Referenzobjekte kommen kann.

Eine weitere wesentliche Eigenschaft des Matching-Algorithmus ist es, auf Basis der berechneten Koreferenzen der Referenzobjekte auch die zwischen diesen Knoten bestehenden Relationen miteinzubeziehen. Dazu wird Algorithmus 4.5.4 benutzt, der eine Liste von verschiedenen Verknüpfungen zweier CRIL-Graphen mit der maximalen Ähnlichkeit berechnet. Die Berechnung enthält die Bildung einer Verknüpfungsmatrix beider Graphen (siehe Abschnitt 4.5 in [H 2003]). Algorithmus 6.1.1 sortiert im Anschluss alle Verknüpfungen anhand eines Schwellenwertes der Ähnlichkeit der Ausgangsgraphen aus. Das dabei verwendete Ähnlichkeitsmaß ist in Definition 4.4.3 in [H 2003] zu finden. Das Ähnlichkeitsmaß bezieht auch die Ähnlichkeit der Kanten des Graphen mit ein.

Zusammenfassend kann gesagt werden, dass bei der Berechnung eines Matches sowohl topologische Aspekte wie die Struktur der Graphen aber auch die taxonomische Ähnlichkeit sowohl der Relationen wie auch der Knoten eine Rolle spielen. Der Matching-Algorithmus liefert immer ein Menge von Matches mit maximalen Ähnlichkeitsgrad, was bedeutet, dass die berechneten Matches die gleiche Bewertung haben. Gibt es keine „guten“ Matches, dann wird aufgrund des Schwellenwerts gar kein Match zurückgeliefert. Innerhalb der Matches mit maximaler Bewertung ist davon auszugehen, dass es Ähnlichkeiten in Form von einer Anzahl von identisch koreferenzierten Instruktionsknoten gibt. Die Ähnlichkeiten beruhen auf der Verwendung der Knotenmatrix, in der alle bekannten Knoten systematisch erfasst werden. Wenn es verschiedene (gleich bewertete) Möglichkeiten gibt, einen Knoten der Instruktion i einem Knoten der Perzeption zuzuordnen, dann führt dies dazu, dass es unter den besten Matches einige gibt, die sich nur in Hinblick auf die unterschiedliche Zuordnung von Instruktionsknoten i unterscheiden. Damit ist innerhalb der Matches mit maximaler Bewertung mit einer Zahl von gleichen Koreferenzen zu rechnen.

Obwohl der Matching-Algorithmus nur Werte oberhalb des Schwellenwerts berechnet, soll auch in den nachfolgenden Definitionen auf diesen nicht mehr zurückgegriffen werden, da Matches innerhalb dieser Arbeit ausschließlich als Ergebnis des Matching-Algorithmus von Bedeutung sind. Es soll daher Grundannahme sein, dass nur von Matches oberhalb des Schwellenwerts berichtet wird. So ist insbesondere die Aussage „In einer Situation gibt es keine Matches.“ so zu verstehen, dass es keine Matches oberhalb des Schwellenwerts gibt.

Definition 6 (Matches) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation. Dann ist matches(s) die Klasse aller Matches der Situation S, die der Matching-Algorithmus berechnet hat. Nachfolgend werde ich, um eine bessere Lesbarkeit zu gewährleisten, statt matches(s) unter Umständen die Kurzform $\mu\sigma(s)$ verwenden.*

Definition 7 (Klasse aller Matches) *Die Klasse aller Matches sei $\mathcal{M}$.*
Anmerkung: es gilt $\mu\sigma(s) \subseteq 2^{\mathcal{M}}$.

In Abschnitt 2.2.3 wird das Ziel einer Aktion eingeführt. Da ein Match während der Verarbeitung des Aktionsplans ausgeführt wird, existiert in jeder Situation auch eine Aktion, die gerade verarbeitet wird. Außerdem läßt sich das Ziel einer Aktion auch im Instruktionsgraphen wiederfinden. Daher ist es möglich einem Match in einer bestimmten Situation einen Knoten aus dem Instruktionsgraphen zuzuordnen, der das Ziel der aktuell auszuführenden Aktion bezeichnet. In jedem Match existiert dann genau eine Koreferenz, die das Ziel der Aktion mit einem Knoten der Perzeption verknüpft.

Definition 8 (Ziel einer Aktion in einem bestimmten Match) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation, $m \in$ matches(s). Das Ziel einer Aktion $\tau : matches(s) \rightarrow \mathcal{K}(P, I')$ des Matches s sei diejenige Koreferenz, die den CRIL-Knoten des Ziels der Aktion aus dem Instruktionsgraph mit einem Objekt aus der Perzeption verbindet.*

Der Matching-Algorithmus erzeugt eine Menge von gleich gewerteten möglichen Matches. Aus diesen wählt der Agent eins aus. Es läßt sich daher das in einer Situation ausgewählte Match definieren.

Definition 9 (Das in einer Situation gewählte Match) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation. Dann ist $\gamma : \mathbb{S} \rightarrow \mathcal{M}$ das durch den geometrischen Agenten gewählte Match in einer Situation oder $\emptyset$, falls entweder kein Match gewählt ist oder in s keine Matches vorliegen.*

Kapitel 3

Wissensrevision

In diesem Kapitel werden die Grundlagen zur Wissensrevision vorgestellt. Wissensrevision ist eine Operation, die widersprüchliche Informationen in eine vorher konsistente Menge von Sätzen einer logischen Sprache integriert. Die Grundlage der in Kapitel 4 vorgestellten Wissesbasis ist die Aussagenlogik, die in Abschnitt 3.1 vorgestellt wird. Im Rahmen der Behandlung der Aussagenlogik werden auch zwei Beweisverfahren vorgestellt, da Implementationen von Wissensbasen einen Deduktionsmechanismus benötigen, der die Konsequenzen einer Menge von expliziten Sätzen berechnet, da die Konsequenzen einer Menge von expliziten Sätzen selbst in der Aussagenlogik nicht endlich ist und somit nicht komplett explizit gespeichert werden kann. Wissensbasen und Wissensrevision werden in den Unterabschnitten 3.2 und 3.2 eingeführt und einige Gruppen möglicher Revisionsoperatoren klassifiziert. Wie das Kernproblem der Revision, die Auswahl einer der konfligierenden Sätze geschieht, wird im Abschnitt über epistemische Verankerung ausgeführt. Den Abschluss dieses Kapitels bildet die Vorstellung von zwei Verfahren zur konkreten Implementation von Revisionsoperatoren.

3.1 Formale Logik

Formale Logik untersucht die Verknüpfung logischer Aussagen und das formale Schließen mithilfe logischen Aussagen. Formale Logik ist innerhalb des hier vorgestellten Ansatzes zur Verbesserung der Robustheit des Verhaltens des Agenten die Grundlage zum Einsatz von Wissensrevision, die in den nachfolgenden Unterabschnitten dieses Kapitels behandelt wird. Es wird an dieser Stelle nicht eine detaillierte Einführung in die Syntax und Semantik formaler Logik erfolgen[1]. Die hier interessanten Aspekte von formaler Logik sind zum einen die Tatsache, dass es verschiedene Logiken gibt, die sich in Hinblick auf ihre Ausdruckstärke und Berechnungskomplexität der Schlußverfahren unterscheiden, zum anderen die Formalisierung an sich, die es in Kapitel 4 erlaubt, den Aufbau der revisionsbereiten Wissensbasis formal zu beschreiben. Da in den nachfolgenden Kapiteln nur die Aussagenlogik Verwendung findet, werde ich mich auf die formale Einführung dieser Logik beschränken.

Aussagenlogik

Die klassische Aussagenlogik behandelt nach [FITTING 1996] das Verhalten von aussagenlogischer Verknüpfungen wie „und“ und „oder“. Auf syntaktischer Ebene existieren dabei atomare Aussagen, deren interne Struktur nicht analysiert wird und die zu einem Wahrheitswert ausgewertet werden können. Die Verknüpfungen, die auch „Junktoren“ genannt werden, werden benutzt, um zusammengesetzte Aussagen auf der Basis von atomaren oder zusammengesetzten Aussagen zu bilden. Dabei werden atomare Aussagen durch ein Aussagensymbol repräsentiert.

[1] Eine Einführung in die Aussagenlogik findet sich unter anderem in [FITTING 1996] und [S 1987]

Definition 10 (Aussagensymbole) *Es sei $\mathcal{L}_{atom}$ eine abzählbare Menge von Aussagensymbolen.*

Neben den Aussagensymbolen besitzen logische Sprachen darüber hinaus logische Konstanten. Die Vereinigung der Menge der Aussagensymbole mit der Menge der logischen Konstanten wird die Menge der atomaren Formeln genannt.

Definition 11 (Logische Konstanten) *Die Menge der logischen Konstanten $\mathcal{L}_{konstanten}$ sei $\{\top, \bot\}$*

Definition 12 (Atomare Formel) *Die Menge der atomaren Formeln $\mathcal{L}_{aform}$ sei*

$$\mathcal{L}_{aform} := \mathcal{L}_{konstanten} \cup \mathcal{L}_{atom}$$

Atomare Formeln können durch Junktoren mit anderen atomaren oder komplexen Formeln verknüpft werden. Es gibt eine Vielzahl möglicher Junktoren, die sich unter anderem anhand der Anzahl der durch sie verknüpften Symbole klassifizieren lassen. Zu den einstelligen Junktoren gehört die Negation „$\neg$“, zu den zweistelligen gehören die Konjunktion „$\wedge$“ die Disjunktion
„$\vee$“ , die Implikation „$\supset$“ , die Biimplikation „$\equiv$“ und das exklusive Oder „$\not\equiv$“

Definition 13 (Logische Junktoren) *Die Menge der einstelligen logischen Junktoren sei $\mathcal{L}_{junär} := \{\neg\}$.*
Die Menge der zweistelligen logischen Junktoren sei $\mathcal{L}_{jbinär} := \{\wedge, \vee, \supset, \equiv, \not\equiv\}$.
Da $\vee$ und $\wedge$ assoziativ und kommutativ sind, sollen für diese beiden Junktoren neben den binären Junktoren auch verallgemeinerte n-stellige Junktoren, die eine beliebige endliche Zahl von Operanden besitzen, eingeführt werden. Zukünftig soll dann nicht mehr zwischen dem binären $\vee$ bzw. dem binären $\wedge$ und ihren verallgemeinerten Varianten unterschieden werden. Die Menge der verallgemeinerten Junktoren sei $\mathcal{L}_{generalisiert} = \{\bigvee, \bigwedge\}$.

Die Menge der logischen Junktoren sei $\mathcal{L}_{junktoren} := \mathcal{L}_{junär} \cup \mathcal{L}_{jbinär} \cup \mathcal{L}_{jnstellig}$

Definition 14 (Menge der logischen Formeln) *Die Menge der logischen Formeln sei die kleinste Menge $\mathcal{L}$, so dass*

1. *Wenn $A \in \mathcal{L}_{aform}$ dann ist $A \in \mathcal{L}$*
2. *Wenn $X \in \mathcal{L}$ und $\circ \in \mathcal{L}_{junär}$, dann ist $\circ X \in \mathcal{L}$*
3. *Wenn $\circ \in \mathcal{L}_{jbinär}$, X und $Y \in \mathcal{L}$, dann ist $(X \circ Y) \in \mathcal{L}$*
4. *Wenn $\circ \in \mathcal{L}_{generalisiert}$, $X_1, \ldots, X_n \in \mathcal{L}, n \in \mathbb{N}, n > 0$. Dann ist $\circ(X_1, X_2, \ldots, X_n) \in \mathcal{L}$*

Die vorrangehenden Definitionen haben die Syntax der Aussagenlogik festgelegt. Da der Inhalt der atomaren Aussagen nicht bekannt ist, ist es üblich, eine Interpretation einer logischen Formel durchzuführen. Dabei werden die atomaren Aussagen auf eine Menge von Wahrheitswerten (nachfolgend mit Tr bezeichnet) abgebildet. Innerhalb der Logik ist die Kardinalität der Menge der Wahrheitswerten ein weiterer Freiheitsgrad. Die klassische Aussagenlogik kennt zwei Wahrheitswerte, die typischerweise als „wahr“ oder „falsch“ oder 1 und 0 bezeichnet werden. Wird eine atomare Aussage von einer Interpretation auf wahr abgebildet so wird in der Interpretation zugewiesen, dass die Aussage wahr ist, wird sie auf falsch abgebildet, so wird in der Interpretation zugewiesen, dass sie falsch ist. Existiert für eine komplexe Formel F aus $\mathcal{L}$ eine Menge B, in der für jede atomare Formel in F eine Abbildung auf die Wahrheitswerte existiert, so nennt man die Funktion $B \rightarrow \{\mathrm{Tr}\}$ eine Belegung von F. Die Auswertung der Junktoren kann im Falle von unären Junktoren durch eine Abbildung Tr$\rightarrow$Tr, im Falle von binären Junktoren durch eine Abbildung Tr$\times$Tr$\rightarrow$Tr erfolgen (die beide nachfolgend mit φ bezeichnet werden). Tabelle 2.1 in [FITTING 1996] stellt die Abbildungen der hier verwendeten Junktoren in Tabellenform dar. Die Interpretation von generalisierten Junktoren kann im Fall von zwei oder mehr

Operanden auf die Interpretation von binären Junktoren zurückgeführt werden. Im Fall von einem Operand können sowohl verallgemeinerte Disjunktion als auch verallgemeinerte Konjunktion auf die Interpretation des einzelnen Operanden reduziert werden. Die Interpretation der leeren Disjunktion entspricht der Interpretation von $\perp$, die der leeren Konjunktion der von $\top$.

Um eine komplexe aussagenlogische Formel auszuwerten, wird in [FITTING 1996] die „boolsche Auswertung“ (Boolean valuation) v eingeführt, die zum einen die logischen Konstanten $\top$ und $\perp$ auf wahr bzw. falsch abbildet, zum anderen die unären und binären Junktoren auf Anwendung von φ auf die Auswertungen der jeweiligen Operanden abbildet. An dieser Stelle wird keine formale Definition zur Auswertung komplexer Formeln gegeben, da die Definition aus [FITTING 1996] übernommen wird. Die boolsche Auswertung soll nachfolgend mit v bezeichnet werden.

Es gibt aussagenlogische Formeln, die aufgrund ihrer Struktur unter allen Belegungen wahr oder unter allen Belegungen falsch sind. Diese werden im Falle von unter allen Belegungen wahren Formeln Tautologien bzw. im Falle von unter allen Belegungen falschen Kontradiktionen genannt. Formeln, die Tautologien sind, werden in [FITTING 1996] als „valid“ oder zu deutsch „gültig“ bezeichnet.

Neben Tautologien und Kontradiktionen gibt es eine Menge von Formeln, die unter mindestens einer, aber nicht unter allen Belegungen als wahr ausgewertet werden. Diese Formeln werden als „kontingent“ bezeichnet. Von Bedeutung ist hier der Zusammenhang, dass die Negation einer tautologischen Formel eine Kontradiktion ist, und die Negation einer Kontradiktion eine Tautologie.

Eine semantische Beziehung zwischen zwei Formeln F und G ist die sog. Äquivalenz, die gilt, wenn jede Belegung, die F wahr macht auch G wahr macht und jede Belegung, die in G wahr macht auch F wahr macht. Es kann mit dem in der Literatur als „Ersetzungstheorem“ bekanntem Theorem gezeigt werden, dass innerhalb einer Formel die Ersetzung einer Teilformel durch eine zur Teilformel äquivalenten Formel an der Auswertung der Ausgangsformel nichts ändert.

Neben semantischen Kategorien wie Tautologien sind im folgenden auch einige syntaktische Strukturen von Bedeutung. So wird eine Formel, die ausschließlich aus einer atomaren Formel oder der Negation einer atomaren Formel besteht, als Literal bezeichnet. Nach [S 1987] wird eine Formel, die aus einer (verallgemeinerten) Disjunktion von (verallgemeinterten) Konjunktionen von Literalen besteht, als in disjunktiver Normalform (DNF) stehend bezeichnet. Eine Konjunktion von Disjunktionen von Literalen steht in konjunktiver Normalform (KNF). Nach [S 1987] ist es möglich, jede aussagenlogische Formel durch Ersetzungen nach dem Ersetzungstheorem in KNF und DNF zu bringen.

Aussagenlogische Beweisverfahren

Neben der Definition, was eine gültige aussagenlogische Formel ist, werden Verfahren benötigt, die prüfen, ob eine Formel gültig ist oder nicht. Ein solches Beweisverfahren was korrekt („sound“) und vollständig ist und terminiert, wird nach [FITTING 1996] Entscheidungsverfahren genannt. Korrekt bedeutet, dass jede Formel, die durch ein Beweisverfahren bewiesen wird, auch gültig ist. Vollständigkeit bedeutet, dass jede gültige Formel durch das Beweisverfahren bewiesen wird, und Termination bedeutet, dass das Beweisverfahren nach endlichen vielen Schritten zu einer Entscheidung kommt. Im Falle der Aussagenlogik existieren verschiedene Beweisverfahren, die diese drei Eigenschaften haben. Ein einfaches Verfahren ist es, zu prüfen ob alle minimalen passenden Belegungen einer Formel diese auch wahr machen, was in [S 1987] „Wahreitstafelmethode“ genannt wird. Dabei steigt die Berechnungskomplexität exponentiell zu der Anzahl der atomaren Aussagensymbole in der zu beweisenden Formel an. Die exponentielle Berechnungskomplexität wird auch im schlechtesten Fall durch alle anderen Beweisverfahren innerhalb der Aussagenlogik erreicht.

Neben Beweisverfahren, die direkt die Gültigkeit einer Formel testen, gibt es auch Verfahren, welche die Kontingenz einer Formel nachweisen können (von Schöning in [S 1987] „Erfüllbarkeitstest“ genannt). Wenn ein Erfüllbarkeitstest auf die Negation einer Formel angewendet wird und eine Belegung gefunden wird, die die negierte Formel wahr macht, dann kann die Ausgangsformel keine Tautologie sein, da die Negation einer Tautologie eine Kontradiktion ist und damit keine Belegung haben darf, die diese wahr macht. Wird keine Belegung gefunden, die die negierte Formel wahr macht, so ist diese eine Kontradiktion und die Ausgangsformel damit eine Tautologie. Ein solches Beweisverfahren wird in [S 1987] „Widerlegungsverfahren“ genannt.

Es gibt ein Fragment der Aussagenlogik für das es nach [S 1987], Kapitel 1.3 ein Beweisverfahren gibt, das die Erfüllbarkeit einer Formel F dieses Fragments mit maximal der Anzahl der atomaren Aussagen von F Schritten feststellt. Dieses Fragment wird Hornlogik genannt und die Formeln, die Teil der Hornlogik sind, Hornformeln. Eine Hornformel ist eine Formel, die in KNF steht und in der jedes Disjunkt höchstens ein positives Literal (nicht negiertes Atom) enthält.

Von den verschiedenen Beweisverfahren für Formeln der Aussagenlogik sind nach [FITTING 1996] zwei zur Automatisierung geeignet. Eins dieser Beweisverfahren, das Tableauverfahren, ist wie ein Widerlegungsverfahren. Die Kernidee des Tableaubeweises ist es, eine disjunktive Normalform schrittweise aufzubauen und in ihr nach möglichen Modellen der Negation der zu beweisenden Formel zu suchen. Wird ein solches Modell gefunden, so ist die Negation der Ausgangsformel erfüllbar und damit die Ausgangsformel keine Tautologie. Der Aufbau der disjunktiven Normalform erfolgt schrittweise durch eine von [FITTING 1996] Tableau-Expansionsregeln genannte Menge von Inferenzregeln(siehe Tabelle 3.1 in [FITTING 1996]). Die durch die Expansion erzeugte Formelmenge wird als Baum dargestellt, wobei ein Knoten eine Formel als Beschriftung erhalten und ein Zweig als eine Konjunktion aller seiner beschriftenden Formeln angesehen wird. Die verschiedenen Zweige des Baums stehen in verallgemeinerter Disjunktion zueinander.

Eine verallgemeinerte Konjunktion von Literalen ist dann eine Kontradiktion, wenn sie entweder $\perp$ enthält oder eine Formel sowohl negiert als auch nicht negiert auftritt. Ist in einem Zweig des Tableaus jede Formel maximal einmal expandiert worden, so wird dieser Zweig strikt expandiert genannt. Ein Zweig, in der eine Formel sowohl negiert als auch nicht negiert vorkommt, oder der $\perp$ enthält, wird geschlossen genannt. Ist in einem Tableau jeder Zweig geschlossen, so ist auch das Tableau geschlossen, was bedeutet, dass die Suche nach einem Modell für die Negation der zu beweisenden Formel gescheitert ist. Existiert kein Modell für die Negation der zu beweisenden Formel, so ist diese eine Kontradiktion und die zu beweisende Formel eine Tautologie. Der Beweis der Korrektheit dieser Aussage liefert Fitting in Kapitel 3.4.

Existiert ein offener Zweig, der nicht mehr strikt expandiert werden kann, so exisitert ein Modell für die Negation dieser Formel, woraus folgerbar ist, dass die zu beweisende Formel nicht gültig ist.

Fitting benutzt als Zeichen für die Beweisbarkeit mit dem Tableauverfahren das Symbol $\vdash_{pt}$. Diese Konvention wird in dieser Arbeit übernommen.

Definition 15 (Beweis mit einem aussagelogischen Tableau) *Sei $f \in \mathcal{L}$, so hat f einen aussagelogischen Tableaubeweis, wenn für $\neg f$ ein geschlossenes aussagelogisches Tableau existiert. Dies wird geschrieben als $\vdash_{pt} f$.*

In der Literatur zu Tableau-Beweisern werden oft die Ableitungsregeln anhand von sog. α und β Formeln definiert (siehe S. 23 in [FITTING 1996]). α und β Formeln kategorisieren Formeln $(X \circ Y)$ und $\neg(X \circ Y)$ anhand ihres primären Junktors $\circ$ in konjunktive (α) und disjunktive (β) Formeln ein. Konjunktive Formeln sind Formeln, deren primärer Junktor in genau einem Fall mit 1 ausgewertet wird, Disjunktive, deren primärer Junktor in genau drei Fällen mit 1 ausgewertet wird. β-Formeln werden durch einen Tableaubeweiser verzweigend expandiert. Die Notwendigkeit, β-Formeln verzweigend zu expandieren, ist verantwortlich für die im schlimmsten Fall exponentielle Berechnungskomplexität des Tableaubeweisers, da für den Fall, dass ein Tableau komplett expandiert werden muss, jede β-Formel zu einer Vervielfachung der Zahl der Zweige führt

Wie in Kapitel 2.6 in [BA01] erwähnt, ist die Tableau-Expansion nicht deterministisch. Es ist nicht festgelegt, welche noch nicht expandierte Formel, die Teil eines Zweigs ist, expandiert werden soll. Außerdem ist nicht festgelegt, in welchem der durch die Expansion einer β-Formel entstandene Zweige nachfolgend expandiert werden soll.

Bei dem Tableau-Beweis von aussagenlogischen Formeln ist eine mehrfache Expansion der gleichen Formel eines Zweigs nicht notwendig. Während eines Tableau-Beweises liegen innerhalb eines Zweigs Br eine Menge von bereits expandierten und noch nicht expandierten Formeln vor. Falls bei einer

Expansion von Br eine Verzweigung notwendig ist, so ist es auch in dem neu erzeugtem Zweig nicht notwendig, dass bereits expandierten Formeln erneut expandiert werden. Deshalb ist es sinnvoll vor Expansionen, die eine Verzweigung erfordern, zunächst Expansionen, die keine Verzweigung erfordern, auszuführen. Werden jedoch die Expansionen, die eine Verzweigung erfordern, früh ausgeführt, so verdoppelt sich die Anzahl der Zweige mit jeder verweigenden Expansion und zusätzlich muss in jedem Zweig jede Formel die nicht verzweigt, expandiert werden, was zu einer großen Redundanz führt. Um die Redundanz zu vermeiden, wird deshalb oft die Heuristik benutzt, zunächst alle nicht verzweigenden Expansionen innerhalb eines Zweiges auszuführen. Die Expansionsreihenfolge hat damit großen Einfluss auf den tatsächlichen Berechnungsaufwand eines Tableaubeweises.

Ein Beispiel für eine gewöhnliche Expansion eines Tableaus findet sich im Anhang B, auf Seite 85.

Optimierungen eines Tableau-Beweises

Es existiert eine Vielzahl an Optimierungen für (aussagenlogische) Tableau Beweiser. Es soll an dieser Stelle nicht eine Übersicht über alle bekannten Verfahren gegeben werden, sondern nur eine kleine Auswahl, die in Kapitel 5 von Bedeutung ist.

Eine einfache Optimierung ist der nicht-atomare Abschluss. Wenn der nicht atomar Abschluss verwendet wird, so wird während der Prüfung, ob ein Zweig geschlossen ist, nicht nur die Menge der Literale auf Kontradiktion untersucht, sondern die Gesamtmenge der in dem Zweig enthaltenen Formeln. Sind nun in einzem Zweig zwei Formeln F und G enthalten so, dass $F = \neg G$, dann wird bereits der Abschluss gebildet, wenn F und G keine Literale sind.

Wenn eine β-Formel expandiert wird, so enstehen im Normalfall zwei alternative Zweige, in der jeweils a oder b als wahr angesehen wird. Beispielsweise wird die Formel $(a \vee b)$ in zwei Zweige mit den Einträgen a bzw. b expandiert. Semantic Branching ist eine Alternative zu der gewöhnlichen Expansionsregel, in deren Rahmen eine Expansion eines der Disjunkte auswählt und in einem Zweig negiert, im anderen Zweig nicht negiert expandiert. Wenn in obigem Beispiel das Disjunkt a ausgewählt wird, so entstehen die Zweige a bzw $\neg a, b$. Semantic Branching wird unter anderem in [HSP99] vorgestellt.

3.2 Wissensbasen und AGM Postulate

Eine Wissensbasis läßt sich aus verschiedenen Perspektiven betrachten. In [RN 2003] führen Russell und Norvig unter anderem die Kategorien von rationalem Handeln und von rationalem Denken ein, um künstliche Intelligenz zu bewerten. Die Wissensbasis, die in Kapitel 4 vorgestellt werden soll, dient dazu, dem GA rationales Handeln aber auch rationales Denken zu ermöglichen. Deshalb soll der Begriff der Wissensbasis zum einem aus der Sicht des (handelnden) geometrischen Agenten, aber auch aus der Sicht von epistemologischen Theorien („epistemological theories“, wie sie Gärdenfors in [G 1988] nennt) untersucht werden. Aus der Sicht des Agenten ist eine Wissensbasis ein internes Modell eines Agenten, das die Außenwelt als eine Menge von Sätzen einer logischen Sprache („sentences“ in [RN 2003], Kapitel 7) unter Berücksichtigung von Hintergrundwissen („Background Knowledge“ in [RN 2003]) und Erfahrung repräsentiert. Die Wissensbasis selbst enthält deklaratives Wissen, dass durch prozedurales Wissen außerhalb der Wissensbasis ergänzt werden kann. Die Wissensbasis besitzt eine Schnittstelle zur Außenwelt, die die beiden Funktionen „Tell“ und „Ask“ umfaßt. Tell dient dazu, der Wissensbasis Informationen mitzuteilen, Ask dazu, Informationen aus der Wissensbasis abzufragen. Die Berechnung der Antwort auf eine Anfrage geschieht dabei auf Basis des vorher mitgeteilten Wissens und des initial vorhandenen Hintergrundwissens. Nach [RN 2003] liefert die Wissensbasis als Reaktion auf Ask eine Aktion („Each time the agent program is called, it does three things. First, it tells the knowledge base what it percieves. Second, is asks the knowledge base what action it should perform. Third, the agent records it's choice with tell and executes the action.“, [RN 2003], Seite 196). Die Beschränkung der Antwort der Revisionsbasis auf eine Aktion kann damit begründet werden, dass Russel und Norvig die Wissensbasis im Kontext eines Agenten sehen, und die Wissensbasis in diesem Fall dem Zweck der Aktionsplanung dient. Dies ist aber nicht notwendigerweise immer der Fall.

Eine epistemologische Theorie behandelt nach [G 1988] die Konzepte von epistemischen Zuständen und Veränderungen, die bestimmten Rationalitätskriterien genügen müssen, Im Kontext einer epistemologischen Theorie ist eine Wissensbasis eine Repräsentationsform eines epistemischen Zustands oder eines „belief state“. Ein Belief-State bezeichnet Gärdenfors in [G 1988] als einen tatsächlichen oder möglichen Zustand eines Individuums zu einem bestimmten Zeitpunkt, der zusätzlich den Rationalitätskriterien über die Veränderungen von epistemischen Zuständen genügen muss. Aus Sicht der von Russell und Norvig in [RN 2003] vertretenen Agententheorie ist die Repräsentation der epistemischen Zustände die Wissensbasis. Die Tell-Funktion verursacht Änderungen der epistemischen Zustände. Im Rest dieses Kapitels wird die Wissensbasis aus Sicht der epistemologischen Theorien betrachtet, wobei auch die dort verwendete Terminologie übernommen wird.

Epistemische Zustände lassen sich unter anderem als Menge von Sätzen einer logischen Sprache, aber auch als Menge von möglichen Welten auffassen. In den nachfolgend vorgestellten Ansätzen werden diese beiden verschiedenen Repräsentationsformen benutzt, um die Änderungen von epistemischen Zuständen zu untersuchen. In diesem Text wird die Definition von epistemischem Zustand aus [GM 1988], der dort als „knowledge set“ bezeichnet wird, übernommen.

Gärdenfors und Makinson definieren das „knowledge set“ als eine Menge K von Sätzen einer logischen Sprache, die unter den boolschen Operatoren Negation, Konjunktion, Disjunktion und Implikation abgeschlossen ist, und die der Vollständigkeitsbedinung genügen, dass alle Sätz, die aus K gefolgert werden können, Element von K sind:

Definition 16 (Knowledge Set) *Sei L eine logische Sprache. Dann ist Kb eine Menge von Sätzen aus L, für die gilt*

1. *Wenn ein Satz B aus Kb gefolgert werden kann, dann ist $B \in Kb$.*

Ist ein Satz C aus einem „knowledge set“ Kb folgerbar, so wird dies mit $Kb \models C$ abgekürzt.

In vielen Artikeln zu epistemologischen Theorien wird eine derartige Repräsentationsform eines epistemischen Zustands auch als Belief-Set bezeichnet.

Abgschlossenheit in klassicher Logik führt dazu, dass es eine unbegrenzte Anzahl von Konsequenzen aus einer endlichen Menge von Sätzen geben kann, was für eine Implementation problematisch ist. Es können im Rahmen einer Implementation eines „knowledge set“ mit Sätzen aus der Aussagenlogik daher nicht alle Sätze und ihre Folgerungen explizit werden. Allerdings ist es möglich, eine endliche Repräsentationsform eines „knowledge set“ durch die Aufnahme einiger expliziter Sätze sowie eines Beweisers, der die Konsequenzen der Sätze, die explizit aufgenommen sind, berechnet, zu finden. In [GR 1995] wird die Unterscheidung der expliziten und der impliziten Einträge der Wissensbasis durch die Einführung der Funktion Cn, welche die Konsequenzen einer Menge von Sätzen bezeichnet, getroffen. Mithilfe von Cn lassen sich „Belief Sets“ und „Belief Bases“ unterscheiden. Belief Sets sollen in dieser Arbeit als Mengen, die ihre Konsequenzen komplett enthalten, also unter anderem auch unendlich sind, Belief Bases dagegen als eine endliche Menge einiger expliziten Einträgen sowie einem Beweiser, der Cn berechnet angesehen werden. Da Belief Bases eine endliche Repräsentationsform von „knowledge sets“ sind, sollen sie als Definition für eine Wissensbasis dienen.

Definition 17 (Wissensbasis) *Sei H eine endliche Menge von Formeln aus $\mathcal{L}$, $\vdash$ ein Beweiser für $\mathcal{L}$, der Cn für Formelmengen aus $\mathcal{L}$ berechnet. H ist genau dann eine Belief Base für ein Belief Set K, wenn $Cn(H) = K$. H wird in diesem Fall als Wissensbasis bezeichnet.*

Anmerkung: die obige Definition ist im wesentlichen aus [GR 1995] (S. 48) übernommen.

In dem Kontext der Verwendung von Belief Bases in einer konkreten Implementation ist es notwendig, die Berechnungskomplexität der Beweisvorgänge zu berücksichtigen. Im Kontext eines Agenten repräsentieren Wissensbasen einen Teil der Umwelt des Agenten. Hier ist es notwendig, darauf zu achten, welches Wissen über die Umwelt des Agenten aufgenommen wird, um den Berechnungsaufwand gering zu halten. Es könnte möglich sein, die Terme, die ein Beweiser zu prüfen hat, auf die Hornlogik zu beschränken, da ein Hornlogikbeweiser Beweise in linearer Zeit durchführen kann. Wird wenig

Wissen in die Wissensbasis aufgenommen, so wird über diese unter Umständen schneller räsoniert werden können. Andererseits muss die Menge der in der Wissensbasis aufgenommenen Sätze groß genug sein, um sinnvolle Ask- und Tell-Anweisungen durchführen zu können. Aus Gründen der Performanz sollen nur Einträge in der Wissensbasis gemacht werden, die zur Formulierung eines Sachverhalts unbedingt notwendig sind. Im Fall des geometrischen Agenten ist z.B. die in Kapitel 4.2 eingeführte Verhaltensäquivalenz ein solcher Sachverhalt. Es gibt verschiedene Möglichkeiten, Verhaltensäquivalenz durch logische Sätze auszudrücken, weshalb in Kapitel 4.3.3 diese Möglichkeiten in Hinblick auf ihre Performanz und Ausdrucksstärke beurteilt werden.

Wie in [AGM 1985] ausgeführt, führt Inkonsistenz innerhalb der Wissensbasis zu dem Problem, dass in einer inkonsistenten Wissensbasis (die absurde Wissensbasis genannt wird) jeder Satz folgerbar ist, was in [AGM 1985] als „useless for information handling purposes“ bezeichnet wird. Das Problem der Inkonsistenz von Wissensbasen führt dazu, dass eine Wissensbasis im Falle eines Tells Vorkehrungen treffen muss, um zu verhindern, dass eine absurde Wissensbasis entsteht. Eine solche Inkonsistenz entsteht, wenn einer Wissensbasis eine Aussage mitgeteilt wird, aus der die komplementäre Formel einer bereits in dem Belief Set der Wissensbasis enthaltenen Formel gefolgert werden kann. Eine Formel F ist zu einer Formel G komplementär, wenn $F \equiv \neg G$ oder $G \equiv \neg F$.

Wissensrevision

Das Problem der Integration von Wissen in eine Wissensbasis wurde von Alchourrón, Gärdenfors und Makinson (nachfolgend abgekürzt mit AGM) in [AGM 1985] formalisiert. Eine Zusammenfassung zum Thema Wissenrevision findet sich unter anderem in [GM 1988] und in [GR 1995]. Nach Gärdenfors und Makinson ([GM 1988]) werden die drei Operatoren auf Wissensbasen Expansion, Revision und Kontraktion genannt. Expansion wird in [GM 1988] als das Hinzufügen eines Satzes F sowie seiner logischen Konsequenzen in eine Wissensbasis charakterisiert, ohne dass dabei die entstehende Menge auf ihre Konsequenzen hin untersucht wird, was insbesondere bedeutet, dass durch Expansion eine absurde Wissensbasis entstehen kann, weshalb die Expansion nur unter der Vorbedingung, dass der hinzuzufügende Satz nicht in Widerspruch zu der bereits existierenden Wissensbasis steht, ausgeführt werden darf. Somit ist die Expansion eine reine Ergänzung. Wenn k eine Wissensbasis ist, so soll die durch die Expansion von k um den Satz x entstehende Wissensbasis als $(k + x)$ bezeichnet werden.

Kontraktion wird charakterisiert als das Zurückziehen eines Satzes aus der Wissenbasis ohne dem Hinzufügen einer weiteren Information, was dazu führt, dass ein explizit in der Wissensbasis eingetragener Satz entfernt werden muss. Die Bezeichnung für die aus der Kontraktion der Wissenbasis k um den Satz x entstehende Wissensbasis sei nachfolgend $(k - x)$.

Revision ist das Hinzufügen eines zur bestehenden Wissensbasis inkonsistenten Satzes, was dazu führt, dass einige der bestehenden Einträge in der Wissensbasis entfernt werden müssen oder die Revision ignoriert werden muss. Die aus der Revision der Wissensbasis k um den Satz x entstehende Wissensbasis soll mit $(k * x)$ bezeichnet werden.

Revision läßt sich nach [GR 1995] durch eine Komposition von Kontraktion und Expansion, die in [AGM 1985] als „Levi-Identität“ bezeichnet wird, ausdrücken. Die Levi-Identität lautet: $(k * x) = ((r - \neg x) + x)$. Ein mithilfe der Levi-Identität implementierter Revisionsoperator übernimmt die Eigenschaften von Expansion und Kontraktion.

Alle drei Operatoren werden durch je eine Menge von Rationalitätspostulaten charakterisiert, die unter anderem in [GR 1995] dargestellt werden. Diese Rationalitätspostulate reichen aber für einen konstruktiven Ansatz nicht aus, da es mehrere mögliche konkrete Operatoren gibt, die den Rationalitätspostulaten entsprechen. Im Rahmen einer Konstruktion eines Revisionsoperators gibt es in der Literatur verschiedene Herangehensweisen. Es gibt zum einen die sog. „Full-Meet Revision“. Neben der Full-Meet Revision existieren fünf weitere Herangehensweisen zur Konstruktion von Revisionsoperatoren. Diese sind nach [GR 1995] die „Partial-Meet Revision“, die Revision mithilfe einer Rangordnung der epistemischen Verankerung von Sätzen, der „Safe-Contraction“sowie der Verwendung von ordinalen Konditionalen. Von Bedeutung ist im Kontext der Konstruktion von Revisionsoperatoren die Frage, ob diese für ein Belief-Set oder eine Belief-Base verwendet werden. Belief-Sets umfassen unendliche Formelmengen, und sind damit für eine technische Umsetzung nicht geeignet. Allerdings lassen sich viele Verfahren, die für Belief-Sets geeignet sind, auch (abgewandelt) auf Belief-Bases anwenden. Die folgenden Ansätze zur Konstruktion eines Revisionsoperators werden zunächst für Belief-Sets dargelegt, wenn nicht explizit von Belief-Bases gesproochen wird.

Belief-Base

Eine Belief-Base ist nach [GR 1995] eine Menge von expliziten Sätzen einer logischen Sprache, die mithilfe eines Beweisers implizites Wissen aus den expliziten Sätzen ableiten können. Es gibt, wie Gärdenfors und Rott ausführen, unterschiedliche Ansichten zu der Frage, ob die Menge der Sätze in der Belief-Base endlich sein soll oder nicht. Da es in dem Kontext dieser Arbeit auch um die Möglichkeit einer Implementation geht, setze ich vorauß, dass alle Belief-Bases eine endliche Menge von expliziten Sätzen haben. Gärdenfors und Rott sind der Ansicht, dass eine Belief-Base ein Belief-Set „aufspannt".

Es gibt einen Unterschied zwischen der Auswirkung der Operatoren auf eine Belief-Base und ein Belief-Set. Dieser wird von Gärdenfors und Rott charakterisiert als „The principal idea of base revision is that syntax matters." ([GR 1995], S. 85) Es kann sein, dass zwei syntaktisch unterschiedliche Belief-Bases das gleiche Belief-Set aufspannen, aber nach einer Kontraktion um die gleiche Aussage die Belief-Sets der aus der Kontraktion resultierenden Belief-Bases unterschiedlich sind. In [GR 1995] wird dazu folgendes Beispiel gebracht: Seien $B^{'}$ und $B^{''}$ zwei Belief-Bases und $B^{'} = \{C, D\}$, $B^{''} = \{C \wedge D\}$. Werden $B^{'}$ und $B^{''}$ jeweils um C kontrahiert, so bleibt in $B^{'}$ D erhalten, während in $B^{''}$ sowohl C als auch D entfernt sind. Das Belief-Set von $(B^{'} - C)$ ist somit zu $(B^{''} - C)$ verschieden.

Das die Durchführung der gleichen Operation auf zwei Wissensbasen mit identischem Belief-Set zwei Wissensbasen mit unterschiedlichem Belief-Set erzeugen kann, führt dazu, dass es im Einzelfall zu prüfen ist, ob ein Ansatz zur Konstruktion eines Revisionsoperators für Belief-Sets auf Grundlage einer modellorientierten Sichtweise sich analog auf eine Belief-Base anwenden läßt.

Full- und Partial Meet Contraction

Sowohl Full- als auch Partial-Meet Revision basieren auf der Konstruktion der Revision über die Levi-Identität und unterscheiden sich in Hinblick auf die Kontraktion, die in beiden Fällen analog zu den Revisionsoperatoren benannt ist. Nachfolgend sei mit $(k \dot{-} x)$ die Maxichoice-Kontraktion bezeichnet. Die Idee hinter der Maxichoice-Kontraktion ist es, eine größte Teilmenge von k, aus der x nicht folgerbar ist, zu ermitteln („Their basic idea was to choose $(A \dot{-} x)$ as a maximal subset of A that fails to imply x.", [AGM 1985], Seite 511). Allerdings entsteht bei der Maxichoice-Revision das Phänomen, dass nach der Revision das entstehende Belief-Set komplett ist, und darüber hinaus zu jeder Aussage y entweder die Disjunktion $(x \vee y)$ oder $(x \vee \neg y)$ in dem durch die Revision $(k \dot{-} x)$ entstehenden Belief-Set enthalten sind. Umgangssprachlich ausgedrückt sind in der Wissensbasis nun Meinungen zu allen Aussagen enthalten, die vorher nicht in der Wissensbasis enthalten waren.

Das Verhalten der Maxichoice-Revision, neue Meinungen zu entwickeln, ist offensichtlich nicht immer gewünscht und aus diesem Grund führen [AGM 1985] die Partial-Meet Kontraktion ein. Die Kernidee der Partial-Meet Kontraktion ist es, eine Selektionsfunktion γ zu verwenden, um die Klasse der wichtigsten maximalen Teilmengen der ursprünglichen Wissensbasis auszuwählen. Die Kontraktion wird als der Durchschnitt aus allen durch γ gewählten maximalen Teilmengen definiert. Um γ zu spezifizieren, führen [AGM 1985] eine Präferenz-Relation über die maximalen Teilmengen der Menge der maximalen Teilmenge der Wissensbasis, welche die zu kontrahierende Aussage nicht implizieren, ein. Diese Präferenzrelation, die von einzelnen Kontraktionen unabhängig sein soll, soll dazu dienen, das beste Element der Menge der die zu kontrahierende Aussage nicht implizierenden Sätze aus der Wissensbasis zu ermitteln. Gegen die Implementation dieser Methode spricht nach [GR 1995] die extrem hohe Berechnungszeit der maximalen relevanten Teilmenge der Wissensbasis.

Auch die Partial-Meet Kontraktion liefert eine große Anzahl möglicher Revisionsoperatoren. Ein Extremfall ist die sogenannte „Full-Meet" Kontraktion, in welcher das durch Anwendung der Full-Meet Kontraktion entstehende Belief-Set auf die zu kontrahierende Aussage und ihre Konsequenzen beschränkt wird, während alle anderen Informationen verloren gehen.

Safe Contraction

Safe Contraction ist eine von Alchourrón und Makinson entwickelte Implementationsvariante einer Revisionsfunktion über die Spezifikation der Kontraktion. Jeder Safe Contraction Operator ist darüber hinaus nach Korrolar 3.3 in [AM 1985] ein Partial Meet Contraction Operator.

Die Kernidee der Safe Contraction ist die Einführung der Klassifikation von im Hinblick auf die Kontraktion sicheren („safe") Elementen. Bei der Kontraktion einer Belief-Base B um eine Aussage x ist das Kernproblem die Auswahl der Teilmenge von B, die x nicht impliziert. Ein Element von B ist nach [AM 1985] sicher in Hinblick auf x, genau dann wenn es nicht ein minimales Element von einer beliebiger Teilmenge C aus B ist, so dass $C \models x$. Zur Bestimmung der Minimalität wird die Existenz einer Hierarchie über einer Menge von Aussagen (wie der expliziten Sätze einer Belief-Base) vorausgesetzt. Eine nicht zirkuläre Hierarchie ist dabei eine Relation $<$, die nicht zirkulär ist, also für keine $b_1, \ldots b_n \in B$ gilt, dass $b_1 < b_2 < \cdots < b_n < b_1$. Die Definition des Kontraktionsoperators erfolgt in [AM 1985] durch die Gleichung $(B - x) = B \cap Cn(B/x)$, wobei B/x die Menge aller Elemente von B, die sicher in Hinblick auf x sind, ist.

In [GR 1995] wird der Hierarchie $<$, die von außerhalb der Wissensbasis vorgegeben werden muss, eine Ähnlichkeit zu der im nächsten Unterabschnitt vorgestellten epistemischen Verankerung zugeschrieben. Darüber hinaus ist aus technischer Sicht zur Implementation der Safe Contraction die Berechnung aller minimalen Teilmengen notwendig, die die zu kontrahierende Aussage enthalten, was ein großer Berechnungsaufwand ist.

Update und Model Change

Katsuno und Mendelzon führen in [KM 1991] einen von der Revision separierten Operator ein, den sie „Update" nennen, und liefern auch ein Verfahren zur Implementation. Nach Ansicht von [KM 1991] existieren zwei unterschiedliche Arten von Veränderungen, die einer Wissensbasis mitgeteilt werden können: zum einen Änderungen der Ansichten über die Welt oder des Wissens über die Welt, zum anderen Veränderungen an der Welt an sich. Aufbauend auf dieser Unterscheidung lassen sich der nach [KM 1991] zum Erfassen der Änderungen der Ansichten über die Welt angemessene Revisionsoperator und der zum Erfassen von Änderungen der Welt geeignete Update-Operator konstruieren.

Katsuno und Mendezon betrachten im Gegensatz zu [AGM 1985] Änderungen an den Modellen des Belief-Sets anstelle von Änderungen am Belief-Set als solches, was in [GR 1995] als „possible worlds model" bezeichnet wird. Ein wesenstliches Merkmal der Wissensrevision, bedingt durch die AGM-Postulate, ist die Minimierung der Änderungen an den Modellen des Belief-Sets, was im Falle der Erfassung von Änderungen in der Welt nicht zufriedenstellend ist. Aus diesem Grund wird in [KM 1991] eine weitere Sammlung von Postulaten, in denen anstelle der Minimierung der Änderungen der Wissensbasis eine Minimierung der Änderungen an den Modellen des Operanden, mit dem das Update ausgeführt werden soll, tritt, für den Update-Operator aufgestellt.

Der Ansatz, auf der Basis der Modelle von Belief-Sets einen (Update)-Operator auf Wissensbasen zu konstruieren, umfasst folgende Punkte :

1. Zwischen zwei Modellen kann eine „Distanz" auf der Basis der Anzahl der unterschiedlichen aussagenlogischen Atome ermittelt werden.

2. Nach diesem Distanzmaß exisitiert zu jedem Modell der Wissenbasis eine Menge von Modellen mit der geringsten Distanz zu der Aussage, die der Operator als Parameter erhält.

3. Die Menge aller Modelle der Wissensbasis, die durch Anwendung des Operator mit einem Parameter entsteht, ist definiert als die Vereinigung aller nächsten Modell zwischen jedem Modell der Wissensbasis und dem Parameter des Operators.

Der geometrische Agent besitzt eine großteilig statische Welt. Nur der Agent selbst bewegt sich. Daher ist die Verwendung des Updates in erster Linie für die Erfassung der Positionsänderung des Agenten sinnvoll. Die Frage, ob das Update in der in Kapitel 4 definierten Wissensbasis Verwendung finden wird, ist daher abhänging, ob die Position des Agenten in dieser explizit erfasst wird.

Ordinale Konditionalfunktionen

Die bisher vorgestellten Klassen von Operatoren, die die AGM-Postulate erfüllen, teilen das Wissen der Wissensbasis in zwei Zustände ein: Wissen, das in der Wissensbasis enthalten ist, und Wissen, das nicht in der Wissensbasis enthalten ist. Aus Sicht des „possible-world model“ ist dies nach [GR 1995] die Folge der Einteilung der möglichen Welten in nur zwei Klassen, wodurch es nicht möglich ist, eine feinere Einteilung der Plausibilität von verschiedenen möglichen Welten zu machen.

Als eine Möglichkeit, die Plausibilität möglicher Welten zu unterteilen, wird unter anderem in [SW88] die Einführung von ordinalen Konditionalfunktionen (nachfolgend mit κ bezeichnet), die eine gegebene Menge von möglichen Welten auf Ordinalzahlen abbildet, vorgeschlagen. Die Zuordnung der möglichen Welten muss unter der Voraussetzung geschehen, dass einige Welten auf die 0 abgebildet werden. Mögliche Welten, denen die 0 zugeordnet wird, sollen als die Plausibelsten angesehen werden. κ läßt sich auf Aussagen A erweitern, indem der Aussage das Minimum des Ergebnis der Konditionalfunktionen der Welten, aus denen A zusammengesetzt ist, zugeordnet wird. Dies hat nach [GR 1995] unter anderem zur Folge, dass für jede Aussage A entweder $\kappa(A)$=0 oder $\kappa(\neg A)$=0 ist.

Ordnet κ der Negation einer Proposition A eine Zahl > 0 zu, so spricht man davon, dass A in dem durch κ repräsentierten epistemischen Zustand akzeptiert wird. Diese einfache Akzeptierung einer Proposition in einem epistemischen Zustand kann um eine Plausibilitätsrelation ergänzt werden. Für die Plausibiltätsrelation zwischen zwei Propositionen A und B in Hinblick auf einen epistemischen Zustand κ gilt genau dann, wenn A plausibler als B in Hinblick auf κ ist, dann ist $\kappa(A) < \kappa(B)$ oder $\kappa(\neg B) < \kappa(\neg A)$.

Die Revisionsfunktion $(\kappa * (A, \alpha))$, die mithilfe der Plausibilitätsrelation definiert werden kann, wird (A, α)-Konditionalisierung genannt. Sie geht jedoch über die reine Revision hinaus, da sich nach [GR 1995] die Kontraktion um A durch eine (A, α)-Konditionalisierung umsetzen läßt. Durch die Plausibiltätsrelation besitzt die hier vorgestellte Repräsentationsform einen Mehrwert gegenüber den bisher erwähnten Formen zur Umsetzung der AGM Postulate. Nach [GR 1995] liegt ihr wesentlicher Nachteil darin, dass eine Revision einer Aussage ohne Plausibilitätsfaktor nicht möglich ist. Ein wesentlicher Vorteil dagegen bestehe darin, auch iterierte Revisionen sinnvoll ausführen zu können. Iterierte Revision mit Operatoren, die innerhalb des durch die AGM-Postulate gesteckten Rahmens liegen, kann unter anderem dazu führen, dass Sätze aus der Wissensbasis entfernt werden, obwohl dies nicht rational ist. In [DP 1996] wird dies durch das folgende Beispiel motiviert : Wenn ein unbekanntes Tier X beobachtet wird und es wie ein Hund zu bellen scheint, dann läßt sich schließen, dass X kein Vogel ist und X nicht fliegen kann. Falls weitere Beobachtungen zeigen, dass X fliegen kann, so wäre über X bekannt, dass X kein Vogel ist und X fliegen kann. Wenn X nun eindeutig als Vogel erkannt werden würde, so wäre es nicht sinnvoll, X die Flugfähigkeit abzusprechen. In [DP 1996] wird ein Operator, der die AGM Postulate erfüllt und ein solches Verhalten zuläßt, konstruiert. Darwiche und Pearl konstruieren mithilfe der (A, α)-Konditionalisierung einen Revisionsoperator, der das in obigem Beispiel geschilderte Problem vermeidet.

Ein Kernproblem aller Modell-basierten Ansätze, wie auch der (A, α)-Konditionalisierung ist es, dass sie in Belief-Bases keine direkte Verwendung finden können, da, wie auf Seite 24 im Abschnitt über die Belief-Base erwähnt, die Syntaxsensitivität von Belief-Bases dazu führt, dass unterschiedliche Belief-Bases das gleiche Belief-Set aufspannen können, aber nach der Anwendung eines Revisionsoperators auf jeweils beiden Belief-Bases die beiden durch die Revision entstandenen Wissensbasen unterschiedliche Belief-Sets haben können.

Die durch ordinale Konditionalfunktionen spezifizierte Revisionsfunktion kennzeichnet die Einteilung des Wissens in akzeptiertes Wissen (dessen Modell(e) κ auf 0 abbildet) und nicht akzeptiertes Wissen, was allerdings durch die Plausibilitätsrelation geordnet ist. Ob ordinale Konditionalfunktionen zur Modellierung der Wissensbasis des Agenten verwendet werden sollen, hängt damit von der Frage ab, ob es gewünscht ist, dass nicht akzeptierte Wissen zu ordnen.

Eine endliche Darstellung der Wissensbasis ist mit ordinalen Konditionalfunktionen nicht zwingend gewährleistet, was ohne weitere Untersuchungen, ob eine endliche Repräsentation möglich ist, eine Computer basierte Implementation unter Umständen nicht zuläßt.

Epistemische Verankerung

Die Grundidee epistemischer Verankerung ist nach [GR 1995] die Beobachtung, dass nicht jeder Satz innerhalb eines Belief-Sets den gleichen Wert für Räsonieren und Problemlösen besitzt. Wenn verschiedene Sätze eines Belief-Sets einen unterschiedlichen Wert besitzen, so wird davon gesprochen, dass sie einen unterschiedlichen Grad („degree“) an epistemischer Verankerung haben. Ausgehend von einer Quantifizierung der epistemischen Verankerung läßt sich eine qualitative Beziehung (Relation) zwischen Sätzen eines Belief-Sets herstellen. Eine wesentliche Eigenschaft der Relation der epistemischer Verankerung ist dabei die Verknüpfung von logischen Beziehungen der Sätze des Belief-Sets mit dem Grad ihrer epistemischen Verankerung. Die epistemische Verankerung selbst stammt von außerhalb der Wissensbasis. So erklären Gärdenfors und Makinson in [GM 1988], dass die epistemische Verankerung einer Aussage die Wichtigkeit der Aussage für Problemlösung und Planung auf Grundlage der Wissensbasis ist und auf diese Art die Datenbankpriorität („database priority“) der Aussage festlegt.

Das Ziel der Revision ist es, im Falle von verschiedenen Aussagen, die in Konjunktion zusammen eine Kontradiktion bilden, diejenigen, die in der Wissensbasis bestehen bleiben, auszuwählen um so zu verhindern, dass die absurde Wissensbasis gebildet wird. Dabei ist die Kernfrage, welche der konfligierenden Aussagen behalten wird. Durch die Einführung der epistemischen Verankerung von Aussagen im Rahmen der Revision die Aussage ausgewählt werden, die epistemisch stärker verankert ist.

Ein Schritt zur Spezifikation eines Revisionsoperators auf der Basis von epistemischer Verankerung ist die Charakterisierung der Relation der epistemischen Verankerung. In [GM 1988] geschieht dies durch die Einführung der Postulate (EE1) bis (EE5). Diese 5 Postulate beinhalten dabei:

1. Die Forderung nach Transitivität der epistemischen Verankerung zwischen drei Aussagen

2. Dominanz, die bedeutet, dass, wenn eine Aussage B aus einer Aussage A folgerbar ist, der Grad von A $\leq$ dem Grad von B sein muss. Gärdenfors und Makinson begründen die Forderung von Dominanz damit, dass, falls entweder A oder B kontrahiert werden sollen, die Entfernung von A die geringere Änderung an der Wissensbasis ist.

3. Konjunktivität, die aussagt, dass für A und B gilt: Der Grad von $A \leq$ dem Grad von $A \wedge B$ oder der Grad von $B \leq$ dem Grad von $A \wedge B$. Dies begründen sie damit, dass im Falle der Kontraktion um die Aussage $(A \wedge B)$ entweder A oder B kontrahiert werden müssen, weshalb der „Informationsverlust entweder mit dem Informationsverlust mit der Aufgabe von A oder mit der Aufgabe von B korrespondieren muss.

4. Minimalität, die aussagt, dass eine Aussage A genau dann nicht Teil des Belief-Sets ist, wenn der Grad von $A \leq$ dem Grad von B für alle B, die Teil des Belief-Sets sind.

5. Maximalität, die verhindert, dass nicht gültige Aussagen innerhalb des Belief-Sets den maximalen Grad besitzen.

In [GR 1995] werden zu (EE1) – (EE5) analoge Postulate für eine strikte Ordnung formuliert. [GM 1988] zeigen mithilfe von zwei Bedingungen, dass es möglich ist, aus einem vorgegebenem Belief-Set und einer Ordnung der epistemischen Verankerung einen Kontraktionsoperator zu konstruieren (und umgekehrt aus einem Belief-Set und einem Kontraktionsoperator eine Rangordnung der epistemischen Verankerung). Mithilfe der Levi-Identität und der (relativ einfach zu realisierenden) Expansion läßt sich aus dem Kontraktionsoperator ein Revisionsoperator definieren.

3.3 Finite Partial Entrenchment Rankings

Grundlagen

Wie eine konkrete Implementation eines Revisionsoperators mithilfe von epistemischer Verankerung aussehen kann, beschreibt Williams in [W 1998]. Bei der Implementation tritt das Problem auf, dass ein auf epistemischer Verankerung beruhender (Revisions)operator zunächst eine Belief-Base erstellt, allerdings dabei die Rangordnung der epistemischen Verankerung verloren geht, womit eine iterierte Revision nicht möglich ist. Des weiteren muss das Problem gelöst werden, ein unendliches Belief-Set mithilfe einer endlichen Belief-Base zu repräsentieren und dabei ebenfalls die unbegrenzte Menge der impliziten (aus der Belief-Base folgerbaren) Sätze zu ordnen.

Die Kernidee von Williams, das Repräsentationsproblem zu lösen, liegt in der Einführung von sogenannten Finite Partial Entrechment Rankings, die in dieser Arbeit mit „endliche partielle Rangordnung der epistemischen Verankerung“ oder kurz als „FPER“ bezeichnet werden. FPER sind Funktionen, die Einträgen einer Belief-Base Werte aus dem Intervall $[0, 1]$ zuordnet, wobei ein größerer Zahlenwert eine stärkere epistemische Verankerung repräsentiert. Die Zuordnung von Sätzen zu einem Zahlenwert wird in einigen Fällen in dieser Arbeit auch als „bewerten“ bezeichnet.

Definition 18 (Endliche partielle Rangordnung der epistemischen Verankerung (FPER)) *Eine endliche partielle Rangordnung der epistemischen Verankerung (nachfolgend abgekürzt mit FPER) ist eine Funktion B aus einer endlichen Teilmenge von Sätzen von $\mathcal{L}$ in das Intervall $[0, 1]$, so dass für alle $\varphi \in dom(B)$ folgende Bedingungen erfüllt werden :*

1. $\{\psi \in dom(B) : B(\varphi) < B(\psi)\} \not\models \varphi$[2]*. Dies bedeutet, dass φ aus keiner Menge von Aussagen, die höher als φ gewertet sind, folgerbar ist.*
2. *Wenn $\models \neg\varphi$, dann $B(\varphi) = 0$. Kontradiktionen sind damit immer mit 0 bewertet.*
3. *$B(\varphi) = 1$ genau dann, wenn $\models \varphi$.*

Eine FPER repräsentiert eine mit epistemischer Verankerung angereicherte Belief-Base. In einer Belief-Base B bilden alle > 0 gewerteten Sätze die Menge der expliziten Informationen, der logische Abschluss Cn(B) die impliziten Informationen. Um nicht nur den Abschluss einer FPER zu berechnen, sondern auch jedem Satz aus Cn eine epistemische Verankerung zuzuordnen, ist ein weiterer Mechanismus notwendig. Williams führt aus diesem Grund die Funktion *degree* ein, die allen nichttautologischen Sätzen eine eindeutige epistemische Verankerung, unabhängig davon, ob diese explizit oder implizit sind, zuordnet. Eine solche Funktion, die aus einer partiellen epistemischen Rangordnung eine vollständige erstellt, ist nicht eindeutig. Wie Williams ausführt, können z.B. $B(\varphi) = 0,2$ und $B(\psi) = 0,4$ sein. Dies sorgt dafür, dass $\varphi < \psi$ ist, aber es spezifiziert nicht, ob $\psi < \varphi \vee \psi$ oder $\psi = \varphi \vee \psi$ ist. Die von Williams vorgeschlagene Definition von *degree* geht dabei den Weg, einem Satz die kleinst mögliche epistemische Verankerung zuzuordnen.

Definition 19 (Degree) *Sei φ ein nichttautologischer Satz, B ein FPER. Dann ist*

$$degree(B, \varphi) = \begin{cases} \textit{größte } j, \textit{ so dass } \{\psi \textit{ explizit in } B \ : B(\psi) \geq j\} \models \varphi, & \textit{wenn } \varphi \in Cn(B) \\ 0, & \textit{sonst} \end{cases}$$

Ein wesentliches Ergebnis von Williams ist darüber hinaus der Nachweis, dass eine FPER ein Weg zur Modellierung einer Rangordnung der epistemischen Verankerung ist ([W 1998], Theorem 15).

Williams nächster Schritt zur Implementation eines auf epistemischer Verankerung basierenden Revisionsoperators ist die Einführung einer von ihr „Adjustment“ genannten Funktion. Ein Adjustment

[2] Die obige Definition weicht in Hinblick auf das verwendete Symbol zur Folgerung von der in [W 1998] verwendeten ab. Williams benutzt das Symbol $\vdash$, was für gewöhnlich zur Bezeichnung eines Beweisverfahrens genutzt wird, anstelle von $\models$, was der üblichen logischen Folgerbarkeit entspricht. Ich habe aus diesem Grund das gängige Symbol $\models$ anstelle von $\vdash$ verwendet.

definiert auf der Basis eines existierenden FPERs B ein neues FPER, was das Ergebnis der Revision von B um die Aussage φ, der eine epistemische Verankerung i zugeordnet ist, ist. Die Unterscheidung zwischen einer reinen Revision und einem Adjustment ist darin begründet, dass die Revision keine Rangordnung der epistemischen Verankerung des Ergebnis erzeugt.

Das Adjustment orientiert sich an der Konstruktion eines Revisionsoperators auf Basis der Levi-Identität, da es aus zwei elementaren Funktionen zusammengesetzt ist, die im wesentlichen Varianten von Expansion und Kontraktion unter Berücksichtung der Rangordnung der epistemischen Verankerung sind.

Definition 20 (Adjustment) *Sei φ eine kontingente Formel aus $\mathcal{L}$ und $0 \leq i < 1$. Dann ist das Adjustment einer endlichen partiellen Rangordnung der epistemischen Verankerung die Funktion *, so dass*

$$B^*(\varphi, i) = \begin{cases} (B^-(\varphi, i)), & \textit{wenn } i \leq \textit{degree}(B, \varphi) \\ \\ (B^-(\neg\varphi, 0))^+(\varphi, i), & \textit{sonst} \end{cases}$$

Das Kontraktionsanalogon (von Williams „Abwärtsbewegung“ benannt) ist definiert als

$$B^-(\varphi, i)(\psi) = \begin{cases} i, & \textit{wenn degree}(B, \varphi) = \textit{degree}(B, \varphi \vee \psi) \textit{ und } B(\psi) > i \\ B(\psi), & \textit{sonst} \end{cases}$$

für alle explizit in B aufgenommenen Formeln. Das Expansionsanalogon (von Williams „Aufwärtsbewegung“ benannt)

$$B^+(\varphi, i)(\psi) = \begin{cases} B(\psi), & \textit{wenn } B(\psi) > i \\ i, & \textit{wenn } \varphi \equiv \psi \textit{oder } B(\psi) \leq i < \textit{degree}(B, \varphi \supset \psi) \\ \textit{degree}(B, \varphi \supset \psi), & \textit{sonst} \end{cases}$$

ist definiert für alle explizit in B aufgenommenen Formeln sowie φ.

Anmerkung: diese Defitinition entspricht Definition 16 aus [W 1998]. Des weiteren ist darauf zu achten, dass ein Adjustment nicht für ein inkonsistentes oder tautologisches φ definiert ist.

Williams charakterisiert das Verhalten dieses Revisionsoperator als Verwendung des kleinst möglichen Maßes von Veränderung („. . . and use a policy for change based on an absolute minimal measure; they transmute a finite partial entrenchment ranking so as to incorporate the desired new information using an absolute minimal measure of change.“, [W 1998], S. 298). Eine weitere Eigenschaft des Adjustments ist nach Williams das Fokusieren auf die Relationen der epistemischen Verankerung, während die absoluten Werte ohne Bedeutung sind.

Das Adjustment hat nach Williams den Nachteil, dass es die Spezifikation von Informationsunabhängkeit erfordert, obwohl es in vielen Bereichen sinnvoll ist, die Abhängigkeiten zu spezifizieren („For the purpose of developing real world applications, a major shortcoming of adjustment is that the order to achieve desired behavior the user must specify when information is independent with respect to change. For many applications it is more natural to specify dependence.“, [W 1998], S. 299). Als Alternative spezifiziert Williams die Maxi-Adjustment Funktion, in welcher Änderungen nur anhand der explizit in der FPER eingetragenen Abhängigkeiten durchgeführt werden. In [W 1996] wird die Einführung des Maxi-Adjustments damit begründet, dass es für einen Systementwickler leichter ist, die Abhängigkeiten von von Informationen zu spezifizieren, anstelle der Unabhängigkeiten. Die Kerneigenschaft des Maxi-Adjustments ist die standardmäßige Annahme der Unabhängigkeit der im FPER enthaltenen Aussagen, was Williams als „closed world assumption with respect to information dependence under change“([W 1996], S. 412) bezeichnet.

Definition 21 (Maxi-Adjustment) *Sei B, ein FPER, $j_0, j_1, \ldots, j_{\mathcal{R}max}$ die aufsteigend geordnete Aufzählung der Bildmenge von B, $\mathcal{R}max$ die Kardinalität der Bildmenge von B, φ ein kontingenter*

Satz, $j_m = degree(B,\varphi)$ und $0 \leq i < \mathcal{R}_{max}$. Dann ist das (φ,i)-Maxi-Adjustment von $B^*(\varphi,i)$ definiert als

$$B^*(\varphi,i) = \begin{cases} (B^-(\varphi,i)) & \text{wenn } i \leq degree(B,\varphi) \\ \\ (B^-(\neg\varphi,0))^+(\varphi,i) & \text{sonst} \end{cases}$$

$B^-(\varphi,i)(\psi)$ wird dabei für alle explizit in B eingetragenen logischen Sätze ψ definiert als

1. *Für epistemisch stärker als φ verankerte ψ ist $B^-(\varphi,i)(\psi) = B(\psi)$. Dies bedeutet, dass sich an der epistemischen Verankerung aller Einträge, die stärker als φ verankert sind, nichts verändert.*

2. *Für Einträge ψ, mit $i < B(\psi) < j_m$ gilt unter der Annahme, dass $B^-(\varphi,i)(\psi)$ für die ψ mit $B(\psi) \geq j_{m-k}$ und $k = -1,0,1,2,\ldots,n-1$ definiert ist, dass diejenigen ψ mit $B(\psi) = j_{m-n}$*

$$B^-(\varphi,i)(\psi) = \begin{cases} & \text{Wenn } \psi \text{ aus } \varphi \text{ folgerbar ist, oder} \\ & \text{wenn } \psi \text{ nicht aus } \varphi \text{ folgerbar ist und } \psi \in \Gamma \\ i & \text{wobei } \Gamma \text{ eine minimale Teilmenge von} \\ & \{\psi : B(\psi) = j_{m-n}\} \text{ so dass} \\ & \{\psi : B^-(\varphi,i)(\psi) > j_{m-n}\} \cup \Gamma \models \varphi \\ B(\psi) & \text{sonst} \end{cases}$$

3. *Im Falle von ψ mit $B(\psi) \leq i$ ist $B^-(\varphi,i)(\psi) = B(\psi)$.*

$B^+(\varphi,i)(\psi)$ wird für alle explizit in B stehenden Formeln sowie φ und ψ wie folgt definiert :

$$B^+(\varphi,i)(\psi) = \begin{cases} B(\psi) & \text{wenn } B(\psi) > i \\ i & \text{wenn } \varphi \equiv \psi \text{ oder } B(\psi) \leq i < degree(B,\varphi \supset \psi) \\ degree(B,\varphi \supset \psi) & \text{sonst} \end{cases}$$

Performanz des Maxi-Adjustments

Während der Ausführung von B^+ und B^- im Rahmen eines Adjustments $B^*(\varphi,i)$ wird jeweils einmal jeder Eintrag in der FPER durchlaufen. In der Revision selbst wird entweder die Kontraktion oder Kontraktion gefolgt von Expansion aufgerufen, und dies für jede explizit im FPER eingetragene Aussage. Da Expansion und Kontraktion ebenfalls jede Aussage bearbeiten ist die Berechnungskomplexität des Adjustments daher polynomiell von der Anzahl der Einträge des FPERs abhängig.

Der Aufwand der Abarbeitung der einzelnen Einträge in B^- und B^+ ist unterschiedlich. Im Rahmen der Bearbeitung von Einträgen, die eine stärkere epistemische Verankerung als der Eintrag, um den das FPER revidiert werden soll, wird die epistemische Verankerung von B übernommen. Dabei ist keine zeitaufwändige Berechnung notwendig. In den anderen Fällen ist zum einen die Berechnung von $\varphi \models \psi$ sowie $\Gamma \models \varphi$, die in der Kontraktion verlangt werden, zum anderen die Berechnung der Äquivalenz zweier Formeln in der Expansion von Bedeutung.

Darüber hinaus wird die Berechnung des Grads der epistemischen Verankerung von Formeln in vielen Fällen notwendig. Exakt einmal wird der Degree von φ in B berechnet. Wird eine Kontraktion um $\neg\varphi$ ausgeführt, so muss mindestens einmal der Degree von $\neg\varphi$ berechnet werden. Besonders innerhalb der Expansion ist eine Vielzahl von Aufrufen der degree-Funktion notwendig, da für jedes ψ mit kleinerer oder gleicher epistemischer Verankerung von i der Degree von $\varphi \supset \psi$ berechnet werden muss. Die Berechnung von Degree sowie der Folgerbarkeit und Äquivalenz wird innerhalb der Aussagenlogik auch unter der Verwendung eines Tableau-Beweisers im schlimmsten Fall exponentiell schlecht sein. Um die Berechnung von degree zu umgehen, empfiehlt Williams wichtige Informationen explizit in die Wissensbasis einzutragen. Da dies aber nur in einer endlichen Zahl von Fällen möglich ist und außerdem dafür die Berechnung des Degrees aller nicht explizit eingetragenen Sätze verlangsamt, ist es ein lohnendes Ziel, die Performanz von degree direkt zu verbessern.

Implementation der Berechnung von Degree

Eine Implementation von *degree(B,φ)* weist Ähnlichkeiten zu der Berechnung der Folgerbarkeit von φ aus einer beliebigen Belief-Base auf. Im Fall einer gewöhnlichen Belief-Base Kb mit einem Tableau Beweiser $\vdash_{pt}$ könnte dies dadurch geschehen, dass Kb $\models \varphi$ auf einen Tautologiebeweis der Implikation der Konjunktion aller Formeln $f_i \in$ Kb auf φ zurückgeführt wird. Es gibt dabei eine Vielzahl von syntaktisch unterschiedlichen Möglichkeiten, die Formel zu bilden, die der Beweiser zu prüfen hat. Allerdings ist ihnen gemein, dass in jeder Formel mindestens alle Elemente von Kb sowie φ enthalten sind. Der Nachteil dieses Vorgehens ist es, dass nicht klar ist, welcher Eintrag oder welche Einträge für den Beweis notwendig sind. Die Feststellung, welche Einträge aus Kb zum Beweis von φ notwendig ist, wäre möglich indem die Potenzmenge aller Einträge aus Kb (nachfolgend bezeichnet mit 2^{Kb}) gebildet wird, und eine kleinste Teilmenge von 2^{Kb} gesucht wird, aus der φ folgerbar ist. Dies führt dazu, dass eine Vielzahl von oftmals ähnlichen Beweisen notwendig ist. Eine Implementation von *degree*(B, φ) erfordet die Berechnung der größten epistemischen Verankerung j, so dass aus der Menge der expliziten Einträgen aus B, die größer oder gleich j sind, φ folgerbar ist. Seien 2^{B_ψ} die Potenzmenge der Formeln aus B und 2^{B_ϵ} die Potenzmenge der epistemischen Verankerung der Formeln aus B. Ein Element $\psi \in 2^{B_\psi}$ *korrespondiert* genau dann mit einem Element $\epsilon \in 2^{B_\epsilon}$, wenn ϵ die epistemische Verankerung von ψ nach B ist.

Eine intuitive, aber rechenintensive Lösung der Berechnung von *degree* ist es, für jedes Element $e \in 2^{B_\psi}$ die Folgerbarkeit $e \vdash_{pt} \varphi$ zu berechnen. Zu der Teilmenge L von 2^{B_ψ}, aus deren Elementen φ folgerbar ist, existiert nun eine Teilmenge K mit korrespondierenden Elementen aus 2^{B_ϵ}. Der *degree* von φ ist nun $\max(\{\min(k)|k \in K\})$.

Das oben geschilderte Verfahren ist aufgrund der Behandlung aller Teilmengen von 2^{B_ψ} sehr rechenaufwendig. Es ist aber nicht notwendig zu prüfen, ob aus allen Teilmengen von 2^{B_ψ} φ folgerbar ist. Beispiel: seien $\psi_1, \psi_2, \psi_3, \psi_4 \in 2^{B_\psi}$ sowie $B(\psi_1) > B(\psi_2)$ und $B(\psi_2) = B(\psi_3) = B(\psi_4)$. Angenommen, φ ist aus $\psi_1 \cup \psi_2$ und $\psi_1 \cup \psi_4$ folgerbar. Dann ist $L = \{\{\psi_1, \psi_2, \psi_3, \psi_4\}, \{\psi_1, \psi_2, \psi_3\}, \{\psi_1, \psi_2\}, \{\psi_1, \psi_3, \psi_4\}, \{\psi_1, \psi_4\}, \{\psi_1, \psi_2, \psi_4\}\}$. Die Elemente von K sind in diesem Beispiel alle gleich, da ihr Wert jeweils $B(\psi_2)$ ist. Es ist jedoch nicht notwendig, für jedes Element der Potenzmenge die Folgerbarkeit von φ zu testen. Wenn es n unterschiedliche Stufen der epistemischen Verankerung in B gibt, dann reicht es aus, die Folgerbarkeit von φ aus n unterschiedlichen Formelmengen zu testen. Für jedes $i \in \text{im}(B)$ sei $\mathcal{F}_i = \{f|f \in \text{dom}(B), B(f) \geq i\}$. Die durch $\mathcal{F}_i$ berechnete Formelmenge enthält alle Formeln, die notwendig sind, um zu prüfen, ob der Degree von φ mindestens i ist. Wenn kein i existiert, so dass $\mathcal{F}_i \vdash_{pt} \varphi$, so ist der Degree von φ 0. Ansonsten ist der Degree das größte i, so dass $\mathcal{F}_i \vdash_{pt} \varphi$.

Algorithmus 1 (Berechnung von Degree) *In Pseudocode ausgedrückt kann die Berechnung von degree*(B, φ) *durch folgendes Verfahren geschehen :*

1. *Initialisiere j mit der höchsten epistemischen Verankerung in B*
2. *Berechne die Menge* $\mathcal{F}_j$
3. *Prüfe, ob* $\mathcal{F}_j \models \varphi$
4. *Wenn Prüfung erfolgreich ist, so beende und gebe j zurück*
5. *Wenn Prüfung nicht erfolgreich so sei* j_{neu} *die größte epistemische Verankerung* $< j$ *oder 0, wenn es keine kleinere epistemische Verankerung als j gibt*
6. *Wenn* j_{neu} *0 ist, so beende und gebe 0 zurück, da* φ *nicht aus B folgerbar ist*
7. *Setze* $j = j_{neu}$ *und fahre mit Schritt 2 fort*

Offensichtlich ist, dass bei obigem Verfahren eine Vielzahl von redundanten Beweisen erfolgt. Muss einer Formel der Grad 0 zugeordnet werden, so müssen insbesondere die epistemisch hoch bewerteten Formeln mehrmals expandiert werden. Der letzte Beweis, der für die Berechnung des Degrees einer Aussage, die der Grad 0 zugeordnet werden muss, notwendig ist, ist identisch mit demjenigen, der in einer Belief-Base ohne epistemische Zusatzinformationen bei der Prüfung der Folgerbarkeit von φ durchgeführt wird. Wird anstelle eines gewöhnlichen Tableau-Beweisers ein Tableau-Beweiser, der eine Spezifikation der **Priorität** der Expansionsschritte erlaubt, verwendet, so kann *degree* in einem Schritt, der alle Formeln aus B umfasst, berechnet werden. Die Durchführung der Berechnung des Degrees einer Formel kann unter der Voraussetzung, dass $\vdash_{pt}$ von außen priorisierbar ist, auf den Beweis von $\mathrm{dom}(B) \vdash_{pt} \varphi$ beschränkt werden.

Innerhalb der Formelmenge, aus der φ gefolgert werden soll, muss dabei jedem explizitem Eintrag aus B eine Priorität entsprechend der epistemischen Verankerung zugewiesen werden. Die Priorisierung hat zum Ziel, dem Tableau-Beweiser zu ermöglichen, das Tableau mit den epistemisch am stärksten verankerten Formeln abzuschließen. Liefert der Beweiser zusätzlich zu der Information, ob φ aus B folgerbar ist, die niedrigste Priorität der verwendeten Formeln mit, so kann daraus der Degree von φ geschlossen werden.

Grundsätzlich kann gegen eine Spezifikation der Reihenfolge der Expanssionschritte eines Tableau-Beweisers eingewendet werden, dass dadurch unter Umständen eine für den Berechnungsaufwand her ungünstige Expansionsreihenfolge entsteht. Aus diesem Grund soll auch der angepasste priorisierte Beweiser in der Lage sein, im Falle der Existenz mehrerer gleich gewerteter Formeln diese nach eigener Expertise zu expandieren. Eine strikte Prioritätsrelation ist deshalb nicht erwünscht. Es bietet sich an, die epistemische Verankerung jeder explizit in B vorhandenen Formel auch als Prioritätslevel für den priorisierten Beweiser zu verwenden.

Es bleibt dann die Frage, wie φ zu priorisieren ist. Eine Lösung besteht darin, φ jeweils die höchste Priorität zuzuordnen. Eine hohe Priorisierung von φ ist auch deshalb notwendig, da die Formelmenge der explizit in B gespeicherten Formeln kontingent ist. Wird eine kontingente Formelmenge einem Tableau-Beweiser übergeben, so kann dieser das Tableau nicht abschließen, da er notwendigerweise ein Modell finden muss. Ein Tableau-Abschluss kann unter der Voraussetzung der Verwendung einer nicht-absurden Wissensbasis ausschließlich durch Einbeziehung von φ geschehen. Die Notwendigkeit der Verwendung φ zum Abschluss des Tableaus bedeutet, dass φ sehr früh expandiert werden sollte, um den Tableau-Beweis nicht überflüssig zu verlängern.

Zur Implementation des prioritätsbasierten Tableau-Beweisers bietet es sich an, einen markierten Tableau-Beweiser zu verwenden und die Priorität als Markierung dem Beweiser zu übergeben. Da es möglich ist, dass der Tableau-Beweiser auch andere Markierungen verwendet, wird zur Unterscheidung von diesen die Prioritätsmarkierung kurz mit „priority“ benannt. Hier entsteht dass Problem, dass die Markierung für eine Gesamtformel gilt. Wird der Beweiser für den Beweis $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \models \varphi$ aufgerufen, so muss es aber möglich sein, jedem α und φ eine eigene Priorität zuzuweisen. Aus diesem Grund ist es notwendig, dass der Tableau-Beweiser ein passend vorgefertigtes Tableau erhält. Dies besteht im Falle des Beweises von $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \models \varphi$ aus einem einzelnen Zweig mit den Einträgen $\alpha_1 \ldots \alpha_n$ und $\neg\varphi$, wobei jeder Formel in diesem Zweig die passende Prioritätsstufe als Markierung zugewiesen wird. Dabei gibt es einen Sonderfall zu beachten: wenn in einem Tableauzweig ein Eintrag X mit priority p_1 existiert und die Anwendung einer Expansionsregel dazu führt, dass ein weiterer Eintrag X mit priority p_2 in den Tableauzweig hinzugefügt werden muss, so ist es im Falle eines Tableaus ohne Prioritätsteuerung aus Gründen der Redundanzvermeidung sinnvoll, den expandierten Eintrag nicht dem Zweig hinzuzufügen. Dies trifft auch für ein prioritätsgesteuertes Tableau zu. Allerdings stellt sich hier die Frage, welche Priorität der Eintrag X haben sollte. Um die Korrektheit der Berechnung des Degrees zu gewährleisten muss im Konfliktfall immer die größere Priorität gewählt werden, da ansonsten der berechnete Degree zu gering ist. Die Notwendigkeit für diese Art der Prioritätsvergabe kann in Beispiel 3 auf Seite 89 eingesehen werden.

Ist der priorisierte Tableau-Beweiser mit einem wie oben beschriebenen Tableau initialisiert worden, so muss der priorisierte Beweiser im Vergleich zu einem gewöhnlichem Tableau-Beweiser um folgende Funktionalität ergänzt werden:

Innerhalb der Initialisierungsphase:

Erstelle Variable p (aktuelle Prioritätsstufe) und setze p auf das Maximum der priority Markierung

Vor der Anwendung einer Expansionsregel : Ermittle die Menge M der Einträge im aktuellen Zweig bei denen priority $\geq$ p ist, und beschränke die Auswahl der besten Formel für die Expansion auf diese Menge.

Wenn keine Formel aus M expandierbar ist, so reduziere p auf die nächst höchste Prioritätsstufe, die kleiner als p ist, prüfe ob der Zweig abgeschlossen werden kann und fahre wenn dies nicht der Fall ist mit der Expansion im aktuellen Zweig fort. Falls es keine kleinere Prioritätsstufe als p gibt, so terminiere und gebe unerfüllbar zurück.

Nach der Anwendung einer Expansionsregel: Füge in den neu expandierten Einträgen die priority p hinzu. Falls ein Eintrag expandiert wurde, der bereits im aktuellen Tableau Zweig existiert, so behalte denjenigen Eintrag mit der größten Priorität.

Zur Prüfung des Abschlusses des aktuellen Zweigs : Beschränke die Abschlussprüfung auf die Menge der Formeln, deren priority $\geq$ p ist.

Erfolgreiche Termination: Falls kein offener Zweig mehr existiert, so gebe erfüllbar mit Prioritätsstufe p zurück.

Anmerkung: die oben geschilderten Ergänzungen berechnen nur dann korrekt den Degree von φ, wenn priority von $\neg\varphi$ mit einem Wert größer oder gleich allen anderen Prioritätsstufen initialisiert wurde. Dies liegt daran, dass innerhalb einer kontingenten Wissensbasis der Abschluss des Tableaus nur unter Einbeziehung von $\neg\varphi$ möglich ist. Es sollte deshalb $\neg\varphi$ eine Priorität von 1 zugewiesen werden.

Kapitel 4

Aufbau einer revisionsbereiten Wissensbasis

4.1 Grundlagen

Aus der Sicht des Agenten besitzt seine Wissensbasis, wie in 3.2 erwähnt, die Funktionen Tell und Ask. Tell dient dazu, der Wissensbasis neue Informationen mitzuteilen, Ask dazu, Wissen des Agenten abzufragen. Welches Wissen des Agenten der revisionsbereiten Wissensbasis mitgeteilt werden soll und welches abgefragt werden kann sind daher zwei grundlegende Fragen, die vor der Spezifikatation der eigentlichen revisionsbereiten Wissensbasis notwendigerweise beantwortet werden müssen.

Neben der Frage, welches Wissen der Agent der revisionsbereiten Wissensbasis mitteilen soll, muss entschieden werden, wie dieses Wissen innerhalb der revisionsbereiten Wissensbasis repräsentiert wird. Grundsätzlich wird in Kapitel 3.2 die Wissensbasis als ein Belief-Set definiert. Da Belief-Sets eine im Normalfall unbegrenzte Formelmenge umfassen, muss zur technischen Realiserung eine endliche Belief-Base verwendet werden. Belief-Bases bestehen aus einer endliche Menge von Sätzen einer logischen Sprache sowie einem Beweiser, der aus diesen expliziten Sätzen die Konsequenzen ableitet und so das unendliche Belief-Set für die explizien Sätze aufspannt. Die Verwendung von Belief-Sets zur Repräsentation scheidet aufgrund ihrer Unbegrenztheit und dem Anspruch dieser Arbeit, eine implementationsfähige Wissensbasis aufzubauen, aus.

Der Aufbau einer revisionsbereiten Wissensbasis erfordert deshalb die Zusammenstellung einer (endlichen) Menge von Sätzen einer logischen Sprache als expliziten Anteil der Wissensbasis. Dabei ist zu beachten, dass der Beweiser selbst im Fall von Aussagenlogik im Hinblick auf die Zahl der verwendeten atomaren Symbole exponentiell langsam arbeitet. Des weiteren besteht im Fall von Belief-Bases das Problem, dass diese syntaxabhängig sind und zwei syntaktisch unterschiedliche Belief-Bases, die das gleiche Belief-Set repräsentieren, nach einer Kontraktion oder Revision unterschiedliche Belief-Sets aufspannen können. Aus diesen zwei Gründen folgt, dass beim Aufbau der Wissensbasis besonders die Syntax der expliziten Einträge der Belief-Base von Bedeutung ist.

Da der Revisionsoperator am besten für die Erfassung der Änderung der Ansichten über die Welt, nicht jedoch für die Erfassung der Änderungen innerhalb der Welt geeignet ist, stellt sich die Frage, ob ergänzend neben einem Wissenrevisionsoperator ein Update-Operator verwendet werden sollte. Da die Welt des Agenten mit Ausnahme des Agenten selbst statisch ist, kommt nur die Veränderung der Position des Agenten selbst als Möglichkeit für ein Update in Frage. Es ist daher zu prüfen, ob die Position des Agenten explizit in der revisionsbereiten Wissensbasis aufgenommen werden sollte.

Es stehen für Belief-Bases eine Vielzahl von unterschiedlichen Operatoren zur Durchführung einer Revision zur Verfügung. Viele von diesen Operatoren besitzen inherente Probleme, insbesondere in Hinblick auf ihre Berechnungskomplexität (siehe dazu Abschnitt 3.2) und auf die (in Frage stehende)endliche Repräsentation. Der unter diesen Möglichkeiten am sinnvollsten erscheinende Revisionsoperator ist das Adjustment innerhalb einer FPER (siehe Abschnitt 3.3). Finite Partial Entrenchment Rankings benutzen Rangordnungen von epistemischer Verankerung, um im Falle eines Konflikts während des Adjustments denjenigen Belief auszuwählen, der aufgegeben werden soll. Dabei soll Belief, der eine starke epistemische Verankerung hat, behalten werden, während der schwächer Verankerte aufgegeben wird. Damit ist es von Bedeutung beim Aufbau der Wissensbasis die ausgewählten expliziten Einträge auch in Hinblick auf ihre epistemische Verankerung beurteilt werden.

Neben der Wahl einer Repräsentationsform der Wissensbasis gibt es einige grundlegende inhaltliche Fragen zum Aufbau der revisionsbereiten Wissensbasis (nachfolgend kurz: Revisionsbasis) des Agenten. Diese Fragen sind:

- Welches Wissen des Agenten kann revidiert werden?
- Welches Wissen aus den verschiedenen Wissensquellen des Agenten soll in die Revisionsbasis aufgenommen werden?
- In welchem Kontext und zu welchem Zeitpunkt sollen die Informationen aus den verschiedenen Wissensquellen in die Revisionsbasis geschrieben werden und wann soll eine Revision möglich ein? Die Beantwortung dieser Frage wird unter anderem darüber entscheiden, ob iterierte Revision möglich sein wird oder nicht.
- Was sind mögliche Auslöser der Revision? Die Beantwortung dieser Frage erfordert es unter anderem festzulegen, um welche Information revidiert werden soll und ob es möglich ist, dass eine einmal verworfene Information erneut als möglich angesehen werden kann.

Eine basale Frage ist, welches Wissen zur Revision anstehen soll. Prinzipiell können die Taxonomien, die Perzeption, der aus der natürlichsprachigen Instruktion erstellte Aktionsplan und der ebenfalls aus der Instruktion erstellte CRIL-Graph sowie die Koreferenzen, die der Matching-Algorithmus geliefert hat, fehlerbehaftet sein. Eine Revision wird immer durch einen Widerspruch ausgelöst. Dieser Widerspruch stammt im Falle des Agenten daher, dass Perzeption, Instruktion und das innere Räsonieren des Agenten nicht miteinander übereinstimmen. Das innere Räsonieren ist im Fall des Agenten der Matching-Algorithmus. Prinzipiell läßt sich die Revision in jedem der drei Teilgebiete Perzeption, Instruktion und dem inneren Räsonieren ansetzen. Werden die Ergebnisse eines der drei Teilbereiche revidiert, so findet hier implizit eine Bewertung der Verlässlichkeit dieses Teilbereichs statt. Die Ergebnisse des Teilbereichs, in dem revidiert wird, wird im Vergleich zu den anderen Teilbereichen als weniger zuverlässig eingestuft. Nachfolgend werde ich die Zuverlässigkeit der drei Teilbereiche untersuchen, um zu ermitteln, welcher die niedrigste Zuverlässigkeit aufweist und daher die erste Wahl für eine Revision ist.

Die Fehlerhaftigkeit der Perzeption kann durch verrauschte oder unzureichende Sensoren erklärt werden. Wie in Kapitel 2 dargelegt wird, ist die Perzeption des Agenten eine Simulation, deren Ergebnis konfigurierbar ist. Dies läßt prinzipiell eine Revision an dieser Stelle zu. Allerdings liefert momentan die Perzeption keine Alternativen. Das Fehlen der Alternativen bedeutet, dass die Entscheidungen des Perzeptionsmoduls als zuverlässig anzusehen sind. Da die Perzeption simuliert wird, können innerhalb der Perzeption keine auf technischen Problemen basierte Mehrdeutigkeiten auftreten. Ich nehme deshalb an, dass eine technische Implementation der Perzeption mit konkreten Problemen ein besserer Ausgangspunkt für eine Revision in diesem Bereich ist. Auch stellt sich die grundsätzlich die Frage, warum ein zumindest dem Anspruch nach universelles Verfahren der Wissensrevision sich mit konkreten Problemen befassen sollte, die ein Mechanismus auf einer anderen Stufe der Verarbeitung hat. Aufgrund der oben genannten Punkte, insbesondere auf das Fehlen von Mehrdeutigkeiten, Alternativen und Unschärfe innerhalb der Perzeption und da eine intensive Beschäftigung mit den Problemen der Bildverarbeitung den Rahmen dieser Arbeit sprengen würde, werde ich die Perzeption als den Zuverlässigsten der drei Teilbereiche ansehen. Sie steht damit auch nicht zur Revision.

Die Fehlerhaftigkeit des Instruktionsplanes und des CRIL-Graphens kann auf einer falschen oder unzureichenden Wegbeschreibung oder einem Fehler innerhalb des Sprachverarbeitungsmoduls der

Instruktionsphase [1], insbesondere während des Parsings, beruhen. Das Parsing liefert dabei alternative Interpretationsmöglichkeiten der Instruktion. Diese Alternativen sind innerhalb der Instruktionsphase aufgrund der durch sie implizierten Unschärfe des Parsings der aussichtsreichste Ansatzpunkt für eine Revision. Allerdings gilt für Fehler innerhalb des Sprachverarbeitungsmoduls das gleiche, wie für die Probleme der Perzeption, die ich oben ausgeführt habe: diese Probleme können als Interna des Sprachverarbeitungsmoduls aufgefasst werden, weshalb die Behandlung dieser Probleme nicht in das Wissensrevisionsmodul gehört und eine konkrete Analyse den Rahmen dieser Arbeit sprengen würde. Darüber hinaus ist der Agent in ein zweiphasiges Modell gegliedert, in dem zunächst die Instruktions verarbeitet und in ein internes Modell überführt wird, während anschließen der Agent auf Basis dieses Modells navigiert. Eine Revision der Instruktion wäre entweder mit der Erinnerung der möglichen Alternativen des Parsings oder mit einem Transfer der Funktionalität der Instruktionsphase in die Navigationsphase verbunden. Da der Agent aber auf Basis von Vorabinstruktionen navigieren soll, kommt ein Transfer von Funktionalität der Instruktionsphase nicht in Frage.

Neben der Fehlerhaftigkeit des Parsings können unzureichende und fehlerhafte Routeninstruktionen eine weitere Fehlerquelle sein. Fehlerhafte oder unzureichende Routeninstruktionen sind ein Thema, das ein umfrangreiches Wissen über die Semantik von Routeninstruktionen, Problemen der umgangsprachlichen Verbalisierung und den natürlichen kognitiven Landkarten des Menschen erfordet. Eine Beschäftigung mit diesem Thema würde ebenfalls den Rahmen dieser Arbeit sprengen. Aus diesem Grund werde ich die Instruktion nicht zur Revision stellen.

Abschließend stehen nur die Ergebnisse des Matching-Algorithmus als gute Möglichkeit zur Revison zur Disposition. Ihre Eignung zur Revision besteht jedoch nicht nur darin, dass Perzeption und Instruktion zur Revision ungeeignet sind. Wird die Revision in diesem Bereich angesetzt, so bedeutet dies, dass der Agent sich der Unsicherheit seiner Entscheidungen bewusst ist. Der Matchingalgorithmus[2] erstellt im Normalfall eine größere Anzahl über dem Schwellenwert liegender Matches. Dass unter Umständen eine große Anzahl von über dem Schwellenwert liegenden Matches auftreten kann, ist unter anderem in Beispiel 2 aus Seite 83 zu erkennen, in dem alle über dem Schwellenwert liegenden Matches innerhalb eines Navigationsdurchlaufs aufgeführt sind. Einige Matches sind unter Umständen sogar exakt gleich bewertet. Dies bedeutet, dass der Agent im Falle von geringen Unterschieden der Bewertung wenig und bei gleicher Bewertung gar keine Anhaltspunkte besitzt, welches Match das zur Instruktion passende ist. Der Agent trifft die Auswahl des Matches willkürlich und unter großer Unsicherheit. Insbesondere im Fall von mehreren gleich bewerteten Matches muss der Agent die Entscheidung ohne jegliche Grundlage treffen. Daher ist es sinnvoll, die Revision hier anzusetzen.

Die Berechnung des Ähnlichkeitsmaßes durch den Matchingalgorithmus bezieht unter anderem auch die taxonomische Ähnlichkeit der durch CRIL-Knoten repräsentierten Konzepten mit ein. Damit sind die berechneten Koreferenzen auch von dem Aufbau der Taxonomien abhängig, was den Schluss erlaubt, dass auch die Taxonomien als mögliche Kandidaten zur Revision in Betracht kommen. Allerdings sind zum einen Taxonomien von einem (menschlichen) Experten erstelle globale Wissenseinheiten, denen damit eine hohe Zuverlässigkeit zugesprochen werden kann. Zum anderen würden Änderungen an den Taxonomien es verlangen, dass der komplette bisherige Navigationsfortschitt aufgegeben werden müsste, da die Änderungen an den Taxonomien auch in den bisher gezogenen Koreferenzen in älteren Situationen die Ähnlichkeitsmaße verändert. Die Änderung des Ähnlichkeitsmaßes in den bisher gezogenen Koreferenzen stellt den kompletten Navigationsfortschritt zur Disposition und verlangt einen kompletten Neubeginn der Navigation zu Anfang der Navigationsphase, was aber aufgrund des damit verbundenen Aufwands erst dann gerechtfertigt erscheint, wenn mögliche andere Fehlerquellen ausgeschlossen worden sind. Aus diesem Grund soll die Taxonomie auch im Rahmen der Revision von Koreferenzen nicht untersucht werden.

Es stellt sich grundsätzlich die Frage, ob der Agent ein Match (auf welchen Grundlagen auch immer) verändern kann, beispielsweise indem er einzelne Koreferenzen austauscht, oder sich auf die Auswahl alternativer Matches beschränken soll. Der Wissensrevisionsmechanismus müsste im ersten Fall die

[1]Siehe 2.1
[2]Siehe 2.4

Begründungen für die Berechnung genau des vorliegenden Matches durch den Matchingalgorithmus kennen und nachvollziehen können. Dies würde erfordern, die Funktionalität des Matchingalgorithmus dem Wissensrevisionsverfahren zur Verfügung zu stellen und sie somit zu duplizieren, was keine sinnvolle Lösung ist. Eine Wahl eines alternativen Matches entspricht allerdings unter den Umständen, die innerhalb des Agenten vorliegen, indirekt einer Manipulation einzelner Koreferenzen. Die einzelnen Matches besitzen, besonders im Falle von gleicher Bewertung durch den Matchingalgorithmus, eine hohe Ähnlichkeit. Manchmal ist im Vergleich zweier Matches nur eine einzelne Koreferenz unterschiedlich. In Situation 6 von Beispiel 3 auf Seite 84 ist es leicht zu erkennen, dass sich nur die Koreferenz eines Instruktionsknoten ändert. Wird ein anderes Match gewählt, ändert sich entsprechend nur eine Koreferenz. Voraussetzung dafür ist, dass die Revisionsbasis so aufgebaut ist, dass strukturähnliche Matches auch nacheinander ausgewählt werden. Auf diese Weise wird das Wissen des Matching-Algorithmus durch das Revisionsmodell weiterverwendet, ohne dass es selbst dort eingebaut werden muss. Auf die Möglichkeit der Strukturähnlichkeit der berechneten Matches ist bereits in 2.4 hingewiesen worden.

Für die Wissensrevision relevante Informationsquellen

Es gibt im weshentlichen folgende Informationsquellen, die zur Einbindung in das Wissensrevisionsmodul des geometrischen Agenten zur Auswahl stehen :

1. Taxonomien.
2. Die CRIL-Graphen aus Instruktion und Perzeption.
3. Der Aktionsplan.
4. Die Koreferenzen, die der Matching-Algorithmus als Ergebnis an jedem Entscheidungspunkt für den zur Aktion gehörenden Teilgraphen des Instruktionsgraphen berechnet.

Der Matching-Algorithmus verwendet dabei die Informationen aus den Taxonomien, des Instruktionsgraphen und des Perzeptionsgraphen[3]. Die Taxonomien werden durch den Matchingalgorithmus zur Berechnung des Ähnlichkeitsmaßes von I- und P-Graphen und in Form der in Koreferenz gesetzten Knoten verwendet, womit Informationen aus den Taxonomien indirekt in den Ähnlichkeitsmaßen weiterverwendet werden. Da sich sowohl die CRIL-Graphen als auch die Taxonomien in den Koreferenzen wiederfinden, werde ich nur die Koreferenzen explizit in die Revisionsbasis aufnehmen. Der CRIL-Graph besteht aus Relationen und Knoten, wobei die Koreferenzen ein bestimmter Typ von Relation ist. Werden nur die Koreferenzen explizit in die Revisionsbasis aufgenommen, so wird diese Relation sowie die Knoten, die koreferenziert werden, in die Revisionsbasis aufgenommen, aber nicht die übrigen Relationen und die Knoten, die nicht koreferenziert werden. Die nicht in der Revisionsbasis aufgenommenen Relationen enthalten im Wesentlichen geometrische und topologische Informationen wie in [H 2003] Abschnitt 5.3.2. dargelegt wird. Die Relationen der CRIL-Graphen sind für das Wissensrevisionsmodul allerdings ohne Bedeutung, da das Revisionsmodul nicht die Plausibilität der generierten Matches auf ihre innere Konsistenz prüfen soll. Eine Plausibilitätsprüfung der berechneten Matches ist eine Verlagerung von Funktionalität des Matching-Algorithmus in die Revsionsbasis. Eine solche Verlagerung ist aber wie oben erwähnt nicht sinnvoll.

Da der Matching-Algorithmus eine Anzahl von alternativen Matches generiert, reicht es aus, wenn die Wissensrevision die Wahl des Matches revidiert und ein alternatives Match auswählt. Dabei sehe ich eine „sinnvolle“ Auswahl des alternativen Matches als das Leistungsmaß für den hier von mir vorgestellten Revisionsmechanismus an. Erinnert der Agent die alternativen Matches, so ist ein erneutes Räsonieren mit Hilfe des Matchingalgorithmus’ überflüssig. Da der Agent die alternativen Matches bisher verwirft[4], muss das Wissensrevisionsmodul diese explizit in die Revisionsbasis eintragen, um ein späteres „Erinnern“ zu ermöglichen.

[3] Siehe Abschnitt 2.4
[4] Siehe Abschnitt 2.3

Die Position des geometrischen Agenten selbst oder die direkt davon abhängige Perzeption wird nach obigem Vorschlag nicht explizit in die Revisionsbasis aufgenommen. Es ist die Frage zu klären, ob es sinnvoll ist, dass ein Update nach einer Positionsänderung und der damit verbundenen Änderung der Perzeption notwendig ist. Aufgrund der statischen Welt des GA kann ohne Bewegung des Agenten sich seine Perzeption nicht ändern. Deshalb besteht keine Notwendigkeit für ein Update ohne Positionsänderung des Agenten.

Es läßt sich aus konstruktivistischer Sicht argumentieren, dass der Agent generell kein Update ausführen kann, da er immer nur seine Interpretation der Welt besitzt. Nachfolgend soll im wesentlichen nach konstruktivistischer Sicht argumentiert werden, warum die einzelnen Beliefs des Agenten nicht für ein Update in Frage kommen.

Die Perzeption findet sich nur indirekt in den Matches wieder. Die Perzeption des Agenten repräsentiert zwar Teile der Welt, allerdings ist die Perzeption bereits mit Annahmen des Agenten über die Welt verknüpft. Als Beispiel seien hier die Taxonomien genannt. Der Agent kann einen Kirchturm nicht als solchen erkennen, sondern wäre gezwungen den Kirchturm als mehrgeschossiges Gebäude zu klassifizieren. Ebenfalls kann ein perzipiertes Objekt falsch in die Taxonomien eingeordnet worden sein, z.b. könnte ein Gebäude mit einem Parterre durch den Perzeptionsmechanismus als ein normales Gebäude mit einem zusätzlichen Stockwerkt klassifiziert werden.

Eine weitere Annahme über die Welt ist innerhalb der Matches die angenommene Koreferenz zwischen dem Instruktions- und Perzeptionsknoten. Darüber hinaus findet sich nur eine Teilmenge der Perzeption innerhalb des Matches wieder, nämlich die, die mit einem Instruktionsknoten koreferenziert werden konnte. Der Agent hat daher bereits eine weitere Annahme getroffen, nämlich dass ein bestimmter Teil der Perzeption wichtiger ist als der Rest. Ein Match enthält damit nicht nur Informationen über die Welt in Form der Perzeptionsanteile sondern auch eine Vielzahl von Annahmen über die Welt. Eine Veränderung des Belief-States läßt sich daher am besten durch die Anwendung des Revisionsoperators ausdrücken, da dieser am besten dafür geignet ist (siehe Abschnitt „Update und Model Change“ in Kapitel 3.2). Da keine Weltinformation, die für ein Update in Frage kommt, explizit in der Revisionsbasis aufgenommen ist, ist die Verwendung eines Update-Operators nicht notwendig.

Als Abschluss dieses Kapitels bleibt die Untersuchung möglicher Revisionsgründe über. Das Revisionsmodul kann selbst nicht entscheiden, wann eine Revision erforderlich ist. Dies muss von „außen“ geschehen. Denkbar ist ein komplexes Analysemodul, das den aktuellen Zustand des Agenten untersucht und prüft, ob eine Revision angestoßen werden sollte. Ich möchte mich allerdings auf ein einfaches Kriterium zum Anstoßen der Revision beschränken: dem Fehlen von Matches oberhalb des benutzen Schwellenwerts. Das Eintreten dieses Falles bedeutet, dass die Beschreibung der Instruktion nicht mehr zuverlässig mit der Perzeption in Verbindung gebracht werden kann. Das Fehlen von Matches bedeutet in diesem Fall, dass zumindest das zuletzt gewählte Match nicht zutrifft. Falls alternative Matches vorliegen, so ist zu prüfen, ob diese zum Ziel führen. Im Abschnitt 4.2 werde ich darlegen, wie aus der Information, dass ein Match nicht zutrifft, auch geschlossen werden kann, dass bestimmte alternative Matches nicht zutreffen. Unter Umständen liegt die Fehlentscheidung auch bereits länger zurück. In diesem Fall muss der Revisionsmechanismus eine Art „Backtracking“ durchführen, und zu einem älteren Entscheidungspunkt zurückkehren.

Im Falle einer Tiefensuche wird ein einmal untersuchter Pfad nicht erneut aufgegriffen. Dieses soll auch im Fall der Wissensrevision so sein. Im Allgemeinen bedeutet die Revision einer Revision, dass die Gründe für die erste Revision in Zweifel gezogen werden. Im unserem Fall ist dies das Fehlen von Matches oberhalb eines bestimmten Schwellenwerts. Eine erneute Revision ist nicht sinnvoll, da das Zurückkehren zu einem Ort, wo es keine sinnvollen Matches gab, nicht dazu führt, dass sich das Fehlen von Matches oberhalb des Schwellenwerts ändert. Ein Argument für eine iterierte Revision an dieser Stelle ist, dass der Agent an dieser Stelle neue Folgeaktionen[5] durchführen kann oder mit einem verändertem Schwellenwert für Matches arbeitet. Allerdings erscheint mir diese Rückkehr zwecks Durchführung von Folgeaktionen kein sinnvoller Weg zu sein. Wenn kein Match gefunden wird, so kann in diesem Fall sofort eine Folgeaktion beginnen, ohne dass der Agent dieses Match erst verwirft und

[5] Siehe 2.3

einen anderen Weg nimmt. Aus diesem Grund treffe ich die Entscheidung, eine wiederholte Revision eines einzelnen Matches nicht durch die Modellierung zu unterstützen.

Einbindung des Wissensrevisionsmoduls in den bestehenden Agenten

In [RN 2003] wird als Ergebnis der Anwendung eines Ask an eine Wissensbasis als Antwort eine Aktion erstellt. Dies kann allerdings für die Revisionsbasis nicht umgesetzt werden, da das Aktionswissen in prozeduraler Form vorliegt und daher nicht deklarativ durch die Wissensbasis ermittelt werden kann, wenn nicht das komplette Aktionsmodul in die das Revisionsmodul verlagert werden soll. Die Auswahl einer Aktion muss also durch ein der Anfrage an die Revisionsbasis nachgeschalteten prozeduralen Anteil des Revisiosmoduls oder innerhalb des das Revisionsmodul aufrufenden Programmabschnitts erfolgen. Da entschieden wurde, dass der Auslöser der Wissensrevision das Fehlen von Matches ist, muss das Revisionsmodul nach der Auflösung der Koreferenzen und vor der Selbstlokalisierung eingebunden werden (siehe dazu Kapitel 2.3 auf Seite 11), da zunächst geprüft werden muss, ob Matches berechnet werden können. Falls keine Matches berechnet werden können, so kann die Selbstlokalisierung nicht ausgeführt werden, weshalb das Revisionsmodul vor Beginn der Selbstlokalisierung eingreifen muss, um einen Abbruch der Navigation zu verhindern, so dies denn möglich ist.

Die Einbindung des Wissensrevisionsmoduls in den bestehenden Agenten erfordert die Spezifikation einer Schnittstelle zwischen dem Agenten und dem Revisionsmodul. Die Schnittstelle des Wissensrevisionsmoduls muss erlauben, dass neben dem gewähltem Match auch alle durch den Matching-Algorithmus berechneten Alternativen in die Revisionsbasis aufgenommen werden. Der Agent muss ermitteln können, welches Match durch das Revisionsmodul bevorzugt wird, und in der Lage sein, die Revision anzustoßen.
Mein Vorschlag für die Einbindung des Revisionsmoduls in den bestehenden Agenten hat folgende Kernpunkte :

- Nachdem der Matching-Algorithmus die Menge der Matches, die oberhalb des eingestellten Schwellenwerts liegt, berechnet hat, werden diese an das Revisionsmodul zwecks Eintragung in die Revisionsbasis weitergeleitet. Dies ist unter anderem erforderlich, weil der Agent bisher die nicht gewählten Matches verwirft. Da bestimmte Funktionen wie γ auf $\mathbb{S}$ definiert sind, muss der Revisionsbasis die gesamte Situation übergeben werden, da ansonsten die Revisionsbasis keinen Zugriff auf diese Informationen hat.

  Aus agententheoretischer Sichtweise von [RN 2003] wird somit eine „Tell“Anweisung ausgeführt, die der Revisionsbasis die Informationen über eine Situation übergibt.

- Nachdem die Matches in die Wissensbasis aufgenommen wurden, fragt der Agent das durch das Wissensrevisionsmodul bevorzugte Match ab. Dies ermöglicht es, dem Wissensrevisionsmodul eine eigene Reihenfolge auf Basis der epistemischen Verankerung zu entwickeln. Aus agententheoretischer Sicht von [RN 2003] wird eine „Ask “-Anweisung ausgeführt, die von der Revisionsbasis die Information, ob ein bestimmtes Match zutrifft, erfragt.

- Liegt ein Revisionsgrund vor, so schließt das Wissensrevisionsmodul eine Teilmenge der bekannten Matches aus und es wird bei der nächsten Anfrage durch die Aktionsplanung das nächst beste gültige Match oder eine leere Menge, im Falle, dass von einem Entscheidungspunkt aus gesehen, keine weiteren Matches oberhalb des Schwellenwerts zur Verfügung stehen, an den Agenten geliefert. Der Agent muss im zweiten Fall dann zu einem älteren Entscheidungspunkt zurückkehren. Wie das Zurückkehren geschen soll, ist Teil der Handlungsrevision, die ich aber explizit von der Wissensrevision trennen und damit auch nicht behandeln möchte.

4.2 Verhaltensäquivalenz

Die Analyse der Aktionen[6] zeigt, dass die vier in CRIL unterschiedenen Aktionenstypen GO, CH_ORIENT, VIEW und BE_AT jeweils ein Zielobjekt besitzen. Der Matching-Algorithmus koreferenziert die Repräsentationen dieser Ziele mit einem wahrgenommenen Objekt aus der Umgebung des Agenten. Stimmen im Falle von zwei alternativen Matches diese Ziele überein, so führt die Aktion auch zum gleichen Ergebnis. Dieses Phänomen, dass bestimmte alternative Matches das gleich Ergebnis haben, werde ich *einfache Verhaltensäquivalenz* von Matches nennen. Diese Definition, dass die Aktion im Falle von Verhaltensäquivalenz zum gleichen Ergebnis führt, möchte ich am Beispiel der GO Anweisung nachweisen. Angenommen, der Agent hat die Anweisung zu einer Kreuzung zu gehen und es liegen zwei unterschiedliche Matches vor, die dazu führen, dass der Agent sich dieser Kreuzung aus zwei unterschiedlichen Richtungen nähert. Der Agent befindet sich in diesem Falle nach der Ausführung des GO-Befehls unter Verwendung des ersten Matches auf der gleichen Position, wie nach der Ausführung des GO-Befehls unter Verwendung des zweiten Matches. Allerdings wird der Agenten einen unterschiedlichen Blickwinkel einnehmen und unterschiedliche Perzepte wahrnehmen.

Aus diesem Grund ist es sinnvoll, den Begriff der *strengen Verhaltensäquivalenz* einzuführen. Im Gegensatz zur einfachen Verhaltensäquivalenz sollen zwei Matches als streng verhaltensäquivalent gelten, wenn diese verhaltensäquivalent sind und der Agent nach Ausführung des Befehls auch die den gleichen Blickwinkel aufweist.

Definition 22 (Einfache Verhaltensäquivalenz) *Sei* $s = (\lambda, A, P, I, I^{'}, K)$ *eine Situation,* m_1, $m_2 \in$ *matches(s)*, $\tau(m_1) = (c_i, c_{p1}, konf_1), \tau(m_2) = (c_i, c_{p2}, konf_2)$ *mit* $c_i \in \mathbb{I}$, $c_{p1}, c_{p2} \in \mathbb{P}$, $konf_1, konf_2 \in \mathbb{R}$. *Dann ist* $ev : \mathcal{M} \times \mathcal{M} \to \{0, 1\}$,

$$ev(m_1, m_2) := \begin{cases} 1 & \textit{wenn } c_{p1} = c_{p2} \\ 0 & \textit{wenn } c_{p1} \neq c_{p2} \end{cases}$$

Zur Definition der strengen Verhaltensäquivalenz möchte ich zunächst eine Hilfsfunktion einführen. Diese Hilfsfunktion soll den Blickwinkel nach Ausführung einer atomaren Anweisung mit einem bestimmten Match spezifizieren.

Definition 23 (Blickwinkel nach Ausführung einer atomaren Anweisung) *Sei* $s = (\lambda, A, P, I, I^{'}, K)$ *eine Situation,* $m \in$ *matches(s). Dann sei* $\beta(s, m)$ *derjenige Blickwinkel, den der Agent nach der Ausführung der Instruktion in s mit dem Match m einnimmt.*

Definition 24 (Strenge Verhaltensäquivalenz) *Sei* $s = (\lambda, A, P, I, I^{'}, K)$ *eine Situation,* $m_1, m_2 \in$ *matches(s),* $sv : \mathcal{M} \times \mathcal{M} \to \{0, 1\}$,

$$sv(m_1, m_2) := \begin{cases} 1 & \textit{wenn } ev(m_1, m_2) = 1 \wedge \beta(s, m_1) = \beta(s, m_2) \\ 0 & \textit{andernfalls} \end{cases}$$

An dieser Stelle möchte ich zeigen, dass sowohl strenge als auch einfache Verhaltensäquivalenz die drei Eigenschaften Reflexivität, Transitivität und Symmetrie einer Äquivalenzrelation aufweist. Im Falle von Reflexivität gilt: Wenn ein Match m_1 identisch mit einem Match m_2 ist, dann ist auch die Zielkoreferenz $\tau(m_1) = \tau(m_2)$. Wenn $\tau(m_1) = \tau(m_2)$ ist, dann sind auch die Tripel $(c_{i1}, c_{p1}, konf_1) = (c_{i2}, c_{p2}, konf_2)$, woraus folgt, dass auch $c_{i1} = c_{i2}$ und $c_{p1} = c_{p2}$ ist. Somit ist die einfache Verhaltensäquivalenz reflexiv. Aufgrund der Definition sind zwei Matches streng verhaltensäquivalent, wenn sie einfach verhaltensäquivalent sind und zudem $\beta(s, m_1) = \beta(s, m_2)$ ist. Unter der Voraussetzung $m_1 = m_2$ und der Eindeutigkeit einer Funktion gilt, dass $\beta(s, m_1) = \beta(s, m_2)$ ist, und somit $\beta(s, m_1) = \beta(s, m_2)$. Da die Reflexivität von einfacher Verhaltensäquivalenz bereits bewiesen ist, ist damit auch strenge Verhaltensäquivalenz reflexiv.

[6]Siehe Abschnitt 2.3

Im Falle von Transitivität gilt: Wenn $ev(m_1, m_2) = 1$ und $ev(m_2, m_3) = 1$, dann ist auch $\tau(m_1) = \tau(m_2) \wedge \tau(m_2) = \tau(m_3)$. Wenn die drei Koreferenzen des Zielknotens identisch ($\tau(m_1) = \tau(m_2) \wedge \tau(m_2) = \tau(m_3)$) sind, so gilt auch, dass $c_{p1} = c_{p2} \wedge c_{p2} = c_{p3}$ und $c_{i1} = c_{i2} \wedge c_{i2} = c_{i3}$. Daraus folgt, dass auch $c_{p1} = c_{p3}$ und $c_{i1} = c_{i3}$ ist. Nach Definition von einfacher Verhaltensäquivalenz ist somit $ev(m_1, m_3) = 1$. Für strenge Verhaltensäquivalenz gilt nach Definition zusätzlich zur einfachen Verhaltensäquivalenz, dass $\beta(s, m_1) = \beta(s, m_2) \wedge \beta(s, m_2) = \beta(s, m_3)$. Damit gilt auch, dass $\beta(s, m_1) = \beta(s, m_3)$ sind. Da die Transitivität der Verhaltensäquivalenz bereits bewiesen ist, folgt mit $\beta(s, m_1) = \beta(s, m_3)$, dass $sv(m_1, m_2) = 1$ ist.

Für Symmetrie gilt: Sind m_1 und m_2 einfach verhaltensäquivalent, so ist auch $\tau(m_1) = \tau(m_2)$ und aufgrund der Symmetrie der Identitätrelation $\tau(m_2) = \tau(m_1)$. Aufgrund der Definition von einfacher Verhaltensäquivalenz ist damit $ev(m_2, m_1) = 1$. Für strenge Verhaltensäquivalenz ist zusätzlich zu zeigen, dass $\beta(s, m_2) = \beta(s, m_1)$ ist. Im Falle von strenger Verhaltensäquivalenz ist die Voraussetzung $\beta(s, m_1) = \beta(s, m_2)$ gegeben, woraus mit der Symmetrie der Identitätsrelation $\beta(s, m_2) = \beta(s, m_1)$ und damit $sv(m_2, m_1) = 1$ folgt. Damit ist sowohl die strenge als auch die einfache Verhaltensäquivalenz symmetrisch.

Aufgrund der Reflexivität, Transitivität und Symmetrie sowohl der strengen als auch der einfachen Verhaltensäquivalenz folgt, dass sowohl strenge als auch einfache Verhaltensäquivalenz Äquivalenzrelationen sind.

Zur Berechnung der einfachen Verhaltensäquivalenz reicht es aus, zu vergleichen, ob der Zielknoten der Instruktion bei beiden Matches mit dem gleichen Knoten der Perzeption koreferenziert wurde. Die Berechnung der strengen Verhaltensäquivalenz ist komplexer, da sie erfordert, dass der Agent in der Lage ist, vorherzusagen, aus welchem Blickwinkel er sich seinem Ziel nähert. Der Blickwinkel zweier alternativen Wege ist bei Erreichen des gleichen Zielpunkts dann gleich, wenn das Ziel aus der gleichen Richtung erreicht wird. So wären zwei Wege, deren letzter Wegabschnitt identisch ist streng verhaltensäquivalenz. Eine Identität der letzten Wegstrecken ist aber nicht zwingend notwendig, da der Zielort auch dann aus der gleichen Richtung erreicht wird, wenn eine der letzten Wegstrecken innerhalb der anderen liegt. Wie die Berechnung von β genau aussieht, soll kein Gegenstand dieser Arbeit sein. Aus diesem Grund ist β auch nicht formal spezifiziert. Meine einzige Anforderung an β ist, dass β eine Situation und ein Match als Parameter übernehmen und daraus den Blickwinkel des Agenten berechnen können muss. Dieser Blickwinkel muss darüber hinaus mit anderen Blickwinkeln auf Identität prüfbar sein.

Da die Berechnung der einfachen Verhaltensäquivalenz zusätzlichen Aufwand im Vergleich zur Berechnung der strengen Verhaltensäquivalenz benötigt oder im allgemeinen Fall ein beliebiger Agent nicht in der Lage sein könnte, strenge Verhaltensäquivalenz zu erkennen, ist es sinnvoll zu prüfen, inwieweit die Verwendung der strengen Verhaltensäquivalenz auch notwendig ist. Wie aus der Definition ersichtlich ist, ist die Menge der streng verhaltensäquivalenten Matches eine Teilmenge der einfach verhaltensäquivalenten Matches und somit jedes streng verhaltensäquivalente Match auch gleichzeitig einfach verhaltensäquivalent. Nachfolgend will ich die Frage untersuchen, ob jedes einfach verhaltensäquivalente Match auch streng verhaltensäquivalent ist.

Der Agent kann durch geschickte Wahl der Entscheidungspunkte verhindern, dass strenge Verhaltensäquivalenz auftreten kann. Das wesentliche Kriterium bei der Wahl der Entscheidungspunkte ist dabei, diese nicht exakt auf Gabelungen oder Kreuzungen zu legen, sondern kurz vor diese. Auf diese Weise wird sichergestellt, dass, falls eine Kreuzung aus verschiedenen Richtungen erreicht werden kann, mit jeder Richtung auch ein separater Zielpunkt verbunden ist.

Eine weitere Möglichkeit, die Verwendung von strenger Verhaltensäquivalenz zu umgehen, kann die Verwendung von bestimmten Folgeaktionen sein. Wenn der Agent nach der Bewegung zu seinem Ziel kein Match findet, kann er sich zunächst um sich selbst drehen und so weitere Informationen über seine Umgebung erhalten. Dreht sich der Agent um sich selbst, so kann er auch den Blickwinkel einnehmen, den er hat, wenn er das Ziel aus einer anderen Richtung erreicht. Durch das Drehen kann so das Erreichen des Ziels über alle anderen einfach verhaltensäquivalente Matches simuliert werden. Wenn so alle einfach verhaltensäquivalente Matches durch das Drehen des Agenten abgedeckt werden, so sind auch alle potentiellen streng verhaltensäquivalente Matches abgedeckt, weil streng verhaltensäquivalente Matches eine Teilmenge der einfach verhaltensäquivalenten Matches sind.

Das Ziel der Verwendung von Verhaltensäquivalenz ist es, Aktionen, die die gleiche Wirkung wie bereits ausgeführte Aktionen haben, nicht auszuführen. Wird daher im Revisionsmodul generell die einfache Verhaltensäquivalenz benutzt, so führt dies dazu, dass die Vollständigkeit des Verfahrens nicht mehr gewährleistet ist. Dies liegt daran, dass unter Umständen Aktionen, die zum Erreichen des Ziels führen, nicht ausgeführt werden. Der restliche Ablauf innerhalb des Revisionsmoduls wird aber durch diese Unterscheidung nicht tangiert werden. Aus diesem Grund werde ich im weiteren nicht mehr zwischen einfacher und strenger Verhaltensäquivalenz unterscheiden. Da sich das Auftreten von strenger Verhaltensäquivalenz vermeiden lässt oder die Auswirkungen von strenger Verhaltensäquivalenz im Falle ihres Auftretens durch Folgeaktionen negieren lassen, empfehle ich für den Fall, dass ein Agent keine strenge Verhaltensäquivalenz berechnen kann oder die Berechnung mit zu größem Aufwand verbunden ist, einfache Verhaltensäquivalenz zu verwenden.

Im Abschluss dieses Abschnitts definiere ich noch die Menge der Matches einer Situation, die zu einem bestimmten Match nicht verhaltensäquivalent sind. Auch hierbei ist es nicht von Bedeutung, ob strenge oder einfache Verhaltensäquivalenz verwendet wird.

Definition 25 (Äquivalenzklasse zu einem Match einer Situation) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation, $m \in$ matches(s), dann sei*

$$\mathcal{VE}(m) := \{m_i \in \mathit{matches(s)}\,|\,sv(m_i, m) \equiv ev(m_i, m)\}$$

die Äquivalenzklasse zu einem Match einer Situation.

Seien $m_1 \dots m_n \in$ matches(s). Dann ist $\mathcal{VEM}(\{m_1, \dots, m_n\}) := \bigcup_{i=1}^{n} \mathcal{VE}(m_i)$

Die Klasse aller Äquivalenzklassen einer Situation sei $\mathcal{VE}(s)$

Nachfolgend die Definition einer Menge aller zueinander nicht verhaltensäquivalenten Matches innerhalb einer Situation insgesamt.

Definition 26 (Menge der nicht verhaltensäquivalenten Matches) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation sowie γ eine Selektionsfunktion, die aus der Äquivalenzklasse eines Matches ein Exemplar selektiert. Dann ist*

$$\mathcal{NVE}_{unique}(s) := \left\{ m_i \,\middle|\, \bigvee_{c \in \mathcal{VE}(s)} \; \coprod_{m \in c} m_i = \gamma(\mathcal{VE}(m)) \right\}$$

$|\mathcal{NVE}_{unique}(s)|$ koreliert mit der Zahl der Äquivalenzklassen der Situation s. Zur Abkürzung soll nachfolgend die Zahl der Äquivalenzklassen einer Situation s auch mit EC(s) abgekürzt werden.

Verallgemeinerung der Verhaltensäquivalenz

Nach der Definition von einfacher oder strenger Verhaltensäquivalenz kann diese als ein Artefakt des Matchingalgorithmus des Agenten aufgefasst werden und damit wäre Verhaltensäquivalenz ein sehr spezifisches Problem mit einem sehr geringen Relevanzbereich. Es läßt sich aber argumentieren, dass Verhaltensäquivalenz nach der Definition ein Spezialfall eines allgemeineren Phänomens ist. Abstrakt betrachtet besteht das Problem, dass durch Verhaltensäquivalenz aufgeworfen wird, darin, dass ein „mentaler“ Prozess eines Agenten verschiedene Ergebnisse produziert, die verschiedene Handlungen des Agenten auslösen. Die durch den mentalen Prozess ausgelösten Handlungen führend jedoch zum gleichen Ergebnis. Dies ist schematisch in Figur 4.1 für den Fall einer Äquivalenzklasse dargestellt. Eine Untersuchung von Verhaltensäquivalenz ist auch im abstakten Fall sinnvoll, falls es möglich ist, dass der mentale Prozess möglicherweise nicht zielführende Handlungen auslöst, die revidiert werden müssen. Prinzipiell gilt, dass Verhaltensäquivalent dann auftreten kann, wenn es Ergebnisse

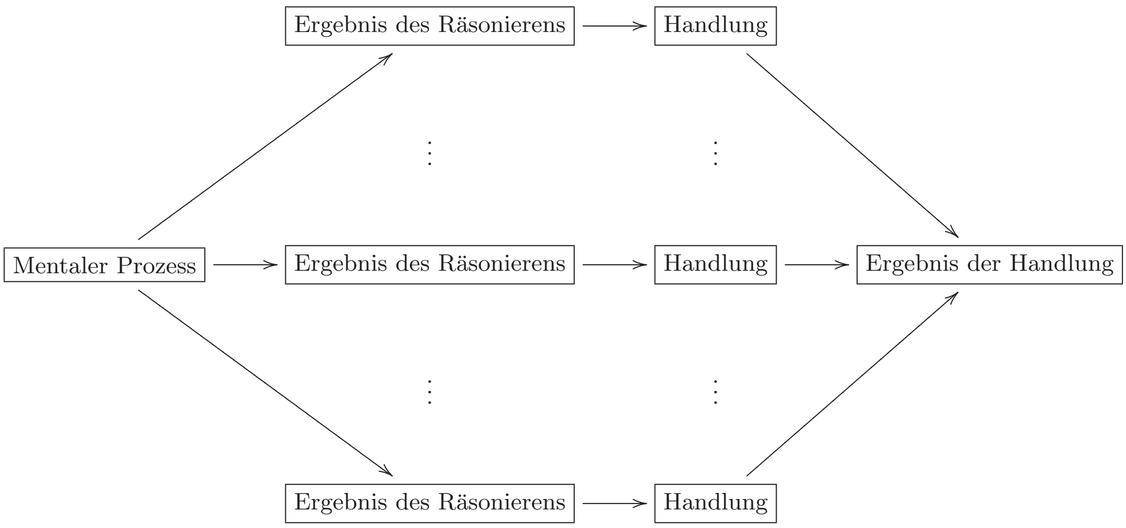


Abbildung 4.1: Verallgemeinerte Verhaltensäquivalenz
Schematische Darstellung des verallgemeinerten Prinzips von Verhaltensäquivalenz für eine Äquivalenzklasse. Ein mentaler Prozess erzeugt eine Vielzahl von Alternativen Handlungsanleitungen, die je eine Handlung auslösen. Die unterschiedlichen Handlungen münden aber in einem einzigem Ergebnis.

gibt, die revidiert werden müssen. Eine Revision von Handlungen oder von Wissen besonders im Fall von Unsicherheit erforderlich, weshalb Verhaltensäquivalenz vor allem im Fall von Räsonieren unter Unsicherheit auftritt. Unsicherheit kann im Fall von Agenten unter anderem durch die eingeschränkte Beobachtbarkeit der Umwelt, wie im Fall des geometrischen Agenten, verursacht werden. Der Agent liefert auch weitere Beispiele für Verhaltensäquivalenz. So existiert auch im Sprachverarbeitungsmodul Unsicherheit, da es beispielsweise Sätze gibt, in denen Satzteile nicht eindeutig bestimmt werden können. Das Sprachverarbeitungsmodul liefert daher eine Reihe von Alternativen unter denen der Agent eine auswählen muss. Wie in Abschnitt 4.1 ausgeführt ist es prinzipiell möglich, die Ergebnisse des Sprachmoduls einer Revision zu unterziehen.

Das Ausschließen von verhaltensäquivalenten Alternativen sorgt dafür, dass der Agent rationaler handelt, da das repetetive Ausführen von Handlungen, die nicht bei der Maximierung des Performanzmaßes helfen, unökonomisch ist.

4.3 Aufbau der Revisionsbasis

4.3.1 Die Übersetzungsfunktion

Die Revisionsbasis ist eine FPER, die in Kapitel 3.3 vorgestellt wird. Eine FPER ist nach Definition eine Funktion von $\mathcal{L}$ in das Intervall $]0,1[$. Wichtig ist, dass für kontingente Sätze die absolute Höhe der epistemischen Verankerung keine, dafür aber die Unterschiede zwischen zwei oder mehr „Ebenen“ der epistemischen Verankerung eine Bedeutung haben. Kontingente Formeln sollen auf das Intervall $[0,1[$ abgebildet werden, Tautologien auf 1 und Kontradiktionen auf 0.

Ich werde die FPER über eine Funktion aufbauen, die einen Teil des Wissens des Agenten auf Tupel von $\mathcal{L}\times]0,1[$ abbildet. Dabei sollen keine konkreten Werte, sondern einzelne Variable benutzt werden, für die bestimmte $>, <$ und $=$ Beziehungen gelten sollen. Diese Art von Bedingungen werden im Abschnitt über die epistemische Verankerung spezifiziert werden. In der Funktion, die Teile des Wissens des Agenten auf Tupel aus $\mathcal{L}\times]0,1[$ abbildet, werden deshalb Variablennamen des Musters $\Lambda_1, \ldots, \Lambda_n$ Verwendung finden.

Die Tupel der Bildmenge der oben genannten Funktionen sollen als Basis für die Definition einer FPER-Funktion dienen. Wie ich unter 4.1 dargelegt habe, werde ich als Hauptwissensquelle die Ergebnisse des Matching-Algorithmus verwenden. Dieser Algorithmus liefert für das aktuell zur Bearbeitung anstehende Teilnetz des Instruktionsgraphen[7] eine Sammlung alternativer Matches von Koreferenzen zur Assoziierung eines Knotens der Instruktion mit einem Knoten der Perzeption[8]. Diese alternative Mengen sind durch ein Ähnlichkeitsmaß bewertet, wobei alternative Matches die gleiche Bewertung haben können.

Die Übersetzungsfunktion soll dabei pro Entscheidungspunkt die Menge der durch den Matching-Algorithmus für diese Situation berechneten Matches in Tupel aus $\mathcal{L}$ und $]0,1[$ übersetzen. Diese Tupel sind per Adjustment der bereits bestehenden FPER hinzuzufügen. Aus Gründen, die ich später erleutern werde, erfordert diese Übersetzungsfunktion neben den Matches einer Situation auch das zuletzt gewählte Match. Ich werde ebenfalls die Begründung für meine Entscheidung, die Revisionsbasis in der nachfolgend definierten Weise aufzubauen, später nachliefern. Nach Definition der Übersetzungsfunktion wird eine alternative Übersetzungsfunktion eingeführt, die in einem Detail von der präferierten Übersetzungsfunktion abweicht.

Definition 27 (Übergeordnete Übersetzungsfunktion) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation und $s_{alt} = (\lambda_{alt}, A_{alt}, P_{alt}, I_{alt}, I'_{alt}, K_{alt})$ die Situation des letzten Entscheidungspunkts vor s, dann ist*

$$\theta : \mathbb{S} \times \mathbb{S} \to 2^{\mathcal{L}\times]0,1[},$$

$$\theta(s, s_{alt}) := \theta_{ex}(s) \cup \theta_m(s) \cup \theta_s(s, s_{alt})$$

die übergeordnete Übersetzungsfunktion.

Die übergeordnete Übersetzungsfunktion besteht aus drei Subfunktionen. θ_{ex} werde ich als „Exklusionsfunktion“, θ_m als „Übersetzungsfunktion für Matches“ und θ_s als „situationsübergreifende Verankerungsfunktion“ bezeichnen. Diese Funktionen werden nachfolgend definiert.

Definition 28 (Abbildung einer Koreferenz auf ein Atom) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation, $L \subset \mathcal{L}_{atom}, |L| = |\mathcal{K}(P, I)|$. Eine Bijektion $\theta_k : \mathcal{K}(P, I) \leftrightarrow L$ ist eine Übersetzungsfunktion für eine einzelne Koreferenz.*

Anmerkung: θ_k kann eine beliebige Bijektion sein.

Definition 29 (Übersetzungsfunktion für ein einzelnes Match) *Sei $s = (\lambda, A, P, I, I', K)$ eine Situation, $m \in$ matches(s), $\{k_1 \ldots k_n\} = m$, $<$ eine beliebige strenge Totalordnung, $k_1 < \cdots < k_n$ und θ_k eine Übersetzungsfunktion für eine einzelne Koreferenz.*

Dann ist $\theta_\mu : \mathcal{M} \to \mathcal{L}$,

$$\theta_\mu(m) := \bigwedge_{i=1}^{n} \theta_k(k_i)$$

die Übersetzungsfunktion für ein einzelnes Match. Die Übersetzungsfunktion für ein einzelnes Match repräsentiert die Menge der Koreferenzen zwischen je einem Knoten der Instruktion und einem Knoten der Perzeption, aus denen ein einzelnes Match zusammengesetzt ist, als eine Konjunktion von atomaren Aussagen. Die Verwendung der strengen Totalordnung $<$ soll sicherstellen, dass $\theta_\mu(m)$ eindeutig ist und hat darüber hinaus keine Bedeutung.

[7] Siehe 2.3
[8] Siehe 2.4

Definition 30 (Übersetzungsfunktion für Matches) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation, $\vee \in \mathcal{L}_{generalisiert}$, $ep_{\mu\sigma}\mathbb{N} \rightarrow \mathbb{Q}$ die Funktion der epistemischen Verankerung für Matches, θ_μ eine Übersetzungsfunktion für ein einzelnes Match, $\{m_1 \dots m_n\}$ = matches(s), $m_1 < m_2 < \dots < m_n$, wobei $<$ eine strenge Totalordnung ist. Dann ist $\theta_m : \mathbb{S} \rightarrow 2^{\mathcal{L} \times]0,1[}$,*

$$\theta_m(s) := \left\{ \left(\bigvee_{k=1}^{i} \theta_\mu(m_k), ep_{\mu\sigma}(i) \right) \middle| \, i \in \{1, \dots, n\} \right\}$$

die Übersetzungsfunktion für Matches. Die Übersetzungsfunktion für Matches erzeugt eine Menge von Disjunktionen, so dass jedem Match aus der strikt geordneten Menge aller Matches eine Disjunktion zugeordnet wird. Dabei erhält das nach der Totalordnung her i. beste Match eine Disjunktion mit i Disjunkten zugeordnet. Die Disjunktion des i. Match besteht aus der Disjunktion der (i-1). Disjunktion und dem i. Match.

Definition 31 (Exklusionsfunktion) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation,θ_k eine Übersetzungsfunktion für eine einzelne Koreferenz und Λ_1 eine Zahl aus $]0, 1[$, die eine epistemische Verankerung repräsentiert. Dann ist $\theta_{ex} : \mathbb{S} \rightarrow 2^{\mathcal{L} \times]0,1[}$,*

$$\theta_{ex}(s) := \{ (\theta_k(\tau(m_i)) \supset \neg\theta_k(\tau(m_k)), \Lambda_1) | \, m_i, m_k \in \mathcal{M}(s), \tau(m_i) \neq \tau(m_k) \}$$

die Exklusionsfunktion. Die Exklusionsfunktion erstellt eine Menge von Implikationen, in denen der wechselseitige Ausschluss der Zielkoreferenzen spezifiziert wird. Diese Definition benutzt implizit einfache Verhaltensäquivalenz.

Definition 32 (Situationsübergreifende Verankerungsfunktion) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation, $s_{alt} = (\lambda_{alt}, A_{alt}, P_{alt}, I_{alt}, I^{'}_{alt}, K_{alt})$ die Situation des letzten Entscheidungspunkts vor s, $\{m_1 \dots m_n\} = \mathcal{NVE}_{unique}(s)$, stop $\in \mathcal{L}_{atom}$ sowie Λ_2 eine Zahl aus $]0, 1[$, die eine epistemische Verankerung repräsentiert. Dann ist $\theta_s : \mathbb{S} \times \mathbb{S} \rightarrow \mathcal{L} \times]0, 1[$,*

$$\theta_s(s, s_{alt}) := \begin{cases} \left(\left(\bigwedge_{i=1}^{n} \neg\theta_k(\tau(m_i)) \right) \supset \neg\theta_k(\tau(\gamma(s_{alt}))), \Lambda_2 \right) & \textit{wenn } s_{alt} \neq s_0 \\ \left(\left(\bigwedge_{i=1}^{n} \neg\theta_k(\tau(m_i)) \right) \supset \textit{stop}, \Lambda_2 \right) & \textit{andernfalls} \end{cases}$$

Anmerkung: $\gamma(s_{alt})$ ist das in der Situation s_{alt} durch den Agenten gewählte Match, siehe Definition Seite 16. Die situationübergreifende Verankerungsfunktion erzeugt eine einzige Implikation, die in der Antezedens aus der Konjunktion der negierten Koreferenzen des Ziels der Aktionen der einzelnen Matches besteht. Die Konsequenz wird entweder aus dem ansonsten unbenutzten Atom stop oder der negierten Koreferenz des Ziels der Aktion der Situation des vorangegangenen Entscheidungspunkts gebildet.

Definition 33 (Übersetzungsfunktion für ein einzelnes Match (alternativ)) *Sei s eine Situation, $L \subset \mathcal{L}_{atom}$, $|L| = |\mu(s)|$. Die Bijektion $\theta^{'}_\mu : \mu(s) \leftrightarrow L$, ist eine alternative Übersetzungsfunktion für ein einzelnes Match, die jedes Match auf ein eindeutig bestimmtes aussagenlogisches Atom abbildet.*

Definition 34 (Übersetzungsfunktion für Matches (alternativ)) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation, $ep_{\mu\sigma}\mathbb{N} \rightarrow \mathbb{Q}$ die Funktion der epistemischen Verankerung für Matches, $\theta^{'}_\mu$ eine Übersetzungsfunktion für ein einzelnes Match, $\{m_1 \dots m_n\}$ = matches(s), $m_1 < m_2 < \dots < m_n$, wobei $<$ eine strenge Totalordnung ist. Dann ist $\theta^{'}_m : \mathbb{S} \rightarrow 2^{\mathcal{L} \times]0,1[}$,*

$$\theta^{'}_m := \left\{ \left(\bigvee_{k=1}^{i} \theta_{match}(m_k), ep_{\mu\sigma}(i) \right) \middle| \, i \in \{1, \dots, n\} \right\}$$

eine alternative Übersetzungsfunktion für Matches.

Definition 35 (Exklusionsfunktion (alternativ)) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation, $\theta^{'}_{\mu}$ eine Übersetzungsfunktion für ein einzelnes Match. Dann ist $\theta^{'}_{ex} : \mathbb{S} \rightarrow 2^{\mathcal{L}\times]0,1[}$,*

$$\theta^{'}_{ex}(s) := \left\{ \left(\theta_{match}(m_i) \supset \neg\theta^{'}_{\mu}(m_k), \Lambda_1\right) \middle| \, m_i, m_k \in \mathcal{M}(s), \neg ev(m_i, m_k) \right\}$$

die alternative Exklusionsfunktion.

Definition 36 (Situationsübergreifende Verankerungsfunktion (alternativ)) *Sei $s = (\lambda,\ A,\ P,\ I,\ I^{'},\ K)$ eine Situation, $s_{alt} = (\lambda_{alt},\ A_{alt},\ P_{alt},\ I_{alt},\ I^{'}_{alt},\ K_{alt})$ die Situation des letzten Entscheidungspunkts vor s, $\{m_1 \ldots m_n\} = \mu\sigma(s)$, $stop \in \mathcal{L}_{atom}$ sowie Λ_2 eine Zahl aus $]0,1[$, die eine epistemische Verankerung repräsentiert. Dann ist $\theta^{'}_{s} : \mathbb{S} \times \mathbb{S} \rightarrow \mathcal{L}\times]0,1[$,*

$$\theta^{'}_{s}(s, s_{alt}) := \begin{cases} \left(\left(\bigwedge\limits_{i=1}^{n} \neg\theta^{'}_{\mu}(m_i)\right) \supset \neg\theta^{'}_{\mu}(\gamma(s_{alt})), \Lambda_2\right) & \textit{wenn } s_{alt} \neq s_0 \\ \left(\left(\bigwedge\limits_{i=1}^{n} \neg\theta^{'}_{\mu}(m_i)\right) \supset stop, \Lambda_2\right) & \textit{andernfalls} \end{cases}$$

.

Definition 37 (Übergeordnete Übersetzungsfunktion (alternativ)) *Sei $s = (\lambda, A, P, I, I^{'}, K)$ eine Situation und $s_{alt} = (\lambda_{alt}, A_{alt}, P_{alt}, I_{alt}, I^{'}_{alt}, K_{alt})$ die Situation des letzten Entscheidungspunkts vor s, dann ist*

$$\theta^{'} : \mathbb{S} \times \mathbb{S} \rightarrow 2^{\mathcal{L}\times]0,1[},$$

$$\theta^{'}(s, s_{alt}) := \theta^{'}_{ex}(s) \cup \theta^{'}_{m}(s) \cup \theta^{'}_{s}(s, s_{alt})$$

eine alternative übergeordnete Übersetzungsfunktion.

Neben der Repräsentation der Wissensinhalte des geometrischen Agenten in der Repräsentationsbasis ist auch die Form des Adjustments, das der Revisionsbasis mitteilt, dass ein Match nicht zutreffen kann, von Bedeutung.

Definition 38 (Adjustment zum Ausschluss eines einzelnen Matches) *Sei $s = (\lambda,\ A,\ P,\ I,\ I^{'},\ K)$ eine Situation,$m \in$matches(s) und Λ_{remove} eine Zahl aus $]0,1[$, die eine epistemische Verankerung repräsentiert. Dann ist die Funktion $\chi : \mathbb{S} \times \mathcal{M} \rightarrow \mathcal{L}\times]0,1[$ mit*

$$\chi(s, m) := (\neg\tau(m), \Lambda_{remove})$$

Nachfolgend wird es notwendig sein, auf den Formelanteil der Tupeln in den von den θ erstellten Mengen zuzugreifen. Deshalb sei an dieser Stelle die Funktion $\psi : 2^{\mathcal{L}\times]0,1[} \rightarrow 2^{\mathcal{L}}$ mit $\psi(\mathcal{F}) := \{x | \text{für alle } (x, y) \in \mathcal{F}\}$ eingeführt.

In den nachfolgenden Unterabschnitten werde ich die Begründung für meine in den obigen Definitionen gewählte Form der Revisionsbasis darlegen. Dabei werde ich verschiedene Alternativen zu der von mir gewählten Form auf ihre Konsequenzen für die Leistungsfähigkeit der Revisionsbasis untersuchen. Da, wie in Kapitel 3.2 erwähnt, eine unterschiedliche Syntax von Belief-Bases mit dem gleichen Belief-Set nach Durchführung einer Kontraktion oder Revision um den gleichen Operanden zwei unterschiedliche Belief-Bases mit unterschiedlichem Belief-Set erzeugen kann, ist es notwendig, zu prüfen, ob eine Revision mit der vorgeschlagenen Revisionsbasis auch die gewünschten Veränderungen an ihrem Belief-Set erzeugt.

Das Wissen des Agenten muss dabei in einer solchen Art in die Revisionsbasis aufgenommen werden, dass der Revisionsmechanismus in der Lage ist, folgende Anforderungen zu erfüllen:

- Es ist hinreichend, dass der Agent in der Lage ist, das Ziel der aktuellen Aktion zu koreferenzieren. Im Prinzip ist es daher möglich, dass das Revisionsmodul eigenständig Matches erstellt, was, wie ich in Abschnitt 4.1 dargelegt habe, nicht sinnvoll ist, da es den Transfer der Funktionalität des Matching-Algorithmus in das Revisionsmodul erfordert. Daher ist es hinreichend, wenn der Revisionsmechanismus auf der Ebene kompletter Matches arbeitet.

  Zu jeder Zeit sollte ein eindeutig bestimmtes „bestes“ Match für die aktuelle Anweisung innerhalb des Aktionsplans verfügbar sein. Dies bedeutet, dass aus den alternativen und unter Umständen gleich bewerteten Matches eine Auswahl getroffen werden muss. Um diese Anforderung zu erfüllen, ist es sinnvoll, die alternativen Matches in eine Totalordnung innerhalb der Wissensbasis zu bringen. Wenn die Information, dass ein Match nicht zutrifft, zu einer Revision führt, sollte die epistemische Verankerung dieses Matches auf Null gesenkt werden. Da die Möglichkeit besteht, dass mehrere Matches ausgeschlossen werden, ist eine strenge Totalordnung der Matches nicht sinnvoll, da diese ausschließen würde, dass mehrere Matches den Rang Null erhalten. Wie in der Definition von θ_m ersichtlich ist, wird dort eine strenge Totalordnung der einzelnen Matches gefordert. Dies hat den Hintergrund, dass eine Funktion eindeutig sein muss und eine Definition von θ_m unter Verwendung einer einfachen Totalordnung unter Zulassung der Reflexivität diese Eindeutigkeit nicht gewährleisten kann. Aus diesem Grund werde ich zwei unterschiedlichen Totalordnungen zum Aufbau der Revisionsbasis verwenden: eine strenge Totalordnung der Matches als Vorbedingung zur Verwendung von θ_m und eine einfache Totalordnung der Matches innerhalb der Revsionsbasis.

- Die Einträge in der Revisionsbasis sollen prägnant sein, um ein schnelles Räsonieren zu ermöglichen, was Williams in [W 1997] mit den Worten „Information should be in its simplest logical form.“(S. 78) ausdrückt. Wie [FITTING 1996] auf Seite 107 schreibt, kann ein Tableaubeweis, ähnlich wie ein Resolutionsbeweis, exponentiell schlecht sein. Die exponentielle Zeit hängt hierbei im Falle von Resolutions-Beweisen von der Komplexität der zu beweisenden Formeln ab ([FITTING 1996], S 105). Dies gilt auch im Fall eines Tableau-Beweises. Der Berechnungsaufwand eines Beweises hängt im Fall eines Tableau-Beweises davon ab, wie oft verzweigend expandiert werden muss. Da die Einträge der Revisionsbasis in vielen Beweisen verwendet werden, ist es sinnvoll die Einträge so zu wählen, dass diese innerhalb eines Beweises möglichst wenig verzweigend expandiert werden müssen. Im Fall eines Tableau-Beweises ist deshalb die Verwendung von $\beta - Formeln$ ungünstig.
  Wie in Abschnitt 3.2 dargelegt wird, erfordert die Abfrage von implizitem Wissen aus der Wissensbasis heraus den Test, ob die Konjunktion aller expliziten Einträge der Wissensbasis die abzufragende Formel impliziert. Im Falle von Wissenbasen mit einer epistemischen Ordnung[9] werden statt einem solchen Beweis unter Umständen mehrere, teils hoch redundante, benötigt. Die Vielzahl an Beweisen, wird benötigt, um die epistemische Verankerung der Aussage, die erfragt werden soll, zu berechnen. Daraus folgt, dass auch die Anzahl der Aussagen, die explizit in die Wissensbasis eingetragen werden, und deren Gruppierung hinsichtlich der epistemischen Verankerung Einfluss auf die Geschwindigkeit des Räsonierens haben wird. Wie in [W 1997] erwähnt, ist es sinnvoll, keine redundanten Informationen innerhalb des FPERs zu verwenden, um die Zahl der expliziten Einträge klein zu halten.

- Verhaltensäquivalenz soll so eingebunden werden, dass die epistemische Verankerung aller Matches aus einer Menge von verhaltensäquivalenten Matches nach der Revision um die Negation eines dieser Matches auf null gesenkt wird, so dass der Agent diese nicht als mögliche alternative Wege ansieht

- Nach dem Ausscheiden aller alternativen Matches innerhalb einer Situation soll das Revisionsmodul in der Lage sein, daraus Rückschlüsse auf vorangegangene Situationen zu ziehen.

[9] Siehe 3.2

- Es sollte eine möglichst große Funktionalität innerhalb der Revisionsbasis realisiert sein und wenig Funktionalität in dem ihr vorgeschaltetem Modul, das die Informationen aus den verschiedenen Wissensquellen zusammenführt und für die Verwendung in der Revisionsbasis aufbereitet.

Um die Wissensbasis prägnant zu halten, sollten nur Matches mit einem Ähnlichkeitsmaß oberhalb des benutzten Schwellenwerts[10] verwendet, da die Restlichen auch nach dem aktuell verwendeten Verfahren verworfen werden.

4.3.2 Zur Repräsentation der Koreferenz in der Revisionsbasis

Ein Match besteht aus einer Sammlung von Koreferenzen zwischen Knoten des Instruktions- und Perzeptionsgraphens. Eine Koreferenz ist zum einen eine Relation, die im Kontext des Agenten die Identität der Repräsentation von zwei Konzepten darstellt[11]. Zum anderen besitzt die Koreferenz, wie in [H 2003] in Abschnitt 1.2.2 dargelegt wird, einen Konfidenzwert im Intervall $[0, 1]$. Der Konfidenzwert der Koreferenz fließt auch in das Ähnlichkeitsmaß des Gesamtmatches ein (siehe [H 2003], Definition 4.4.2). Das Ähnlichkeitsmaß der einzelnen Matches werde ich bei der Berechnung der epistemischen Verankerung verwenden und daher auf die Aufnahme des Konfidenzwerts der einzelnen Koreferenzen in die Revisionsbasis verzichten, da der Konfidenzwert indirekt bereits bei der Berechnung der epistemischen Verankerung der Matches verwendet wird. Durch den Wegfall des Konfidenzwerts der Koreferenz bleibt der Aspekt der Relation zur Aufnahme in die Revisionsbasis.

Wird eine logische Sprache, die Relationen beinhaltet, in der Revisionsbasis verwendet, so lassen sich Koreferenzen direkt als Relation in der Revisionsbasis abbilden. Die Abbildung der Koreferenzen als Relation bedeutet, dass die koreferenzierten Knoten aus dem Instruktions- und Perzeptionsgraph als Individuenkonstanten aufgefasst werden. In den Relationen werden allerdings keine Variable verwendet. Da es möglich ist, variablenfreie Relationen als atomare Aussagen einer aussagenlogischer Sprache aufzufassen, werde ich Koreferenzen also atomare Aussagen in die Revisionsbasis aufnehmen. Dies ist möglich, weil, wie Schöning in [S 1987] auf S. 81 formuliert, sich variablenfreie Formeln in Herbrand-Expansionen aussagenlogisch interpretieren lassen, da die Terme keine Bedeutung haben (bezogen auf die Herbrand Expansion einer Formel (E/F):

> Man beachte, dass die Formeln in E(F) letzlich wie aussagenlogische Formeln behandelt werden können, dass sie keine Variablen enthalten. Anstelle von $A_1, A_2, A_3 \dots$ wird sozusagen ein anderes Bezeichnungssystem für die atomaren Formeln verwendet. Bei der Angabe einer Struktur für die Formeln in E(F) genügt es, die Wahrheitswerte der atomaren Formeln in E(F) anzugeben. Die Angabe eines Grundbereichs und der Interpretation der Terme ist überflüssig.

Die Koreferenzen besitzen alle Eigenschaften (sie sind variablenfrei, und haben keine Quantoren) der Formeln in der Herbrand-Expansion, weshalb Schönings Argumentation auch im Falle der Koreferenzen anwendbar ist.

In der Definition von θ_k wird die Abbildung von Koreferenzen als Atom der Aussagenlogik für die Wissensbasis durchgeführt. Ich gehe dabei davon aus, dass eine Koreferenz immer auf das gleiche Atom abgebildet wird.

Die koreferenzierten Knoten werden durch diese Entscheidung nicht explizit in der Revisionsbasis Verwendung finden.

Anmerkung: in den Beispielen und im Anhang stelle ich Koreferenzen aus Gründen der Übersichtlichkeit in dem Format „k“i vor, wobei i eine natürliche Zahl > 0 ist. Liegen zwei Koreferenzen mit unterschiedlichem Index vor, so ist davon auszugehen, dass die Koreferenzen verschieden sind, ist der Index gleich, so wird auch die gleiche Koreferenz bezeichnet.

---

[10] Siehe 2.4
[11] Siehe 2.2.2

4.3.3 Zur Repräsentation eines einzelnen Matches und der Verhaltensäquivalenz in der Revisionsbasis

Ich sehe prinzipiell zwei Möglichkeiten, ein einzelnes Match in der Revisionsbasis zu repräsentieren:

- Ein einzelnes Match wird als atomare Aussage modelliert. Dies hat einerseits zur Folge, dass die durch den Beweiser zu leistende Arbeit minimiert wird, da die Modellierung des Matches als atomare Aussage die denkbar kleinste Einheit innerhalb der Aussagenlogik darstellt. Anderseits geht durch diese Modellierung ein Teil des Nutzens der Revision verloren. Dies wird in diesem Unterabschnitt und Abschnitt 5.1 genauer untersucht. Durch $\theta^{'}$ wird eine Revisionsbasis auf Grundlage der Repräsentation von einzelnen Matches als atomare Aussage spezifiziert.

- Ein Match wird als eine Konjunktion von Koreferenzen dargestellt. Diese Koreferenzen werden wie in Unterabschnitt 4.3.2 dargelegt als atomare Aussagen repräsentiert und durch eine Konjunktion miteinander verknüpft. Dies hat den Vorteil, dass auch Informationen über einzelne Koreferenzen verarbeitet werden können, allerdings steigt der Rechenaufwand für die Beweise an. Die Wirkung der Konjunktion ist nach Williams in [W 1997], dass die Konjunktion von Aussagen dazu führt, dass die Konjunkte zusammengebunden werden, und eine Kontraktion um ein Konjunkt zum „Fall" aller Konjunkte führt, die nicht explizit in der Revisionsbasis existieren oder aus anderen Sätzen ableitbar sind. Ergebnisse der Performanzuntersuchung sind in Abschnitt 5.2 dargestellt. θ spezifiziert eine Revisionsbasis auf der Grundlage der Repräsentation von Matches als Konjunktion von Koreferenzen.

Beide oben aufgeführten Möglichkeiten zur Umsetzung eines einzelnen Matches und ihre Auswirkungen werden in diesem Unterabschnitt, sowie in Kapitel 5 abgewogen. Da die Frage nach der Repräsentation eines einzelnen Matches eng mit der Frage nach der Notationsform der Verhaltensäquivalenz in der Revisionsbasis zusammenhängt, werden beide Fragestellungen zusammen erörtert.

Verhaltensäquivalenz

Verhaltensäquivalenz besitzt inherent zwei wesentliche Aspekte, die in der Revisionsbasis umgesetzt werden sollen. Zum einem soll Verhaltensäquivalenz das gleiche Verhalten des Agenten bei der Wahl unterschiedlicher Matches modellieren. Die erste Anforderung an die Revisionsbasis ist daher, das gleiche Verhalten des Agenten bei der Wahl dieser Matches zu erfassen. Der zweite Aspekt der Verhaltensäquivalenz ist der wechselseitige Ausschluss von nicht verhaltensäquivalenten Matches. Die zweite Anforderung an die Notation der Verhaltensäquivalenz ist daher, diese Exklusionsbeziehung ausdrücken zu können. Es ist dabei möglich, dass die epistemische Verankerung von Matches, die exklusiv zu dem gewählten Match sind, 0 ist, aber gleichzeitig auch die epistemische Verankerung der Negation dieser Matches ebenfalls 0 ist. Wünschenswert ist es, dass die epistemische Verankerung der Negation der exklusiven Matches ebenfalls größer als 0 ist, da dies den wechselseitigen Ausschluss des Matches akurater wiederspiegelt.

Es ist jedes Match zu jedem anderen Match exklusiv. Der wechselseitige Ausschluss aller Matches untereinander beruht auf der Tatsache, dass sich jedes Match in mindestens einer Koreferenz unterscheidet. Außerdem ist sind alle Koreferenzen, die den gleichen Knoten der Instruktion mit einem unterschiedlichen Knoten der Perzeption koreferenzieren, zueinander exklusiv. Dass Wissen über die Exklusivitätsbeziehung der Koreferenzen könnte direkt für jede Koreferenz in die Revisionsbasis eingetragen werden, ähnlich wie es in θ_{ex} für die Koreferenzen des Ziels der Aktion geschieht, wodurch der wechselseitige Ausschluss der Matches implizit ableitbar wäre. Die Repräsentation dieses wechselseitigen Auschlusses ist die exakte Umsetzung der bekannten Informationen über Matches und dem Matchingalgorithmus. Allerdings ist die Aufnahme der Exklusionsbeziehung zwischen allen Koreferenzen, die den gleichen Instruktionsknoten mit unterschiedlichen Knoten der Perzeption koreferenzieren durch die Einführung einer Vielzahl von zusätzlichen Implikationen in die Revisionsbasis aus Gründen der Performanz nicht wünschenswert. Des weiteren ist wie im Abschnitt über die Verhaltensäquivalenz erwähnt, für den Agenten in erster Linie die Koreferenz, die den Instruktionsknoten des Ziels der Aktion enthält, von Bedeutung, weshalb ich die Repräsentation des wechselseitigen Auschlusses aller Koreferenzen nicht in die Revisionsbasis aufgenommen habe. Um dafür zu sorgen, dass in jeder Situation maximal ein Match eine epistemische Verankerung > 0 besitzen kann, soll statt dessen die Totalordnung, die in der Definition von θ_m gefordert wird, umgesetzt werden.

In Zusammenhang mit der Exklusionsbeziehung steht die Frage, welche Matches nach einer Revision als Alternative angeboten werden sollen. Diese Frage wird durch die Benutzung der in Abschnitt 4.3 erwähnten strengen Totalordnung gelöst. Diese strenge Totalordnung muss ebenfalls in der Revisionsbasis umgesetzt werden. Wie die Matches in eine Totalordnung gebracht werden können, ist an dieser Stelle nicht von Bedeutung. Es ist im Folgendem aber eine Voraussetzung, dass die Matches in einer strengen Totalordnung zueinander stehen. Aufgrund der Grundsatzentscheidung, einmal ausgeschlossene Matches nicht wieder in Betracht zu ziehen, wird die strenge Totalordnung der Matches im Falle der Notwendigkeit von Revisionen einmal durchlaufen. Dies führt dazu, dass immer ein oder kein Match eine epistemische Verankerung > 0 haben muss. Einmal augeschlossene Matches besitzen eine epistemische Verankerung von 0. Sind mehrere Matches ausgeschlossen, so besitzen alle ausgeschlossenen Matches eine epistemische Verankerung von 0. Dies widerspricht den Eigenschaften der strengen Totalordnung. Aus diesem Grund muss innerhalb der Wissensbasis eine (einfache) Totalordnung umgesetzt werden. Wird der wechselseitige Ausschluss von verhaltensäquivalenten Matches umgesetzt, so haben in jeder Situation die Negationen aller nicht zu dem Match, das eine epistemische Verankerung > 0 besitzt, verhaltensäquivalenten Matches eine epistemische Verankerung > 0. Der Negation von Matches, die zu dem Match, das eine epistemische Verankerung > 0 besizt, verhaltensäquivalent sind, kann aber keine epistemische Verankerung > 0 zugewiesen werden. Damit sind weder die Negation von Matches, die zum dem Match mit einer positivem epistemischen Verankerung verhaltensäquivalent sind, noch sie selbst aus der Revisionsbasis ableitbar. Diese Konstellation kann so interpretiert werden, dass die Revisionsbasis keine Informationen über die zum aktuell als zutreffend angesehenen Match verhaltensäquivalenten Matches besitzt. Innerhalb des Kontext des Agenten ist aber die Interpretation: „Es spielt keine Rolle“ zutreffender.

Aus den Anforderungen zur Umsetzung der strengen Totalordnung in der Revisionsbasis folgt des weiteren, dass innerhalb der Revisionsbasis nach Eintragung der Matches einer Situation das nach der strengen Totalordnung beste Match eine epistemische Verankerung > 0 besitzen muss. Auch muss sichergestellt werden, dass das Durchführen von Revisionen die Reihenfolge der nicht durch die Revision betroffenen Matches innerhalb ihrer Totalordnung nicht ändert. Nach einer Revision um ein einzelnes Match sollen alle zu diesem Match verhaltensäquivalenten Matches mit 0 verankert sein.

Zusammengefasst ergeben sich folgende Anforderungen an die Repräsentation der Verhaltensäquivalenz unter Berücksichtigung der Umstände, in denen die Revisionsbasis eingesetzt wird:

1. Die epistemische Verankerung aller Matches aus der gleichen Verhaltensäquivalenzklasse muss nach dem Eintreffen der Information, dass eins von ihnen nicht zutrifft, auf 0 gesetzt werden.

2. Es kann maximal ein Match von der Revisionsbasis gefolgert werden, was bedeutet, dass maximal einem Match durch die Revisionsbasis eine epistemische Verankerung > 0 zugeordnet werden kann.

3. Das Match mit einer epistemischen Verankerung > 0 ist das nach der Totalordnung und dem gegenwärtigen Informationsstand des Agenten her beste. Falls kein bestes Match nach gegenwärtigem Informationsstand existiert, besitzt kein Match eine epistemische Verankerung > 0.

4. Direkt nach dem Erstellen der Revisionsbasis in einer Situation besitzt ein Match eine epistemische Verankerung > 0, wenn der Matcher mindestens ein Match liefert.

5. Nach dem Durchführen einer Revision, die zum Ausschluss des gewählten Matches führt, wird ein alternatives Match in Betracht gezogen, solange noch ein zu den bisher revidierten nicht verhaltensäquivalentes Match existiert.

6. Die epistemische Verankerung der Negation aller zum aktuell gewählten Match nicht verhaltensäquivalenten Matches, ist größer als 0.

Es soll nachfolgend gezeigt werden, dass die oben genannten Anforderungen zur Repräsentation von Verhaltensäquivalenz durch die Revisionsbasis erfüllt werden. Dabei ist auch die Einbeziehung des Adjustments von Bedeutung.

Aufgrund meiner Grundsatzentscheidung, einmal ausgeschlossene Matches nicht erneut in Betracht zu ziehen, werden ausschließlich Adjustments durchgeführt, in denen ein Match oder eine Koreferenz ausgeschlossen werden. Es kann damit insbesondere keine Adjustments geben, durch die ein einmal ausgeschlossenes Match wieder in betracht gezogen wird. Nach Definition wird durch θ_m eine Menge von Disjunktion erzeugt. Da jede Disjunktion innerhalb der von θ_m erzeugten Menge eine eindeutige Anzahl von Disjunkten hat, lassen sich die Disjunktionen anhand der Zahl ihrer Disjunkte nummerieren. Nachfolgend sei mit der i-ten Disjunktion $\theta_m(s,i)$ diejenige Disjunktion mit i Disjunkten bezeichnet.
Nach einem rekursivem Schema ist $\theta_m(s,i)$ definiert als:

$$\theta_m(s,i) := \left\{ \begin{array}{ll} \theta_\mu(m_1) & \text{falls } i = 1 \\ \theta_m(s,i-1) \vee \theta_\mu(m_i) & \text{falls } i > 1 \\ \top & \text{andernfalls} \end{array} \right\}$$

Aus dem rekursivem Definitionssschema können folgende Eigenschaften von θ_m erkannt werden :

(TC1) Wenn $i \leq |\text{matches}(s)|$, dann existiert das it beste Match und die Disjunktion $\theta_m(s,i)$

(TC2) Die Disjunktion $\theta_m(s,i)$ die dem it bestem Match zugeordnet ist, besitzt i Disjunkte und das i. Disjunkt repräsentiert das i. beste Match

(TC3) Die ersten i-1 Disjunkte in $\theta_m(s,i)$ repräsentieren alle Matches, die nach der Totalordnung her besser als das it beste Match sind.

Beweisskizze 1 (Die Anforderungen 2, 4 werden durch θ_m erfüllt) *Offensichtlich ist m_1 direkt nach Eintragung von θ_m in die Revisionsbasis folgerbar, da der Eintrag $\theta_m(1)$, der aus $\theta_\mu(m_1)$ besteht, explizit mit positiver epistemischer Verankerung in der Revisionsbasis eingetragen ist. Damit ist Anforderung 4 erfüllt. m_1 ist nach Definition von θ_m das nach der Totalordnung her größte Match.*

***Behauptung**: Anforderung 2 wird durch θ_m erfüllt.*
Beweis:
Sei s eine Situation, $\vee(\psi_1,\ldots,\psi_k)$ mit $i \in \{1,\ldots,k\}, k \leq |matches(s)|$ eine Disjunktion. Wenn in $\vee(\psi_1,\ldots,\psi_k)$ die ersten $k-1$ Disjunkte ausgeschlossen sind (was bedeutet, dass sie nicht aus der Revisionsbasis folgerbar sind), aber bekannt ist, dass die Disjunktion insgesamt wahr ist, dann ist das Disjunkt ψ_k folgerbar. Innerhalb der Disjunktion $\theta_m(s,i)$ ist daher das i. Disjunkt folgerbar, wenn die ersten i-1 Disjunkte ausgeschlossen worden sind. Das i. Disjunkt repräsentiert das Match m_i. Daher wird das Match m_i dann, einen positiven Grad epistemischer Verankerung erhalten, wenn zum einen eine Disjunktion aus θ_m einen positiven Grad epistemischer Verankerung besitzt, und in dieser Disjunktion alle anderen Disjunkte ausgeschlossen werden können. Des weiteren ist es weder möglich, ein Match aus den in θ_{ex} und θ_s definierten Formeln noch aus deren Vereinigung zu folgern, da in diesen die Koreferenzen, die nicht den Instruktionsknoten, der das Ziel der Aktion repräsentiert, weder explizit aufgenommen sind noch implizit gefolgert werden können.

Die Erfüllung von Anforderung 1 durch die Revisionsbasis verlangt zu zeigen, dass nach dem Ausschluss des Matches m_i gilt: $\displaystyle\forall_{m_l \in \mathcal{VE}(m_i)} \text{degree}(\theta_\mu(m_l)) = 0$. Das Adjustment zum Ausschluss eines Matches m einer Situation s ist so definiert, dass es einen Eintrag erzeugt, der die Negation der Koreferenz des Ziels der Aktion in Situation s des Matches m mit einer positiven epistemischen Verankerung

Λ_{remove} einträgt. Nach der Definition gilt, dass ein einzelnes Match als eine Konjunktion von atomaren Aussagen repräsentiert wird. Des weiteren gilt, dass einer Konjunktion durch die Revisionsbasis der minimale Grad der epistemischen Verankerung zugewiesen wird, der den Konjunkten zugeordnet wird ($\text{degree}(\psi_1 \wedge \cdots \wedge \psi_n) = \min(\text{degree}(\psi_1), \ldots, \text{degree}(\psi_n))$ (was aus Anforderung (EE3) an die epistemische Verankerung in [W 1998] auf S. 290 folgt). Unter der Voraussetzung, dass Λ_{remove} größer ist, als die epistemische Verankerung mit der ein Match m verankert ist, gilt, dass ein Adjustment um $\chi(s, m)$ dazu führt, dass der Konjunktion, die m repräsentiert, eine epistemische Verankerung von 0 zugeordnet wird. Die Revisionsbasis kann aufgrund des expliziten Eintrags von $\neg\tau(m)$ mit positiver epistemischer Verankerung folgern, dass jede Konjunktion k, die $\tau(m)$ als Konjunkt enthält, nicht nur nicht zutreffen kann sondern auch $\text{degree}(\neg k) > 0$ ist. Die Folgerbarkeit der Negation von k kann damit begründet werden, dass die Negation einer Konjunktion k nach den de morganschen Regeln äquivalent zu der Disjunktion der negierten Konjunkten von k ist ($\neg \wedge (\psi_1, \ldots, \psi_n) \equiv \vee(\neg\psi_1, \ldots, \neg\psi_n)$). Die Revisionsbasis ordnet jeder Disjunktion mindestens das Maximum des Degrees der einzelnen Disjunkte zu ($\text{degree}(\vee(\psi_1, \ldots, \psi_n)) = \max(\text{degree}(\psi_1), \ldots, \text{degree}(\psi_n))$) (siehe dazu auch S. 294 in [W 1998]). Die Negation aller Konjunktionen, die $\tau(m)$ als Konjunkt enthalten, können daher als Disjunktion repräsentiert werden, die $\neg\tau(m)$ als Disjunkt enthält $\neg \wedge (\ldots, \tau(m), \ldots) \equiv \vee(\ldots, \neg\tau(m), \ldots)$. Da durch das Adjustment $\neg\tau$ mit einer epistemischen Verankerung von Λ_{remove} in der Revisionsbasis eingetragen ist, muss die Revisionsbasis die Negation aller Konjunktionen, die τ enthalten, mindestens den degree Λ_{remove} zuordnen. Da sowohl nach Definition von einfacher als auch strenger Verhaltensäquivalenz verhaltensäquivalente Matches das Ziel der Aktion der Situation s gleich koreferenzieren, werden nach einem Adjustment um $\neg\tau(m)$ auch alle zu m verhaltensäquivalenten Matches ausgeschlossen.

Korollar 1 (Anforderung 1 wird erfüllt) *Anforderung 1 wird durch ein Adjustment um die Negation der Zielkoreferenz eines Matches m in Kombination mit der Repräsentation eines Matches durch eine Konjunktion von Koreferenzen erfüllt. Enthält die Äquivalenzklasse von m mehr als ein Element, so werden die übrigen Matches aus der Äquivalenzklasse ebenfalls mit 0 gewertet. Ebenfalls kann das nicht-Zutreffen aller zum Match m verhaltensäquivalenten Matches aus der Revisionsbasis deduziert werden.*

Beweisskizze 2 (Die Anforderungen 3 und 5 werden durch θ_m erfüllt) *Sei s eine Situation. Aufgrund des Eintrags $\theta_m(s, 1)$ ist das nach der Totalordnung her beste Match m_1 im Initialzustand folgerbar, falls ein solches Match existiert. Damit ist die Anforderung 3 im Initializustand der Revisionsbasis erfüllt. Falls kein solches Match existiert ist die Anforderung 3 trivialerweise erfüllt. Da im Initialzustand der Revisionsbasis einer Situation noch keine Revision durchgeführt wurde, íst trivialerweise auch Anforderung 5 im erfüllt.*

Da die Anforderungen 3 und 5 im Initialzustand der Revisionsbasis erfüllt sind, muss nur gezeigt werden, dass die Eigenschaften auch nach dem Ausschluss einer beliebiger Anzahl von Matches erfüllt sind. Es gilt aufgrund der Annahmen, dass ein ausgeschlossenes Match nicht erneut in Betracht gezogen wird, woraus insbesondere folgt, dass es keine Adjustments gibts, die ein einmal ausgeschlossenes Match oder eine einmal ausgeschlossene Koreferenz wieder folgerbar machen. Daher existiert innerhalb einer Situation eine monoton wachsende Menge von ausgeschlossenen Matches, die nachfolgend mit e_1 bezeichnet wird. Da Anforderung 1 bewiesen ist, gilt, dass es neben den explizit ausgeschlossenen Matches auch eine Menge von Matches e_2 gibt, die zu den explizit ausgeschlossenen Matches verhaltensäquivalent ist. Dabei ist $e_1 \subseteq e_2 \subseteq matches(s)$. Sei $m_i, 0 < i \leq |matches(s)|$ das momentan gewählte Match.

1. *Angenommen, es existiert nach dem Adjustment um $\chi(s, m_i)$ kein weiteres bestes Match. Es gilt damit, dass $e_2 = matches(s)$. Da Anforderung 1 erfüllt ist, gilt, dass innerhalb der Revisionsbasis kein Match mehr folgerbar ist und, da es keine Adjustments geben kann, die ein Match wieder in Betracht ziehen, es auch später nicht mehr folgerbar sein wird. Damit ist Anforderung 3 erfüllt. Anforderung 5 ist trivialerweise erfüllt, da keine mögliche Alternative existiert, die in Betracht gezogen werden könnte.*

2. *Angenommen, es existiert nach dem Adjustment um $\chi(s, m_i)$ das nächst beste Match $m_k, i < k \leq |matches(s)|$. Damit existiert nach TC1 auch die Disjunktion $\theta_m(s,k)$. Aufgrund von TC2 repräsentiert das k. Disjunkt aus $\theta_m(s,k)$ das Match m_k. Die ersten k-1 Disjunkte in $\theta_m(s,k)$ repräsentieren nach TC3 Matches, die nach der Totalordnung für Matches besser als m_k sind. Da nach Anforderung 3 gilt, dass m_k das beste Match ist, was nicht verhaltensäquivalent zu bisher explizit ausgeschlossenen Matches ist, werden innerhalb der Disjunktion $\theta_m(s,k)$ die ersten k-1 Disjunkte als falsch ausgewertet. Wenn bekannt ist, dass eine Disjunktion $\vee(\psi_1, \ldots, \psi_k)$ wahr ist und die Disjunkte $\psi_1, \ldots \psi_{k-1}$ nicht wahr sind, so kann das verbleibende Disjunkt ψ_k gefolgert werden. Damit kann in der Disjunktion $\theta_m(s,k)$ das kte Disjunkt gefolgert werden, welches die Repräsentation des kten Match ist ($\theta_\mu(m_k)$). Da es keine Adjustments geben kann, die ein einmal ausgeschlossenes Match oder eine einmal ausgeschlossene Koreferenz erneut in Betracht ziehen, ist damit Anforderung 5 erfüllt. Um zu zeigen, dass Anforderung 3 erfüllt ist, muss nachgewiesen werden, dass*

$$\bigvee_{m_j \in matches(s)} j < k \supset degree(\theta_\mu(m_k)) = 0$$

gilt. Wenn m_k das verbleibende beste Match ist, so ist jedes Match $m_j \in e_2$. Die Revisionsbasis erfüllt Anforderung 1. Daher ist der Degree der Repräsentation jedes Matches m_j 0, da jedes m_j verhaltensäquivalent zu einem explizit ausgeschlossenem Match ist. Damit ist Anforderung 4 erfüllt.

Adjustment

Die einzelnen Disjunkte der Einträge lassen sich sowohl durch ein Adjustment um die Negation eines Matches, wie auch durch das Adjustment um die Negation einer der Koreferenzen des das Disjunkt repräsentierenden Matches ausschließen, vorausgesetzt die Matches sind nicht als atomare Aussage repräsentiert. Ein Adjustment um die Negation einer Koreferenz ist dabei sinnvoller als die Revision um die Negation eines ganzen Matches, da die Revisionsbasis dadurch alle anderen Matches, die ebenfalls die Koreferenz, um deren Negation das Adjustment durchgeführt wird, enthalten, ausschließen kann. Da aus der Information, dass das gewählte Match m nicht zutrifft, geschlossen werden kann, dass $\tau(m)$ nicht korrekt ist, kann ein Adjustment um $\neg\tau(m)$ durchgeführt werden. Dies führt dazu, dass alle zu m verhaltensäquivalenten Matches ebenfalls ausgeschlossen werden. Dies ist der Vorteil, den die Repräsentation eines einzelnen Matches durch eine Konjunktion von Koreferenzen bietet. Werden einzelne Matches jeweils als atomare Aussage repräsentiert, so ist es notwendig, die Verhaltensäquivalenz explizit in die Revisionsbasis einzutragen. Dies erfordert allerdings die Berechnung von $\mathcal{VE}$ für jedes Match einer Situation außerhalb der Revisionsbasis. Da eine „Auslagerung“ von Funktionalität aus der Revisionsbasis in das vorgeschaltete Modul zur Zusammenführung von Wissen aus den verschiedenen Wissenaquellen des Agenten bedeutet, ist dies aber nicht sinnvoll.

Exklusionsfunktion

Da die Anforderung 3 durch θ_m erfüllt wird, gilt, dass wenn die Revisionsbasis ausschließlich durch θ_m spezifiziert wird, in jeder Situation höchstens ein Match eine epistemische Verankerung > 0 besitzen kann, aber es der Revisionsbasis unmöglich ist zu deduzieren, dass die Negation der nicht ausgeschlossenen Matches, die exklusiv zu dem aktuell gewähltem sind, einen positiven Degree besitzt. Für Matches, die nicht explizit ausgeschlossen wurden, gilt, dass die Revisionsbasis weder für das Match selbst, noch für die Negation des Matches einen positiven Degree ableiten kann, was bedeutet, dass die Revisionsbasis keine Informationen über sie hat. Wenn die Negation der nicht ausgeschlossenen Matches, die exklusiv zum gewähltem Match sind, keinen positiven Degree besitzt, so repräsentiert das nicht die wechselseitige Exklusivität, die zwischen allen Matches besteht. Um den wechselseitigen Ausschluss zwischen zwei Matches zu repräsentieren wurde die Funktion θ_{ex} eingeführt.

Die Funktion θ_{ex} erzeugt eine Menge von Konditionalen, in denen je die Koreferenz des Zielknotens der Instruktion eines Matches m die Negation der Koreferenz des Zielknotens der Instruktion eines zu m nicht verhaltensäquivalenten Matches impliziert. Jedes einzelne dieser Konditionale hat zwei Effekte:

1. Wenn ein Match, das die Koreferenz der Antezedens enthält, einen positiven Degree besitzt, so wird jedes Match, das die Koreferenz der Konsequenz enthält, einen Degree von 0 erhalten. Dies ist allerdings ein überflüssiger Nebeneffekt, da bereits θ_m Anforderung 3 erfüllt und so je Situation nur ein Match einen Degree > 0 haben kann.

2. Wenn ein Match, das die Koreferenz der Antezedenz enthält, einen positiven Degree besitzt, so ist der Degree der Negation aller Matches, die eine Koreferenz der Konsequenz enthalten, größer als 0.

Korollar 2 (Anforderung 7 wird durch θ_{ex} erfüllt)

Aufgrund des Modus Tollens ist es nicht notwendig, dass, wenn bereits eine Implikation der Form $\tau(m_1) \supset \neg\tau(m_2)$, in der Revisionsbasis existiert, eine zusätzliche Implikation der Form $\tau(m_2) \supset \neg\tau(m_1)$ eingefügt wird. Die implizite Anwendung des Modus Tollens durch den Beweiser wird durch die Funktion θ_{ex} genutzt, um so die Größe des durch θ_{ex} erstellten expliziten Revisionsbasiseintrags zu reduzieren. Trotz der Reduzierung der Größe des Eintrags in der Revisionsbasis werden alle notwendigen wechselseitigen Ausschlüsse durch die in $theta_{ex}$ verwendete Implikation abgedeckt. Notwendig ist der wechselseitige Ausschluss aller Matches aus der Menge $\mathcal{NVE}_{\text{unique}}$, aus denen nach den Vorbedingungen der Definition von θ_{ex} die Koreferenzen von Antezedens und Konsequenz der Implikation stammen und die auch alle durch θ_{ex} paarweise erfasst werden. Aufgrund der Tatsache, dass Verhaltensäquivalenz eine Äquivalenzrelation und damit transitiv und symmetrisch ist, ist es ausreichend, dass sich der wechselseitige Ausschluss auf die Elemente von $\mathcal{NVE}_{\text{unique}}$ beschränkt. Die Beschränkung des wechselseitigen Ausschluss auf Elemente von $\mathcal{NVE}_{\text{unique}}$ hat zur Folge, dass Matches einer Verhaltensäquivalenzklasse sich wechselseitig nicht ausschließen. Allerdings erfüllt die Revisionsbasis auch mit θ_m Anforderung 3, weshalb es auch unter dieser Einschränkung der Definition θ_{ex} nicht dazu kommen kann, dass die Revisionsbasis dem Agenten mehr als ein Match anbietet. Es kann allerdings der Negation der restlichen Matches aus der Verhaltensäquivalenzklasse des aktuell gewählten Matches nicht ein positiver Degree zugeordnet werden.

Es ist möglich, in θ_{ex} anstelle der Koreferenzen des Instruktionsziels, komplette Matches in der Implikation zu verwenden.[12] Dies kann unter dem Umstand, dass Matches generell als atomare Aussagen oder als Konjunktion von Koreferenzen repräsentiert werden, geschehen. Wenn Matches generell als atomare Aussagen repräsentiert werden, dann kann ein Adjusment nicht um eine einzelne Koreferenz ausgeführt werden, was zu dem auf Seite 54 geschilderten negativen Auswirkungen führt. Wird ein Match als Konjunktion von Koreferenzen repräsentiert, so werden Implikationen, die komplette Matches ausschließen, dadurch zum einem unnötig lang, zum anderen reicht der wechselseitige Ausschluss aller Matches aus $\mathcal{NVE}_{\text{unique}}$ nicht aus, um der Negation von nicht zum aktuell gewählten Match verhaltensäquivalenten Matches einen Degree > 0 zuzuordnen, was ich an folgendem Beispiel illustrieren werde:

Seien $\{m_1, m_{eq_1}, \ldots, m_{eq_n}\} = \mu\sigma(s), s \in \mathbb{S}, m_{eq_1} \ldots m_{eq_n} \in \mathcal{VE}(m_{eq_1}), \neg ev(m_1, m_2)$, was bedeutet, dass es in einer Situation s die Matches m_1 sowie die Matches $m_{eq_1} \ldots m_{eq_n}$ gibt, wobei die letzteren untereinander verhaltensäquivalent sind, während m_1 zu dieser Gruppe von Matches nicht verhaltensäquivalent ist. Wenn die Exklusionsfunktion ganze Matches in Antezedens und Konsequenz benutzt, so ergibt sich $\theta_{ex}(s) = m_1 \subset \neg m_{eq_1}$. Ist $\gamma(s) = m_1$, dann ist der Degree von $\neg(m_{eq_2}) \ldots \neg(m_{eq_n})$ 0, da die Revisionsbasis nicht die Transitivität der Verhaltensäquivalenz von $\mathcal{VE}(m_2)$ deduzieren kann.

Es wäre in dem Fall, dass die Exklusionsregeln auf Basis eines Matches definiert werden, notwendig die Revisionsbasis entweder um einen oder mehrere explizite Einträge, mit der die Transitivität der

[12] in der Form $m_i \supset \neg m_k$

Verhaltensäquivalenz durch die Revisionsbasis deduzierbar ist, zu erweitern oder in den Exklusionsregeln die Implikation nicht nur für alle Matches aus $\mathcal{NVE}_{unique}$ sondern für alle Matches der Situation zu definieren.

Wie in der Definition von θ_{ex} verlangt, ist es zur Reduzierung der Rechenzeit im Rahmen der Verwendung der Revisionsbasis wichtig, dass nicht beliebige Koreferenzen in den durch θ_{ex} definierten Funktionen verwendet wird, sondern jeweils die Koreferenz, die den Zielknoten der Instruktion beinhaltet. Es sollten auf ausschließlich Koreferenzen als exklusiv zueinander spezifiziert werden, die den gleichen Knoten aus der Instruktion mit unterschiedlichen Knoten der Perzeption koreferenziert. Zwar schließen sich generell Koreferenzen aus, die den gleichen Knoten der Instruktion mit unterschiedlichen Knoten der Perzeption koreferenzieren, allerdings hat der wechselseitige Ausschluss von anderen Koreferenzen als der Koreferenz des Ziels der Aktion kein nach außen sichtbares unterschiedliches Ergebnis (siehe Abschnitt 4.2).

Zur Abwägung, ob ein einzelnes Match als atomare Aussage oder als Konjunktion von Koreferenzen in der Revisionsbasis repräsentiert wird sowie zur Motivation hinter meiner Wahl der Exklusionsregeln, läßt sich folgendes Fazit ziehen:

Die Repräsentation eines Matches in der Revisionsbasis durch eine atomare Aussage ist auf den ersten Blick kürzer als die Repräsentation durch eine Konjunktion von Koreferenzen. Allerdings wird es durch die Repräsentation des einzelnes Matches als atomare Aussage unmöglich, in θ_{ex} die Exklusivität von Matches durch einzelne Koreferenzen zu definieren. θ_{ex} läßt sich auch auf der Basis von einzelnen Matches definieren, allerdings wird dazu eine erheblich größere Anzahl von Einträgen in der Revisionsbasis benötigt. Ebenfalls kann das Adjustment, das den Ausschluss eines Matches m in der Revisionsbasis anstoßen soll, auf die Negation von $\tau(m)$ beschränkt werden.

4.3.4 Situationsübergreifende Revision

In den beiden vorangegangenen Unterabschnitten sind die Grundlagen des Aufbaus der Revisionsbasis erklärt worden. Dabei ist bisher die in Abschnitt 4.3 aufgestellte Forderung, dass die Revisionsbasis in der Lage sein soll, aus dem Ausscheiden von Matches einer Situation Rückschlüsse auf die vorangegangene Situation(en) zu ziehen, nicht umgesetzt worden. Wie in Abschnitt 4.1 ausgeführt, läßt das Fehlen von Matches in einer Situation den Schluss zu, dass der Agent in einer früheren Situation eine falsche Entscheidung getroffen hat. Die falsche Entscheidung besteht darin, dass der Agent ein Match gewählt hat, das dazu führt, dass er nicht der Instruktion folgen kann. Dies bedeutet, dass aus dem Fehlen von Matches in einer Situation s der Schluss gezogen wird, dass die Wahl zumindest des Matches in der Situation direkt vor s nicht zutrifft. Die Situation direkt vor s werde ich in Zukunft mit s_{alt} abkürzen. Falls der Fehler bereits in einer der Situation vor s_{alt} gemacht wurde, so kann dies nicht automatisch geschlossen werden. Der Agent muss in jeder Situation s erst die nicht verhaltensäquivalenten Matches nach dem Ausschluss des gewähltem Match in Erwägung ziehen, bevor der Rückschluss auf s_{alt} möglich ist. Es ergeben sich folgende Anforderungen, welche die situationsübergreifenden Revision erfüllen muss:

1. Bevor eine situationsübergreifende Revision von s auf s_{alt} angestoßen wird, müssen alle nicht verhaltensäquivalenten Matches in s ausgeschlossen worden sein.

2. Eine situationsübergreifende Revision von s muss immer auf s_{alt} erfolgen. Falls der Fehler, der zum Fehlen von passenden Matches geführt hat, in einer Situation vor s_{alt} liegt, so muss der Revisionsmechanismus in s_{alt} diesen Schluss ziehen.

3. Die situationsübergreifende Revision soll Verhaltensäquivalenz berücksichtigen.

θ_s erfüllt diese drei genannten Anforderungen, wenn die durch θ_s generierten Einträge in der Revisionsbasis eingetragen sind. $\theta_s(s, s_{\text{alt}})$ erzeugt einen Eintrag bestehend aus einer Implikation, deren Antezedens sich aus der Konjunktion der Zielkoreferenz aller nicht zueinander verhaltensäquivalenten

Matches der Situation s zusammensetzt. Die Konsequens ist die Negation der Zielkoreferenz des in s_{alt} gewählten Matches.

Für jedes Match $m_1 \in \text{matches}(s)$ gilt, dass exakt ein Match $m_2 \in \mathcal{NVE}_{\text{unique}}(s)$ existiert, so dass $ev(m_1, m_2)$, womit auch $\tau(m_1) = \tau(m_2)$. Diese Aussage kann direkt aus der Definition von einfacher Verhaltensäquivalenz und $\mathcal{NVE}_{\text{unique}}$ abgeleitet werden. Wenn jedes Match aus $\mathcal{NVE}_{\text{unique}}(s)$ ausgeschlossen ist, sind somit auch alle Matches aus matches(s) ausgeschlossen. Die in θ_s definierte Implikation ist damit genau dann wahr, wenn alle Matches in s ausgeschlossen sind, was die erste der drei Anforderungen an die situationsübergreifende Revision erfüllt.

Nach der Definition von θ_s besteht die Konsequenz der Implikation aus der Negation der Zielkoreferenz des in s_{alt} gewählten Matches. Wenn die Antezendens wahr ist, so kann der Revisionsmechanismus diese Negation folgern, was die gleichen Veränderungen in der Revisionsbasis wie eine von außerhalb der Revisionsbasis angestoßene Revision um $\neg\theta_k(\tau(\gamma(s_{\text{alt}})))$ auslößt. Dass das Anstoßen der Revision von außen überflüssig ist, zeigt, dass die Revisionsbasis den restlichen Modulen des Agenten einen Teil des Räsonierens abnimmt. Da die Konsequenz sich auf ein Element von s_{alt} bezieht, ist die zweite Anforderung an die situationsübergreifende Revision durch θ_s erfüllt.

θ_s selbst berücksichtigt nicht explizit die Verhaltensäquivalenz bei Ausschluss des gewähltem Matches in der vorrangegangenen Situation, da sie z.B. nicht verlangt, dass alle zu $\gamma(s_{\text{alt}})$ verhaltensäquivalenten Matches ebenfalls ausgeschlossen werden oder dass nach dem Ausschluss von $\gamma(s_{\text{alt}})$ ein nächstbestes Match gewählt wird, falls den ein solches existiert. Sei $s_{\text{älter}}$ nun die Situation vor s_{alt}. Wenn, wie in der Voraussetzungen erwähnt, in der Revisionsbasis die durch $\theta(s_{\text{alt}}, s_{\text{älter}})$ erstellten Einträge existieren, nutzt θ_s die Eigenschaften der dort spezifizierten Einträge aus. Deshalb erfüllt θ_s Anforderung 3.

Die atomare Aussage stop dient dazu, innerhalb der ersten Situation einen Bezugspunkt für die Situationsübegreifende Verankerungsfunktion zu haben. Der Agent kann durch die Abfrage des Degrees von stop ermitteln, ob nach Kenntnisstand der Wissensbasis der Agent weiter navigieren soll oder der Navigationsversuch fehlgeschlagen ist. Besitzt stop einen positiven Degree, so ist die Navigation fehlgeschlagen. stop wird einen positiven Degree erhalten, wenn der Agent alle ihm möglich erscheinenden Wegalternativen untersucht hat und trotzdem nicht zum Ziel gekommen ist.

4.4 Epistemische Verankerung

Die absolute Höhe der epistemischen Verankerung hat, wie in 3.3 erwähnt, keine direkte Bedeutung, vielmehr der relative Unterschied von verschiedenen Einträgen innerhalb des FPER. Deshalb wurden in der situationsübergreifenden Verankerungsfunktion sowie der Exklusionsfunktion zwei Ebenen der epistemischen Verankerung eingeführt. In der Übersetzungsfunktion für Matches wurde $ep_{\mu\sigma}$ eingeführt, um die epistemische Verankerung der sukzessiv aufsteigenden Disjunktionen zu bezeichnen. Die durch $ep_{\mu\sigma}(n)$ erzeugten Ebenen der epistemischen Verankerung sollen mit $\Lambda_{\mu\sigma n}$ bezeichnet sein. Des weiteren ist die epistemische Verankerung eines Adjustments zum Ausschließen eines Matches von Bedeutung. Diese sei mit Λ_{remove} bezeichnet. Ziel dieses Abschnitts ist es, alle Bedingungen zu sammeln, die durch die Ebenen der epistemischen Verankerung zu erfüllen sind. Offensichtlich gelten zunächst einige formale Grundbedingungen:

(CA1) Alle Λ_i müssen rationale Zahlen sein, da technische Implementation einer Turingmaschine endlich sind.

(CA2) Alle Λ_i müssen ≥ 0 sein, da sie innerhalb des für die epistemische Verankerung einer FPER definierten Intervalls liegen müssen.

(CA3) Alle Λ_i müssen < 1 sein, da sie keine Tautologien sind.

Insgesamt existieren zwei Klassen von Ebenen der epistemischen Verankerung: solche, die Einträgen zugeordnet sind, die auf relativ unsicherem Wissen beruhen, also insbesondere $\Lambda_{\mu\sigma 1} \ldots$ und solche, die aufgrund der Verhältnisse in der Welt des Agenten eine sehr starke Verankerung haben müssen.

Die Exklusionsfunktion ist ein naheliegender Kandidat für epistemisch sehr stark verankerte Einträge. Es gilt aufgrund der Bedingungen des Aktionsmoduls, dass der Agent immer genau ein Match aus einer Menge von Alternativen auswählen muss, da jedem Knoten aus dem Instruktionsgraph ein eindeutiger Knoten der Perzeption zugeordnet werden sollte. Das Revisionsmodul muss diesen Sachverhalt widerspiegeln. Eine niedrige epistemische Verankerung der Exklusion alternativer Matchs würde die Exklusion zwischen den Matches zur Disposition stellen, nämlich genau dann, wenn in der Wissensbasis die jeweils ausgeschlossenen Matches explizit mit einem höhreren Wert als ihre Exklusion eingetragen sind. Dementsprechend muss Λ_1 die höchste Ebene der epistemischen Verankerung nach den Tautologien bekommen.

Der zweite Kandidat mit Einträgen von hoher epistemischer Verankerung ist die situationsübergreifende Verankerungsfunktion. Die hohe epistemische Verankerung der situationsübergreifenden Verankerungsfunktion kann durch die Annahme einer Closed World Assumption (siehe [PMG 1998] Kapitel 7.4), begründet werden. Die Closed World Assumption sagt aus, dass das Wissen eines Agenten komplett ist. Beim geometrischen Agenten ist dies nicht der Fall. Allerdings sind alle Matches, die der Agent berechnet, ein Produkt des Agenten selbst. Prinzipiell kennt der Agent alle Matches, die in einer Situation in Frage kommen. Er grenzt seine Auswahl zwar aufgrund eines Schwellenwerts ihrer Bewertung ein, allerdings läßt sich auch die eingeschränkte Menge als die Gesamtmenge **aller** guten Kandidaten auffassen. Damit läßt sich eine lokale Closed World Assumption rechtfertigen. Der Einwand, es könnte das zutreffende Match durch den Schwellenwert ausgeschlossen werden, läßt sich mit der Annahme begegnen, dass der Agent von vornherein mit einem ausreichend kleinen Schwellenwert, um dies auszuschließen, konfiguriert wurde. Weitere Gründe, die eine Verringerung der epistemischen Verankerung der Einträge der situationsübergreifenden Verankerung begründen könnten, sind die Annahme, dass der Agent in der Berechnung der Matches einer Situation ungenügend oder falsch instruiert ist oder eine fehlerhafte Perzeption besitzt. Diese Fälle wurden jedoch zu Beginn dieses Kapitels ausgeschlossen. Dadurch scheint es zusammen mit der lokalen Closed World Assumption gerechtfertigt zu sein, der situationsübergreifenden Verankerung eine sehr hohe epistemische Verankerung zuzuordnen.

Dem Erkennen, dass das gewählte Match nicht zutrifft, kann unter den (zu Beginn dieses Kapitels getroffenen) Annahmen ebenfalls eine hohe epistemische Verankerung zugeschrieben werden. Da der Agent nur innerhalb seiner Auswahl revidieren und die Perzeption und Instruktion als korrekt ansehen soll, muss das Fehlen eines Matches als zutreffend angesehen werden.

Es stellt sich die Frage, ob es einen qualitativen Unterschied zwischen der Ebene der epistemischen Verankerung der Exklusion, der situationsübergreifenden Verankerung und Λ_{remove} geben sollte. Zum einen gilt, dass es keine Subsumptionsbeziehung zwischen den jeweiligen Einträgen gibt. Dies wäre nach der Empfehlung von Williams in [W 1997] ein Ausschlusskriterium für eine gleiche epistemische Verankerung. Für eine gleiche epistemische Verankerung spricht eine mögliche Verringerung der Berechnungszeit von degree. In dem in dieser Arbeit vorgestellten beschleunigten Verfahren zur Berechnung des Degrees mithilfe eines einzelnen Tableau-Beweises gilt, dass aufgrund der Priorisierung unter Umständen in einer ungünstigen Reihenfolge expandiert wird. Wenn die epistemische Verankerung von Λ_1, Λ_2 und Λ_{remove} jedoch identisch sind, dann ist auch die Priorisierung für den Λ_1 und Λ_2 Λ_{remove} zugeordneten Sätzen identisch, was wiederum es dem Tableau-Beweiser überlässt, eine angemessene Reihenfolge der Expansion zu finden. Aus diesem Grund ergeben sich folgende Bedingungen:

(CTC1) $\Lambda_1 = \Lambda_2 = \Lambda_{\text{remove}}$

(CTC2) $\Lambda_1 > \max\left(\{\Lambda_{\mu\sigma 1} ... \Lambda_{\mu\sigma\infty}\}\right)$

Die epistemische Verankerung von $\Lambda_{\mu\sigma 1} \dots \Lambda_{\mu\sigma n}$ muss kleiner als die von Λ_{remove} sein, da die $\Lambda_{\mu\sigma i}$ zugeordneten Formeln durch diejenigen, die Λ_{remove} zugeordnet sind, ausgeschlossen werden sollen. Ist $\Lambda_{\mu\sigma i} \geq \Lambda_{remove}$, so kann dieser Ausschluss nicht geleistet werden, da nach der Definition von $B^{-}(\varphi, i)$ Formeln, die epistemisch stärker als die zu revidierende Formel verankert sind, nicht verändert werden.

$\Lambda_{\mu\sigma 1}$ ist einem einzelnen Match zugeordnet, $\Lambda_{\mu\sigma 2}$ einer Disjunktion aus zwei Matches, usw. Da es unter anderem die Intention dieser Modellierung ist, die Matches einer Situation gemäß ihrer

Totalordnung der Reihe nach auszuwählen, muss dabei $\Lambda_{\mu\sigma i} < \Lambda_{\mu\sigma k}$ sein für alle $i < k$. Wenn $\Lambda_{\mu\sigma 1} = \cdots = \Lambda_{\mu\sigma n}$ ist, so wird gemäß der Definition von $B^{-}(\varphi, i)$ jede Disjunktion der Matches im Falle einer Revision um die Negation einer Koreferenz auf 0 herabgestuft. Diese Herabstufung ist damit begründet, dass die Koreferenz in mindestens einem der Matches der Disjunktion enthalten ist und damit dieses Match ausgeschlossen wird. Alle Disjunktionen, die dieses Match enthalten, sind aus diesem folgerbar, weshalb diese Disjunktionen ebenfalls auf 0 herabgestuft werden.

4.5 Wahl einer logischen Sprache

Bei der Wahl einer logischen Sprache spielen im wesentlichen die Antagonisten Ausdrucksstärke und Komplexität sowie die Frage der Termination der für die gewählte Sprache zur Verfügung stehenden Beweisverfahren eine Rolle. Die Ausdrucksstärke der Sprache, die für die Wissensbasis verwendet wird, muss ausreichen, um das notwendige Wissen auszudrücken. Wie in diesem Kapitel an verschiedenen Stellen dargelegt, umfassen die Wissensquellen, die explizit in der Wissensbasis ausgedrückt werden müssen, Ausdrücke der Aussagenlogik und Relationen. Wie bereits auf Seite 49 ausgeführt, lassen sich, unter der Bedingung, dass keine freien Variablen in den Relationen auftreten, Relationen als atomare Aussagen darstellen. Es bleibt die Frage, ob die spezifizierte Wissensbasis sich auf das aussagenlogische Fragment der Hornlogik beschränken läßt, da für dieses Fragment ein effizientes Beweisverfahren (Markierungsalgorithmus) existiert.

Laut Definition ist eine aussagenlogische Formel eine Hornformel, wenn sie in in KNF ist und jedes Disjunkt höchstens ein positives Literal enthält. Dass sich im Falle der durch die θ-Funktionen spezifizierten Wissensbasis die Formelmengen nicht auf Hornformeln beschränken soll mit einem Gegenbeispiel gezeigt werden. Wie in der Definition von degree erkennbar, muss bei der Berechnung von degree(φ) gezeigt werden, dass aus der Formelmenge der Wissensbasis die Formel φ, deren Grad der epistemischen Verankerung berechnet werden soll, aus dieser folgt. Aufgrund der unterschiedlichen Ebenen der epistemischen Verankerung muss dabei zunächst geprüft werden, ob φ aus den Formeln mit der höchsten epistemischen Verankerung folgerbar ist (siehe dazu Algorithmus 1 in Abschnitt 3.3 auf Seite 31). Nach Abschnitt 4.4 sind die Formeln mit der höchsten epistemischen Verankerung die durch θ_{ex} und θ_s erzeugten Formeln. Wenn nun degree(φ) berechnet werden soll, ist es notwendig zu prüfen, ob $\{\theta_{ex}(s)\} \cup \{\theta_s(s, s_{\text{alt}})\} \models \varphi$.

Sei nun s = Situation 3, Beispiel 1, s_{alt} = Situation 2, Beispiel 1. Die in s enthaltenen Matches sind: $m_1 = \{k_{25}, k_{21}, k_{23}, k_{24}\}$ und $m_2 = \{k_{26}, k_{22}, k_{23}, k_{24}\}$.

Wird der Degree von φ berechnet, so muss unter Berücksichtigung der epistemischen Verankerung geprüft werden, ob folgende Formel erfüllbar ist:

$$f = \quad (k_{25} \supset \neg k_{26}) \wedge ((\neg k_{25} \wedge \neg k_{26}) \supset \neg k_9) \wedge \neg\varphi$$

Wird f in KNF umgeformt, so resultiert dies in :

$$\begin{aligned} f &\equiv (\neg k_{25} \vee \neg k_{26}) \wedge (\neg(\neg k_{25} \wedge \neg k_{26}) \vee \neg k_9) \wedge \neg\varphi \\ &\equiv (\neg k_{25} \vee \neg k_{26}) \wedge (k_{25} \vee k_{26} \vee \neg k_9) \wedge \neg\varphi \end{aligned}$$

Für die Negation von f gilt:

$$\begin{aligned} \neg f &= \neg((\neg k_{25} \vee \neg k_{26}) \wedge (\neg(\neg k_{25} \wedge \neg k_{26}) \vee \neg k_9) \wedge \neg\varphi) \\ \neg f &\equiv (\neg(\neg k_{25} \vee \neg k_{26}) \vee \neg(\neg(\neg k_{25} \wedge \neg k_{26}) \vee \neg k_9) \vee \neg\neg\varphi) \\ &\equiv ((\neg\neg k_{25} \wedge \neg\neg k_{26}) \vee (\neg\neg(\neg k_{25} \wedge \neg k_{26}) \wedge \neg\neg k_9) \vee \varphi) \\ &\equiv ((k_{25} \wedge k_{26}) \vee (\neg k_{25} \wedge \neg k_{26} \wedge k_9) \vee \varphi) \\ &\equiv ((k_{25} \vee \neg k_{25}) \wedge (k_{25} \vee \neg k_{26}) \wedge (k_{25} \vee k_9) \wedge (k_{26} \vee \neg k_{25}) \wedge (k_{26} \vee \neg k_{26}) \wedge (k_{26} \vee k_9)) \vee \varphi \\ &\equiv ((k_{25} \vee \neg k_{26}) \wedge (k_{25} \vee k_9) \wedge (k_{26} \vee \neg k_{25}) \wedge (k_{26} \vee k_9)) \vee \varphi \\ &\equiv (k_{25} \vee \neg k_{26} \vee \varphi) \wedge (k_{25} \vee k_9 \vee \varphi) \wedge (k_{26} \vee \neg k_{25} \vee \varphi) \wedge (k_{26} \vee k_9 \vee \varphi) \end{aligned}$$

Offensichtlich sind f und $\neg f$ nach der Äquivalenzumformung in konjunktiver Normalform und offensichtlich enthält im Fall von f die Klausel $(k_{25} \vee k_{26} \vee \neg k_9)$ mehr als ein positives Literal, im Fall von $\neg f$ potentiell jede Klausel, aber auf jeden Fall die Klauseln $(k_{25} \vee k_9 \vee \varphi)$ und $(k_{26} \vee k_9 \vee \varphi)$. Damit sind weder f noch $\neg f$ Hornformeln, weshalb der Markierungsalgorithmus zur Bestimmung der Erfüllbarkeit von f nicht herangezogen werden kann. Auch ist es nicht möglich, mithilfe des Markierungsalgorithmus die Unerfüllbarkeit von $\neg f$ zu zeigen.

Die Verwendung eines Beweisers auf Basis des Markierungsalgorithmus ist daher nicht möglich, wenn nicht zusätzlich φ so eingeschränkt werden soll, dass durch φ f und $\neg f$ in Hornformeln umgeformt werden können. Nun wird es zum einen nicht trivial sein, eine Einschränkung für φ zu finden, die im allgemeinen Fall dazu führt, dass f oder $\neg f$ unter Einbeziehung des eingeschränkten φ in Hornformeln umgeformt werden können. Zum anderen bedeutet eine derartige Einschränkung, dass nicht mehr der Grad der epistemischen Verankerung für jedes φ berechnet werden kann, was dazu führt, dass der Agent nicht jede Anfrage an die Revisionsbasis stellen kann. Dadurch würde die Revisionsbasis weniger Nutzen aus dem Revisionsmodul ziehen können, was aber nicht sinnvoll ist.

Es ist damit gezeigt, dass als Sprache für die Wissensbasis mindestens die Aussagenlogik verwendet werden muss.

Kapitel 5

Evaluation der Wissensrevision

5.1 Reduzierung der Alternativen

„Verbessertes Backtracking“

Der Hauptnutzen von Verhaltensäquivalenz besteht darin, dass, nachdem ein Navigationsfehler erkannt wurde, alle zu dem Match, dessen Wahl den Navigationsfehler ausgelöst hat, verhaltensäquivalenten Matches ausgeschlossen werden können, ohne dass der Agent sie ablaufen muss. Wird der ganze Navigationsprozess von einem Startpunkt und mithilfe der Instruktion bis zu dem gewünschten Zielpunkt oder einem Punkt, an dem der Agent die Navigation erfolglos beendet, betrachtet, so kann man dies als einen Suchprozess auffassen. Aufgrund der statischen Welt und, da der Agent aufgrund der Existenz von atomaren Aktionen einen diskreten Zustandübergang besitzt, bietet es sich an, den Suchprozess als eine Variante der Graphensuche aufzufassen.

Nach [PMG 1998], Kapitel 4 kann der Graph im Rahmen einer Graphensuche einen Zustandsraum repräsentieren. Im Kontext des geometrischen Agenten stellen die Knoten des Graphen eine Situation dar, eine Kante die Wahl eines Matches durch den Agenten. Wenn der Graph als bidirektional angesehen wird, bedeutet dies, dass der Agent nach der Wahl eines Matches wieder in eine frühere Situation zurückzukehren kann. Da der Agent bisher nicht in der Lage ist, selbst wieder den Weg zurück zu gehen, spiegelt die Kante in eine alte Situation nur die gedachte Möglichkeit des Zurückkehrens wieder. Das Zurückkehren zu einem bereits gewähltem Knoten wird im Rahmen der Tiefensuche in der Literatur als Backtracking bezeichnet (siehe [PMG 1998], S. 125). Aus Sicht der Graphensuche führt der Agent ein Backtracking aus, wenn er zu einer vorangegangenen Situation zurückkehrt.

Im Rahmen der Navigation des Agenten gibt es einen wesentlichen Punkt zu beachten: Da der Agent über kein komplettes Wissen über seine Umgebung verfügt, ist es nicht möglich, in einer beliebigen Situation mehr als den nächsten Navgiationsschritt zu planen, ohne den Agenten eine tatsächliche Aktion ausführen zu lassen. Die Graphensuche des Agenten wird damit regelmäßig unterbrochen.

In der momentanen Implementation, die noch keine Wissensrevision nach dem in Kapitel 4 beschriebenem Schema benutzt, läßt sich das Verhalten des Agenten als unidirektionale Tiefensuche auffassen. Der Agent benutzt dazu bei der Auswahl des zu einer Instruktion passenden Matches das durch den Matcher verwendete Ähnlichkeitsmaß. Dies kann als eine heuristische Auswahl aufgefasst werden, die allerdings noch einen Nichtdeterminusmus enthält, da es gleich bewertete alternative Matches gibt. Da der Agent nicht in der Lage ist, in eine alte Situation zurückzukehren, ist der Graph unidirektional. Des weiteren erinnert der Agent nicht, dass es in einer älteren Situation unter Umständen Alternativen zu dem gewähltem Match gibt, wodurch er nicht in der Lage ist, zu erkennen, ob es sinnvoll ist, die Wahl des Matches, was ihn in diese Situation geführt hat, zu revidieren.

Ein hypothetisches Beispiel unter Verwendung des in Kapitel 4 eingefuhrten Wissensrevisionsmechanismus wird in Abbildung 5.1 dargestellt.

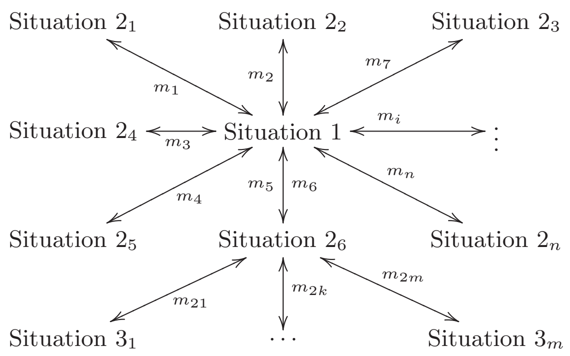


Abbildung 5.1: Navigation als Graphensuche im Problemraum. $m_1 \dots m_n$ sind die Matches passend zu Teilinstruktion 1, $i < n$. $m_{21} \dots m_{2m}$ sind die Matches passend zu der Teilinstruktion 2_6,

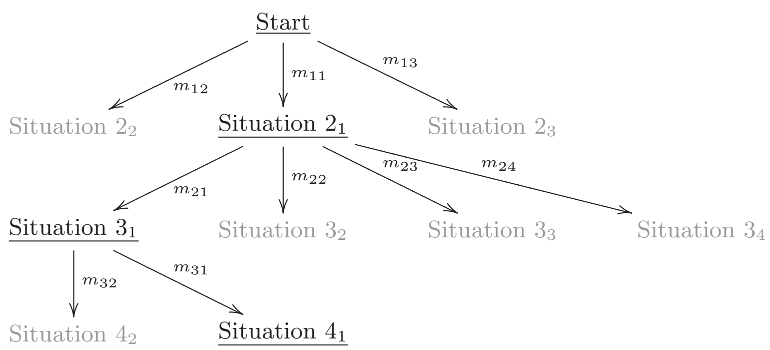


Abbildung 5.2: Ein möglicher Navigationsvorgang ohne Einsatz von Wissensrevision - basierend auf NBTest01 (siehe Seite 81). Nicht gewählte Matches werden vom Agenten nicht erinnert.

Die beiden Kernunterschiede zwischen der Navigation ohne und mit Einsatz von der in Kapitel 4 dargelegten Wissenrevision ist zum einen die Möglichkeit des Agenten, Alternativen in vorangegangenen Situationen zu erinnern und zum anderen die zusätzliche Ausschlussmöglichkeit im Fehlerfall von zu der falschen Wahl nicht verhaltensäquivalenten Matches. Angenommen, der Agent wäre in der Lage im Fehlerfall zu einer alten Situation zurückzukehren und eine beliebige Alternative zu wählen, so bestünde dennoch das Problem, dass er unter Umständen eine große Menge von unnötiger Laufarbeit durch die Wahl von Matches, die zum gleichen Ziel führen, zu bewältigen hat. Dies wird durch die Verwendung der in Kapitel 4 definierten Revisionsbasis vermieden, was aus Sicht der Graphensuche formuliert als eine Verbesserung des Backtracking auffassen läßt.

Beim Ausschluß von Matches einer Situation sind zwei Fälle zu unterscheiden: Zum einen ist es ein Maß für den Nutzen der Revision, wenn die Zahl der Ausschlüsse von verhaltensäquivalenten Matches groß ist. Dagegen wäre es schädlich, wenn Matches ausgeschlossen werden, deren Wahl zu dem oder einem der richtigen Zielorte führt. Der Ausschluss von Matches, die zum Ziel führen, kann bei der Verwendung von strenger Verhaltensäquivalenz nicht geschehen, da nur Matches streng verhaltensäquivalent sind, die exakt die gleiche Perzeption verursachen. Es bleibt daher zu untersuchen, in welchen Situationen und in welchem Ausmaß, also wie stark Verhaltensäquivalenz anteilig an der Gesamtzahl aller Matches einer Situation, auftritt.

Auftreten von Verhaltensäquivalenz

Um zu zeigen, dass die in Kapitel 4 vorgeschlagene Wissensbasis und ihre Einbindung in den geometrischen Agenten einen Nutzen im Sinne eines rationaleren Verhaltens des Agenten hat, wird an dieser Stelle zunächst an einigen Beispielen motiviert, unter welchen Umständen Verhaltensäquivalenz auftritt. Im Anhang ist das Beispiel 1 auf Seite 81 zu finden. In diesem Beispiel ist erkennbar, dass in keiner Situation eine Verhaltensäquivalenz zwischen verschiedenen Matches besteht. Das Beispiel 1 ist ein Protokoll des Navigierens des Agenten auf dem Weg von der Mensa nach Haus E. Das Umfeld, das der Agent im Laufe der Navigation bei diesem Beispiel zu sehen bekommt, ist von der Anzahl der Objekte her eingeschränkt. Insbesondere in den Situationen 2 und 3 sieht der Agent nur sehr wenige Wege. Daher existiert nur eine geringe Zahl von Alternativen aus der Perzeption, die der Matchingalgorithmus den Knoten der Instruktion zuordnen kann. Offensichtlich hat hier die Verwendung von Verhaltensäquivalenz bei der Revision keinen Vorteil.

Ein komplett anderes Bild ergibt die Situation 1 in Beispiel 2 auf Seite 83. In Beispiel 2 gibt es sieben Klassen von verhaltensäquivalenten Matches, die sich neben der Zielkoreferenz in zwei weiteren Koreferenzen unterscheidet. Auffallend ist dabei, dass zwischen vielen Äquivalenzklassen eine große Strukturähnlichkeit besteht. Beispielsweise bestehen die Äquivalenzklassen von k_{140} und k_{141} aus folgenden Einträgen:

Äquivalenzklasse zu k_{140}					Äquivalenzklasse zu k_{141}				
k_{140}	k_{101}	k_{136}	k_8	k_{137}	k_{141}	k_{101}	k_{136}	k_8	k_{137}
k_{140}	k_{101}	k_{136}	k_8	k_5	k_{141}	k_{101}	k_{136}	k_8	k_5
k_{140}	k_{101}	k_{136}	k_8	k_{138}	k_{141}	k_{101}	k_{136}	k_8	k_{138}
k_{140}	k_{102}	k_{136}	k_8	k_{137}	k_{141}	k_{102}	k_{136}	k_8	k_{137}
k_{140}	k_{102}	k_{136}	k_8	k_5	k_{141}	k_{102}	k_{136}	k_8	k_5
k_{140}	k_{102}	k_{136}	k_8	k_{138}	k_{141}	k_{102}	k_{136}	k_8	k_{138}

Wie erkennbar ist, existiert zu jedem Match aus der Äquivalenzklasse zu k_{140} ein analoges Match aus der Äquivalenzklasse zu k_{141}, das sich nur in der Zielkoreferenz unterscheidet. Dies kann mit Definition eines Matches und der Funktionsweise des Matchingalgorithmus erklärt werden. Ein Match selbst besteht aus einer Menge von Zuordnungen von Knoten der Instruktion zu Knoten der Perzeption. Prinzipiell kann jeder Knoten der Instruktion mit jedem Knoten der Perzeption koreferenziert werden. Allerdings bezieht der Matchingalgorithmus auch Taxonomien und die Struktur der Graphen bei der Berechnung der Koreferenzen mit ein, so dass nicht jede Kombinationsmöglichkeit von Instruktions- und Perzeptionsknoten in Frage kommt. Es ist zu erwarten, dass der Agent insbesondere dann, wenn die Perzeption eine große Vielzahl von Objekten und damit eine Vielzahl unterschiedlicher Zuordnungsmöglichkeiten liefert, auch mit einer großen Anzahl von ähnlichen Matches umgehen muss. Ein zweiter Faktor für die Anzahl von Matches ist die Größe des Instruktionsgraphen. Existieren dort viele Knoten, so kann es mehr Koreferenzen geben, solange die Perzeption ausreichende Wahlmöglichkeiten bietet. In Situation 1 aus Beispiel 2 existieren 42 verschiedene Matches und sieben unterschiedliche Äquivalenzklassen. Verwendet der Agent die in Kapitel 4 vorgeschlagenen Revisionsbasis, so wird der Explorationsaufwand auf den einer Situation mit sieben Matches und ohne Verwendung von Verhaltensäquivalenz reduziert. Prinzipiell gilt, dass eine große Anzahl von Perzeptionsknoten, die der Matchingalgorithmus als gute Kandidaten für einen bestimmten Instruktionsknoten ansieht, zu weiteren Permutationen führen. Die Zahl der Äquivalenzklassen entspricht im Falle der einfachen Verhaltensäquivalenz der Zahl der unterschiedlichen Koreferenzen des „Ziel“instruktionsknotens. Im Fall der strengen Verhaltensäquivalenz ist die Zahl der Äquivalenzklassen $\leq$ der Zahl der unterschiedlichen Koreferenzen des Zielinstruktionsknotens.

Es ist naheliegend aus den Beispielen 1 und 2 sowie der Definition eines Matches im Zusammenhang mit der Funktionsweise des Matchingalgorithmus folgende Aussagen zum Auftreten von Verhaltensäquivalenz abzuleiten :

1. Die Zahl der Äquivalenzklassen und die Größe der Äquivalenzklassen hängt von der Zahl der Permutationen der Matches ab, die der Matchingalgorithmus liefert. Eine Permutation der

„Ziel“koreferenz führt zu einer neuen Äquivalenzklasse. Die Permutationen aller anderen Koreferenzen vergrößert die Größe der Äquivalenzklassen.

2. Wird ein großer Bereich perzipiert, so besteht eine höhere Wahrscheinlichkeit für eine große Zahl von Permutationen.

3. Da der Agent zur Bewegung auf Wege angewiesen ist, führt besonders eine große Auswahl von Wegen bei der Ausführung der Befehle GO und FOLLOW zu einer großen Zahl von Äquivalenzklassen. Die Überrepräsentation von Bewegungsbefehlen kann in den im Anhang D Aktionsplänen eingesehen werden.

Verhaltensäquivalenz in verschiedenen Repräsentationsformen

In Abschnitt 4.3.3 wird die Frage aufgeworfen, ob ein Match als Konjunktion von Koreferenzen oder als atomare Aussage in der Revisionsbasis repräsentiert werden soll. Für den Fall, dass ein Match als atomare Aussage repräsentiert wird, ergibt sich als Folge, dass ein Adjustment nicht um die Negation einer Koreferenz, sondern um die Negation eines Matches ausgeführt werden muss. Wird in einem solchen Fall ein Match durch ein Adjustment um die Negation eines Matches ausgeschlossen, so wird das innerhalb der Striktordnung als nächstes in Frage kommende Match ein positiver Degree zugeordnet. Dies geschieht durch die von θ_m erstellten Einträge. Allerdings besteht das Problem, dass durch das Adjustment zwar ein einzelnes Match ausgeschlossen wird, nicht jedoch die anderen Vertreter der Äquivalenzklasse des ausgeschlossenen Matches. Dies bedeutet, dass das „verbesserte Backtracking“ nicht durch eine Repräsentation von Matches als atomare Aussagen umgesetzt werden kann. Um den Ausschluss der restlichen Vertreter der Äquivalenzklasse des ausgeschlossenen Matches zu erreichen ist es daher notwendig, in der Revisionsbasis zusätzliche Einträge zur Repräsentation der Verhaltensäquivalenz vorzunehmen.

5.2 Performanz

Komplexitätskriterien

In Abschnitt 4.3 wurde der strukturelle Aufbau der Revisionsbasis festgelegt und verschiedene Alternativen zu der gewählten Struktur vorgestellt. Auf Situation 1 in Beispiel 1 angewendet ergibt sich die auf Seite 81 dargestellte Revisionsbasis. Ist die Revisionsbasis mit den Einträgen aus Situation 1, Beispiel 1 gefüllt, so ist für die Berechnung des Degrees einer Formel φ, zunächst ohne Berücksichtigung der epistemischen Verankerung, nach dem in Kapitel 3.3 vorgestelltem Algorithmus zur Berechnung des Degrees mithilfe eines priorisiereten Tableau-Beweisers, diese Folgerbarkeit zu prüfen :

$$\theta(\text{Situation } 1, s_0) \vdash_{pt} \varphi$$

Die Berechnung der Folgerbarkeit ist auch analog dazu im allgemeinen Fall für zwei beliebige Situation s_1, s_2 zu prüfen Das initiale Tableau besteht nach dem Algorithmus zur Berechnung des Degrees aus einem einzelnen Zweig mit allen Elementen aus $\theta(\text{Situation } 1, \emptyset)$ als Einträgen sowie dem zusätzlichem Eintrag $\neg\varphi$. Anhand dieses initialen Tableaus läßt sich ablesen, wie umfangreich die komplette (strikte) Expansion aller Einträge ist. Ein Tableau wird während eines Tableaubeweises nicht notwendigerweise komplett strikt expandiert, allerdings ist die komplette strikte Expansion die Obergrenze des Gesamtaufwands.

Es gibt im wesentlichen zwei Maße zur Abschätzung des Aufwands eines Tableaubeweises: Zum einen die Höhe des komplett expandierten Tableaus, zum anderen der Verzweigungsfaktor des komplett expandierten Tableaus. Der Verzweigungsfaktor entspricht der Gesamtzahl der Zweige im komplett expandierten Tableau. Gegen die Verwendung der Höhe des Tableaus und des Verzweigungsfaktors sprechen aber verschiedene Argumente: zum einen beziehen beide weder die Priorisierung eines zur

Berechnung von degree optimierten Tableaubeweisers mit ein, zum anderen sind beide Maße allenfalls für eine sehr grobe Abschätzung des „Worst-Case“ zu verwenden. Die komplette Expansion eines Tableaus wird nur ein auf das äußerste Minimum an Funktionalität beschränkter Tableaubeweiser durchführen. Insbesondere bleibt dabei die Redundanz der zu expandierenden Einträge unberücksichtigt. In der durch θ aufgebauten Revisionsbasis ist allerdings insbesondere durch die von θ_m erzeugten Disjunktionen eine große Redundanz vorhanden. Im Falle der von θ_m erzeugten Disjunktionen müssten zur vollständigen Expansion innerhalb einer Situation s $|\text{matches}(s)|!$ Expansionen durchgeführt werden. Angewendet auf Situation 1 in Beispiel 3, wo 42 Matches zur Auswahl stehen, ergäbe sich bereits durch diese Disjunktionen ein Verzweigungsfaktor von ca. $1,405 \cdot 10^{51}$, der nahelegt, dass eine Berechnung des Degrees von φ ein in absehbarer (Rechen)Zeit unmögliches Vorhaben ist.

Um ein angemesseneres Bild des Rechenaufwands zu bekommen, soll statt dessen versucht werden zu skizzieren, wie unter Einhaltung der Priorisierung ein Tableaubeweis zur Berechnung von φ durchgeführt werden kann. Dabei sollen auch bestimmte Optimierungen am Tabelau- Beweiser geprüft werden, die es diesem erlauben, ein Tableau frühzeitig zu schließen. Insbesondere soll auf die Möglichkeiten des „Semantic Branching“ eingegangen werden. Als Maß für den Berechnungsaufwand soll die Zahl der erzeugten Zweige $Z : \mathbb{S} \to \mathbb{N}$, die mit der Berechnungszeit koreliert, sowie das Maximum der offenen Zweige $M : \mathbb{S} \to \mathbb{N}$, das mit dem maximalen Speicherbedarf koreliert, verwendet werden. Nicht berücksichtigt werden soll dabei zunächst die Möglichkeit, dass es zusätzliche Einträge aus Adjustments und anderen Situationen geben kann. Falls mehr als eine Situation in der Revisionsbasis enthalten ist, so ist die Zahl der Gesamtverzweigungen, die zur Berechnung von φ notwendig sind, das Produkt aus der Zahl der Verzweigungen, die zum Abschluss der Einträge aus jeder einzelnen Situation notwendig sind. Des weiteren soll davon ausgegangen werden, dass φ nicht aus der Revisionsbasis folgerbar ist.

Schema zur Analyse der Tableau-Komplexität

Nachfolgend soll unter Annahme einer Breitensuche jeweils für alle Formeln einer Prioritätsstufe eine komplette Expansion durchgeführt werden. Breitensuche bedeutet insbesondere, dass die Expansion einer Formel in alle offenen Zweige in denen diese Expansion möglich ist, gleichzeitig durchgeführt wird. Die Annahme einer Breitensuche entspricht nicht der Realität, da ein Tableaubeweiser normalerweise eine Tiefensuche durchführt. Allerdings gibt es das Grundproblem, dass ein Tableau mit nichtdeterministischen Expansionsregeln arbeitet (siehe auch Abschnitt 3.1). Deshalb besteht in einem Tableau für gewöhnlich deutlich mehr als eine Möglichkeit zum einen in der Frage der Anwendung der nächsten Expansionsregel, zum anderen in der Frage der Wahl des zu bearbeitenden Zweiges nach Expansion einer β-Formel. Da der Aufwand, jede mögliche Expansionsreihenfolge auszuprobieren, zu groß ist, soll deshalb durch die Wahl der Breitensuche das Problem des Nichtdeterminimus umgangen werden. Die Priorisierung hilft bereits, den Nichtdeterminismus einzuschränken, da sie es erlaubt, erst eine komplette Expansion aller Einträge einer Prioritätsstufe durchzuführen. Allerdings besitzen Formeln aus θ_s und θ_{ex} die gleiche Prioritätsstufe, weshalb zunächst die Expansion auf θ_{ex} beschränkt werden soll. Dies ist eine willkürliche Entscheidung. Allerdings ist die Zahl der offenen Zweige und die Breite des resultierenden Tableaus nach Abschluss der Breitensuche auf der höchsten Prioritätsstufe nicht von dieser Entscheidung abhängig.

Die Expansion der Formeln aus θ wird nachfolgend zunächst an einem einfachen Beispiel illustriert. Die Beispiele befinden sich in Anhang C ab Seite 91. Ausgehend von diesem Beispiel soll dann jeweils untersucht werden, welche Zweige geschlossen werden können. Die nicht geschlossenen Zweige sollen dann in Typen eingeteilt werden und die Typen in Hinblick auf die Formeln, die in Zweigen des jeweiligen Typs enthalten sind, untersucht werden. Die Typisierung der offenen Zweige soll eine Abstrahierung vom Beispiel weg hin zum allgemeinen Fall enthalten. Für den allgemeinen Fall soll darüber hinaus die Zahl der insgesamt geöffneten Zweige bestimmt werden. Aufgrund des exponentiellen Wachstums der Zweige ist auch mit einem exponentiellem Wachstum der Typen der nicht geschlossenen Zweige zu rechnen. Falls durch eine zu große Zahl von Typen die weitere Analyse des Tableaus eine dieser Arbeit angemessenen Größe überschreitet, so soll auf die Einbeziehung des bereits expandierten Tableaus verzichtet werden.

Nachfolgend wird bei der Expansion das Semantic Branching verwendet. Die folgenden Grafiken, die Expansionen anhand von Beispielen illustrieren sollen, zeigen den Endzustand nach kompletter Expansion aller wichtigen Teilformeln des Tableaus. Bei der Expansion werden Formeln, die bereits im Tableau vorhanden sind, nicht noch einmal eingefügt. Da es an einigen Stellen zur Verbesserung des Verständnis der Struktur des Tableaus wichtig sein kann, bestimmte Einträge auch mehrfach zuzulassen, so werden diese duplikaten Einträge mit grauer Schrift dargestellt.

Als Optimierung des Tableau-Beweises soll neben Semantic Branching auch der Nicht-Atomare Abschluss verwendet werden. Darüber hinaus sollen Disjunktionen, in denen bereits ein Disjunkt im Zweig b, in den die Disjunktion expandiert werden soll, existiert, nicht expandiert werden. Wenn eins der Disjunkte bereits in b existiert, so entsteht durch die Expansion der Disjunktion auch mindestens ein Zweig, der identisch mit b ist. Falls es möglich ist, b im Verlauf der Expansion abzuschließen, so können auch alle Zweige, die durch die Expansion der Disjunktion in b enstanden sind, abgeschlossen werden. Wenn b nicht abgeschlossen werden kann, muss generell nicht weiter expandiert werden. Das entfallen lassen der Expanseion einer Disjunktion mit einem Disjunkt, was bereits in einem Zweig b enthalten ist, wird nachfolgend als „Zweigersparnis“ bezeichnet. Zusätzlich sollen nicht-verzweigende Expansionen bei der Expansion bevorzugt werden, so dass davon ausgegangen werden kann, dass β-Formeln nur dann expandiert werden, wenn bereits alle α-Formeln auf dem aktuellen Prioritätslevel expandiert sind.

Es kann vorkommen, dass die Expansion von Formeln aus θ_{ex}, θ_m oder θ_s äquivalente, aber für einen Tableaubeweiser leichter zu verarbeitende Alternativen existieren. Tritt ein solcher Fall ein, so soll eine alternative Definition von θ_{ex}, θ_m oder θ_s in der leichter zu verarbeitenden Form aufgestellt werden. Die bestehenden Definitionen sollen aber weiterhin ihre Gültigkeit behalten, da sie eine für Menschen relativ einfach verständliche Formulierung besitzen. Die ergänzten Tableaus repräsentieren dagegen eine für Tableau-Beweiser optimierte technische Sicht.

Die Gesamtzahl der erzeugten Zweige einer Situation s soll mit der Funktion Z, $Z : \mathbb{S} \to \mathbb{N}$ bezeichnet werden. Z läßt sich in drei Anteile zerlegen:
Seien $Z^{'} : \mathbb{S} \to \mathbb{N}$, $Z^{''} : \mathbb{S} \to \mathbb{N}$ und $Z^{'''} : \mathbb{S} \to \mathbb{N}$ Funktionen, dann ist $Z(s) := Z^{'}(s) + Z^{''}(s) + Z^{'''}(s)$, wobei $Z^{'}(s)$ die Zahl der durch θ_{ex} erzeugten Zweige , $Z^{''}(s)$ die Zahl der durch θ_s erzeugten Zweige und $Z^{'''}$ die Zahl der durch θ_m erzeugten Zweige ist. Hierbei ist zu beachten, dass die Zahl der durch $Z^{'}, Z^{''}$ und $Z^{'''}$ erstellten Zweige davon abhängt, wieviele offene Zweige vor der Expansion von θ_{ex}, θ_s und θ_m im Tableau existieren. Damit hängt das Ergebnis der drei Teilfunktionen von Z von der Reihenfolge ab, in der θ_{ex}, θ_s und θ_m expandiert werden. Die Reihenfolge der Expansion ist, wie oben erwähnt, auf θ_{ex} gefolgt von θ_s gefolgt von θ_m festgelegt. Die Ergebnisse von $Z^{'}, Z^{''}$ und $Z^{'''}$ setzen daher zwingend diese Reihenfolge voraus.

Als grundlegende Feststellung gilt, dass das Tableau mit einem Zweig initialisiert wird, also mindestens ein Zweig erzeugt wird. Wird in einem Zweig eine verzweigende Expansion ausgeführt, so ist zu beachten, dass der Zweig, in dem die verzeigende Expansion ausgeführt wird, weiter bestehen bleibt. Wenn beispielsweise eine Disjunktion mit n Disjunkten ausgeführt wird, so werden nur n-1 neue Zweige erzeugt. Nicht verzweigende Expansionen erzeugen generell keine neuen Zweige.

Unter Umständen kann es möglich sein, dass aufgrund von notwendigen Abstraktionen Z nur geschätzt werden kann. Dabei soll der Aufwand überschätzt werden, weshalb die tatsächliche Zahl der erzeugten Zweige kleiner als Z ist.

Expansion der durch θ_{ex} und θ_s erzeugten Formeln

Aufgrund der in Abschnitt 4.4 eingeführten Einschränkungen zur epistemischen Verankerung müssen durch einen in Abschnitt 3.3 skizzierten priorisierten Tableaubeweiser die durch die Funktionen θ_{ex} und θ_{S} Einträge am frühsten expandiert werden. Die Einträge von θ_{ex} bestehen aus Implikationen mit je einem Literal im Antezedens und Konsequens. Eine Implikation ist eine β-Formel, die verzweigend expandiert werden muss. Die Komponenten der Expansion der Implikation sind nach [FITTING 1996], Seite 23, $\beta_1 = \neg X, \beta_2 = Y$. Es existieren in jeder Situation s $|\theta_{ex}(s)| = \frac{(EC(s)-1)\cdot EC(s)}{2}$ Implikationen. Wird die Expansion dieser Formeln vorgenommen, so ergeben sich insgesamt $2^{|\theta_{ex}(s)|}$ Zweige im resultierenden Tableau. Allerdings lassen sich alle Implikationen mit dem gleichen Antezedens zu einer Implikationen zusammenfassen, was zu einer alternativen, aber logisch zur ersten Definition äquivalenten, Formelmenge führt.

Definition 39 (Exklusionsfunktion (äquivalente Fassung)) *Sei $s = (\lambda, P, I, I^{'}, K)$ eine Situation, θ_k eine Übersetzungsfunktion für eine einzelne Koreferenz, $\{m_1, \ldots, m_l\} = \mathcal{NVE}_{unique}(s)$, $m_1 < \cdots < m_l$, wobei $<$ die strike Totalordnung für Matches ist und Λ_1 eine Zahl aus $]0,1[$, die eine epistemische Verankerung repräsentiert. Wird θ_{ex} so definiert, dass θ_{ex} zusammgengefasste Implikationen erstellt, so ergibt sich*
$\theta_{ex} : \mathbb{S} \rightarrow 2^{\mathcal{L}\times]0,1[}$,

$$\theta_{ex}(s) := \left\{ \left(\theta_k(\tau(m_i)) \supset \bigwedge_{j=i+1}^{l} \neg\theta_k(\tau(m_j)), \Lambda_1 \right) \middle| \, i \in \{1, \ldots, l-1\} \right\}$$

die Exklusionsfunktion.

Diese alternative Definition von θ_{ex} erzeugt insgesamt $EC(s)$ Implikationen, so dass bei einer Verwendung der alternativen Defintion nur unter Einsatz eines nicht optimierten Tableaubeweisers $2^{EC(s)}$ Zweige erzeugt werden, da die generalisierte Konjunktion innerhalb der Konsequens nicht verzweigend expandiert werden muss. Angenommen, $\{m_1, m_2, m_3, m_4\}$ $= \mathcal{NVE}_{\text{unique}}(s)$ und $\tau(m_1) = k_1, \tau(m_2) = k_2, \tau(m_3) = k_3, \tau(m_4) = k_4$. Dann hat das vollständig expandierte Tableau unter Verwendung von Semantic Branching (siehe Abschnitt 3.1 auf Seite 21) von $\theta_{ex}(s)$ die in Abbildung C.1 dargestellte Struktur. In dem Tableau stammen alle doppelten Negationen und damit auch alle positiven Literale aus dem Semantic Branching, während alle negativen Literale auf den syntaktischen Anteilen der Expansion beruhen. An dem in Abbildung C.1 dargestelltem Tableau ist zu erkennen, dass viele der durch die Expansion der Formeln von θ_{ex} erzeugten Zweige direkt geschlossen werden können, was ohne Verwendung von Semantic Branching nicht möglich wäre. Die nicht geschlossenen Zweige besitzen das Merkmal, dass für alle Matches m_i aus $EC(s)$ je ein offener Zweig, der nicht $\neg\tau(m_i)$ enthält, existiert.

In Abbildung C.2 wird die Expansion der durch θ_{ex} erzeugten Formeln unter Verwendung der Zweigersparnis dargestellt. Man kann erkennen, dass nur diejenigen Zweige erstellt werden, die bei der Expansion ohne Verwendung der Zweigersparnis nicht geschlossen werden konnten. Damit ist gezeigt, dass zum einen Semantic Branching die Zahl der offenen Zweige reduziert, aber zumindest in diesem Fall die Zweigersparnis bereits die Zahl der erstellten Zweige reduziert.

Das Zwischenergebnis unter Einbeziehung der Expansion der Formeln von θ_{ex} ist, dass

$$Z^{'}(s) = 1 + (EC(s) - 1) = EC(s)$$

beträgt, wobei zum einen der im initialen Tableau erzeugte Zweig gezählt wird, und zum anderen durch die Expansion von θ_{ex} EC(s)-1 neue Zweige erzeugt werden. Insgesamt exisitieren nach der Expansion von θ_{ex} EC(s) offene Zweige.

Da die durch θ_s erzeugte Implikation die gleiche epistemische Verankerung wie die durch θ_{ex} erzeugten Formeln besitzt, muss diese Implikation nun in jeden offenen Zweig expandiert werden. Angenommen, die s vorangegangene Situation ist s_0. Dann ist die durch $\theta_s(s)$ erzeugte Implikation $(\wedge(\neg k_1, \neg k_2, \neg k_3, \neg k_4) \supset \text{stop})$. Der Übersichtlichkeit wegen ist in Abbildung C.3 exemplarisch die Expansion der durch θ_s erzeugten Implikation in einen der Zweige aus Abbildung C.1 durchgeführt. Dabei läßt sich erkennen, dass die Expansion von $(\wedge(\neg k_1, \neg k_2, \neg k_3, \neg k_4) \supset \text{stop})$ mehrere Schritte erfordert. Der Hauptoperator ist die Implikation, die verzweigend expandiert werden muss. Das Antezedens der Implikation ist eine Konjunktion, die allerdings aufgrund der Expansionsregeln negiert werden muss und somit ebenfalls verzweigend expandiert werden sollte. Es ist sinnvoll, die Implikation direkt in eine verallgemeinerte Disjunktion umzuformen, um so Ableitungsschritte zu sparen.

Definition 40 (Situationsübergreifende Verankerungsfunktion (äquivalente Fassung)) *Sei* $s = (\lambda, A, P, I, I', K)$ *eine Situation,* $s_{alt} = (\lambda_{alt}, A_{alt}, P_{alt}, I_{alt}, I'_{alt}, K_{alt})$ *die Situation des letzten Entscheidungspunkts vor s,* $\{m_1 \dots m_n\} = \mathcal{NVE}_{unique}(s)$*,* $stop \in \mathcal{L}_{atom}$ *sowie* Λ_2 *eine Zahl aus* $]0,1[$*, die eine epistemische Verankerung repräsentiert. Dann ist* $\theta_s : \mathbb{S} \times \mathbb{S} \rightarrow \mathcal{L} \times]0,1[$*,*

$$\theta_s(s, s_{alt}) := \begin{cases} \left(\left(\bigvee_{i=1}^{n} \theta_k(\tau(m_i))\right) \vee \neg\theta_k(\tau(\gamma(s_{alt}))), \Lambda_2\right) & \textit{wenn } s_{alt} \neq s_0 \\ \left(\left(\bigvee_{i=1}^{n} \theta_k(\tau(m_i))\right) \vee stop, \Lambda_2\right) & \textit{andernfalls} \end{cases}$$

Die Expansion der durch die alternative Definition von θ_s erzeugten Formel $\vee(k_1, k_2, k_3, k_4, \text{stop})$ in den linken Zweig aus Abbildung C.2 ist in Abbildung C.4 dargestellt. Erkennbar ist, dass insgesamt $|EC(s)|$ neue Zweige erzeugt werden, von denen aber nur zwei nicht sofort geschlossen werden können. Der eine Zweig enthält das Literal stop (im allgemeinen Fall eine negierte Koreferenz, die ansonsten innerhalb dieser Situation nicht von Bedeutung ist) und die Negation der Zielkoreferenz aller Matches aus $\mathcal{NVE}_{\text{unique}}$. Der andere Zweig enthält die Negation von $EC(s) - 1$ Zielkoreferenzen sowie die Zielkoreferenz der verbleibenden Äquivalenzklasse.

Die in Abbildung C.4 dargestellte Expansion ist ein Sonderfall des linken Zweigs aus Abbildung C.2, der sich von den übrigen Zweigen in der Hinsicht unterscheidet, dass in ihm kein positives Literal, das die Korerefenz des Ziels der Aktion repräsentiert, enthalten ist. In allen anderen Zweigen, in denen ein solches positives Literal existiert, muss aufgrund der Zweigersparnisregel gar keine Expansion ausgeführt werden. Die Anwendung der Zweigersparnis ist nur unter Verwendung der alternativen Definition von θ_s möglich.

Die nach der Expansion von θ_s spezifizierten Formel verbleibenden Zweige lassen sich wie folgt charakterisieren:

(BC1) Es gibt genau $EC(s) + 1$ offene Zweige, wobei insgesamt durch die Expansion von θ_s EC(s) neue Zweige erzeugt werden.

(BC2) Die nicht geschlossenen Zweige lassen sich in zwei Typen einteilen: Zweige vom Typ br_{stop} und Zweige vom Typ $br_i, 0 < i \leq EC(s)$.

(BC3) Es existiert exakt ein offener Zweig, der aus einem Literal besteht, das entweder stop oder die Negation der Koreferenz des Ziels einer anderen Situation repräsentiert, sowie der Negation der Zielkoreferenzen aller Matches aus $\mathcal{NVE}_{\text{unique}}(s)$. Dieser Zweig soll nachfolgend mit br_{stop} bezeichnet werden. Es gilt:

$$\left\{\neg\theta_k(\tau(m)) \,\middle|\, m \in \mathcal{NVE}_{\text{unique}}(s)\right\} \subseteq br_{stop} \wedge stop \in br_{stop}$$

(BC4) Es existieren genau EC(s) offene Zweige, die dadurch gekennzeichnet sind, dass sie genau ein positives Literal enthalten, welches das Ziel der Aktion eines der Matches aus $\mathcal{NVE}_{\text{unique}}$ repräsentiert, sowie EC(s)-1 negative Literale, welche den Ausschluss der Ziele der verbleibenden Matches aus $\mathcal{NVE}_{\text{unique}}$ repräsentieren. Man kann die Zweige daher anhand ihres positiven Literals charakterisieren. Nachfolgend soll mit $br_i, 0 < i \leq EC(s)$ der Zweig bezeichnet werden, der das nach der Totalordnung der Matches her das Ziel des i. Match aus $\mathcal{NVE}_{\text{unique}}$ als positives Literal enthält:

Seien $\{m_1, \dots, m_l\} = \mathcal{NVE}_{\text{unique}}(s)$ und $m_1 < \dots < m_l$ nach der Totalordnung der Matches. Die Funktion $b : \mathcal{M} \rightarrow 2^{\mathcal{L}}$ ordnet dabei jedem Match eine Menge eine Menge von Formeln zu. Dabei haben die dem Match $m_i \in \mathcal{NVE}_{\text{unique}}(s)$ zugeordneten Mengen unter anderem folgenden Inhalt:

$$\left(\{\theta_k(\tau(m_i))\} \cup \left\{\neg\theta_k(\tau(m)) \,\middle|\, m \in \left(\mathcal{NVE}_{\text{unique}}(s) \setminus \{m_i\}\right)\right\}\right) \subseteq b(m_i)$$

Da die Zahl der erzeugten neuen Zweige $EC(s)$ beträgt, beträgt auch θ_s

$$Z^{''}(s) = EC(s)$$

Es sind nach Abschluss der Expansion auf dem höchsten Prioritätslevel EC(s)+1 Zweige offen.

Expansion der durch θ_m erzeugten Disjunktionen

Die in Abschnitt 4.3.3 eingeführte rekursive Variante von $\theta_m(s, i), 0 < 1 \leq |\text{matches}(s)|$, die einen direkten Zugriff auf ein Disjunkt mit i Disjunkten erlaubt, soll auch an dieser Stelle verwendet werden. Aus der Definition der Constraints zur epistemischen Verankerung geht hervor, dass die Disjunktion nach der Zahl ihrer Disjunkte geordnet in absteigender Reihenfolge expandiert werden müssen, womit die Disjunktion mit den meisten Disjunkten zuerst expandiert werden muss. Disjunktionen müssen verzweigend expandiert werden. Die einzelnen Disjunkte enthalten Konjunktionen von Literalen, die einzelne Matches repräsentieren. Jede der Konjunktionen enthält auch genau eine der Zielkoreferenzen der Matches aus $\mathcal{NVE}_{\text{unique}}(s)$, die in den existierenden Zweigen, die aus der Expansion der durch θ_s und θ_{ex} erzeugten Formeln stammen, teilweise als positives, teilweise als negatives Literal bereits enthalten sind. In den oben verwendeten Beispiel wurde bisher nicht die Menge matches(s) definiert. Es der Einfachheit angenommen, dass matches$(s) = \mathcal{NVE}_{\text{unique}}(s)$.

Es soll zunächst die Disjunkion $\theta_m(s, 4) = \vee(\wedge(k_1, \dots), \wedge(k_2, \dots), \wedge(k_3, \dots), \wedge(k_4, \dots))$ in einen Zweig vom Typ br_{stop} expandiert werden, wobei nur die Einträge, die nach BC3 erfasst werden, in den Zweig aufgenommen werden. Das aus der Expansion resultierende Tableau findet sich in Abbildung C.5. Auch mit der Beschränkung auf durch BC3 erfasste Einträge lassen sich alle durch die Expansion der durch $\theta_m(s, 6)$ neu entstandenen Zweige schließen. Der Abschluss der Zweige ist deshalb möglich, weil br_{stop} für jedes Match $m \in$ matches(s) die Negation der Zielkoreferenz $\neg\theta_k(\tau(m))$ enthält. Durch die Expansion der Disjunktion $\theta_m(s, 6)$ entstehen 4 neue Zweige, die aber je genau für ein Match $m \in$ matches(s) das positive Literal $\theta_k(\tau(m))$ enthalten, wodurch in jedem Zweig ein Abschluss bereits an dieser Stelle der Expansion möglich ist. Eine Expansion der Disjunktion $\theta_m(s, 3), \theta_m(s, 2), \theta_m(s, 1)$ führt ebenfalls dazu, dass jeweils alle neu entstandenen Zweige geschlossen werden, da in jedem neu entstandenen Zweig ein $\theta_k(\tau(m))$ für ein beliebigs $m \in$ matches(s) expandiert wird und in br_{stop} bereits ein dazu passendes negatives Literal existiert, da nach BC3 $\neg\theta_k(\tau(m_i)) \in br_{stop}$ für alle $m_i \in \mathcal{NVE}_{\text{unique}}(s)$ und

$$\underset{(m_i \in \text{matches}(s))}{\forall} \; \underset{\left(m_j \in \mathcal{NVE}_{\text{unique}}(s)\right)}{\exists} \tau(m_i) = \tau(m_j)$$

(Die Zielkoreferenz zu jedem Match einer Situation s ist in einem der Matches aus $\mathcal{NVE}_{\text{unique}}(s)$ enthalten). Es werden bei der Expansion der Disjunktion $\theta_m(s, |\text{matches}(s)|)$ in den einzelnen Zweig vom Typ br_{stop} insgesamt $|\text{matches}(s)| - 1$ neue Zweige erzeugt, aber alle im nächsten Schritt wieder geschlossen.

Da der Abschluss der durch die Expansion von durch $\theta_m(s, |\text{matches}(s)|)$ erzeugten Formeln in einen Zweig vom Typ br_{stop} mit einer für beliebige Situationen passenden Erklärung begründet wurde, werden auch im allgemeinen Fall für eine beliebige Situation s der einzelne Zweig vom Typ br_{stop} nach der Expansion der durch $\theta_m s, |\text{matches}(s)|)$ erzeugten Disjunktionen abgeschlossen.

Nun muss die Disjunktion $\theta_m(s, 4) = \vee(\wedge(k_1, \dots), \wedge(k_2, \dots), \wedge(k_3, \dots), \wedge(k_4, \dots))$ in einen Zweig vom Typ br_i expandiert. Aus dem Beispiel soll der Zweig br_4 gewählt werden, wobei nur Einträge, die durch BC4 erfasst werden, in den Zweig aufgenommen werden. Es soll an dieser Stelle auch der Fall Äquivalenzklassen mit einer Kardinalität > 1 untersucht werden. Deshalb soll das bisher verwendete Beispiel leicht verändert werden: matches$(s) = \{m_1, m_2, m_3, m_4, m_5, m_6\}$ mit $m_1 < \cdots < m_6$ nach der Totalordnung der Matches und $ev(m_1, m_5)$ und $ev(m_2, m_3)$. Die bisher verwendeten Zielkoreferenzen verteilen sich wie folgt auf die Matches: $\tau(m_1) = \tau(m_5) = k_1, \tau(m_2) = \tau(m_3) =$

$k_2, \tau(m_4) = k_3, \tau(m_6) = k_4$. θ_m liefert in diesem Fall $\vee(\wedge(k_1,\dots), \wedge(k_2,\dots), \wedge(k_2,\dots), \wedge(k_3,\dots), \wedge(k_1,\dots), \wedge(k_4,\dots))$. Dabei ist zu beachten, dass die scheinbar doppelten Einträge $\wedge(k_1,\dots)$ und $\wedge(k_2,\dots)$ nicht syntaktisch oder semantisch identisch anzusehen sind, da sich auch verhaltensäquivalente Matches in mindestens einer Koreferenz unterscheiden. Die Koreferenzen, die nicht das Ziel der Aktion repräsentieren, sind an dieser Stelle nicht von Bedeutung und werden deshalb durch ... abgekürzt.

Das durch die Expansion von $\theta_m(s, 6)$ in den Zweig br_4 entstandene Tableau ist in Abbildung C.6 dargestellt.

An dem expandierten Tableau ist erkennbar, dass nicht alle erzeugten Zweige geschlossen werden können. Der Zweig, der nicht geschlossen werden kann, ist dadurch charakterisiert, dass in ihn das Disjunkt expandiert wurde, welches das Match m_4 repräsentiert, dessen Zielkoreferenz in dem Zweig br_4 nicht ausgeschlossen ist. Wird die Disjunktion $\theta_m(s, 6)$ in den Zweig br_1 expandiert, so ergibt sich das in Abbildung C.7 dargestellte Tableau. Hier ist es im Gegensatz zum ersten Beispiel nicht möglich, zwei Zweige abzuschließen. Die beiden nicht abschließbaren Zweige enstanden aus Disjunkten, die verhaltensäquivalente Matches repräsentieren. Es entsteht ein ähnliches, aber im Detail anderes Tableau für jeden Zweig $br_i, 0 < i \le EC(s)$ durch die Expansion von $\theta_m(s, 6)$.

Die nicht abgeschlossenen Zweige entstammen jeweils aus einem Zweig vom Typ $br_i, 0 < i \le EC(s)$ und übernehmen dadurch auch alle Einträge und Eigenschaften von diesen Zweigen. Zudem ist in ihnen jeweils eine Konjunktion enthalten, die ein Match $m_k \in \text{matches}(s)$ repräsentiert.

Nachfolgend soll der offene Zweig, der aus der Expansion des Matches $m_k \in \text{matches}(s), 0 < k \le |\text{matches}(s)|$ entstanden ist, mit brm_k bezeichnet werden. In Abbildung C.6 ist der offene Zweig nach dieser Notation brm_6, in Abbildung C.7 dagegen sind es brm_1 und brm_5. Für jedes Match $m_k \in \text{matches}(s), 0 < k \le |\text{matches}(s)|$ existiert mindestens ein Zweig vom Typ brm_k.

Zweige vom Typ brm_k haben folgende Eigenschaften:

(BCM1) Da jeder Zweig vom Typ $brm_k, 0 < k \le |\text{matches}(s)|$ aus einem Zweig vom Typ $br_i, 0 < i \le EC(s)$ hervorgegangen ist, übernimmt auch jeder Zweig vom Typ brm_k die Eigenschaften desjenigen Zweigs br_i so dass $ev(m_i, m_k)$.

(BCM2) Jeder Zweig vom Typ $brm_k, 0 < k \le |\text{matches}(s)|$ enthält eine Konjunktion, die das Match $m_k \in |\text{matches}(s)|$ repräsentiert:

$$\theta_\mu(m_k) \in brm_k$$

Wenn brm_k aus dem Zweig $br_i, 0 < i \le EC(s)$ hervorgegangen ist, so ist das Match m_k verhaltensäquivalent zu dem Match m_i, da ansonsten das Match m_k eine Koreferenz besitzt, die in Zweig br_i aufgrund der Eigenschaft (BC4) bereits ausgeschlossen ist (als negatives Literal enthalten ist).

(BCM3) Jeder Zweig $brm_k, 0 < k \le |\text{matches}(s)|$ enthält aufgrund des Semantic Branching die Negation aller nach der Totalordnung $<$ her besseren Matches:

$$\{\neg\theta_\mu(m_i)\,|m_i \in \text{matches}(s) \wedge m_i < m_k\} \subseteq brm_k$$

Da ein Match als eine Konjunktion von Koreferenzen repräsentiert wird, muss die Negation eines Matches verzweigend expandiert werden.

Ein Match ist eine Sammlung von Koreferenzen. Da vom konkreten Aufbau der Matches abstrahiert werden soll, kann an dieser Stelle nur rekapituliert werden, dass verhaltensäquivalente Matches mindestens eine gemeinsame Koreferenz besitzen (die Koreferenz des Ziels der Aktion), während im Fall von nicht verhaltensäquivalenten Matches dies nicht der Fall ist. Die in (BCM3) beschriebenen negierten Konjunktionen in einem Zweig vom Typ brm_k enthalten, wenn sie nicht verhaltensäquivalent zu m_k sind aufgrund der Eigenschaft (BC4) eines der bereits im Zweig vorkommenden negativen Literale, womit die Zweigersparnisregel angwendet werden kann. Aus diesem Grund müssen in einem Zweig vom Typ brm_k die negierten Konjunktionen, die in Eigenschaft (BCM3) beschrieben sind, nur dann expandiert werden, wenn sie ein zu Match m_k verhaltensäquivalentes Match repräsentieren. Die Zahl der negierten Konjunktionen pro offenem Zweig vom Typ brm_k hängt davon ab, wie groß die die Äquivalenzklasse von m_k ist. Wenn m_k das einzige Match der Äquivalenzklasse ist, dann existieren in brm_k keine negierten Konjunktionen (siehe brm_6 in C.6)). Im allgemeinen Fall existiert für jedes Match m_j einer Äquivalenzklasse ein offener Zweig. Die Zahl der negierten Konjunktionen pro Zweig hängt von der Stellung des Matches in der Totalordnung aller Matches ab - im Zweig, der die Konjunktion enthält, die das beste Match der Äquivalenzklasse enthält existiert keine, während im Zweig, in dem das schlechteste Match repräsentiert ist, genau $|\mathcal{VE}(m_j)|$. Die Details der totalen Ordnung der Matches soll nicht Teil der Performanzanalyse sein, weshalb zukünftig davon abstrahiert wird. Da die Zahl der Koreferenzen pro Match im Fall von gleich gewerteten Matches mit hoher Wahrscheinlichkeit gleich ist, ist auch die Zahl der Konjunkte der negierten Konjunktionen identisch. Die Zahl der Koreferenzen pro Match soll nachfolgend mit b bezeichnet werden Sei $m_i \in \text{matches}(s)$. Dann gibt es pro Äquivalenzklasse $ec \in \mathcal{VE}(s)$ maximal

$$\sum_{i=0}^{|ec|-1} b^i$$

offene Zweige nach Expansion der negierten Konjunktionen. Diese Zahl ist eine geometrische Reihe. Die Summenformel für geometrische Reihen hat im allgemeinen Fall nach [BR99] (Seite 18, Def. (1.51b)) die Form

$$\sum_{i=0}^{n} q^i = \frac{q^{n+1}-1}{q-1}$$

Angewendet auf die vorliegende geometrische Reihe gibt es pro Äquivalenzklasse $ec \in \mathcal{VE}(s)$ daher maximal

$$\frac{b^{|ec|}-1}{b-1}$$

offene Zweige nach Expansion der negierten Konjunktionen. Um die Gesamtzahl der offenen Zweige nach der Expansion der negierten Konjunktionen für alle (Verhaltens)Äquivalenzklassen zu berechnen, muss nun die oben genannte Summe für alle Äquivalenzklassen berechnet werden: Sei $\mathcal{VE}(s) = \{ec_1, \ldots, ec_n\}$. Dann ergeben sich insgesamt maximal

$$\sum_{k=1}^{n} \frac{b^{|ec_k|}-1}{b-1}$$

offene Zweige. Für die Berechnung von Z''' ist neben der Zahl der offenen Zweige die Zahl der neu erzeugten Zweige von Bedeutung. Zunächst wird in jeden Zweig vom Typ $br_i, 0 < i < EC(s)$ die Disjunktion $\theta_m(s, |\text{matches}(s)|)$ expandiert, was insgesamt $EC(s) \cdot (|\text{matches}(s)| - 1)$ neue Zweige erzeugt. In diesen Zweigen existieren nun abhängig von der Größe der Äquivalenzklassen verschiedene Disjunktionen mit jeweils b Disjunkten die expandiert werden müssen. Jede Expansion einer solchen Disjunktion erzeugt b-1 neue Zweige. Es gilt daher, dass die Zahl der neu erzeugten Zweige folgende Zahl ist:

$$\begin{cases} \displaystyle\sum_{k=1}^{n} \frac{(b-1)^{|ec_k|}-1}{b-2} = \frac{1}{b-2}\sum_{k=1}^{n}\Big((b-1)^{|ec_k|}-1\Big) & \text{falls } b > 2 \\ \displaystyle\sum_{k=1}^{n}\sum_{i=1}^{|ec_k|} i = \sum_{k=1}^{n} \frac{|ec_k|\cdot(|ec_k|+1)}{2} = \sum_{k=1}^{n}\frac{|ec_k|^2+|ec_k|}{2} = \frac{1}{2}\cdot\sum_{k=1}^{n}\big(|ec_k|^2+|ec_k|\big) & \text{falls } b = 2 \\ 0 & \text{andernfalls} \end{cases}$$

Zur Umformung im Fall von $b = 2$ wird die Summe der ersten $|ec_k|$ Zahlen $\sum_{i=1}^{|ec_k|} i$ durch die Anwendung der Gaußschen Summenformel (siehe [BR99], S. 18 Def. (1.52)) zusammengefasst. Um nachfolgend die Referenz auf diese Zahl zu erleichtern, soll die Funktion $Z'''_{\mathcal{VE}}$ eingeführt werden: Sei

$\mathcal{VE}(s) = \{ec_1, \ldots, ec_n\}$. Dann existieren nach der Expansion von $\theta_m(s, |\text{matches}(s)|)$ verschiedene negierte Konjunktionen, die expandiert und nicht geschlossen werden können. Die Zahl der durch die Expansion dieser negierten Konjunktionen neu erzeugten Zweige soll durch die Funktion $Z'''_{\mathcal{VE}} : \mathbb{S} \to \mathcal{N}$ mit

$$Z'''_{\mathcal{VE}}(s) := \begin{cases} \frac{1}{b-2} \sum_{k=1}^{n} \left((b-1)^{|ec_k|} - 1\right) & \text{falls } b > 2 \\ \frac{1}{2} \cdot \sum_{k=1}^{n} \left(|ec_k|^2 + |ec_k|\right) & \text{falls } b = 2 \\ 0 & \text{andernfalls} \end{cases}$$

bezeichnet werden.

An dieser Stelle ist anzumerken, dass es möglich ist, noch detailierter die negierten Konjunktionen zu untersuchen, da dadurch unter Umständen weitere der oben gezählten Zweige geschlossen werden könnten. Diese Untersuchung erfordert jedoch Annahmen über die Struktur der Matches, insbesondere in Hinblick auf mögliche identische Komponenten zu treffen. Eine derart detaillierte Untersuchung soll aber nicht vorgenommen werden, zum einen weil ansonsten Z dadurch von einer zu großen Zahl von Parametern abhängig gemacht wird, zum anderen soll der Rahmen dieser Arbeit nicht gesprengt werden.

Es ist nun bekannt, wie sich die Expansion der Disjunktion $\theta_m(s, |\text{matches}(s)|)$ auf das Tableau auswirkt. Die Expansion der restlichen Disjunktionen $\theta_m(s, i), 0 < i < |\text{matches}(s)|$ ist aufgrund der redundanten Struktur der Disjunktionen erheblich weniger umfangreich. Die Disjunktion $\theta_m(s, |\text{matches}(s)| - 1)$ enthält die ersten $|\text{matches}(s)| - 1$ Disjunkte der Disjunktion $\theta_m(s, |\text{matches}(s)|)$. Damit müssen die ersten $|\text{matches}(s)| - 1$ Zweige aufgrund der Zweigersparnisregel nicht weiter expandiert werden, da sie eines der Disjunkte von $\theta_m(s, |\text{matches}(s)| - 1)$ bereits enthalten. Eine Ausnahme bildet das nach der Totalordnung der Matches her schlechteste Match, was nicht in der nachfolgenden Disjunktion enthalten ist. Die Expansion dafür ist für unser Beispiel in Abbildung C.8 dargestellt. Es ist aufgrund der Abstraktion nicht bekannt, wieviele Zweige vom Typ $brm_{|\text{matches}(s)|}$ existieren, weshalb hier eine Überschätzung des Aufwands stattfinden soll. Sei $ecm \in \mathcal{VE}(s)$ eine oder (falls eine solche existiert) diejenige Äquivalenzklasse mit der größten Kardinalität. Dann existieren maximal $\sum_{i=1}^{|ecm|} (b^{i-1})$ Zweige vom Typ $brm_{|\text{matches}(s)|}$, weshalb durch die Expansion von $\theta_m(s, |\text{matches}(s)| - 1)$ maximal

$$\left(\sum_{i=1}^{|ecm|} (b^{i-1}) \right) \cdot |\text{matches}(s) - 1|$$

geöffnet werden. Alle neu entstandenen Zweige können geschlossen werden, da aufgrund der Eigenschaft (BCM3) $brm_{|\text{matches}(s)|}$ bereits die Negation der neu hinzugekommenen Konjunktionen enthält.

Die Argumentation, die für die Expansion der Disjunktion $\theta_m(s, |\text{matches}(s)| - 1)$ verwendete wurde, um zu zeigen, in den meisten Zweigen keine Expansion notwendig ist, und dass die durch die Expansion in den Zweig $brm_{|\text{matches}(s)|}$ des schlechtesten Matches entstandenen Zweige sofort geschlossen werden können, läßt sich verallgemeinern: Sei $0 < i < \text{matches}(s)$ und $\theta_m(s, i)$ die nach der Priorisierung her höchstehende Formel zur Expansion und $ecm \in \mathcal{VE}(s)$ eine oder (falls eine solche existiert) diejenige Äquivalenzklasse mit der größten Kardinalität. Dann werden bei der Expansion von $\theta_m(s, i)$ maximal

$$\left(\sum_{k=1}^{|ecm|} b^{k-1} \right) \cdot (i-1) = \left(\sum_{k=0}^{|ecm|-1} b^{k} \right) \cdot (i-1) = \frac{b^{|ecm|} - 1}{b-1} \cdot (i-1)$$

neue Zweige geöffnet. Die neu eröffneten Zweige können noch auf der aktuellen Prioritätsstufe wieder geschlossen werden.

Es ist anzumerken, dass es prinzipiell möglich ist, die Zahl der neu entstandenen Zweige genauer als mit $\frac{b^{|ecm|}-1}{b-1} \cdot (i-1)$ zu berechnen. Die genauere Berechnung setzt jedoch Kenntnis der Totalordnung der Matches voraus. Mithilfe der Totalordnung der Matches ist es möglich zu berechnen, wie viele Matches zu dem in Zweig brm_i repräsentierten Match m_i verhaltensäquvalent sind, und wieviele Matches aus $\mathcal{VE}(m_i)$ nach der Totalordnung der Matches her besser sind. Von der Zahl der besseren verhaltensäquivalenten Matches hängt die Zahl der negierten Konjunktionen, auf die nicht die Zweigersparnisregel angewendet werden kann, und damit die Zahl der Zweige vom Typ brm_i ab. Die Aufnahme der Totalordnung der Matches als Parameter für die Komplexitätsberechnung erscheint aber nicht sinnvoll, da diese Totalordnung von außerhalb des Revisionsmoduls vorgegeben wird.

Die Gesamtzahl der bei der Expansion von θ_m neu geöffneten Zweige ist die Summe aus:

- $|\mu\sigma(s)| - 1$ (Aufgrund der Expansion von $\theta_m(s, |\text{matches}(s)|)$ in den Zweig br_{stop})
- $EC(s) \cdot (|\mu\sigma(s)| - 1)$ (Aufgrund der Expansion von $\theta_m(s, |\text{matches}(s)|)$ in die Zweige vom Typ $br_i, 0 < i \leq |EC(s)|$ - es bleiben aber nur $|\mu\sigma(s)|$ Zweige geöffnet)
- Es existieren nach der Expansion von $\theta_m(s, |\text{matches}(s)|)$ verschiedene negierte Konjunktionen, die expandiert und nicht geschlossen werden können. Die Zahl der durch die Expansion dieser negierten Konjunktionen neu erzeugten Zweige beträgt $Z^{'''}_{\mathcal{VE}}(s)$.
- Sei $ecm \in \mathcal{VE}(s)$ eine oder (falls eine solche existiert) diejenige Äquivalenzklasse mit der größten Kardinalität. Für alle $0 < i < |\mu\sigma(s)|$ führt die Expansion von $\theta_m(s, i)$ dann zu der Öffnung von maximal $\frac{b^{|ecm|}-1}{b-1} \cdot (i-1)$ neuen Zweigen

Es gilt damit:

$$\begin{aligned} Z^{'''}(s) \quad = \quad & |\mu\sigma(s)| - 1 + EC(s) \cdot (|\mu\sigma(s)| - 1) + Z^{'''}_{\mathcal{VE}}(s) + \\ & \left(\sum_{i=1}^{|\mu\sigma(s)-1|} \frac{b^{|ecm|}-1}{b-1} \cdot (i-1) \right) \end{aligned}$$

Womit die Zahl der insgesamt geöffneten Zweige Z(s)

$$\begin{aligned} Z(s) \quad = \quad & EC(s) + \\ & EC(s) + \\ & |\mu\sigma(s)| - 1 + EC(s) \cdot (|\mu\sigma(s)| - 1) + Z^{'''}_{\mathcal{VE}}(s) + \\ & \left(\sum_{i=1}^{|\mu\sigma(s)-1|} \frac{b^{|ecm|}-1}{b-1} \cdot (i-1) \right) \end{aligned}$$

beträgt.

Auswertung der Performanzanalyse

Die Komplexität eines Beweises mit der in Kapitel 4 vorgestellten Revisionsbasis hängt im wesentlichen von der Größe der Äquivalenzklassen, die in einer Situation erkannt werden, ab. Die gewählten Umstände, vor allem die Entscheidung, eine Breitensuche, die es erlaubt, den Nichtdeterminusmus der Expansion unberücksichtigt zu lassen, zu verwenden, liefern eine Überschätzung des Gesamtaufwands für einen Tableaubeweis. Allerdings ist diese Abschätzung erheblich genauer als eine Abschätzung der Performanz mithilfe des Branchingfaktors. Bei der Auflösung des Nichtdeterminusmus unberücksichtigt geblieben ist das Semantic Branching.

Angewendet auf die Situation 1, NBTest01 beträgt die Summe der Zahl der Zweige Z(Situation 1) = 21. Im Vergleich dazu beträgt die Summe der Zahl der Zweige in Situation 1, NBTest05 mit Z(Situation 1)= $1,9227158 \cdot 10^7$ wesentlich mehr.

Die Funktion Z ist unter bestimmten Umständen, insbesondere dem Umstand der Breitensuche unter Berücksichtigung der Prioritätsebenen, ermittelt worden. Diese Umstände entsprechen nicht der Realität eines tatsächslichen Tableaubeweises, in dem normalerweise eine Tiefensuche durchgeführt wird. Es ist anzunehmen, dass ein Tableaubeweiser, der eine Tiefensuche durchführt, erheblich schneller Abschlüsse bilden bzw. einen nicht abgeschlossenen Zweig finden kann, weshalb der der tatsächliche Rechenaufwand erheblich geringer ausfallen wird. Des weiteren sind bestimmte Eigenschaften der Revisionsbasis nicht berücksichtigt worden, insbesondere wurde von der Reihenfolge der Matches abstrahiert, ebenso von der internen Struktur der Matches (mit Ausnahme der Koreferenz des Ziels der Aktion), weshalb der tatsächliche Berechnungsaufwand geringer ausfallen wird.

Es ist nun kritisierbar, dass die Annahme einer Breitensuche zur Bestimmung des Berechnungsaufwands nicht sinnvoll ist. Allerdings beruht die Annahme der Breitensuche auf dem Nichtdeterminusmus der Auswahl des zu bearbeitenden Zweiges nach der Expansion einer β-Formel sowie dem Nichtdeterminusmus der Auswahl der zu expandierenden Formel innerhalb eines Zweiges. Eine Analyse ohne Breitensuche müsste beide Nichtdeterminismen berücksichtigen. Die Analyse wird dabei insbesondere aufgrund der großen Zahl von alternativen Zweigen nach der Expansion einer β-Formel eine große Zahl von Sonderfällen umfassen. Die große Zahl der Sonderfälle verhindert dabei eine eingehende Analyse ohne einen vertretbaren Untersuchungsaufwand zu übersteigen. Als weiterer Kritikpunkt kann die Verwendung der einfachen Verhaltensäquivalenz genannt werden, die implizit in θ_{ex} vorgenommen wird. Angenommen, es sollte strenge Verhaltensäquivalenz verwendet werden, so könnte diese nicht mit θ_{ex} ausgedrückt werden, da der Blickwinkel nicht berücksichtigt wird. Allerdings kann, wie in Abschnitt 4.2 ausgeführt, durch die Wahl von Folgeaktionen wie Drehung um die eigenen Achse, der Agent bei Bedarf unterschiedliche Blickwinkel einnehmen und so das Erreichen eines Ortes auf unterschiedlichen Wegen mit wenig Aufwand simulieren.

Die von mir gewählte Form der Untersuchung der Performanz der Revisionsbasis ist erheblich genauer als gewöhnliche Performanz- und Komplexitätsmaße wie der Verzweigungsfaktor. Der Verzweigungsfaktor wäre bereits für die Disjunktionen aus θ_m in Situation 1, NBTest05 mit ca. $1,405 \cdot 10^{51}$ signifikant höher als die mit Z berechnete Überschätzung der neu geöffneten Zweige aller durch θ erzeugten Formeln, die Zweigersparnis und Semantic Branching berücksichtigt. Neben Zweigersparnis und Semantic Branching existieren noch weitere Optimierungen für Tableau-Beweiser wie z.B. das Dependency Directed Backtracking (vorgestellt unter anderem in [HSP99]). Allerdings besteht auch für diese Optimierungen das Problem, dass für sie kein angemessenes Performanz- oder Komplexitätsmaß existiert, weshalb ebenfalls eine weitere Detailuntersuchung der Expansion nach dem Muster der obigen Tableauexpansion notwendig ist.

Ein wichtiges Fazit der Performanzanalyse ist, dass selbst Formeln mit sehr hohem Verweigungsfaktor, wie sie durch die Revisionsbasis erzeugt werden, mit einem Tableau-Beweiser, der bestimmte Optimierungen besitzt, mit einem akzeptablem Rechenaufwand verarbeitet werden können. Im Fall der Revisionsbasis sollte der verwendete Beweiser daher mindestens die Zweigersparnis und Semantic Branching unterstützen, um die Revisionsbasis sinnvoll verwendbar zu machen. Nicht-atomare Abschlüsse müssen dagegen nicht verwendet werden, da sie mit einer Ausnahme an keiner Stelle der Analyse Verwendung gefunden haben. In Abbildung C.8 ist erkennbar, dass nach der Expansion der Konjunktionen in die Zweige an dieser Stelle bereits ein nicht atomarer Abschluss möglich ist. Allerdings erfordert der nicht-atomare Abschluss nur die Expansion der Konjunktion. Da die Konjunktion nicht verzweigend expandiert wird, ist der Zusatzaufwand der Expansion dieser Formel im Vergleich mit dem insgesamt größeren Zusatzaufwand durch Prüfung des nicht atomaren Abschlusses vertretbar.

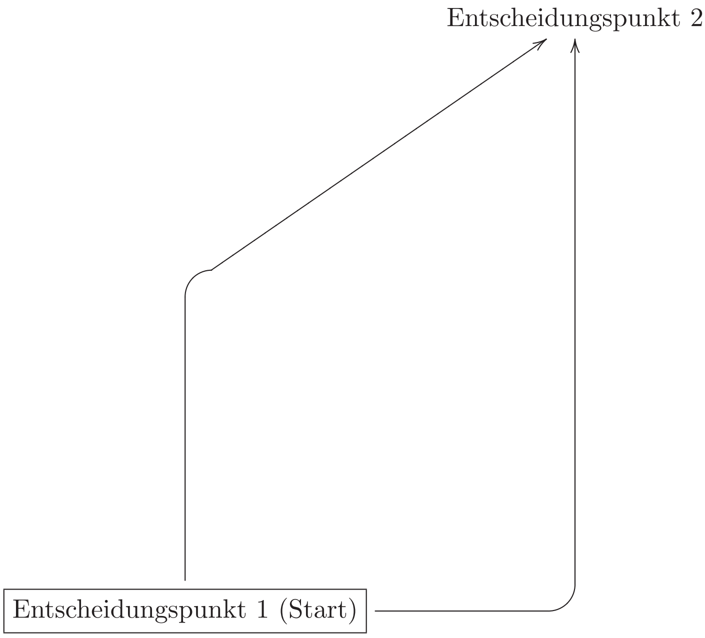


Abbildung 5.3: Verhaltensäquivalenz im nächsten Schritt
Der Agent erkennt, wenn er an Entscheidungspunkt 1 angekommen ist, die Verhaltensäquivalenz der beiden Alternativen, die jeweils zu Entscheidungspunkt 2 führen.

5.3 Erweiterbarkeit

Analyse weiterer Fälle von Verhaltensäquivalenz

Die in Abschnitt 4.2 eingeführte Definition von Verhaltensäquivalenz klassifiziert eine Anzahl von unterschiedlichen Matches als verhaltensäquivalent, wenn die Wahl von ihnen jeweils zum gleichen Ergebnis führt. Als formales Kriterium für Verhaltensäquivalenz wurde dabei im Fall von einfacher Verhaltensäquivalenz die Koreferenzierung des gleichen Perzeptionsknotens zu dem Instruktionsknoten, der das Ziel einer Aktion repräsentiert, im Fall von strenger Verhaltensäquivalenz wird zusätzlich nach Ausführung der Aktion der gleiche Blickwinkel gefordert. Auffallend dabei ist, dass strenge und einfache Verhaltensäquivalenz nur bis zum nächsten Entscheidungspunkt voraus berechnet wird und das Erkennen von Verhaltensäquivalenz ausschließlich von der aktuellen Perzeption abhängig ist.

Es ist allerdings denkbar, dass es darüber hinaus Fälle von Verhaltensäquivalenz gibt, die erst nach Erreichen von späteren Entscheidungspunkten als solche erkannt werden können. Abbildung 5.3 zeigt einen solchen bereits im ersten Schritt offensichtlichen Fall von Verhaltensäquivalenz.

Ein Vergleich von Abbildung 5.3 und 5.4 zeigen im Prinzip eine ähnliche Situation. Offensichtlich besteht der einzige Unterschied zwischen ihnen darin, dass ein Haus die Sicht des sich auf Entscheidungspunkt 1 befindenen Agenten soweit blockiert, dass Entscheidungspunkt 2 und die zwei Zugangsmöglichkeiten zu Entscheidungspunkt 2 nicht erkennbar sind. Daher erkennt der Agent nicht, dass die Wege über die Entscheidungspunkte 1a und 1b verhaltensäquivalent sind.

Gelangt der Agent in Abbildung 5.4 auf einer der Alternativen nach Entscheidungspunkt 2 und stellt dort fest, dass ein weiterer Weg nicht möglich ist, so ist es denkbar, dass der Agent auf Entscheidungspunkt 1 zurückkehrt und die andere Alternative wählt. Besäße der Agent ein räumliches Gedächtnis so wäre er bereits nach Erreichen des Entscheidungspunkts 1a oder 1b in der Lage, die Verhaltensäquivalenz der Wegen zwischen den Entscheidungspunkten 1a und 2 sowie 1b und 2 zu erkennen und kann an dieser Stelle bereits das Navigieren abbrechen.

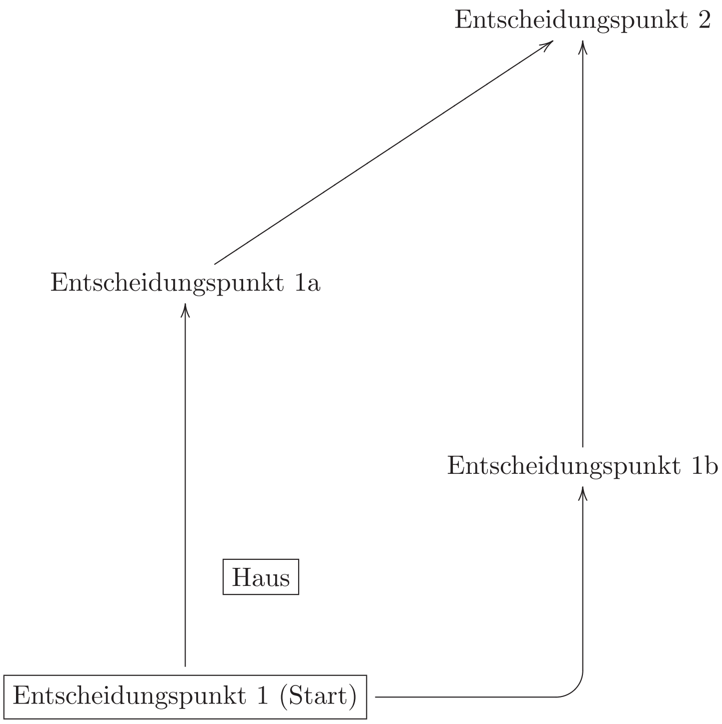


Abbildung 5.4: Verhaltensäquivalenz im übernächsten Schritt
Ein Haus blockiert die Sicht auf Entscheidungspunkt 2. Der Agent kann auf Entscheidungspunkt 1 angekommen nicht feststellen, dass er zwei verhaltensäquivalente Alternativen zu Auswahl hat.

Die Revisionsbasis unterstützt das spätere Erkennen von Verhaltensäquivalenz, da es möglich ist, ein Adjustment durchzuführen, das zwei Koreferenzen über eine Biimplikation verknüpft. Werden zwei Koreferenzen mit einem Degree von Λ_1 über eine Biimplikation verknüpft und ist eine der beiden Koreferenzen bereits mit einem Degree von Λ_{remove} ausgeschlossen, so wird die Revisionsbasis auch die Negation der nicht explizit ausgeschlossenen Koreferenz folgern können, was den gleichen Effekt besitzt, wie das explizite Entfernen von dieser. Alternativ kann anstelle der Biimplikation auch ein zusätzlicher Eintrag gemacht werden, welcher aus der Negation der Zielkoreferenz des gewählten Matches die Negation der Zielkoreferenz der Negation des nicht gewählten, aber später als zum gewähltem Match verhaltensäquivalent erkannten Matches besteht.

Die Revisionsbasis ermöglicht, wie das oben aufgeführte Beispiel zeigt, damit eine nachträgliche Erkennung von Verhaltensäquivalenz. Die Voraussetzung dafür ist, dass ihr dafür von außen mitgeteilt wird, welche Koreferenzen die gleiche Bedeutung besitzen. Die Aufgabe der Erkennung von Verhaltensäquivalenz über mehrere Schritte hinweg kann durch ein zusätzliches Analysemodul geschehen. Kriterien für den Aufbau eines solchen Analysemoduls sollen in dieser Arbeit nicht behandelt werden.

Ausschluss beliebiger Koreferenzen

In der durch θ aufgebauten Revisionsbasis ist das Adjustment auf den Ausschluss der „Ziel“-Koreferenz beschränkt. Der Ausschluss der Zielkoreferenz ist möglich, weil der Agent ein Ausschlusskriterium, nämlich das Fehlen von Matches oberhalb eines Schwellenwerts in der direkt folgenden Situation, für die Zielkoreferenz besitzt. Die Folge des Ausschlusses ist der „Ausfall“ aller Matches, die die ausgeschlossene Zielkoreferenz enthalten, wobei die von außen vorgegebene Totalordnung nicht verändert wird. Ist die ausgeschlossene Koreferenz in dem momentan gewählten Match enthalten, so wird auch dieses ausgeschlossen und ein Alternatives gewählt, falls eine Alternative existiert. Die Revision erfolgt damit nach den in Abschnitt 4.3.3 auf Seite 51 aufgestellten Rationalitätskriterien.

Es ist denkbar, dass weitere Kriterien zum Ausschluss der Zielkoreferenz aber auch von anderen Koreferenzen existieren. Falls diese Kriterien existieren, so ist die Revisionsbasis in der Lage nach einer Revision um eine beliebige Koreferenz diese unter Anlehnung an die in Abschnitt 4.3.3 aufgeführten Rationalitätskriterien auszuschließen. Die Rationalitätskriterien umfassen dabei unter anderem:

- Wird eine Koreferenz ausgeschlossen, so werden auch alle Matches ausgeschlossen, welche die ausgeschlossene Koreferenz enthalten.
- Der Ausschluss verändert nicht die von außen vorgegebene Totalordnung.
- Wird das gewählte Match ausgeschlossen, so wird wenn möglich, eine Alternative gewählt.
- Es soll maximal ein Match pro Situation eine epistemische Verankerung > 0 besitzen.

Die Begründung, warum die oben genannten Rationalitätskriterien auch bei dem Ausschluss beliebiger Koreferenzen gültig sind, kann analog zu den Rationalitätskriterien aus Abschnitt 4.3.3 erfolgen und soll an dieser Stelle nicht wiederholt werden.

Kapitel 6

Zusammenfassung und Ausblick

Zusammenfassung

Die in Kapitel 4 vorstellte Revisionsbasis ermöglicht dem Agenten sein mentales Modell im Fall einer Fehlentscheidung während der Navigationsphase zu revidieren. Die Revisionsbasis umfasst einen vom Umfang her kleinen Ausschnitt des Gesamtwissens des Agenten, wodurch der in Kapitel 3 ausgewählte Revisionsoperator in der Lage ist auch im Fall von komplexen Situationen eine Chance hat, eine annehmbare Berechnungszeit einzuhalten. Dafür ist allerdings die Verwendung von verschiedenen Optimierungen notwendig, angefangen bei der Verwendung eines von außen priorisierbaren Tableaubeweisers zur schnellen Berechnung der Funktion degree. Wie die Performanzanalyse in Kapitel 5, muss ein Tableaubeweiser darüber hinaus über eine Zweigersparnisregel und Semantic Branching verwenden, um die Inhalte der Revisionsbasis in annehmbarer Zeit zu bearbeiten. Die Performanzanalyse zeigt auch, wie Formeln, die von herkömmlichen Komplexitätsmaßen wie dem Verzeigungsfaktor her scheinbar nicht in der näheren Zukunft mit Tableau-Beweisern behandelbar sind, durch Ausnutzung von Redundanz und einigen Optimierungen mit annehmbarem Aufwand beweisbar sind. Eine Voraussetzung dafür ist jedoch eine detaillierte Untersuchung des schrittweisen Aufbaus des Tableaus.

Neben der Untersuchung der reinen technischen Machbarkeit im Rahmen der Performanzanalyse bietet die Revisionsbasis darüber hinaus die Grundlage für eine später hinzukommende Handlungsrevision. Die Handlungsrevision kann es dem Agenten ermöglichen, einen Instruktionsplan für eine erfolgreiche Navigationsphase zu verwenden, wenn dies ohne Handlungsrevision nicht möglich ist. Damit ist der Agent in der Lage, seine Instruktionen robuster zu verarbeiten und sein Performanzmaß (siehe 2.1) zu verbessern. Die Revisionsbasis erlaubt sogar über die reine Revision der Handlung hinausgehende Schlüsse, da sie Verhaltensäquivalenz berücksichtigt. Durch die Berücksichtigung der Verhaltensäquivalenz verhindert, dass der Agent zu einem Ort, der sich einmal als nicht zur Instruktionsanweisung passend erwiesen hat, zurückkehrt, wodurch der Agent insgesamt kleinere Wegstrecken zurücklegt. Die Reduzierung der zurückelegten Wegstrecke unter Beibehaltung der gleichen Routeninstruktion kann, wie in Kapitel 2.1 ausgeführt, als ein Zuwachs an rationalem Verhalten gewertet werden.

Ansatzpunkte für Erweiterungen am geometrischen Agenten

Der offensichtlichste Ansatzpunkt zur Erweiterung des geometrischen Agenten ist die Hinzufügung des Handlungsrevisionsmoduls, um das Revisionsmoduls auch aktiv nutzen zu können. Im Rahmen des Handlungsrevisionsmoduls kann es sinnvoll sein, den Agenten Teile der Perzeption erinnern zu lassen, damit der Agent in der Lage ist, in eine einmal erreiche Situation zurückzukehren.

Das Revisionsmodul erkennt bisher ausschließlich Fälle von Verhaltensäquivalenz, wenn das gleiche Ziel durch die Wahl verschiedener Matches einer Situation erreichbar ist. Es ist ein zusätzliches Diagnosemodul zur Erkennung von Fällen von Verhaltensäquivalenz, denkbar, in dem auch Verhaltensäquivalenz erkannt wird, wenn der Agent mehr als einen Schritt des Aktionsplans ausführen muss, um das gleiche Ziel auf unterschiedlichem Weg zu erreichen. Das Revisionsmodul ermöglicht die Repräsentation von Verhaltensäquivalenz auch wenn diese nicht sofort erkannt wird.

Die Untersuchung der Performanz der Revisionsbasis hat gezeigt, dass es schwierig ist, den tatsächlichen Berechnungsaufwand eines Tableau-Beweises zu ermitteln. Dies liegt insbesondere daran, dass es keine einfachen Komplexitätskriterien mit Ausnahme von Verzweigungsfaktor und Tableau-Tiefe gibt. Problematisch ist, dass diese Komplexitätskriterien keine der möglichen Optimierungen eines Tableau-Beweises einbeziehen können. Es erscheint daher sinnvoll, für die Untersuchung des Berechnungsaufwands eines Tableau-Beweises ein formales Verfahren zu entwickeln, in dem insbesondere die grundlegenden Optimierungen erfasst werden können.

Obwohl das Ergebnis der Untersuchung der Performanz auch in komplexen Fällen zeigt, dass es Tableau-Beweiser mit rudimentären Optimierungen schaffen, die Revisionsbasis in annehmbarer Zeit zu verarbeiten, ist gerade in den komplexen Beispielen erkennbar, dass die Performanz die obere Grenze des Annehmbaren erreicht. Es ist daher sinnvoll, weitere Optimierungen zu untersuchen. Bei der Expansion eines Tableaus gibt es prinzipiell zwei Probleme: der Nichtdeterminismus bei der Expansion eines Zweigs sowie der Nichtdeterminismus bei der Wahl des Zweigs, der nach einer verzweigenden Expansion weiter expandiert wird. Dadurch ist beispielsweise das Semantic Branching mit der Reihenfolge der Untersuchung einzelner alternativer Zweige einer verzweigenden Expansion verknüpft. Die Reihenfolge der Untersuchung der alternativen Zweige wird nichtdeterministisch bestimmt, weshalb auch die Anwendung des Semantic Branching eine Vielzahl von möglichen Verzweigungen erzeugt. Es wäre sinnvoll, ein eindeutiges Kriterium, bspw. eine Heuristik zu benutzen, durch die der Nichtdeterminismus aus der Tableau-Expansion genommen wird. Das Ziel einer solchen Heuristik muss nicht sein, die zur Minimierung der Tableaukomplexität her notwendigerweise beste Expansionsreihenfolge festzulegen. Eine solches, die Komplexität des Tableau-Baums gering haltendes Kriterium ist zwar wünschenswert, allerdings im Rahmen der Untersuchung des Berechnungsaufwands eines Tableau-Beweises nicht zwingend notwendig.

Anhang A

Beispiele von Matches

Vorbemerkung: die Koreferenzen werden, um eine bessere Übersicht zu gewährleisten, in dem Format k_i, wobei $i \in (N)^+$. Die Koreferenz zum „Ziel“ der Aktion wird farblich hinterlegt. Sollten weitere Hervorhebungen in einigen Beispielen notwendig sein, so werden diese lokal erklärt. Die Koreferenzen der einzelnen Matches sind so angeordnet, dass innerhalb einer Spalte in einer Situation immer die gleichen Instruktionsknoten koreferenziert werden. In jeder Situation wird das an oberster Stelle aufgeführte Match gewählt.

Beispiel 1 - NBTest01

Dieses Beispiel entspringt der Routenbeschreibung NBTest01, die den Agenten von der Mensa zu Haus E navigieren soll. Die verwendete Konfiguration ist Sommer, Simulation 1 und mit eingeschalteten Labels. Die Routeninstruktion lautet: „geh nach links dot geh zwischen haus b und haus c durch dot geh nach rechts dot dann stehst du vor haus e dot“

Situation 1 (S1)	k_1	k_4	k_5	k_8		
	k_2	k_4	k_6	k_8		
	k_3	k_4	k_7	k_8		
Situation 2 (S2)	k_9	k_{13}	k_{17}	k_{18}	k_{19}	k_{20}
	k_{10}	k_{14}	k_{17}	k_{18}	k_{19}	k_{20}
	k_{11}	k_{15}	k_{17}	k_{18}	k_{19}	k_{20}
	k_{12}	k_{16}	k_{17}	k_{18}	k_{19}	k_{20}
Situation 3 (S3)	k_{25}	k_{21}	k_{23}	k_{24}		
	k_{26}	k_{22}	k_{23}	k_{24}		

Die Wissensbasis, die aus Situation 1 dieses Beispiels resultiert ist:

$$\theta(\text{Situation 1}, \emptyset) = \theta_{\text{ex}}(\text{Situation 1}) \cup \theta_m(\text{Situation 1}) \cup \theta_s(s, s_0)$$

.

Dabei ist

$$\theta_{\text{ex}}\,(\text{S1}) = \left\{ \begin{array}{l} \Big(k_1 \supset \neg k_2, \Lambda_1\Big), \\ \Big(k_1 \supset \neg k_3, \Lambda_1\Big), \\ \Big(k_2 \supset \neg k_3, \Lambda_1\Big) \end{array} \right\}$$

wobei

$$\theta_m\,(\text{S1}) = \left\{ \begin{array}{l} \Big((k_1 \wedge k_4 \wedge k_5 \wedge k_8), ep_{\mu\sigma}(1)\Big), \\ \Big(((k_1 \wedge k_4 \wedge k_5 \wedge k_8) \vee (k_2 \wedge k_4 \wedge k_6 \wedge k_8)), ep_{\mu\sigma}(2)\Big), \\ \Big(((k_1 \wedge k_4 \wedge k_5 \wedge k_8) \vee (k_2 \wedge k_4 \wedge k_6 \wedge k_8) \vee (k_3 \wedge k_4 \wedge k_7 \wedge k_8)), ep_{\mu\sigma}(3)\Big) \end{array} \right\}$$

$$\theta_s(\text{S1}, s_0) = \Big\{ \Big((\neg k_1 \wedge \neg k_2 \wedge \neg k_3) \supset \text{stop}, \Lambda_2\Big) \Big\}$$

$$\theta(\text{S1}, s_0) = \left\{ \begin{array}{l} \Big(k_1 \supset \neg k_2, \Lambda_1\Big), \Big(k_1 \supset \neg k_3, \Lambda_1\Big), \Big(k_2 \supset \neg k_3, \Lambda_1\Big), \\ \Big((\neg k_1 \wedge \neg k_2 \wedge \neg k_3) \supset \text{stop}, \Lambda_2\Big), \\ \Big(((k_1 \wedge k_4 \wedge k_5 \wedge k_8) \vee (k_2 \wedge k_4 \wedge k_6 \wedge k_8) \vee (k_3 \wedge k_4 \wedge k_7 \wedge k_8)), ep_{\mu\sigma}(3)\Big), \\ \Big(((k_1 \wedge k_4 \wedge k_5 \wedge k_8) \vee (k_2 \wedge k_4 \wedge k_6 \wedge k_8)), ep_{\mu\sigma}(2)\Big), \\ \Big((k_1 \wedge k_4 \wedge k_5 \wedge k_8), ep_{\mu\sigma}(1)\Big) \end{array} \right\}$$

Die Einträge, die in Situation 2 der Revisionsbasis hinzugefügt werden, sind :

$$\theta \left(\begin{array}{c} \text{S2,} \\ \text{S1} \end{array} \right) = \left\{ \begin{array}{l} \Big(k_9 \supset \neg k_{10}, \Lambda_1\Big), \Big(k_9 \supset \neg k_{11}, \Lambda_1\Big), \Big(k_9 \supset \neg k_{12}, \Lambda_1\Big), \\ \Big(k_{10} \supset \neg k_{11}, \Lambda_1\Big), \Big(k_{10} \supset \neg k_{12}, \Lambda_1\Big), \Big(k_{11} \supset \neg k_{12}, \Lambda_1\Big), \\ \Big((\neg k_9 \wedge \neg k_{10} \wedge \neg k_{11} \wedge \neg k_{12}) \supset k_1, \Lambda_2\Big), \\ \Big((k_9 \wedge k_{13} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee (k_{10} \wedge k_{14} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee \\ (k_{11} \wedge k_{15} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee (k_{12} \wedge k_{16} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}), ep_{\mu\sigma}(4)\Big), \\ \Big((k_9 \wedge k_{13} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee (k_{10} \wedge k_{14} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee \\ (k_{11} \wedge k_{15} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}), ep_{\mu\sigma}(3)\Big), \\ \Big((k_9 \wedge k_{13} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}) \vee (k_{10} \wedge k_{14} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}), ep_{\mu\sigma}(2)\Big), \\ \Big((k_9 \wedge k_{13} \wedge k_{17} \wedge k_{18} \wedge k_{19} \wedge k_{20}), ep_{\mu\sigma}(1)\Big) \end{array} \right\}$$

Die Revisionsbasis in Situation 2 besteht aus $\theta(\text{S2}, \text{S1}) \cup \theta(\text{S1}, s_0)$. In Situation 3 besteht die Revisionsbasis aus $\theta(\text{S3}, \text{S2}) \cup \theta(\text{S2}, \text{S1}) \cup \theta(\text{S1}, s_0)$ mit

$$\theta \left(\begin{array}{c} \text{S3,} \\ \text{S2} \end{array} \right) = \left\{ \begin{array}{l} \Big(k_{25} \supset \neg k_{26}, \Lambda_1\Big), \Big((\neg k_{25} \wedge \neg k_{26}, \Lambda_2\Big), \\ \Big((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}), ep_{\mu\sigma}(2)\Big), \\ \Big((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}), ep_{\mu\sigma}(1)\Big) \end{array} \right\}$$

Beispiel 2 - NBTest05

Dieses Beispiel entspringt der Routenbeschreibung NBTest05, die den Agenten vom Pförtner zu Haus E lotsen soll. Die verwendete Konfiguration ist Sommer, Simulation 1 und mit eingeschalteten Labels. Es gibt in Situation 1 insgesamt 7 Äquivalenzklassen mit je 6 Einträgen. Insgesamt gibt es 42 Matches. Die Routeninstruktion lautet: „geh zu haus f dot geh zwischen haus f und haus d durch dot geh nach links dot dann stehst du vor haus e dot“

Situation 1	k_{135}	k_{101}	k_{136}	k_8	k_{137}
	k_{135}	k_{101}	k_{136}	k_8	k_5
	k_{135}	k_{101}	k_{136}	k_8	k_{138}
	k_{135}	k_{102}	k_{136}	k_8	k_{137}
	k_{135}	k_{102}	k_{136}	k_8	k_5
	k_{135}	k_{102}	k_{136}	k_8	k_{138}
	k_{139}	k_{101}	k_{136}	k_8	k_{137}
	k_{139}	k_{101}	k_{136}	k_8	k_5
	k_{139}	k_{101}	k_{136}	k_8	k_{138}
	k_{139}	k_{102}	k_{136}	k_8	k_{137}
	k_{139}	k_{102}	k_{136}	k_8	k_5
	k_{139}	k_{102}	k_{136}	k_8	k_{138}
	k_{140}	k_{101}	k_{136}	k_8	k_{137}
	k_{141}	k_{101}	k_{136}	k_8	k_{137}
	k_{142}	k_{101}	k_{136}	k_8	k_{137}
	k_{143}	k_{101}	k_{136}	k_8	k_{137}
	k_{140}	k_{101}	k_{136}	k_8	k_5
	k_{141}	k_{101}	k_{136}	k_8	k_5
	k_{142}	k_{101}	k_{136}	k_8	k_5
	k_{143}	k_{101}	k_{136}	k_8	k_5
	k_{140}	k_{101}	k_{136}	k_8	k_{138}
	k_{141}	k_{101}	k_{136}	k_8	k_{138}
	k_{142}	k_{101}	k_{136}	k_8	k_{138}
	k_{143}	k_{101}	k_{136}	k_8	k_{138}
	k_{140}	k_{102}	k_{136}	k_8	k_{137}
	k_{141}	k_{102}	k_{136}	k_8	k_{137}
	k_{142}	k_{102}	k_{136}	k_8	k_{137}
	k_{143}	k_{102}	k_{136}	k_8	k_{137}
	k_{140}	k_{102}	k_{136}	k_8	k_5
	k_{141}	k_{102}	k_{136}	k_8	k_5
	k_{142}	k_{102}	k_{136}	k_8	k_5
	k_{143}	k_{102}	k_{136}	k_8	k_5
	k_{140}	k_{102}	k_{136}	k_8	k_{138}
	k_{141}	k_{102}	k_{136}	k_8	k_{138}
	k_{142}	k_{102}	k_{136}	k_8	k_{138}
	k_{143}	k_{102}	k_{136}	k_8	k_{138}
	k_{144}	k_{101}	k_{136}	k_8	k_{137}
	k_{144}	k_{101}	k_{136}	k_8	k_5
	k_{144}	k_{101}	k_{136}	k_8	k_{138}
	k_{144}	k_{102}	k_{136}	k_8	k_{137}
	k_{144}	k_{102}	k_{136}	k_8	k_5
	k_{144}	k_{102}	k_{136}	k_8	k_{138}

Situation 2	(Folgeaktionen)			
Situation 3	k_{148}	k_{145}	k_{146}	k_{147}
	k_{150}	k_{149}	k_{146}	k_{147}
	k_{151}	k_{153}	k_{146}	k_{147}
	k_{154}	k_{124}	k_{146}	k_{147}
	k_{155}	k_{152}	k_{146}	k_{147}
Situation 4	k_{157}	k_{156}		
Situation 5	k_{158}	k_{159}		
	k_{160}	k_{159}		
	k_{161}	k_{162}		
	k_{163}	k_{162}		
	k_{164}	k_{165}		
	k_{166}	k_{165}		
	k_{167}	k_{168}		
	k_{169}	k_{168}		
	k_{170}	k_{171}		
	k_{172}	k_{171}		
	k_{173}	k_{174}		
	k_{175}	k_{174}		
	k_{176}	k_{177}		
	k_{178}	k_{177}		
	k_{179}	k_{180}		
	k_{181}	k_{180}		
	k_{182}	k_{183}		
	k_{184}	k_{183}		
	k_{185}	k_{186}		
	k_{187}	k_{186}		

Beispiel 3 - NBTest03

Dieses Beispiel zeigt den Verlauf der Navigation des Agenten aus Beispiel NBTest03, vom Pförtner nach Haus F(2). Die Konfiguration ist Sommer, Simulation 1 und die Labes sind eingeschaltet. Die Instruktion lautet: „geh nach links dot geh zwischen haus b und haus c durch dot geh nach rechts dot wenn du bei haus e bist dreh dich nach rechts dot geh zwischen haus c und haus e durch dot dann stehst du vor haus f dot“

Situation 1	k_{61}	k_{59}	k_{60}	k_{64}		
	k_{62}	k_{59}	k_{60}	k_{65}		
	k_{63}	k_{59}	k_{60}	k_{66}		
Situation 2	k_{67}	k_{71}	k_{72}	k_{76}	k_{77}	k_{78}
	k_{68}	k_{71}	k_{73}	k_{76}	k_{77}	k_{78}
	k_{69}	k_{71}	k_{74}	k_{76}	k_{77}	k_{78}
	k_{70}	k_{71}	k_{75}	k_{76}	k_{77}	k_{78}
Situation 3	k_{81}	k_{79}	k_{83}	k_{84}		
	k_{82}	k_{80}	k_{83}	k_{84}		
Situation 4	(Folgeaktionen)					
Situation 5	k_{85}	k_{86}	k_{87}	k_{88}		
Situation 6	k_{98}	k_{89}	k_{90}	k_{91}	k_{92}	k_{93}
	k_{98}	k_{89}	k_{90}	k_{91}	k_{92}	k_{94}
	k_{98}	k_{89}	k_{90}	k_{91}	k_{92}	k_{95}
	k_{98}	k_{89}	k_{90}	k_{91}	k_{92}	k_{96}
	k_{98}	k_{89}	k_{90}	k_{91}	k_{92}	k_{97}
Situation 7	k_{99}	k_{100}				

Anhang B

Beispiele von Tableau-Beweisen

Beispiel 1 - reguläre Expansion

Das erste Beispiel der regulären Expansion zeigt, wie die Anfrage, ob eine bestimmte Koreferenz k_{26} aus der Revisionsbasis folgerbar ist, innerhalb des Tableau-Beweisers abgearbeitet werden könnte. Eine solche Anfrage ist unter anderem zur Berechnung eines Degrees, im nachfolgenden Beispiel von k_{26} denkbar. Die Verwendeten Daten stammen dabei aus Beispiel 1, Situation 3. Zur Vereinfachung wurden die Einträge der vorangegangenen Situationen sowie die Einträge zur situationsübergreifenden Verankerung aus dem Tableau-Beweis weggelassen. Es kann sein, dass an bestimmten Stellen Formeln in einen Tableau-Zweig expandiert werden, die dort schon existieren. An Stelle der Formeln soll dann im resultierenden Tableau $\Box$ erscheinen, um Expansionen besser zu illustrieren. Der Tableau-Beweis soll für die Formel $((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26}$ durchgeführt werden. Dazu ist zunächst die Negation der Formel notwendig, so dass das Tableau mit der Formel $\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26})$ initialisiert wird. Das Ausgangstableau bietet nur eine Expansionsmöglichkeit, nämlích die (nicht-verzweigende) Expansion der negierten Implikation.

Expansion 1 ($\neg(X \supset Y)$)

$$
\begin{array}{c}
\neg\Big(\underline{((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26}))} \supset \underline{k_{26}}\Big) \\
| \\
(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26}) \\
| \\
\neg k_{26}
\end{array}
$$

Das resultierende Tableau besitzt nur die Möglichkeit, die verallgemeinerte Konjunktion des zweiten Tableau-Eintrags zu expandieren. Ich gehe zur Einfachheit an dieser Stelle davon aus, dass der Tableau-Beweiser in der Lage ist, verallgemeinerte Konjunktionen zu verarbeiten. Ansonsten wären anstelle einer einzelnen Expansion insgesamt zwei notwendig.

Expansion 2 (verallgemeinertes $\wedge$):

$$
\begin{array}{c}
\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26}) \\
| \\
\underline{(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})} \wedge \underline{((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))} \wedge \underline{(k_{25} \supset \neg k_{26})} \\
| \\
\neg k_{26} \\
| \\
(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \\
| \\
((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \\
| \\
(k_{25} \supset \neg k_{26})
\end{array}
$$

Von den drei dem Tableau hinzugefügten Einträgen müssen zwei verzweigend expandiert werden. Da eine verzweigende Expansion an dieser Stelle dazu führen würde, dass die Expansion des nicht verzweigenden Eintrags $(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$ dann in jedem entstandenen Zweig durchgeführt werden müsste, wird zunächst der nicht verzweigende Eintrag expandiert.

Expansion 3 (verallgemeinertes $\wedge$):

$$
\begin{array}{c}
\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26}) \\
| \\
(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26}) \\
| \\
\neg k_{26} \\
| \\
(\underline{k_{25}} \wedge \underline{k_{21}} \wedge \underline{k_{23}} \wedge \underline{k_{24}}) \\
| \\
((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \\
| \\
(k_{25} \supset \neg k_{26}) \\
| \\
k_{25} \\
| \\
k_{21} \\
| \\
k_{23} \\
| \\
k_{24}
\end{array}
$$

Es können an dieser Stelle zwei Einträge expandiert werden, die beide Verzweigen. Es wird daher willkürlich der Eintrag $(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})$ expandiert.

Expansion 4 ($(X \vee Y)$)

$$
\begin{array}{c}
\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26}) \\
| \\
(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26}) \\
| \\
\neg k_{26} \\
| \\
(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \\
| \\
\Big(\underline{(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})} \vee \underline{(k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})}\Big) \\
| \\
(k_{25} \supset \neg k_{26}) \\
| \\
k_{25} \\
| \\
k_{21} \\
| \\
k_{23} \\
| \\
k_{24}
\end{array}
$$

$$
\begin{array}{cc}
| & \diagdown \\
\square & (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})
\end{array}
$$

Die beiden resultierende Zweige besitzen jeweils einen Eintrag mit einer verallgemeinerten Konjunktion als Inhalt. Aus dem gemeinsamen Zweig verbleibt zusätzlich die verzweigende Expansion von $(k_{25} \supset \neg k_{26})$. Da die Konjunktion nicht verzweigend expandiert wird, in beiden Zweigen jeweils zunächst die Konjunktion expandiert werden.

Expansion 5 (verallgemeinertes $\wedge$)

$$\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24} \wedge)((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26})$$

$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})$$

$$\neg k_{26}$$

$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$$

$$((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))$$

$$(k_{25} \supset \neg k_{26})$$

$$k_{25}$$

$$k_{21}$$

$$k_{23}$$

$$\begin{array}{ccc} k_{24} & & (\underline{k_{26}} \wedge \underline{k_{22}} \wedge \underline{k_{23}} \wedge \underline{k_{24}}) \\ \Box & & k_{26} \\ & & k_{22} \end{array}$$

Im linken Zweig ergibt die Expansion Einträge, die bereits vorhanden sind und deshalb nicht erneut hinzugefügt werden müssen. Der rechte Zweig ist bereits abgeschlossen, da in ihm sowohl k_{26} als auch $\neg k_{26}$ enthalten sind. Er muss nicht weiter expandiert werden. Im linken Zweig steht die verzweigende Expansion von $(k_{25} \supset \neg k_{26})$ aus.

Expansion 6 $(X \supset Y)$

$$\neg(((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26})$$

$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})$$

$$\neg k_{26}$$

$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$$

$$((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))$$

$$(\underline{k_{25}} \supset \underline{\neg k_{26}})$$

$$k_{25}$$

$$k_{21}$$

$$k_{23}$$

$$\begin{array}{ccc} k_{24} & & (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}) \\ \Box & & k_{26} \\ & & k_{22} \\ \neg k_{25} & & \neg k_{26} \end{array}$$

Der resultierende linke Zweig ist durch k_{25} und $\neg k_{25}$ abgeschlossen. Der rechte Zweig ist nicht abgeschlossen und es existiert kein Eintrag, der expandierbar ist und noch nicht expandiert wurde. Er kann daher nicht abgeschlossen werden, was bedeutet, dass $((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \wedge ((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24})) \wedge (k_{25} \supset \neg k_{26})) \supset k_{26}$ nicht gültig ist und somit k_{26} nicht aus der Wissensbasis folgerbar ist.

Beispiel 2 - Expansion mit Priorität

Das folgende Beispiel soll zeigen, wie eine Expansion mit von Außen vorgegebenen Prioritäten sich negativ auf die Expansionsreihenfolge auswirken kann. Als Ausgangspunkt soll die gleiche Formel wie in Tableau-Beispiel 1 bewiesen werden. Allerdings wird das Tableau gemäß des Abschnitt 3.3 auf Seite 31 beschriebenem Verfahrens mit einem vorgefertigtem Tableau initialisiert und expandiert. Das vorgefertigte Tableau entspricht dabei im wesentlichen dem Ergebnis von Expansionsschritt 2 im ersten Beispiel, allerdings sind die einzelnen Tableau-Einträge mit einer Prioritätsstufe versehen. Die Ausgangsformel und die Ergebnisse der 1. Expansion sind für den Beweis überflüssig, da sie bereits einmal expandiert worden sind, und werden daher weggelassen.

Die Prioritätsstufen der einzelnen Tableau-Einträge sind durch einen farbigen Hintergrund gekennzeichnet. Es gibt die Stufen Rot, Gelb und Grün, wobei die Priorität von Rot > Gelb > Grün ist. Die aktuelle Prioritätsstufe p wird mit rot initialisiert.

Das Ausgangstableau

$$\neg k_{26}$$
$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$$
$$((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))$$
$$(k_{25} \supset \neg k_{26})$$

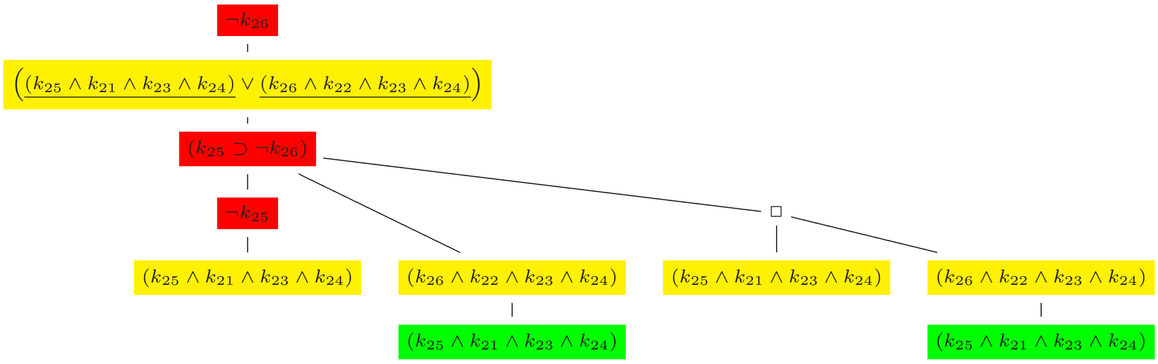
Das Ausgangstableau bietet im Gegensatz zu dem korrespondierenden Tableau in Beispiel 1 keine Möglichkeit, den zu expandierenden Tableaueintrag zu wählten. Der Eintrag $\neg k_{26}$ kann nicht weiter expandiert werden und so muss die verzweigende Expansion $(k_{25} \supset \neg k_{26})$ ausgeführt werden.

Expansion 1 $(X \supset Y)$:

$$\neg k_{26}$$
$$(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$$
$$((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))$$
$$(k_{25} \supset \neg k_{26})$$
$$\neg k_{25} \qquad \square$$

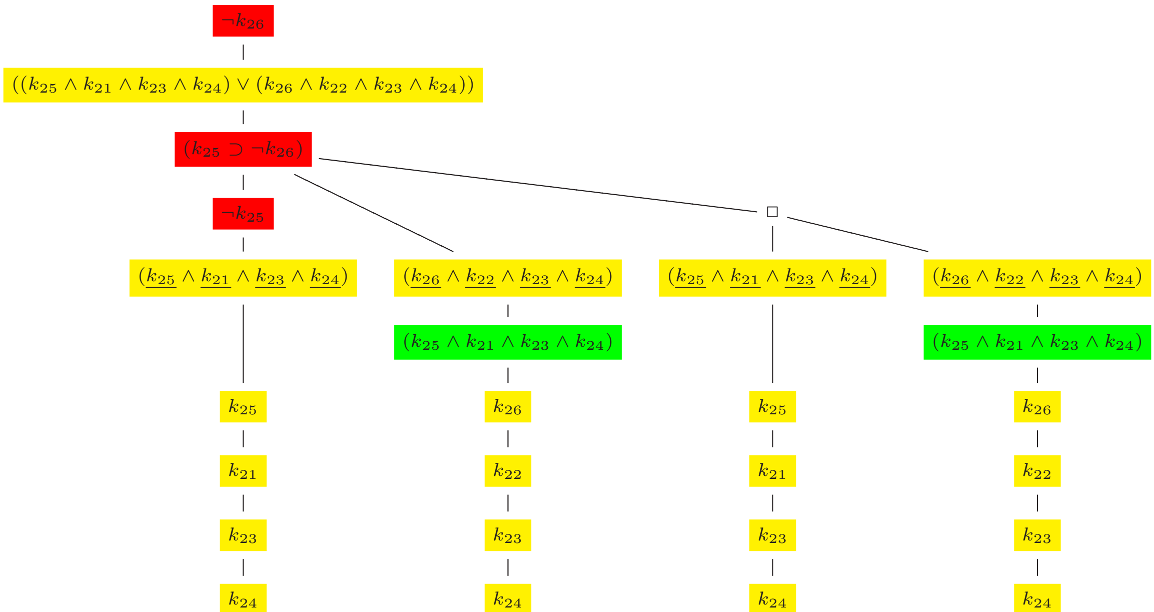
Die beiden neuen Einträge wurden mit der aktuellen Prioritätsstufe Rot von p initialisiert. Ein Abschluss kann mit den Einträgen auf aktueller Prioritätsstufe nicht gebildet werden. Es existiert keine weitere Expansionsmöglichkeit mit der Prioritätsstufe Rot. p wird daher auf Prioriotätsstufe Gelb gesetzt. Auf dieser Prioritätsstufe läßt sich der Eintrag $((k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24}) \vee (k_{26} \wedge k_{22} \wedge k_{23} \wedge k_{24}))$ in beide Zweige verzweigend expandieren.

Expansion 2+3 $(X \vee Y)$:



Zu beachten ist, dass in zwei Zweigen die Einträge $(k_{25} \wedge k_{21} \wedge k_{23} \wedge k_{24})$ durch syntaktisch analoge Einträge mit der höherer Prioritätsstufe Gelb ersetzt wurden. Dies muss geschehen, um zu gewährleisten, dass der berechnete Degree nicht zu klein ist. Es verbleiben in jedem Zweig auf Prioritätsstufe Gelb die verallgemeinerten Konjunktionen zur Expansion.

Expansion 4+5+6+7 (verallgemeinerte Konjunktion):



Es können die beiden linken Zweige geschlossen werden, da in die Literale $\neg k_{26}$ und k_{26} bzw. $\neg k_{25}$ und k_{25} auftreten. Der äußerste rechte Zweig kann aufgrund von $\neg k_{26}$ und k_{26} geschlossen werden. Da keine expandierbaren Formeln mit Prioritätsstufe Gelb existieren, wird p nun auf Prioritätsstufe Grün gesetzt. Der verbleibende Tableauzweig kann auch mit Priotitätsstufe Grün nicht abgeschlossen werden, weshalb k_{26} nicht aus der Wissensbasis gefolgert werden kann und somit der Degree von k_{26} 0 ist. Auffallend ist, dass für die Expansion des priorisierten Tableaus 7 Expansion notwendig sind, während innerhalb der Expansion des unpriorisierten Tableaus in Beispiel 1 ab Schritt 2 nur 4 Expansionen notwendig sind.

Beispiel 3 - priorisierte Expansion (2)

Es soll der priorisierte Tableaubeweis von $\left\{ (a \vee b), a \right\} \models_{pt} (a \vee b)$ durchgeführt werden, wobei die beiden Prioritätsstufen Rot und Gelb mit Rot $>$ Gelb existieren. Dabei soll gezeigt werden, warum es notwendig ist, dass sich bei der Expansion im Fall von bis auf die Priorisierung identischen Einträgen im Konfliktfall die höhere Priorität durchsetzen muss, um den korrekten Degree zu berechnen.

Nach dem Algorithmus des priorisierten Tableau-Beweises, ist das Ausgangstableau

$$\begin{array}{c} (a \vee b) \\ | \\ a \\ | \\ \neg(a \vee b) \end{array}$$

Da zunächst auf der höchsten Prioritätsstufe expandiert wird, kann an dieser Stelle sowohl $(a \vee b)$ als auch $\neg(a \vee b)$ expandiert werden. Da ein Eintrag verzweigend, der andere nicht verzweigend expandiert werden kann, soll an dierser Stelle die Wahl auf $\neg(a \vee b)$ fallen.

Expansion 1 $\neg(X \vee Y)$:

$$\begin{array}{c} (a \vee b) \\ | \\ a \\ | \\ \neg(\underline{a} \vee \underline{b}) \\ | \\ \neg a \\ | \\ \neg b \end{array}$$

Ein nicht priorisiertes Tableau kann an dieser Stelle Aufgrund des Konflikts von a und $\neg a$ abgeschlossen werden. Die Abschlussprüfung von priorisierten Tableaus verlangt aber, dass die konfligierenden Einträge auf der aktuellen oder einer höheren Prioritätsstufe liegen müssen. Da die aktuelle Prioritätstufe Rot ist, kann an dieser Stelle kein Abschluss gebildet werden. Es bleibt auf Prioritätsstufe Rot der verzweigende Eintrag $(a \vee b)$ zu expandieren.

Expansion 2 $(X \vee Y)$**:**

$$\begin{array}{ccc} (\underline{a} \vee \underline{b}) & & \\ | & & \\ \neg(a \vee b) & & \\ | & & \\ \neg a & & \\ | & & \\ \neg b & & \\ | & \searrow & \\ a & & a \\ & & | \\ & & b \end{array}$$

Es wurde im linken Zweig der bestehende Eintrag a durch a ersetzt, da nach Anforderung des Algorithmus sich die höhere Priorität durchsetzt. Damit können beide neu entstandenen Zweige aufgrund der Konflikte a und $\neg a$ bzw. b und $\neg b$ mit der Prioritätstufe Rot abgeschlossen werden. Wäre im linken Zweig der Eintrag a bestehen geblieben, so hätte dieser Zweig erst nach der Reduzierung der aktuellen Prioritätsstufe abgeschlossen werden können, was dazu gef“uhrt hätte, dass ein zu niedriger Degree von $(a \vee b)$ berechnet worden wäre.

Anhang C

Tableaus aus der Performanzuntersuchung

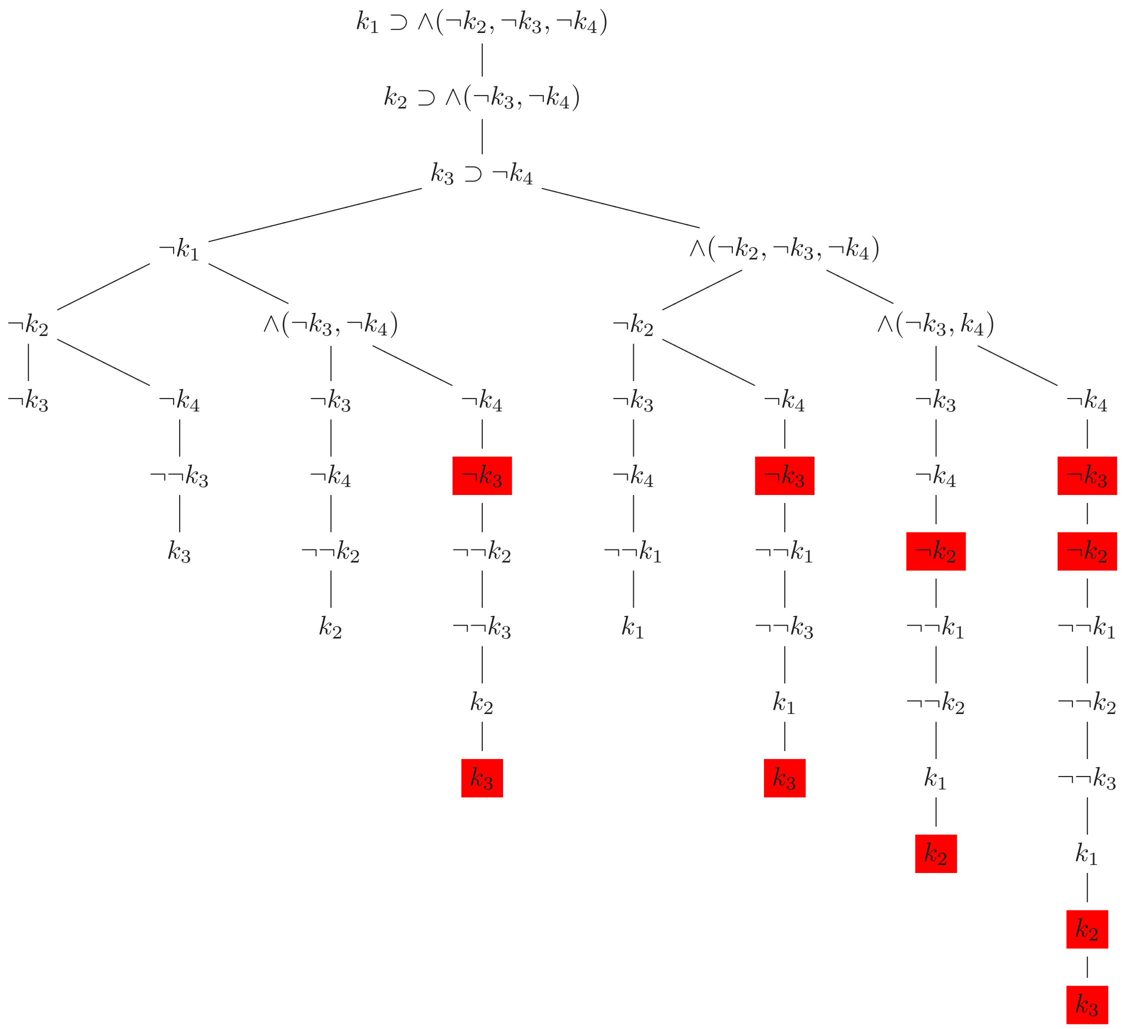


Abbildung C.1: Das aus der Expansion von θ_{ex} resultierende Tableau - ohne Einbeziehung der Zweigersparnis

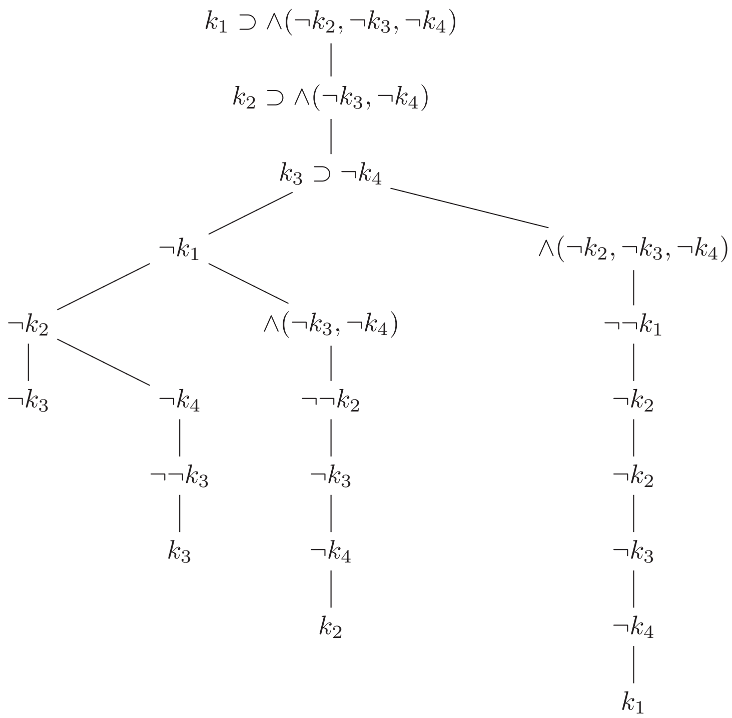


Abbildung C.2: Das aus der Expansion von θ_{ex} resultierende Tableau - unter Einbeziehung der Zweigersparnis. Im rechten Zweig, der durch die Expansion von $k_1 \supset \wedge(\neg k_2, \neg k_3, \neg k_4)$ entstanden ist, liegen bereits die Koreferenzen $\neg k_2$ und $\neg k_3$ vor. Da die Expansion von $k_2 \supset \wedge(\neg k_3, \neg k_4)$ diesem Zweig das Literal $\neg k_2$ hinzufügen würde, greift die Zweigersparnisregel. Das gleich gilt für $k_3 \supset \neg k_4$ und dem Literal $\neg k_3$ Analoges trifft auf die übrigen Zweige ebenfalls zu.

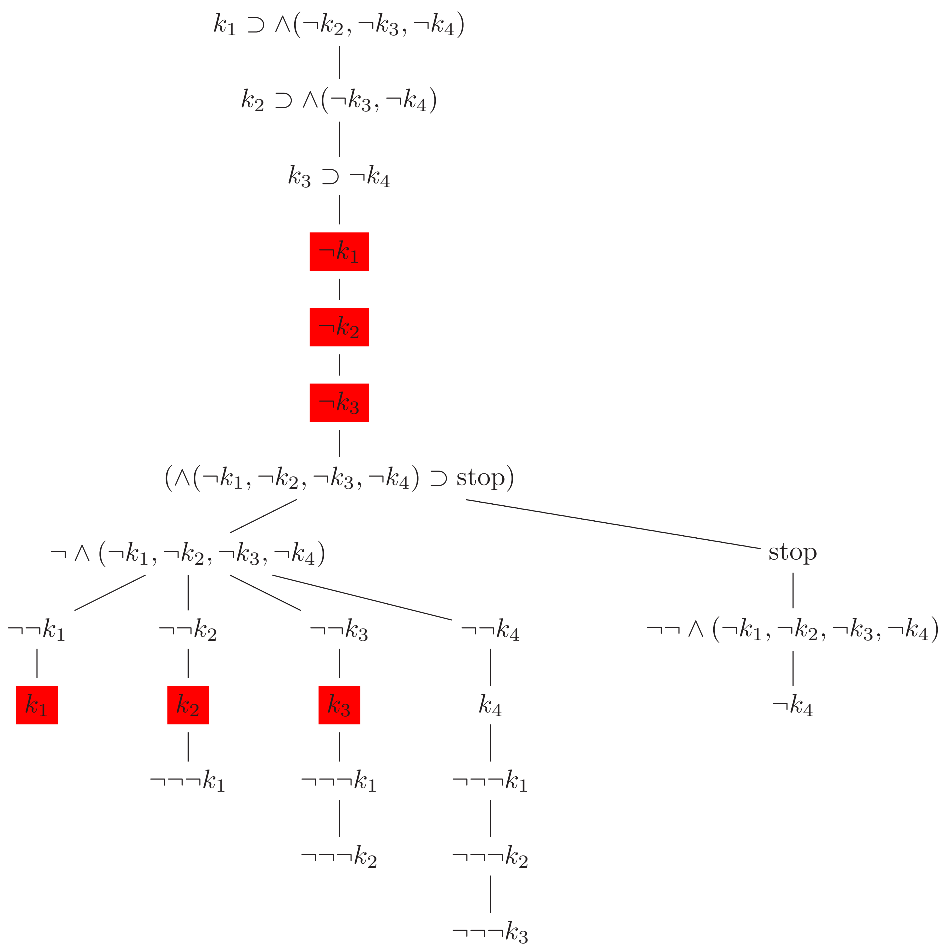


Abbildung C.3: Expansion der durch θ_s erzeugten Implikation in einen der offenen Zweige aus Abbildung C.1 - nach der ursprünglichen Definition von θ_s

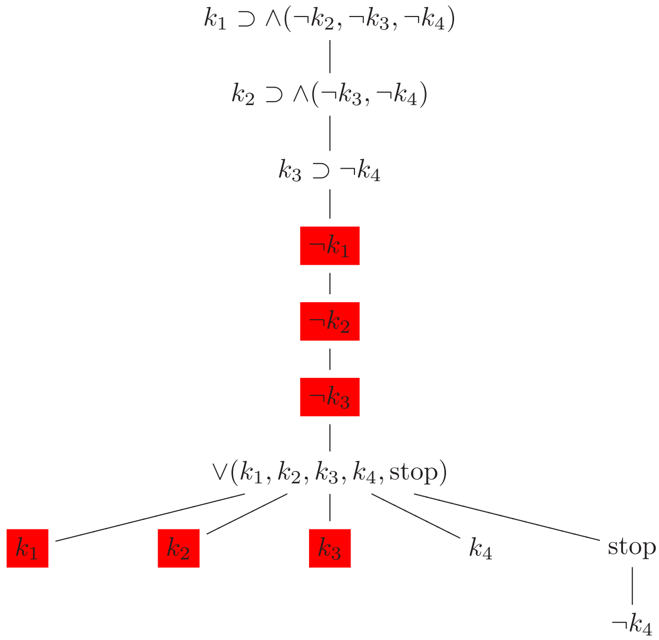


Abbildung C.4: Expansion der durch θ_s erzeugten Implikation in einen der offenen Zweige aus Abbildung C.1
- unter Verwendung der modifizierten Definition von θ_s

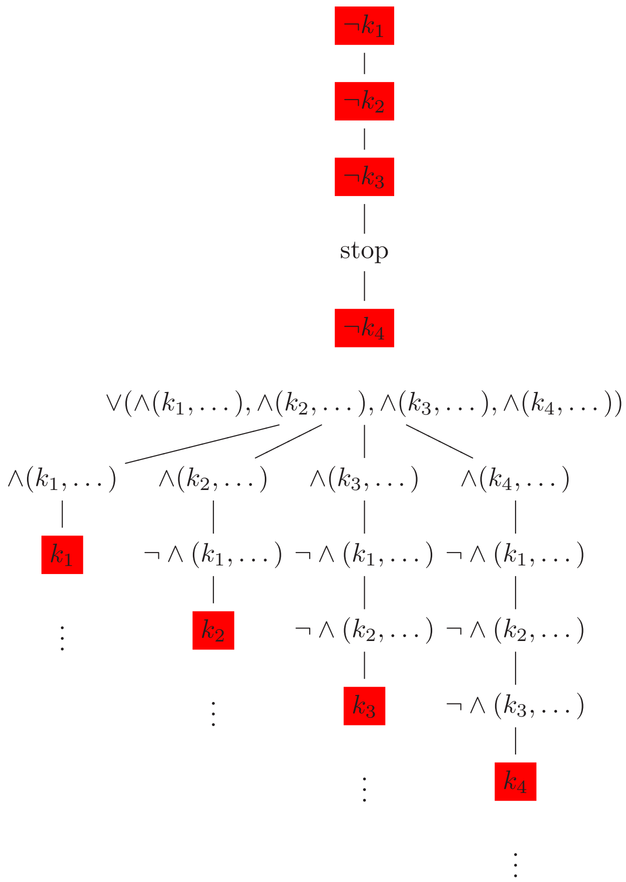


Abbildung C.5: Expansion der höchst gewerteten Disjunktion des Beispiels in einen Zweig vom Typ br_{stop}

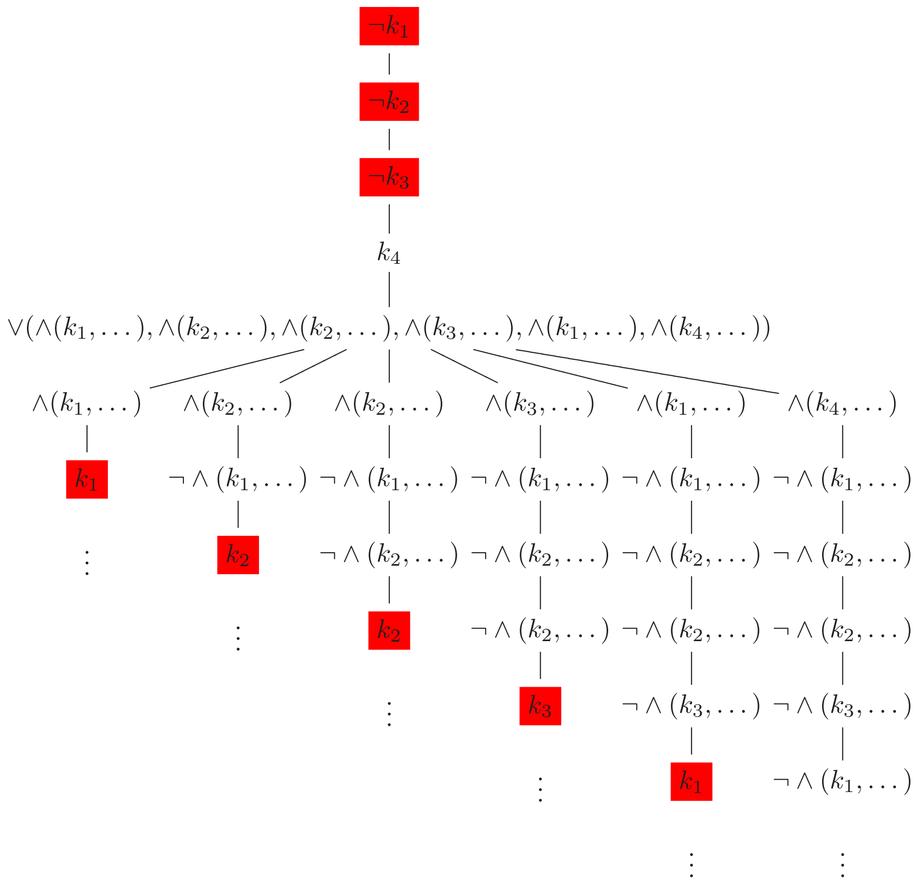


Abbildung C.6: Expansion der höchst gewerteten Disjunktion $\theta_m(s,6)$ des Beispiels in einen Zweig vom Typ br_i - hier br_4

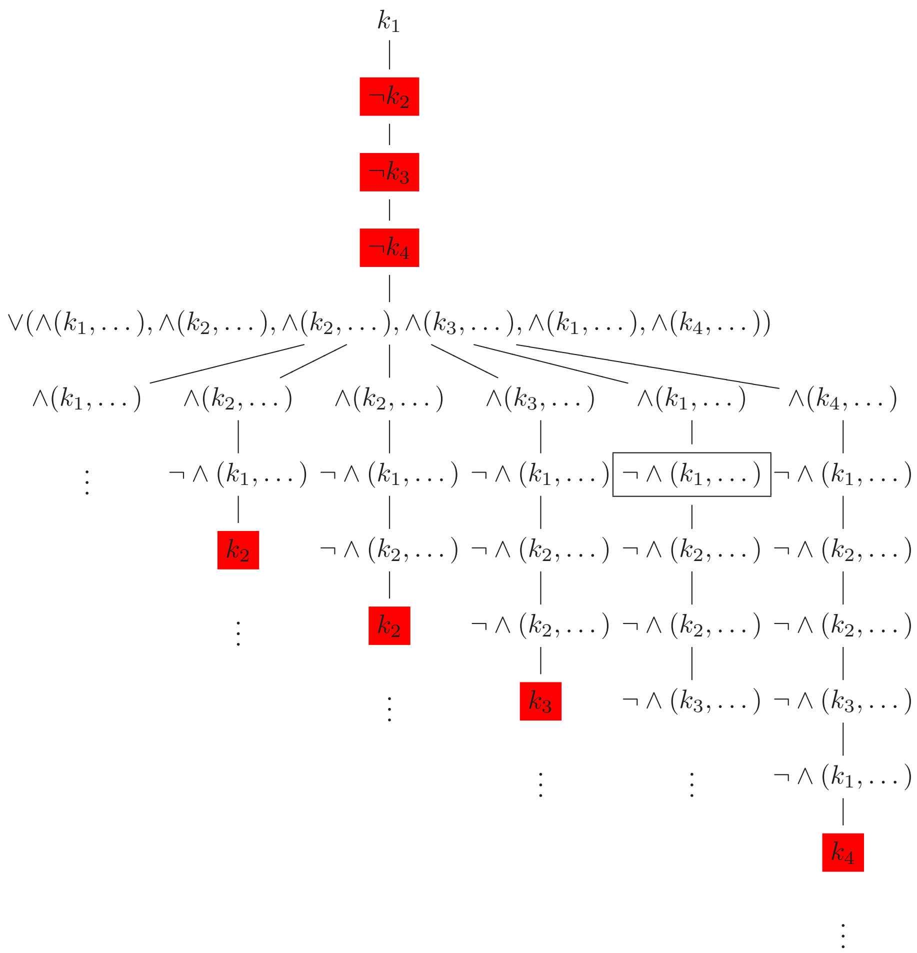


Abbildung C.7: Expansion der höchst gewerteten Disjunktion $\theta_m(s, 6)$ des Beispiels in einen Zweig vom Typ br_i - hier br_1. Die umrahmte negierte Konjunktion muss expandiert werden. Die umrahmte negierte Konjunktion repräsentiert den Ausschluss eines Matches, das verhaltensäquivalent zu dem in Zweig brm_5 expandierten Match ist.

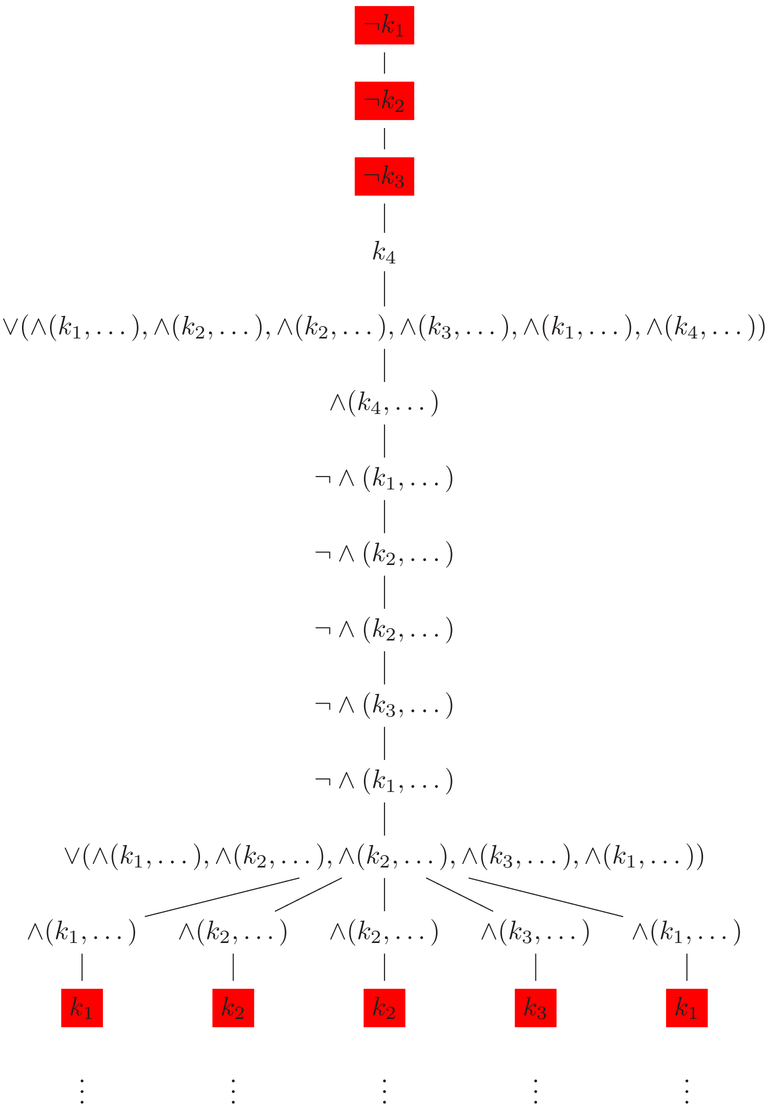


Abbildung C.8: Expansion der Disjunktion $\theta_m(s,3)$ des Beispiels in einen Zweig vom Typ brm_6. Es ist erkennbar, dass alle neu entstandenen Zweige geschlossen werden können

Anhang D

Aktionspläne

Nachfolgend werden die Aktionspläne zu einigen Instrukionen dargestellt. Bei der Darstellung wird die Temporale Reihenfolge nicht explizit spezifiziert. Es wird angenommen, dass die Aktionen in der Reihenfolge ihrer Nummerierung ausgeführt werden. Die Instruktion lautet: „geh nach links dot geh zwischen haus b und haus c durch dot geh nach rechts dot dann stehst du vor haus e dot “

Aktionsplan zu NBTest01

1. GO (w_5)
2. GO (w_6)
3. GO (w_4)
4. BE_AT (r_8)

In obigem Beispiel ist erkennbar, dass der Agent im wesentlichen durch GO Anweisungen instruiert wird. Jede GO Anweisung erwartet einen Wegpunkt als Ziel. Eine Ausnahme ist die letzte Anweisung, die den Agenten veranlasst zu prüfen, ob er sich in seinem Ziel befindet. Die natürlichsprachliche Instruktion lautet: „geh zu haus f dot geh zwischen haus f und haus d durch dot geh nach links dot dann stehst du vor haus e dot“

Aktionsplan zu NBTest05

1. GO (w_5)
2. GO (w_6)
3. GO (w_4)
4. BE_AT (r_8)

Der Aktionsplan zu NBTest05 ist identisch zu dem von NBTest01.

Aktionsplan zu CETest01

Die Instruktion CETest01 soll den Agenten vom Eingang aus nach Haus E navigieren lassen.

1. BE_AT (r_{11})

2. VIEW (b_3)
3. GO (w_7)
4. GO (w_5)
5. GO (w_6)
6. VIEW (b_2)

Der Aktionsplan CETest01 umfasst ein größeres Inventar an Anweisungstypen. Es ist erkennbar, dass auch hier die GO Anweisungen überwiegen.

Literaturverzeichnis

[AM 1985] ALCHOURRÓN, C. E. und MAKINSON, D. On the Logic of Theory Change: Safe Contraction In : *Studio Logica, Volume 44 Number 4* S. 405 - 422, 1985

[AGM 1985] ALCHOURRÓN, C. E., GÄRDENFORS, P. und MAKINSON, D. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*. S 510 - 530. 1985

[BA01] BEN-ARI, M. *Mathematical Logic for Computer Science*, Springer, London. 2nd Edition

[B 2005] BITTKOWSKI, N. *Aktionsplanung und -steuerung unter Unsicherheit bei der Navigation eines Geometrischen Agenten mit Hilfe von Wegbeschreibungen*, FB Informatik, Universität Hamburg. Hamburg, 2005

[BR99] BRONSTEIN, I.N., SEMENDJAJEW, K.A., MUSIOL, G., MÜHLIG, H., *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 4. Auflage, 1999

[DP 1996] DARWICHE, A. und PEARL, J.. On the Logic of Iterated Belief Revision. *Proceedings of the 5th Conference on Theoretical Aspects of Reasoning about Knowledge*. Morgan Kaufman, Pacific Grove, CA, 1994. S 5 - 23´

[FITTING 1996] FITTING, M. : *First-Order Logic and Automated Theorem Proving*. Second Edition. Springer Verlag New York. 1996

[GM 1988] GÄRDENFORS, P. und MAKINSON, D. Revisions of knowledge systems using epistemic entrenchment. *Proceedings of the 2nd conference on Theoretical aspects of reasoning about knowledge*. Pacific Grove, California. S 83 - 95. 1988

[G 1988] GÄRDENFORS, *P. Knowledge in Flux: Modeling the Dynamics of Epistemic States.* Bradford Books, The MIT Press, Cambridge Massachusetts. 1988.

[GOTTWALD 1989] GOTTWALD, S. *Mehrwertige Logik. Berlin.* Akademieverlag (1989)

[GR 1995] Belief Revision. GÄRDENFORS, P. und ROTT, H. In *Handbook of Logic in Artificial Intelligence and Logic Programming. Volume IV.* Dov M. Gabbay, Christopher J. Hogger und John A. Robinson (eds). Oxford. Oxford University Press. S 35 - 132 , 1995

[H 1986] HABEL, Ch. : *Prinzipien der Referentialität.* Springer, Berlin 1986

[H 2003] HELWICH, J. H. *Graphenbasierte Navigation eines Geometrischen Agenten: Integration von Perzeption und Instruktion.* FB Informatik, Universität Hamburg. Hamburg, 2003

[HSP99] HORROCKS, I. und PATEL-SCHNEIDER, P.F. Optimising Description Logic Subsumption. *Journal of Logic and Computation, 9:3*, June 1999, S 267-293

[KM 1991] KATSUNO, H. und MENDELZON, A. On the Differences Between Updating a Knowledge Base and Revising it. In : J.F. Allen, R.Fikes und E. Sanderwall (eds). *KR"91:Principles of Knowledge Representation and Reasoning*, Morgan Kaufman, San Mateo, California, 1991. S. 387-394

[PMG 1998] POOLE, D., MACKWORTH A. , GOEBEL G.G. Computational Intelligence, Oxford University Press, 1998

[RN 2003] RUSSELL, S. und P. NORVIG (2003). *Artificial Intelligence. A Modern Approach.* Prentice Hall, Upper Saddle River, N.J., USA.

[S 1987] SCHÖNING, U. *Logik für Informatiker.* Spektrum Akademischer Verlag; Heidelberg, Berlin; 5. Auflage, 1987

[SW88] SPOHN, W. Ordinal Conditional Functions. A Dynamic Theory of Epistemic States, in: W.L. Harper, B. Skyrms (Hg.), *Causation in Decision, Belief Change, and Statistics, vol. II,* Kluwer, Dordrecht 1988, S.105-134

[TSEHK 2003] TSCHANDER, L.; SCHMIDTKE, H. R.;ESCHENBACH, C.; HABEL, CH.; KULIK, L. A Geometric Agent Following Route Instructions. In *Spatial Cognition III*, Springer Berlin/Heildelberg. 2003

[W 1998] WILLIAMS, M.-A. Applications of Belief Revision. In *Transactions and Change in Logic Databases* von Freitag et al(eds). Springer Verlag Berlin, 1998, S 285-314

[W 1997] WILLIAMS, M, A. Anytime belief revision. *Proc. 15th Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, Aug. 23-29, 74-79. 1997

[W 1996] WILLIAMS, M. A. Towards a Practical Approach to Belief Revision: Reason-Based change, Luigia Carlucci Aiello und C. Shapiro (Hrsg.), *Proceedings of the Fifth International Conference on Principles of Knowledge on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann Publishers, 412-421, 1996

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbtständig und ohne fremde Hilfe angefertigt und mich anderer als der als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnomen wurden, sind als solche kenntlich gemacht. Ich bin mit der Einstellung in den Bestand der Bibliothek des Departments Informatik einverstanden.

Hamburg, den