Formale Untersuchung der Konstruktion einer Closed-World-Assumption in beschreibungslogischen Sprachen

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

von

Oliver Gries
OliverGries@gmx.net
Studiengang Informatik
Matr.-Nr.: 5225314

Erstbetreuerin: Dr. Carola Eschenbach
Universität Hamburg

Zweitbetreuer: Prof. Dr. Ralf Möller
Technische Universität Hamburg-Harburg

Hamburg, den 15. Mai 2008

Zusammenfassung

Die Menge der impliziten Assertionen, die in beschreibungslogischen Sprachen ermittelt werden kann, enthält oft weniger Elemente als erwünscht, weil viele Schlussfolgerungen, die natürlich erscheinen, nicht logisch korrekt sind. In diesen Fällen besteht häufig Interesse an einer Closed-World-Assumption (CWA). Da beschreibungslogische Wissensbasen Konzeptassertionen mit qualifizierten Rollenrestriktionen enthalten, ist die Bildung einer CWA allerdings aufwändig. Zusätzlich können unter dieser Annahme durch die Spezifikation indefiniter Informationen Inkonsistenzen entstehen. Ein Lösungsansatz dieser Problematik ist die Erweiterung beschreibungslogischer Anfragen um einen epistemischen Operator. Anhand der Anfragesprache $\mathcal{ALCK}$ sowie der new Racer Query Language (nRQL) wird dargestellt, in welchen Fällen durch die Berücksichtigung dieses Operators eine lokale CWA getroffen werden kann. Der Lösungsansatz, der den Schwerpunkt dieser Untersuchung bildet, ist die Einbeziehung einer Generalized Closed-World-Assumption (GCWA). Dazu erfolgt die mögliche Identifikation von unvollständig spezifizierten Rollenfüllern mit allen Konstanten eines erweiterten Domänenabschlusses. Es wird ein Verfahren für die Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ vorgestellt, dass in polynomieller Zeit entscheidet, ob eine Assertion unter der GCWA folgt oder nicht. Im Anschluss daran wird ein Verfahren vorgestellt, dass eine lokale GCWA in den Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ durch Überführung von Konzeptassertionen in eine prädikatenlogische Klauselform einbezieht.

Danksagung

Ich möchte mich bedanken bei

- Dr. Carola Eschenbach für die intensive Betreuung und ihren Anspruch an diese Arbeit
- Prof. Ralf Möller für sein Interesse an epistemischen Operatoren
- meinen Eltern Volker Gries und Ulla Gries, geb. Comes, die mir das Studium finanziert und mir viele Freiheiten gelassen haben
- Karin Lohmüller für die liebevolle Unterstützung

Inhaltsverzeichnis

Kapitel 1

Einführung

1.1 Problematik

Beschreibungslogische Sprachen sind in den meisten Fällen entscheidbare Fragmente der Prädikatenlogik. In endlicher Zeit kann entschieden werden, ob ein Axiom logisch aus einer Wissensbasis folgt oder nicht. Es können terminologische Axiome spezifiziert werden, die es ermöglichen, ein neues Konzept innerhalb einer Begriffshierarchie zu klassifizieren, und es können assertionale Axiome explizit festgelegt oder implizit geschlussfolgert werden.

Die Menge der impliziten Assertionen, die in beschreibungslogischen Sprachen ermittelt werden kann, enthält allerdings oft weniger Elemente als erwünscht, weil viele Schlussfolgerungen, die natürlich erscheinen, nicht logisch korrekt sind. Ein einfaches Beispiel, das diese Problematik ausführt, können Verweise zwischen Internetseiten sein:

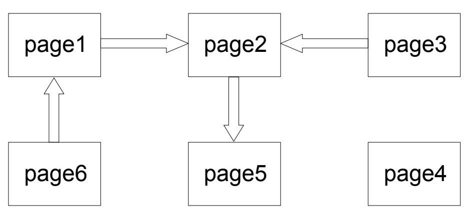


Abbildung 1.1: Internetseiten und ihre Verweise aufeinander

Seiten, auf die verwiesen wird, sind in diesem Beispiel page1, page2 und page5. Besteht Interesse daran, auf welche Seiten *nicht* verwiesen wird, gibt es durch logische Beweisverfahren keine Ergebnisse, weil dem betrachteten Ausschnitt der Welt jederzeit weitere Verweise hinzugefügt werden könnten. Mit der **Closed-World-Assumption (CWA)** kann das erwünschte Resultat erzielt werden. Es handelt sich dabei um die (natürliche) Annahme, dass eine atomare Formel nicht gilt, wenn nicht bewiesen werden kann, dass sie gilt. Sie betrachtet den aktuellen Wissensbestand als vollständig. In relationalen Datenbanken wird diese Annahme vorausgesetzt, in Beschreibungslogiken (und vielen anderen Logiken) hingegen nicht.

Ein Reasoning Service, der in diesem Zusammenhang von Interesse ist, ist der *Assertionstest*, symbolisch $KB \models \alpha$, der prüft, ob eine Assertion α logisch aus einer Wissensbasis KB folgt oder nicht. Da dieser Service keine CWA berücksichtigt, können sich (wie in dem Beispiel der Verweise auf Internetseiten) unerwünschte Fehlschläge von Beweisen ergeben. Aufgrund der Open-World-Assumption logischer Formalismen kann nichts versichert werden, das über den aktuellen Wissensbestand hinausgeht.

Das einführende Beispiel zeigt nur eines von einer Vielzahl von grundsätzlichen Problemen. Beschreibungslogische Wissensbasen bestehen zudem in den meisten Fällen aus Daten, die vielschichtiger und komplexer strukturiert sind. Insbesondere durch die Möglichkeit der Spezifikation qualifizierter Rollenrestriktionen sind Verschachtelungen von Sprachausdrücken möglich, die die Konstruktion einer CWA erschweren. Eine noch größere Problematik stellen indefinite beschreibungslogische Wissensbasen dar. Dabei handelt es sich um Wissensbasen, in denen Konzeptdisjunktionen spezifiziert sind oder unbekannte, aber existente Rollenfüller berücksichtigt werden müssen. Ist eine beschreibungslogische Wissensbasis indefinit, ist nicht gewährleistet, dass die Konsistenz unter der CWA erhalten bleibt.

1.2 Stand der Wissenschaft

Im Kontext der Prädikatenlogik sind Annahmen einer geschlossenen Welt ein weit erforschtes Gebiet. Die CWA nach [Reiter, 1978] ist die grundlegende Definition einer Annahme einer geschlossenen Welt. Eine Wissensbasis wird unter dieser Annahme für alle atomaren Formeln, die aus ihr nicht logisch folgen, um entsprechende negierte atomare Formeln *implizit* vervollständigt (ein ähnlicher Formalismus ist die Negation-as-failure-Semantik für logische Programmiersprachen [Clark, 1978]). In [Reiter, 1978] wird bewiesen, dass die CWA in Horn-Wissensbasen die Konsistenz erhält, dass diese Eigenschaft aber nicht in indefiniten Wissenbasen gilt. Um diese Problematik zu umgehen, wird in [Minker, 1982] die Generalized Closed-World-Assumption (GCWA) definiert, die die logische Folgerung von Klauseln aus einer indefiniten Wissensbasis einbezieht. Die Berechnung negierter atomarer Formeln, die unter Berücksichtigung der GCWA aus einer Wissensbasis folgen, ist allerdings äußerst aufwändig. Verfahren, die Anfragen unter dieser Annahme beantworten, sind in [Fernández und Minker, 1992] aufgeführt. Beispielsweise erfolgt in [Grant und Minker, 1986] die Beantwortung konjunktiver Anfragen unter der GCWA an eine extensionale Wissensbasis und in [Henschen und Park, 1988] die Ausführung von einfachen Anfragen unter dieser Annahme an Wissensbasen, die auch intensionales Wissen enthalten können. Neben der GCWA gibt es viele weitere generalisierte Annahmen, wie zum Beispiel die Weak Generalized Closed-World-Assumption (WGCWA) [Rajasekar et al., 1989]. Andere Ansätze, durch die Annahmen einer geschlossenen Welt berücksichtigt werden können, sind die Einbeziehung von Vervollständigungsaxiomen [Clark, 1978] [Reiter, 1984] sowie die Verwendung eines epistemischen Operators **K** [Levesque, 1984] [Reiter, 1990].

Im Kontext von beschreibungslogischen Sprachen ist aufgrund unvollständig spezifizierter Rollenfüller durch Existenzrestriktionen $\exists R.C$ und mindestens-Anzahlrestriktionen $(\geq n\, R)$ zunächst nicht festgelegt, wie eine Erweiterung einer Wissensbasis um negierte atomare Formeln erfolgen kann. Der einzige vorliegende nicht-monotone Ansatz, der eine Erweiterung einer beschreibungslogischen Wissensbasis um eine Annahme einer geschlossenen Welt berücksichtigt, ist die Einbeziehung einer Zirkumskription in Sprachen mit Existenzrestriktionen [Cadoli et al., 1990]. Ausführliche Untersuchungen über Annahmen einer geschlossenen Welt in beschreibungslogischen Sprachen wurden allerdings nicht vorgefunden.

Stattdessen liegt der Schwerpunkt der Untersuchungen auf einer Erweiterung der Anfragesprache. In [Donini et al., 1992] und [Schaerf, 1994a] werden $\mathcal{ALC}$-Anfragen um den von Levesque und Reiter eingeführten epistemischen Operator $\mathbf{K}$ erweitert. Diese Untersuchungen werden insbesondere in [Donini et al., 1998] und [Donini et al., 2002] vertieft und in [Grimm und Motik, 2005] im Kontext des Semantic Web dargestellt. Die Ergänzung beschreibungslogischer Anfrageassertionen um $\mathbf{K}$ ermöglicht sowohl die ausschließliche Berücksichtigung einer Wissensbasis bekannter Rollenfüller als auch die Ausgabe von Assertionen, für die nicht bewiesen werden kann, dass sie aus einer Wissensbasis logisch folgen. Letzteres entspricht der Einbeziehung einer Negation-as-failure-Semantik.

Eine Anfragesprache, die einen Negation-as-failure-Operator für eine sehr ausdrucksstarke beschreibungslogische Sprache bereitstellt, ist die new Racer Query Language (nRQL) [Racer Systems, 2007] [Wessel und Möller, 2005]. Insbesondere durch die Verwendung dieses Operators können in nRQL viele epistemische Anfragen formuliert werden.

1.3 Vorstellung und Einordnung der Ergebnisse

In dieser Arbeit werden für ausgewählte beschreibungslogische Sprachen Verfahren formalisiert, die unter Berücksichtigung einer Form der CWA entscheiden, ob eine Assertion aus einer Wissensbasis folgt oder nicht.

Entwickelte Verfahren für atomare Wissensbasen und $\mathcal{AL}_0$-Wissensbasen, die die CWA nach [Reiter, 1978] einbeziehen, entsprechen im Wesentlichen den in [Brachman und Levesque, 2004, Kapitel 16] vorgestellten rekursiven Verfahren für lebendige prädikatenlogische Wissensbasen und berücksichtigen Anfragen mit Werte-, Existenz- und Anzahlrestriktionen.

Für die Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ wird ein polynomielles Verfahren vorgestellt, das korrekt und (unter Einschränkungen der Wissensbasis) vollständig entscheidet, ob eine Assertion unter der GCWA aus einer Wissensbasis folgt oder nicht. Da alle Rollenrestriktionen, die zu der Einbeziehung von unvollständig spezifizierten Rollenfüllern führen, unqualifiziert sind, konnte das Verfahren ohne die direkte Berücksichtigung von Klauseln formalisiert werden.

Für die aussagenlogisch abgeschlossenen Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ wird eine GCWA durch Überführung der Anfrage sowie aller in einer Wissensbasis spezifizierten Assertionen in eine prädikatenlogische Klauselform berücksichtigt. Da aus der Anfrage resultierende negative Literale direkt auf Folgerung unter der GCWA überprüft werden können, kann die Einbeziehung dieser GCWA lokal erfolgen.

Die Ergebnisse dieser Arbeit sind relevant, da sich aktuelle wissenschaftliche Untersuchungen im Wesentlichen mit epistemischen Anfragen beschäftigen und bei der Ausführung dieser Anfragen keine Annahmen getroffen werden können. Stattdessen werden bei der Ausführung epistemischer Anfragen ausschließlich Aussagen über lokale Überprüfungen, ob eine Assertion in allen Modellen der verwendeten Wissensbasis gilt oder nicht, zu einer Aussage zusammengeführt, so dass ein Benutzer im Allgemeinen nicht weiß, ob konsistent angenommen werden kann, dass gefolgerte Assertionen ohne einen vorangestellten epistemischen Operator gelten. Im Gegensatz zu einer Anfrage unter Berücksichtigung einer generalisierten Annahme einer geschlossenen Welt sind zum Beispiel die epistemischen Anfragen $KB \models \neg \mathbf{K} A(a)$ und $KB \models \neg \mathbf{K} B(a)$ mit $KB = \{(A \sqcup B)(a)\}$ erfolgreich, obwohl $KB \cup \{\neg A(a), \neg B(a)\}$ inkonsistent ist.

Die Verfahren für atomare Wissensbasen und $\mathcal{AL}_0$-Wissensbasen können als Grundlage der Berücksichtigung einer CWA angesehen werden, werden aber aufgrund ausdrucksstärkerer Anfragesprachen relationaler Datenbanken vermutlich selten von Nutzen sein. Dagegen kann das Verfahren, das für die nicht ausdrucksstarken Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ in polynomieller Zeit entscheidet, ob eine Assertion unter der GCWA aus einer Wissensbasis folgt, für bestimmte zu modellierende Domänen von Bedeutung sein. Das für die Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ vorgestellte Verfahren schließlich eignet sich nicht für große Wissensbestände, da die Ausführung hochgradig exponentielle Berechnungszeiten erfordert. Es zeigt aber auf, wie eine Annahme einer geschlossenen Welt in ausdrucksstärkeren beschreibungslogischen Sprachen erfolgen kann. In weniger großen Wissensbeständen ist ein solches Verfahren insbesondere wegen der lokalen Berücksichtigung der GCWA von Bedeutung.

1.4 Übersicht über die Arbeit

In Kapitel 2 werden die Syntax und Semantik der untersuchten beschreibungslogischen Sprachen sowie verwendete Reasoning Services und Tableau-Kalküle erläutert. Im Anschluss daran (Kapitel 3) werden die Auswirkungen der Open-World-Assumption (OWA) auf beschreibungslogische Assertionstests betrachtet. Anhand vieler Beispiele wird dargestellt, aus welchen Gründen diese Reasoning Services oft unerwünscht fehlschlagen. Außerdem wird gezeigt, dass unter der OWA bei Anfragen mit Existenzrestriktionen Fallunterscheidungen berücksichtigt werden müssen. In Kapitel 4 werden Annahmen einer geschlossenen Welt im Kontext der Prädikatenlogik untersucht. Es werden die Definitionen der CWA [Reiter, 1978], der GCWA [Minker, 1982] und der WGCWA [Rajasekar et al., 1989] sowie wichtige Eigenschaften dieser Annahmen erläutert. Das Kapitel schließt mit der Darstellung der Problematik ab, die sich durch die Berücksichtigung dieser Annahmen in beschreibungslogischen Sprachen ergibt.

Das fünfte Kapitel beschäftigt sich mit der Erweiterung beschreibungslogischer Anfragen um einen epistemischen Operator. Nach der Definition der Syntax und Semantik epistemischer Anfragen wird anhand vieler Beispielanfragen aufgezeigt, in welchen Fällen diese Anfragen eine lokale CWA berücksichtigen. Das darauf folgende Kapitel beschäftigt sich mit der *new Racer Query Language (nRQL)*. Es wird gezeigt, dass in nRQL viele epistemische Anfragen formuliert werden können. Insbesondere wird dabei die Negation-as-failure-Semantik dieser Anfragesprache untersucht.

In Kapitel 7 werden für ausgewählte beschreibungslogische Sprachen Verfahren konstruiert, die die Ausführung von Assertionstests unter Berücksichtigung von Annahmen einer geschlossenen Welt ermöglichen. Zunächst werden Sprachen mit atomaren und $\mathcal{AL}_0$-Wissensbasen untersucht. Es wird gezeigt, dass $\mathcal{ALCN}$-Anfragen an diese Wissensbasen unter Berücksichtigung der CWA nach [Reiter, 1978] korrekt und vollständig beantwortet werden können. Für alle in dieser Arbeit untersuchten beschreibungslogischen Sprachen, die $\mathcal{AL}_0$ erweitern, wird die GCWA [Minker, 1982] einbezogen. Um Inkonsistenzen durch einzubeziehende Skolem-Konstanten zu vermeiden, wird für die Berücksichtigung von Rollenrestriktionen ein um neue Konstanten erweiterter Domänenabschluss verwendet. Es wird ein Verfahren für die Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ vorgestellt, dass in polynomieller Zeit entscheidet, ob eine Assertion unter der GCWA aus einer Wissensbasis folgt oder nicht. Im Anschluss daran erfolgt die Darstellung eines Verfahrens, dass die Berücksichtigung einer lokalen GCWA für die Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ durch Überführung von Konzeptassertionen in eine prädikatenlogische Klauselform ermöglicht. Die Arbeit schließt in Kapitel 8 mit einer Zusammenfassung der Ergebnisse sowie einem Ausblick auf weitere Untersuchungen ab.

Kapitel 2

Beschreibungslogiken

Dieses Kapitel ist eine Einführung in Beschreibungslogiken. In Abschnitt 2.1 werden Ausdrücke beschreibungslogischer Sprachen behandelt, die in dieser Untersuchung relevant sind. Danach folgen eine ausführliche Definition der Semantik dieser Ausdrücke (Abschnitt 2.2) und eine Erläuterung der in dieser Arbeit verwendeten Reasoning Services (Abschnitt 2.3). Das Kapitel schließt mit der Darstellung beschreibungslogischer Tableau-Kalküle (Abschnitt 2.4).

2.1 Sprachausdrücke

Das Vokabular beschreibungslogischer Sprachen stellt *Konzepte*, *Rollen* und *Konstanten* bereit. Konzepte bezeichnen Mengen von Objekten, Rollen binäre Relationen zwischen Objekten und Konstanten ein spezielles Objekt. Bei Konzepten handelt es sich um *atomare Konzepte* oder um *komplexe Konzepte*, auch Konzeptbeschreibungen genannt, die mit Konzeptkonstruktoren aus atomaren Konzepten und Rollen zusammengesetzt sind. Jede Teilzeichenkette einer Konzeptbeschreibung, die ein Konzept ist, wird *Teilkonzept* genannt. Im Weiteren repräsentieren A ein atomares Konzept, P und R atomare Rollen, C und D Konzeptbeschreibungen und a und b Konstanten. In den beiden folgenden Unterabschnitten werden die in dieser Arbeit verwendeten beschreibungslogischen Sprachen und Wissensbasen erläutert.

2.1.1 Verwendete Sprachen

Eine beschreibungslogische Sprache ergibt sich aus den Konzeptkonstruktoren, die sie zur Verfügung stellt. Die hier verwendeten Sprachen basieren auf denen der $\mathcal{AL}$-Familie (attributive languages) [Schmidt-Schauß und Smolka, 1991]. Zunächst wird die Sprache $\mathcal{AL}_0$ eingeführt [Donini et al., 1992]. Konzeptbeschreibungen in dieser Sprache werden durch die folgende Syntaxregel geformt:

$$C, D \longrightarrow \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid \forall R.C$$

$\mathcal{AL}_0$ besitzt die geringste Ausdrucksstärke aller in dieser Arbeit verwendeten beschreibungslogischen Sprachen. Außer atomaren Konzepten und Rollen beinhaltet sie die logischen Konstanten $\top$ und $\bot$, Negationen von atomaren Konzepten ($\neg A$), Konjunktionen von Konzepten ($C \sqcap D$) und Werterestriktionen ($\forall R.C$).

Die Sprache $\mathcal{AL}$ ist die grundlegende Sprache, die von praktischem Interesse ist [Schmidt-Schauß und Smolka, 1991]. Sie erweitert $\mathcal{AL}_0$ um unqualifizierte Existenzrestriktionen $\exists R.\top$:

$$C, D \longrightarrow \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.\top$$

Ausdrucksstärkere Sprachen ergeben sich durch das Hinzufügen weiterer Konzeptkonstruktoren zu der Sprache $\mathcal{AL}$. Dazu gehören Disjunktionen von Konzepten $C \sqcup D$ (bezeichnet durch $\mathcal{U}$), qualifizierte Existenzrestriktionen $\exists R.C$ (bezeichnet durch $\mathcal{E}$), negierte Konzeptbeschreibungen $\neg C$ (bezeichnet durch $\mathcal{C}$) und Anzahlrestriktionen $(\geq n\, R)$ und $(\leq n\, R)$ (bezeichnet durch $\mathcal{N}$). Durch die Äquivalenzen $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ und $\exists R.C \equiv \neg\forall R.\neg C$ können Konzeptdisjunktionen und qualifizierte Existenzrestriktionen durch negierte Konzeptbeschreibungen ausgedrückt werden, so dass die Verwendung dieser Operatoren in Sprachen mit $\mathcal{C}$ (an Stelle von $\mathcal{CUE}$) notiert wird. So entspricht beispielsweise die Sprache $\mathcal{ALC}$ der Erweiterung von $\mathcal{AL}$ um Disjunktionen von Konzepten, qualifizierten Existenzrestriktionen und negierten Konzeptbeschreibungen. Die beschreibungslogischen Sprachen, die in dieser Arbeit behandelt werden, sind:

- Sprachen mit polynomieller Zeitkomplexität: $\mathcal{AL}_0$, $\mathcal{AL}$ und $\mathcal{ALN}$
- aussagenlogisch abgeschlossene Sprachen: $\mathcal{ALC}$ und $\mathcal{ALCN}$

2.1.2 Wissensbasen

Eine beschreibungslogische Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$ besteht aus einer TBox $\mathcal{T}$ und einer ABox $\mathcal{A}$. Sowohl die TBox als auch die ABox bestehen aus einer Menge von *Axiomen*. Die TBox beschreibt die Terminologie einer Wissensbasis und enthält intensionales Wissen. In dieser Arbeit wird angenommen, dass die TBox ausschließlich aus terminologischen Axiomen der Form

$$A \sqsubseteq C, \qquad A \equiv C$$

besteht, in denen A ein atomares Konzept und C ein beliebiges Konzept der jeweiligen Sprache bezeichnen. Axiome der Form $A \sqsubseteq C$ werden *Inklusionen* und Axiome der Form $A \equiv C$ werden *Definitionen* genannt. Es wird weiterhin angenommen, dass jedes atomare Konzept höchstens einmal auf der linken Seite dieser Axiome vorkommt und dass die TBox keine Zyklen enthält [Baader und Nutt, 2003]. Terminologische Axiome mit Rollen, $R \sqsubseteq S$ bzw. $R \equiv S$, werden in dieser Arbeit nicht behandelt. Die ABox beschreibt Zusicherungen (Assertionen) über Konstanten und enthält extensionales Wissen. Eine ABox besteht in dieser Untersuchung aus Axiomen der Form

$$C(a), \qquad R(a,b).$$

Der Buchstabe C steht für eine Konzeptbeschreibung, R für eine atomare Rolle und a und b für Konstanten der Sprache. Axiome der Form $C(a)$ werden als *Konzeptassertionen* und Axiome der Form $R(a,b)$ als *Rollenassertionen* bezeichnet. In einer stark vereinfachten Betrachtungsweise kann die ABox als Instanz einer relationalen Datenbank betrachtet werden, die nur unäre und binäre Relationen enthält. Es wird angenommen, dass die betrachtete TBox bei jedem Inferenzprozess eliminiert wird [Baader und Nutt, 2003]. Eine beschreibungslogische Wissensbasis wird auf diese Art und Weise zu einer bzgl. einer TBox expandierten ABox $\mathcal{A}_\mathcal{T}$ reduziert, so dass $KB = (\{\}, \mathcal{A}_\mathcal{T})$.

2.2 Interpretationen und Modelle

Die Semantik von beschreibungslogischen Sprachen wird mit *Interpretationen* $\mathcal{I} = (\triangle^\mathcal{I}, \cdot^\mathcal{I})$ definiert. Die Domäne der Interpretation $\triangle^\mathcal{I}$ ist eine nicht leere Menge von Objekten eines betrachteten Weltausschnitts. Die Interpretationsfunktion $\cdot^\mathcal{I}$ bildet Konstanten auf Objekte der Domäne ($a^\mathcal{I} \in \triangle^\mathcal{I}$), atomare Konzepte auf Teilmengen der Domäne ($A^\mathcal{I} \subseteq \triangle^\mathcal{I}$) und atomare Rollen auf Teilmengen des Kreuzproduktes der Domäne ab ($R^\mathcal{I} \subseteq \triangle^\mathcal{I} \times \triangle^\mathcal{I}$). Wenn nicht anders angegeben, gilt die *Unique-Name-Assumption (UNA)* für alle in KB vorkommenden Konstanten. Das ist die Annahme, dass unterschiedliche Konstanten unterschiedliche Objekte bezeichnen, so dass $a \neq b \rightarrow a^\mathcal{I} \neq b^\mathcal{I}$. Die Definition der Interpretationsfunktion $\cdot^\mathcal{I}$ wird für komplexe Konzepte erweitert. Die Interpretation der Konzeptkonstruktoren der Sprache $\mathcal{AL}_0$ ist definiert durch:

$$\begin{array}{lcl}
\top^\mathcal{I} & = & \triangle^\mathcal{I} \\
\bot^\mathcal{I} & = & \emptyset \\
(\neg A)^\mathcal{I} & = & \triangle^\mathcal{I} \setminus A^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} & = & C^\mathcal{I} \cap D^\mathcal{I} \\
(\forall R.C)^\mathcal{I} & = & \{a \in \triangle^\mathcal{I} \mid (\forall b)\,[(a,b) \in R^\mathcal{I} \rightarrow b \in C^\mathcal{I}]\}
\end{array}$$

Die Interpretationsfunktion wird für Konzeptkonstruktoren weiterer in dieser Arbeit behandelter Sprachen wie folgt definiert:

$$\begin{array}{lcl}
(\exists R.\top)^\mathcal{I} & = & \{a \in \triangle^\mathcal{I} \mid (\exists b)\,[(a,b) \in R^\mathcal{I}]\} \\
(\exists R.C)^\mathcal{I} & = & \{a \in \triangle^\mathcal{I} \mid (\exists b)\,[(a,b) \in R^\mathcal{I} \wedge b \in C^\mathcal{I}]\} \\
(\neg C)^\mathcal{I} & = & \triangle^\mathcal{I} \setminus C^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} & = & C^\mathcal{I} \cup D^\mathcal{I} \\
(\geq n\, R)^\mathcal{I} & = & \{a \in \triangle^\mathcal{I} \mid card\{b \mid (a,b) \in R^\mathcal{I}\} \geq n\} \\
(\leq n\, R)^\mathcal{I} & = & \{a \in \triangle^\mathcal{I} \mid card\{b \mid (a,b) \in R^\mathcal{I}\} \leq n\}
\end{array}$$

$card\{b \mid (a,b) \in R^\mathcal{I}\}$ bildet eine Rolle R mit der Konstanten a als erstem Tupelelement auf die Anzahl der Rollenfüller b von a ab.

Die Interpretation wird auf Axiome φ einer Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$ erweitert. Für terminologische Axiome gilt: Eine Inklusion $A \sqsubseteq C$ ist erfüllt in $\mathcal{I}$, wenn $A^\mathcal{I} \subseteq C^\mathcal{I}$ und eine Definition $A \equiv C$, wenn $A^\mathcal{I} = C^\mathcal{I}$. Erfüllt eine Interpretation $\mathcal{I}$ diese Axiome, wird das mit $\mathcal{I} \models A \sqsubseteq C$ bzw. $\mathcal{I} \models A \equiv C$ notiert. Assertionale Axiome $C(a)$ und $R(a, b)$ sind erfüllt in $\mathcal{I}$, wenn $a^\mathcal{I} \in C^\mathcal{I}$ bzw. $(a, b)^\mathcal{I} \in R^\mathcal{I}$ (notiert mit $\mathcal{I} \models C(a)$ bzw. $\mathcal{I} \models R(a, b)$). Interpretationen, die Axiome erfüllen, sind *Modelle* dieser Axiome. Erfüllt eine Interpretation eine Menge $\mathcal{T}$ von terminologischen Axiomen, ist sie ein Modell von $\mathcal{T}$ (notiert mit $\mathcal{I} \models \mathcal{T}$), erfüllt sie die Menge aller Axiome einer ABox $\mathcal{A}$, ist sie ein Modell dieser ABox (notiert mit $\mathcal{I} \models \mathcal{A}$). Eine Interpretation $\mathcal{I}$ ist ein Modell für eine Wissensbasis KB, wenn sie ein Modell für $\mathcal{A}$ und $\mathcal{T}$ ist (notiert mit $\mathcal{I} \models KB$). Gibt es ein Modell für KB, ist KB *erfüllbar*.

2.3 Reasoning Services

Ein Axiom φ *folgt logisch* aus einer beschreibungslogischen Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$, symbolisch $KB \models \varphi$, wenn φ in allen Modellen der Wissensbasis gilt, und φ folgt nicht logisch aus KB, $KB \not\models \varphi$, wenn es nicht in allen Modellen der Wissensbasis gilt.

Definition 2.3.1 *Sei KB eine beschreibungslogische Wissensbasis und α eine beliebige Assertion. Ein Assertionstest ist ein Inferenzprozess, der prüft, ob $KB \models \alpha$ oder $KB \not\models \alpha$. Im ersten Fall ist der Assertionstest erfolgreich, im zweiten Fall nicht.*

Inferenzprozesse wie der Assertionstest werden im Folgenden, bedingt durch die Dienste, die sie anhand von darstellbaren Ergebnissen leisten, auch *Reasoning Services* genannt. Dadurch, dass die TBox $\mathcal{T}$ bei jedem Inferenzprozess eliminiert wird (siehe Abschnitt 2.1.2), wird bei Beginn der Durchführung eines Assertionstests (und bei Beginn aller weiteren hier behandelten Reasoning Services) die Wissensbasis KB durch die bzgl. $\mathcal{T}$ expandierte ABox $\mathcal{A}_\mathcal{T}$ ersetzt, und es wird geprüft, ob $\mathcal{A}_\mathcal{T} \models \alpha_\mathcal{T}$. Da die deduktive Komponente nicht mehr in der Wissensbasis $\mathcal{A}_\mathcal{T}$ enthalten ist, muss explizit berücksichtigt werden, dass die Assertion α ebenfalls bzgl. $\mathcal{T}$ expandiert wird. Dadurch ist sichergestellt, dass $KB \models \alpha$ *gdw.* $\mathcal{A}_\mathcal{T} \models \alpha_\mathcal{T}$ (vgl. [Baader und Nutt, 2003]).

Der Assertionstest ist der zentrale Reasoning Service dieser Arbeit. Dadurch, dass eine Assertion α entweder eine Konzeptassertion oder eine Rollenassertion ist, ist der Assertionstest eine Verallgemeinerung weiterer Reasoning Services. Der Service, der prüft, ob $KB \models C(a)$, wird *Instance-Check* und der, der prüft, ob $KB \models R(a, b)$ *Relationentest* genannt. Zusätzlich sind *negierte Relationentests* $KB \models \neg R(a, b)$ erlaubt. Der Instance-Check ist ein grundlegender Reasoning Service für eine ABox. Er wird ausführlich in [Donini et al., 1994] behandelt. Der Instance-Check wird in dieser Untersuchung häufig an Stelle des Assertionstests verwendet, weil üblicherweise selten ein Bedarf für die direkte Prüfung eines (negierten) Relationentests besteht und der Instance-Check – bedingt durch das Auftreten von Rollenrestriktionen in Konzeptbeschreibungen – implizit Relationentests durchführt.

Beispiel 2.3.2 Sei $KB = (\mathcal{T}, \mathcal{A})$ die folgende beschreibungslogische Wissensbasis:

$$\mathcal{T} = \{Tochtervater \equiv Vater \sqcap \forall hatKind.Weiblich\}$$
$$\mathcal{A} = \{Tochtervater(john), hatKind(john, susy)\}$$

Soll der Instance-Check $KB \models Weiblich(susy)$ auf Erfolg überprüft werden, wird zunächst KB durch die Expansion von $\mathcal{A}$ bzgl. $\mathcal{T}$ ersetzt (die Expansion des atomaren Konzepts *Weiblich* bzgl. $\mathcal{T}$ hat keine Auswirkung). Dann ist $\mathcal{A}_{\mathcal{T}} = \{(Vater \sqcap \forall hatKind.Weiblich)(john), hatKind(john, susy)\}$, und es wird geprüft ob $\mathcal{A}_{\mathcal{T}} \models Weiblich(susy)$. Wie sich leicht erkennen lässt, gilt diese logische Folgerung.

Ein Service, der aus dem Instance-Check hervorgeht, ist der *Retrieval Service*. Dieser bestimmt alle Konstanten, die Instanz eines gegebenen Konzeptes C sind, also die Menge aller derjenigen Konstanten a, für die gilt: $KB \models C(a)$. Er gibt nicht an, ob eine bestimmte Bedingung erfüllt ist oder nicht, sondern für welche Konstanten sie erfüllt ist und ermöglicht so die Formulierung von Anfragen mit einer freien Variable an Beschreibungslogiken (Assertionstests entsprechen einer „ja/nein“-Anfrage ohne freie Variablen). Durch die Möglichkeit der Darstellung von Konzeptkonjunktionen sowie Restriktionen über Rollen ist mit einem Retrieval Service die Beantwortung vieler *konjunktiver Anfragen* [Abiteboul et al., 1995] möglich. Dieser Service kann durch die zusätzliche Berücksichtigung entsprechender (negierter) Relationentests auf beliebige Assertionen erweitert werden und wird dann *erweiterter Retrieval Service* genannt.

Die Voraussetzung für eine sinnvolle Anwendung aller der hier betrachteten Reasoning Services ist der Erfolg der Überprüfung einer Wissensbasis auf Erfüllbarkeit. Eine Wissensbasis ist erfüllbar, wenn es keine Konstante a mit $KB \models \bot(a)$ gibt. Ist diese Bedingung nicht erfüllt, enthält die Wissensbasis widersprüchliche Aussagen, und es kann jedes Axiom aus der Wissensbasis logisch gefolgert werden, so dass die Ausführung eines weiteren Reasoning Services irrelevant ist.

Erweiterung der Anfragesprache Häufig entspricht bei der Ausführung eines Assertionstests $KB \models \alpha$ die Anfragesprache der Assertion α der Sprache der Wissensbasis KB. Es ist allerdings auch möglich, unterschiedliche Sprachen für die Formulierung von Anfragen und für die Spezifikation der Wissensbasis zu verwenden. Die Anfragesprache sollte dabei mindestens so ausdrucksstark wie die Sprache der Wissensbasis KB sein, damit für alle explizit spezifizierten Zusicherungen nachgewiesen werden kann, dass sie in allen Modellen von KB gelten [Schaerf, 1994a]. Die Erweiterung einer Anfragesprache bezüglich der Sprache einer Wissensbasis führt in vielen Fällen zu einer größeren Ausdrucksstärke, ohne die Berechnungskomplexität eines Assertionstests erheblich zu vergrößern. Aus der Kombination der Sprache der Wissensbasis und der erweiterten Anfragesprache ergeben sich zusätzlich zu den in Abschnitt 2.1.1 aufgeführten Sprachen weitere Sprachen. Diese Sprachen werden notiert, indem die Anfragesprache hinter der Sprache der Wissensbasis, getrennt durch „/“, angegeben wird. Die Sprache, die eine $\mathcal{AL}_0$-Wissensbasis einbezieht, aber $\mathcal{ALCN}$-Anfragekonzepte erlaubt, wird beispielsweise mit $\mathcal{AL}_0/\mathcal{ALCN}$ bezeichnet. Sollte dagegen die Anfragesprache mit der Sprache der Wissensbasis übereinstimmen, wird das nicht gesondert dargestellt.

2.4 Beweisverfahren: Tableau-Kalkül

Logische Folgerungen können nachgewiesen und Assertionstests dadurch auf Erfolg überprüft werden. Tableau-Kalküle für Beschreibungslogiken ermöglichen solche Beweise aufgrund von Widersprüchen. Initiale Formelmengen – bei den vorgestellten Reasoning Services sind das die bzgl. einer TBox $\mathcal{T}$ expandierte ABox $\mathcal{A}_\mathcal{T}$ evtl. zuzüglich einer negierten bzgl. $\mathcal{T}$ expandierten Anfrageassertion $\neg\alpha_\mathcal{T}$ – werden durch korrekte Inferenzregeln expandiert. Durch diese Expansion entstehen vergrößerte Formelmengen, die auf einen Widerspruch überprüft werden können. Enthält die expandierte Formelmenge einen Widerspruch, ist die initiale Formelmenge unerfüllbar und der entsprechende Assertionstest ist erfolgreich [Baader und Nutt, 2003].

Die Reduktion aller der in dieser Untersuchung verwendeten Reasoning Services auf Unerfüllbarkeit ist in Abschnitt 2.4.1 dargestellt. Abschnitt 2.4.2 stellt den Tableau-Kalkül für die Sprache $\mathcal{ALC}$ vor, und in Abschnitt 2.4.3 wird dieser Kalkül für die Sprache $\mathcal{ALCN}$ um Anzahlrestriktionen erweitert. Dabei wird jeweils auf entsprechende Literatur verwiesen, in der wichtige Eigenschaften wie Korrektheit, Vollständigkeit und (bei adäquater Reihenfolge der Expansionsregeln) Termination dieser Verfahren nachgewiesen wurden.

2.4.1 Reduktionen auf Unerfüllbarkeit

Die grundlegende Reduktion eines Reasoning Services auf Unerfüllbarkeit erfolgt durch die Negation der Erfüllbarkeit einer beschreibungslogischen Wissensbasis KB. $KB = (\mathcal{T}, \mathcal{A})$ ist unerfüllbar, symbolisch $KB \models \bot(a)$ für eine beliebige Konstante a, wenn die bzgl. $\mathcal{T}$ expandierte ABox $\mathcal{A}_\mathcal{T}$ unerfüllbar ist. Der Instance-Check, der Relationentest und der negierte Relationentest können wie folgt auf Unerfüllbarkeitsprüfung reduziert werden:

$$\begin{array}{lll} KB \models C(a) & gdw. & \mathcal{A}_\mathcal{T} \cup \{\neg C_\mathcal{T}(a)\} \text{ unerfüllbar} \\ KB \models R(a,b) & gdw. & \mathcal{A}_\mathcal{T} \cup \{\neg R(a,b)\} \text{ unerfüllbar} \\ KB \models \neg R(a,b) & gdw. & \mathcal{A}_\mathcal{T} \cup \{R(a,b)\} \text{ unerfüllbar} \end{array}$$

Die Negation von atomaren Rollenassertionen $\neg R(a,b)$ kann in den behandelten Sprachen für einen Tableau-Beweis verwendet werden. Die Reduktionen des Assertionstests, des Retrieval Services und des erweiterten Retrieval Services auf Unerfüllbarkeitsprüfung ergeben sich aus diesen Reduktionen.

Für die Prüfung vieler Inferenzprobleme muss aufgrund des Konzeptes $\neg C_\mathcal{T}(a)$ die Negation von allen Konzepten der jeweiligen Sprache ausgedrückt werden können. Es werden deswegen Tableau-Kalküle verwendet, die aussagenlogisch geschlossen sind, d.h. die sowohl Expansionsregeln für Konjunktionen als auch für die Negation aller vorkommenden Konzepte ermöglichen [Baader und Nutt, 2003, Seite 67ff]. Dadurch, dass die Konjunktion in allen Sprachen enthalten ist und die Negation von Konzepten mit vollständigen Existenzrestriktionen und Konzeptdisjunktionen ausgedrückt werden kann (und umgekehrt), sind das in nahezu allen Fällen Tableau-Kalküle für $\mathcal{ALC}$ und ausdrucksstärkere Sprachen. Ein Tableau-Kalkül kann auch für eine eingeschränkte Sprache verwendet werden (beispielsweise der $\mathcal{ALC}$-Tableau-Kalkül für die Sprache $\mathcal{AL}$). Allerdings kann sich in diesem Fall für alle Instance-Checks mit Anfragekonzepten, deren Negation nicht in der Sprache enthalten sind, eine größere Berechnungskomplexität ergeben.

2.4.2 Tableau-Kalkül für $\mathcal{ALC}$

Der hier vorgestellte Tableau-Kalkül verwendet beschreibungslogische Formelmengen (im Folgenden auch Zweige genannt) in Form von ABoxen, die neben Konzepten, Rollen und Konstanten auch Skolem-Konstanten enthalten. Letztere werden mit *new* notiert, um aufzuzeigen, dass sie neue Konstanten sind (eine detaillierte Ausführung zu Skolem-Konstanten ist in den Abschnitten 4.5.2 und 7.2 zu finden). Ein Zweig $\mathcal{A}$ des $\mathcal{ALC}$-Tableau-Kalküls besteht aus ABox-Axiomen der Form:

$$C(x), \qquad R(x,y), \qquad a \neq b$$

Die Meta-Variablen x, y stehen entweder für Konstanten aus KB oder Skolem-Konstanten und a und b nur für Konstanten aus KB. Der initiale Zweig enthält bei Beginn eines Beweises die Assertionen, die sich durch Reduktion auf Unerfüllbarkeit ergeben, sowie bei Annahme der UNA für alle auftretenden Konstanten die Ungleichheitsaxiome $a \neq b$. Bei Ausführung eines Relationentests kommt durch die entsprechende Reduktion auf einen Unerfüllbarkeitsbeweis auch die atomare Rollennegation der Anfragerelation $\neg R(a,b)$ in dem initialen Zweig vor. Nach der Eliminierung der TBox werden zu Beginn eines jeden Beweises alle Konzepte des initialen Zweiges rekursiv in eine *Negationsnormalform (NNF)* transformiert:

$$\begin{array}{lcl}
\neg\top & \Rightarrow & \bot \\
\neg\bot & \Rightarrow & \top \\
\neg\neg C & \Rightarrow & C \\
\neg(C \sqcap D) & \Rightarrow & \neg C \sqcup \neg D \\
\neg(C \sqcup D) & \Rightarrow & \neg C \sqcap \neg D \\
\neg\forall R.C & \Rightarrow & \exists R.\neg C \\
\neg\exists R.C & \Rightarrow & \forall R.\neg C
\end{array}$$

Ist keine dieser Umformungsregeln mehr anwendbar, ist das Konzept in NNF. Diese Umformung kann in linearer Zeit vollzogen werden. Ist ein Konzept in NNF, kommen Negationen ausschließlich vor atomaren Konzepten vor.

Im Verlauf eines Beweises wird die Formelmenge eines Zweiges $\mathcal{A}$ mit *Expansionsregeln* erweitert. Auf diese Art und Weise werden zusätzliche Erfüllbarkeitsbedingungen berücksichtigt. In Tabelle 2.1 sind die Expansionsregeln für die Sprache $\mathcal{ALC}$ zusammengefasst. Sie ergeben sich aus der Interpretation der entsprechenden Konzeptkonstruktoren (siehe Abschnitt 2.2).

$\rightarrow$ die $\sqcap$-Regel	
Bedingung:	$\mathcal{A}$ enthält $(C \sqcap D)(x)$, aber nicht $C(x)$ und $D(x)$.
resultierender Zweig:	$\mathcal{A}' = \mathcal{A} \cup \{C(x), D(x)\}$.
$\rightarrow$ die $\sqcup$-Regel	
Bedingung:	$\mathcal{A}$ enthält $(C \sqcup D)(x)$, aber weder $C(x)$ noch $D(x)$.
resultierende Zweige:	$\mathcal{A}' = \mathcal{A} \cup \{C(x)\}$, $\mathcal{A}'' = \mathcal{A} \cup \{D(x)\}$.
$\rightarrow$ die $\forall$-Regel	
Bedingung:	$\mathcal{A}$ enthält $\forall R.C(x)$ und $R(x, y)$, aber nicht $C(y)$.
resultierender Zweig:	$\mathcal{A}' = \mathcal{A} \cup \{C(y)\}$.
$\rightarrow$ die $\exists$-Regel	
Bedingung:	$\mathcal{A}$ enthält $\exists R.C(x)$, aber es gibt keine (Skolem-)Konstante y, so dass $R(x, y)$ und $C(y)$ in $\mathcal{A}$ enthalten sind.
resultierender Zweig:	$\mathcal{A}' = \mathcal{A} \cup \{R(x, new), C(new)\}$ und *new* ist eine Skolem-Konstante, die nicht in $\mathcal{A}$ vorkommt.

Tabelle 2.1: Expansionsregeln für den Tableau-Kalkül in $\mathcal{ALC}$

Aufgrund der NNF müssen entsprechende duale Negationsregeln für $\sqcap$, $\sqcup$, $\forall$ und $\exists$ nicht berücksichtigt werden. Es werden deterministische Expansionsregeln von nicht-deterministischen sowie erzeugende von nicht-erzeugenden Regeln unterschieden. Nicht-deterministische Regeln ersetzen den aktuellen Zweig durch eine Menge von Zweigen und erzeugende Regeln führen Skolem-Konstanten ein. Eine nicht-determinstische Regel ist die $\sqcup$-Regel, und alle übrigen Regeln sind deterministisch. Eine erzeugende Regel ist die $\exists$-Regel, und alle anderen Regeln sind nicht-erzeugend.

Eine beschreibungslogische Formel ist unter Berücksichtigung dieses Kalküls unerfüllbar, wenn alle Zweige abgeschlossen sind. Ein Zweig dieses Kalküls ist abgeschlossen, wenn er einen der folgenden Widersprüche enthält:

$$\begin{array}{ll} \text{(i)} & \bot(x) \in \mathcal{A}; \\ \text{(ii)} & \{A(x), \neg A(x)\} \subseteq \mathcal{A} \end{array}$$

Der $\mathcal{ALC}$-Tableau-Kalkül ist korrekt, vollständig, terminiert und ist PSPACE-vollständig. Beweise für die Korrektheit und Vollständigkeit sind z.B. in [Baader und Nutt, 2003] zu finden und für Termination und PSPACE-Vollständigkeit in [Baader und Sattler, 2000].

2.4.3 Tableau-Kalkül für $\mathcal{ALCN}$

Der Kalkül für die Sprache $\mathcal{ALCN}$ ist eine Erweiterung des Kalküls für $\mathcal{ALC}$ um Regeln für Anzahlrestriktionen $(\geq n\,R)$ und $(\leq n\,R)$. Die Regeln zur Erstellung der Negationsnormalform werden für Konzeptbeschreibungen mit Anzahlrestriktionen wie folgt erweitert:

$$\neg(\geq n\,R) \quad \Rightarrow \quad \begin{cases} (\leq (n-1)\,R) & \text{für} \quad n > 0 \\ \bot & \text{für} \quad n = 0 \end{cases}$$

$$\neg(\leq n\,R) \quad \Rightarrow \quad (\geq (n+1)\,R)$$

Für die Ausführung von Beweisen ergeben sich weitere Expansionsregeln, die in Tabelle 2.2 dargestellt sind. Die $\geq$-Regel ist erzeugend und die $\leq$-Regel nicht-deterministisch.

$\rightarrow$ die $\geq$-Regel	
Bedingung:	$\mathcal{A}$ enthält $(\geq n\,R)(x)$, und es gibt keine (Skolem-)Konstanten $z_1, ..., z_n$ mit $R(x, z_i) \in \mathcal{A}$ für $1 \leq i \leq n$ und $z_i \neq z_j \in \mathcal{A}$ für $1 \leq i \neq j \leq n$.
result. Zweig:	$\mathcal{A}' = \mathcal{A} \cup \{R(x, new_i) \mid 1 \leq i \leq n\} \cup \{new_i \neq new_j \mid 1 \leq i \neq j \leq n\}$ und $new_1, ..., new_n$ sind unterschiedliche Skolem-Konstanten, die nicht in $\mathcal{A}$ vorkommen.
$\rightarrow$ die $\leq$-Regel	
Bedingung:	$\mathcal{A}$ enthält $(\leq n\,R)(x)$ und $R(x, y_1), ..., R(x, y_{n+1})$ für unterschiedliche y_i, und mindestens ein Axiom $y_i \neq y_j$ ist nicht in $\mathcal{A}$ enthalten für $i \neq j$.
result. Zweige:	Es ergeben sich die ABoxen $\mathcal{A}_{i,j}$ durch Ersetzung von y_i durch y_j in $\mathcal{A}$ für jedes mögliche Paar von Variablen y_i, y_j mit $i > j$ und $y_i \neq y_j \notin \mathcal{A}$.

Tabelle 2.2: Expansionsregeln für Anzahlrestriktionen $\mathcal{N}$

Die ABox-Axiome, die in einem Zweig in $\mathcal{ALCN}$ auftreten können, erweitern die Ungleichheitsaxiome eines $\mathcal{ALC}$-Zweiges um Ungleichheitsaxiome für Skolem-Konstanten (diese entstehen bei der Ausführung der $\geq$-Regel), so dass zusätzlich Axiome der Form $x \neq y$ für Variablen x, y in einem Zweig enthalten sein können. Ein weiterer Abschluss, der zusätzlich zu den Abschlüssen (i) und (ii) des vorherigen Unterabschnittes berücksichtigt werden muss, ist gegeben durch:

(iii) $\{\leq n\,R)(x)\} \cup \{R(x, y_i) \mid 1 \leq i \leq n+1\} \cup \{y_i \neq y_j \mid 1 \leq i \neq j \leq n+1\} \subseteq \mathcal{A}$;

Um bei Verwendung des $\mathcal{ALCN}$-Kalküls Berechnungen in PSPACE zu ermöglichen, ist eine Expansionsstrategie, die in [Baader und Nutt, 2003] dargestellt ist, hilfreich. Dort sind auch Korrektheit, Vollständigkeit und Termination des Kalküls in dieser Sprache aufgezeigt.

Kapitel 3

Der Assertionstest unter der Open-World-Assumption

Der Assertionstest und die weiteren Services, aus denen er besteht oder die diesen erweitern, wurden in Kapitel 2 vorgestellt. Alle diese Reasoning Services berücksichtigen die logische Folgerung, so dass alle möglichen Fälle einbezogen werden. Das entspricht der Annahme einer offenen Welt. Die Vor- und Nachteile der Open-World-Assumption (OWA) und ihr Bezug zu logischer Folgerung werden in Abschnitt 3.1 diskutiert. In Abschnitt 3.2 wird das unerwünschte Fehlschlagen vieler Assertionstests unter der OWA dargestellt. In den einzelnen Unterabschnitten wird für alle Konzeptkonstruktoren, die diese Problematik betrifft, nahegebracht, aus welchen Gründen viele Inferenzen nicht gezogen werden können. Das Kapitel schließt in Abschnitt 3.3 mit der Feststellung ab, dass sich zusätzlich zu der bereits intraktablen Berechnungskomplexität in vielen beschreibungslogischen Sprachen eine weitere Komplexitätsquelle durch Fallunterscheidungen beim Schließen mit Existenzrestriktionen unter der OWA ergibt.

3.1 Die Open-World-Assumption (OWA)

In einem zu modellierenden Weltausschnitt (auch Domäne genannt) kann nicht sichergestellt werden, dass die repräsentierte Information vollständig ist. Es gibt viele Sachverhalte, die nicht bekannt sind und deswegen nicht spezifiziert werden können. Das beste Beispiel dafür ist das World-Wide-Web: Die immense Anzahl von Web-Ressourcen führt dazu, dass jeder repräsentierte Weltausschnitt nur eine sehr kleine Teilmenge aller Informationen dieser Ressourcen beinhalten kann. Aus diesem Grund entscheidet man sich gerne für die Annahme einer offenen Welt. Alle Informationen, die (noch) nicht bekannt sind, können zu einem späteren Zeitpunkt – mit der Eigenschaft, dass gültige Formeln weiterhin gültig bleiben – hinzugefügt werden. Das ist die Eigenschaft der Monotonie. Die Open-World-Assumption (OWA) basiert auf *logischer Folgerung*. Das bedeutet, dass nur solche Formeln bewiesen werden können, die unter Betrachtung aller Möglichkeiten, d.h. in allen Modellen der Wissensbasis, gelten. Schlussfolgerungen, die auf Annahmen beruhen, für die es eigentlich keine Evidenz gibt, führen zu einer inkorrekten Nutzung der Informationen, die beispielsweise im World-Wide-Web enthalten sind. Bezogen auf das einführende Beispiel der Verweise auf Internetseiten in Kapitel 1 kann bei der Modellierung dieser Domäne im Allgemeinen nicht bekannt sein, dass es keine Verweise gibt. Wenn es Verweise gibt, wie die auf Seite 1, 2 und 5 (siehe Abbildung 1.1), gibt es allerdings einen Beweis dafür, weil durch das Hinzufügen weiterer Informationen diese Daten in keinem Fall ungültig werden können.

Folgt eine Formel nicht logisch aus einer Wissensbasis, ist nicht gewährleistet, dass die negierte Formel logisch aus dieser Wissensbasis folgt. Sowohl die Formel als auch ihre Negation können in einigen Modellen gelten und in anderen nicht. An Stelle von „nein“ könnte das System deswegen in beiden Fällen ebenso mit „weiß ich nicht“ antworten. Die Wissenslücke, die sich dadurch ergibt, ist häufig größer als die Menge der Formeln, die nachgewiesen werden können (der folgende Abschnitt enthält eine Darstellung dieser Problematik).

Um logische Folgerungen beweisen zu können, werden für alle in dieser Arbeit untersuchten beschreibungslogischen Sprachen die in Kapitel 2 vorgestellten Tableau-Kalküle verwendet, weil diese korrekt und vollständig sind.

In Anlehnung an die OWA wird gerne auf die Unique-Name-Assumption (UNA) verzichtet. Die Annahme, dass unterschiedliche Konstanten unterschiedliche Objekte bezeichnen, ist in einer offenen Welt nicht sinnvoll. Es werden keine Annahmen getroffen, die sich später als falsch herausstellen könnten. Wird die UNA dennoch verwendet, kann zum Beispiel ein und dasselbe Objekt nicht in unterschiedlichen natürlichen Sprachen repräsentiert werden. Außerdem können sich in einer offenen Welt wie dem World-Wide-Web unter der UNA Widersprüche ergeben. Enthält eine Wissensbasis beispielsweise die Assertionen

$$\mathit{hasFather}(\mathit{Francesca}, \mathit{Salvatore}), \quad (\leq 1\, \mathit{hasFather})(\mathit{Francesca})$$

und wird um $\mathit{hasFather}(\mathit{Francesca}, \mathit{Pietro})$ erweitert, ist sie inkonsistent, wenn zusätzlich $\mathit{Salvatore} \neq \mathit{Pietro}$ gilt.

3.2 Unerwünschtes Fehlschlagen von Assertionstests

In beschreibungslogischen Sprachen können bei Verwendung vieler Konstruktoren häufig nicht die Schlüsse gezogen werden, die erwartet werden. In diesem Abschnitt soll an einigen Beispielen aufgezeigt werden, aus welchen Gründen Assertionstests $KB \models \alpha$ mit Assertionen α, die diese Konstruktoren beinhalten, oft unerwünscht fehlschlagen. Dabei handelt es sich nicht um unbedeutende Konstruktoren, und eine große Anzahl an Inferenzen, die in Beschreibungslogiken vermutlich nach Ansicht vieler Anwender als „intelligent“ bezeichnet werden, können nicht gezogen werden.[1]

[1] Beispielsweise haben die Benutzer von *RacerPro* eine Form der Inferenz gefordert, die Annahmen trifft, die sie auch treffen würden. Das Ergebnis ist ein Negation-as-failure-Mechanismus, der in Kapitel 6 dargestellt ist.

Die Konzeptkonstruktoren, mit denen bei der Ausführung eines Assertionstests unter der OWA nicht die erwünschten Inferenzen gezogen werden können, sind Werterestriktionen $\forall R.C$ und negierte Existenzrestriktionen $\neg\exists R.C$ (Abschnitt 3.2.1), Anzahlrestriktionen $(\leq n\, R)$ und $(\geq n\, R)$ (Abschnitte 3.2.2 und 3.2.4) sowie atomare Negationen $\neg A$ (Abschnitt 3.2.3). Zu den letztgenannten gehört auch der Rollenkonstruktor für negierte atomare Rollen $\neg R$. Bei Verwendung dieser Konstruktoren in Anfrageassertionen können zum Beispiel in der Weinontologie des W3C nur deswegen erwartete Inferenzen unter der OWA gezogen werden, weil Konzeptassertionen, die für das Treffen dieser Inferenzen notwendig sind, dort explizit spezifiziert wurden.[2] Da die oben genannten Konstruktoren, mit Ausnahme der Negation atomarer Rollen, Konzeptkonstruktoren sind, handelt es sich bei den meisten der erforderten Assertionstests um Instance-Checks. Es gelten die folgenden semantischen Äquivalenzen:

$$\begin{array}{rcl} \forall R.C & \equiv & \neg\exists R.\neg C, \\ (\leq n\, R) & \equiv & \neg(\geq n+1\, R), \\ \multicolumn{3}{c}{\forall R.\bot \equiv \neg\exists R.\top \equiv (\leq 0\, R) \equiv \neg(\geq 1\, R).} \end{array}$$

Wie in den Unterabschnitten 3.2.1 und 3.2.2 zu finden ist, führen alle Konzeptkonstruktoren, die Bestandteil einer solchen Äquivalenz sind und in der Anfrageassertion eines Instance-Checks vorkommen, in einem Tableau-Zweig zu einer erzeugenden Regel, deren Ausführung unter der OWA nur durch explizite Spezifikation spezieller Konzeptassertionen in der Wissensbasis zu einem abgeschlossenen Zweig führen kann. In den darauf folgenden Unterabschnitten wird gezeigt, inwiefern atomare Negationen sowie mindestens-Anzahlrestriktionen in Anfrageassertionen andere Probleme darstellen.

3.2.1 Werterestriktionen und negierte Existenzrestriktionen

Werterestriktionen (und negierte Existenzrestriktionen) werden zuerst untersucht, weil dieser Operator charakteristisch für die meisten beschreibungslogischen Sprachen ist. Er ist Bestandteil sowohl von $\mathcal{FL}^-$ [Brachman und Levesque, 2004] als auch von $\mathcal{AL}$ [Schmidt-Schauß und Smolka, 1991], zweier etablierter grundlegender beschreibungslogischer Sprachen. Als Beispiel können drei assertionale Wissensbasen betrachtet werden:

$$\begin{array}{rcl} KB_1 & = & \{hatKind(vater, kind_1), hatKind(vater, kind_2), Weiblich(kind_1), \\ & & \neg Weiblich(kind_2)\} \\ KB_2 & = & \{hatKind(vater, kind), Weiblich(kind)\} \\ KB_3 & = & \{\} \end{array}$$

Die Anfrageassertion $\forall hatKind.Weiblich(vater)$ kann für keine der drei Wissensbasen bewiesen werden. Für KB_1 ist das durchaus erwünscht. In allen Modellen dieser Wissensbasis gilt

[2] www.w3.org/TR/owl-guide/wine.rdf. Es erfolgen explizite Zusicherungen der Form $(\leq 1\, R)(a)$, mit denen bei Existenz eines Rollenfüllers, der Instanz eines Konzeptes C ist, $\forall R.C(a)$ logisch gefolgert werden kann.

die negierte Konzeptbeschreibung $\neg\forall hatKind.Weiblich(vater) \equiv \exists hatKind.\neg Weiblich(vater)$. Für KB_2 könnte allerdings ein Beweis erwartet werden, weil alle bekannten Kinder des Vaters weiblich sind. Da es Modelle gibt, in denen der Vater weitere Kinder hat, die nicht weiblich sind, ergibt sich unter der OWA aber kein Beweis. Aus dem gleichen Grund lässt sich für KB_3, in der keine Kinder repräsentiert sind, nicht folgern, dass der Vater nur weibliche Kinder hat. Das Fehlschlagen solcher grundlegender Instance-Checks hat auch Auswirkungen auf Instance-Checks mit komplexeren Konzeptassertionen, die Werterestriktionen enthalten (beispielsweise auf einen Instance-Check mit der Konzeptassertion $\exists R_1.\forall R_2.C(a)$).

Im Folgenden wird anhand des Tableau-Kalküls gezeigt, aus welchen Gründen bei Durchführung eines Assertionstests mit Werterestriktionen nicht die erwarteten Inferenzen getroffen werden können. Beispielsweise könnte die Frage lauten: $KB_2 \models \forall hatKind.Weiblich(vater)$? Wird dieser Instance-Check auf Unerfüllbarkeit reduziert, ergibt sich die Formelmenge $\{hatKind(vater, kind), Weiblich(kind)\} \cup \{\neg\forall hatKind.Weiblich(vater)\}$. Nach Anwendung der Regeln zur Bildung der Negationsnormalform (NNF) entsteht ein initialer Tableau-Zweig, der mit der einmaligen Anwendung der $\exists$-Regel vollständig expandiert werden kann (Abbildung 3.1). Die Skolem-Konstante *new* kann nicht mit *kind* identifiziert werden. Das hat zur Folge, dass sich nicht der Abschluss $\{Weiblich(kind), \neg Weiblich(kind)\}$ ergeben kann. Bezogen auf dieses Beispiel wäre ein Abschluss in $\mathcal{ALCN}$ nur durch explizite Spezifikation der Anfrageassertion $\forall hatKind.Weiblich(vater)$ oder von $(\leq 1\, hatKind)(vater)$ in der Wissensbasis möglich.

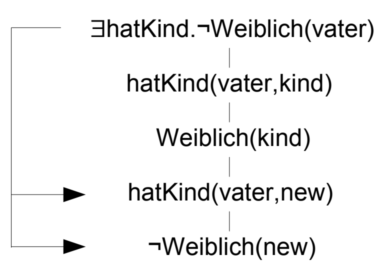


Abbildung 3.1: Scheitern eines Instance-Checks mit Werterestriktionen

Das Beispiel der Verweise auf Internetseiten, dessen Problematik in der Einführung aufgezeigt wurde, ist ein weiteres für das unerwünschte Scheitern von Assertionstests mit Werterestriktionen. In diesem Fall wurde das Problem allerdings mit negierten Existenzrestriktionen dargestellt. Abbildung 1.1 repräsentiert eine ABox mit folgenden Zusicherungen:

$$\begin{array}{cc} Page(page1). \quad Page(page2). \quad Page(page3). \\ Page(page4). \quad Page(page5). \quad Page(page6). \\ isLinkedFrom(page2, page1). \quad isLinkedFrom(page2, page3). \\ isLinkedFrom(page5, page2). \quad isLinkedFrom(page1, page6). \end{array}$$

Seiten, auf die verwiesen wird, sind $page1$, $page2$ und $page5$. Besteht Interesse daran, auf welche Seiten nicht verwiesen wird, kann ein Retrieval Service $\{a \mid \mathcal{A} \models \neg\exists isLinkedFrom.Page(a)\}$ ausgeführt werden, der für jede in der ABox $\mathcal{A}$ vorkommende Konstante a einen entsprechenden Instance-Check ausführt und die Konstanten in einer Menge ausgibt, für die dieser Instance-Check erfolgreich ist. Unter der OWA gibt es für diesen Service allerdings keine Ergebnisse, weil in einer offenen Welt nicht alle Verweise auf eine Seite bekannt sein können. Wird der Versuch unternommen, diesen Service mit dem Tableau-Kalkül für die Seiten zu beweisen, von denen nicht bekannt ist, dass auf sie verwiesen wird, kann der jeweilige Zweig nicht abgeschlossen werden. Das Fehlschlagen eines solchen Beweises wird in Abbildung 3.2 exemplarisch mit einem Instance-Check für $page3$ gezeigt. Das Anfragekonzept wird zunächst negiert, um auf Unerfüllbarkeit überprüft zu werden. Die NNF ist anschließend durch $\neg\neg\exists isLinkedFrom.Page \Rightarrow \exists isLinkedFrom.Page$ entstanden.

Abbildung 3.2: Scheitern eines Instance-Checks mit negierten Existenzrestriktionen

3.2.2 Höchstens-Anzahlrestriktionen

Anfrageassertionen mit höchstens-Anzahlrestriktionen können unter der OWA in vielen Fällen zu keinem Beweis führen, obwohl der aktuelle Wissensbestand einen Beweis erwarten lassen würde. Beispielsweise könnte die Wissensbasis

$$KB = \{hatKind(john, susy), hatKind(john, charles)\}$$

lauten, und es könnte geprüft werden, ob John höchstens zwei Kinder hat. Das entspricht dem Instance-Check $KB \models (\leq 2\, hatKind)(john)$. Durch die Reduktion auf Unerfüllbarkeit und anschließende Umformung in NNF ergibt sich $(\geq 3\, hatKind)(john)$. Die Anwendung der $\geq$-Regel führt dann in einem Tableau-Zweig zu keinem Abschluss (Abbildung 3.3). Die Formel kann nicht bewiesen werden, weil John in einem zukünftigen Zustand der Wissensbasis

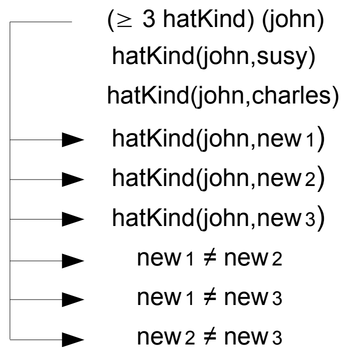


Abbildung 3.3: Scheitern eines Instance-Checks mit höchstens-Anzahlrestriktionen

durchaus weitere Kinder haben könnte. Bedingt durch die unqualifizierte Form der Anzahlrestriktionen, die in den Sprachen dieser Arbeit verwendet wird, stellt dieses Beispiel – im Gegensatz zu Werterestriktionen bzw. negierten Existenzrestriktionen – das Ausmaß der Problematik dar. Höchstens-Anzahlrestriktionen in komplexeren Anfrageassertionen, beispielsweise $\forall R_1.\exists R_2.(\leq 4\, R_3)(a)$, beziehen sich ausschließlich auf andere Konstanten, die Rollenfüller der aufgeführten Werte- und Existenzrestriktionen sind.

3.2.3 Atomare Negationen

Wird angenommen, dass eine Wissensbasis *KB* vollständiges Wissen bzgl. atomarer Konzepte repräsentiert, dann würde für jedes atomare Konzept, das nicht repräsentiert wurde, die Negation dieses Konzepts aus *KB* gefolgert werden können. Unter der OWA können diese Schlussfolgerungen aber nicht getroffen werden. Beispielsweise könnte ein Arzt wissen wollen, welches Medikament ein Schmerzmittel, aber kein Gerinnungshemmer ist [Drummond und Shearer, 2006]. Er könnte diese Anfrage an folgende Tabelle T stellen:

Medikament	Effekt
Aspirin	Schmerzmittel Gerinnungshemmer
Warfarin	Gerinnungshemmer
Paracetamol	Schmerzmittel

Die Tabelle ähnelt der einer Datenbank. Eine solche würde die Anfrage des Arztes mit „Paracetamol" beantworten. Ein Retrieval-Service unter der OWA, $\{a \mid T \models \mathit{Schmerzmittel} \sqcap \neg \mathit{Gerinnungshemmer}(a)\}$, führt dagegen zu keinem Ergebnis, weil es Modelle gibt, in denen Paracetamol ein Gerinnungshemmer ist. Ein initialer Tableau-Zweig für den Instance-Check mit der Konstanten *paracetamol* würde außer den Daten aus der Tabelle die negierte

Form des Anfragekonzeptes $\neg Schmerzmittel \sqcup Gerinnungshemmer(paracetamol)$ beinhalten. Nach der Expansion dieser Formel durch die $\sqcup$-Regel kann der Zweig, der die Formel $Gerinnungshemmer(paracetamol)$ enthält, nicht abgeschlossen werden (Abbildung 3.4).

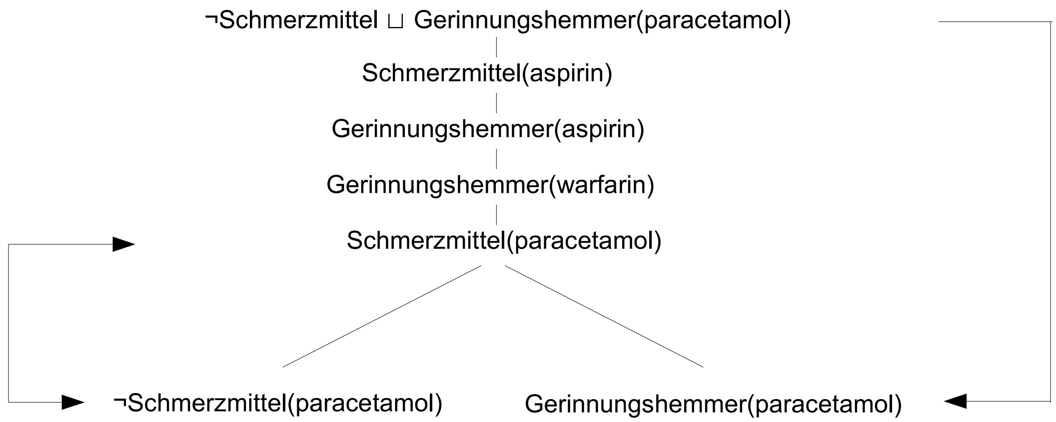


Abbildung 3.4: Scheitern eines Instance-Checks mit atomarer Negation

Es gibt viele Möglichkeiten, negierte atomare Konzepte in Anfragen zu formulieren. Dazu gehören auch Anfrageassertionen mit atomaren Negationen im Skopus einer Werte- oder Existenzrestriktion. Obwohl Assertionstests mit diesem Konstruktor unter der OWA oft unerwünscht fehlschlagen, gibt es Situationen, in denen Wissensbasen nicht ausreichend spezifiziert werden können und ein solches Fehlschlagen dringend erforderlich ist. Sollte in der oben dargestellten Tabelle zum Beispiel nicht angegeben sein, dass Aspirin ein Gerinnungshemmer ist, würde eine Datenbank fälschlicherweise annehmen, dass das nicht der Fall ist.

Entsprechend zu Assertionstests mit negierten atomaren Konzepten ergibt sich häufig ein Fehlschlagen von Assertionstests mit negierten atomaren Rollen. Bezogen auf die in Abschnitt 3.2.1 dargestellte ABox der Verweise auf Internetseiten kann zum Beispiel nicht bewiesen werden, dass $\mathcal{A} \models \neg isLinkedFrom(page3, page1)$ gilt.

3.2.4 Mindestens-Anzahlrestriktionen

Die Problematik, die sich in offenen Welten durch Anfragen mit dem Konstruktor $(\geq n\, R)$ ergibt, hat andere Ursachen als die der bereits vorgestellten Konstruktoren. Unerwünschte Fehlschläge von Assertionstests ergeben sich in diesem Fall dadurch, dass die Unique-Name-Assumption (UNA) nicht gilt. Wird zum Beispiel an die Wissensbasis aus Abschnitt 3.2.2

$$KB = \{hatKind(john, susy), hatKind(john, charles)\}$$

die Frage gestellt, ob John mindestens zwei Kinder hat, kann sich für einen Tableau-Zweig mit dieser Wissensbasis und der Negationsnormalform der negierten Anfrageassertion, die durch

$\neg(\geq 2\, hatKind) \Rightarrow (\leq 1\, hatKind)$ entsteht, kein Abschluss der Art (iii) (siehe Abschnitt 2.4.3) ergeben, weil nicht bekannt ist, dass $susy \neq charles$ gilt.

Die explizite Unterscheidung von bezeichneten Objekten wird im Kontext des Semantic Web aber dennoch oft vorgenommen. In der bereits erwähnten Weinontologie wird das mit <owl:differentFrom> oder mit <owl:AllDifferent> vollzogen.

3.3 Fallanalyse bei Anfragen mit Existenzrestriktionen

In diesem Abschnitt wird gezeigt, dass eine Anfrageassertion unter der OWA in jedem Modell gelten kann, ohne dass sie sich auf gleiche Konstanten in den einzelnen Modellen bezieht. Die Berücksichtigung solcher Fallunterscheidungen kann sehr aufwändig sein. Ein beliebtes Beispiel zur Darstellung von Fallunterscheidungen ist das von Oedipus [Baader und Nutt, 2003]. In der griechischen Mythologie tötete Oedipus seinen Vater, heiratete seine Mutter Iokaste und hatte mit dieser den Sohn Polyneikes gezeugt. Dieser wiederum hatte Thersandros als Sohn, von dem bekannt ist, das er seinen Vater nicht getötet hat. Die ABox dieser Familienstruktur ist in Abbildung 3.5 dargestellt und enthält folgende Zusicherungen:

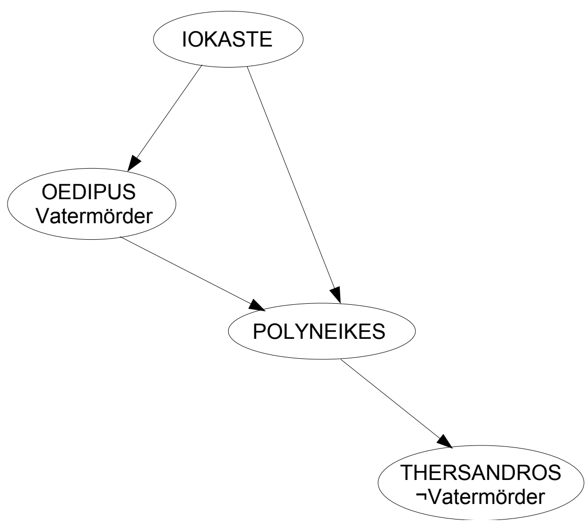


Abbildung 3.5: Graphische Darstellung der ABox des Oedipus-Beispiels

$$\begin{array}{cc} hatKind(iokaste, oedipus). & hatKind(iokaste, polyneikes). \\ hatKind(oedipus, polyneikes). & hatKind(polyneikes, thersandros). \\ Vaterm\ddot{o}rder(oedipus). & \neg Vaterm\ddot{o}rder(thersandros). \end{array}$$

Im Folgenden wird der Instance-Check diskutiert, durch den entschieden werden soll, ob Iokaste einen Vatermörder als Kind hat, der wiederum ein Kind hat, das kein Vatermörder ist:

$$\mathcal{A} \models \exists hatKind.(Vaterm\ddot{o}rder \sqcap \exists hatKind.\neg Vaterm\ddot{o}rder)(iokaste)$$

Für beide möglichen Rollenfüller der Relation *hatKind* bzgl. Iokaste, Oedipus und Polyneikes, gilt die Anfrageassertion dieses Instance-Checks nicht in allen Modellen der ABox. Wird der Rollenfüller Oedipus betrachtet, gilt nicht in allen Modellen, dass sein Sohn Polyneikes kein Vatermörder ist, und wird der Rollenfüller Polyneikes betrachtet, so gilt für diesen nicht in allen Modellen, dass er selbst ein Vatermörder ist, obwohl sein Sohn in allen Modellen kein Vatermörder ist. Aufgrund dieser Argumentation könnte angenommen werden, dass der Instance-Check nicht erfolgreich ist. Allerdings folgt die angegebene Konzeptassertion logisch aus der ABox. Relevante Modelle unterscheiden sich ausschließlich dadurch, dass entweder $Vaterm\ddot{o}rder(polyneikes)$ oder $\neg Vaterm\ddot{o}rder(polyneikes)$ gilt. Die Anfrageassertion gilt in Modellen, in denen Oedipus der Rollenfüller ist, wenn Polyneikes kein Vatermörder ist und sie gilt in Modellen, in denen Polyneikes der Rollenfüller ist, wenn er ein Vatermörder ist. In Abbildung 3.6 ist ein Tableau-Beweis für den Oedipus-Assertionstest dargestellt. Der erste Abschluss $\{Vaterm\ddot{o}rder(oedipus), \neg Vaterm\ddot{o}rder(oedipus)\}$ ergibt sich für alle Modelle, in denen Polyneikes kein Vatermörder ist und der zweite $\{Vaterm\ddot{o}rder(polyneikes), \neg Vaterm\ddot{o}rder(polyneikes)\}$ für alle, in denen er einer ist. In Kombination mit dem zweiten Abschluss wird durch den dritten $\{\neg Vaterm\ddot{o}rder(thersandros), Vaterm\ddot{o}rder(thersandros)\}$ sichergestellt, dass Polyneikes ein Kind hat, das kein Vatermörder ist, wenn er selbst ein Vatermörder ist.

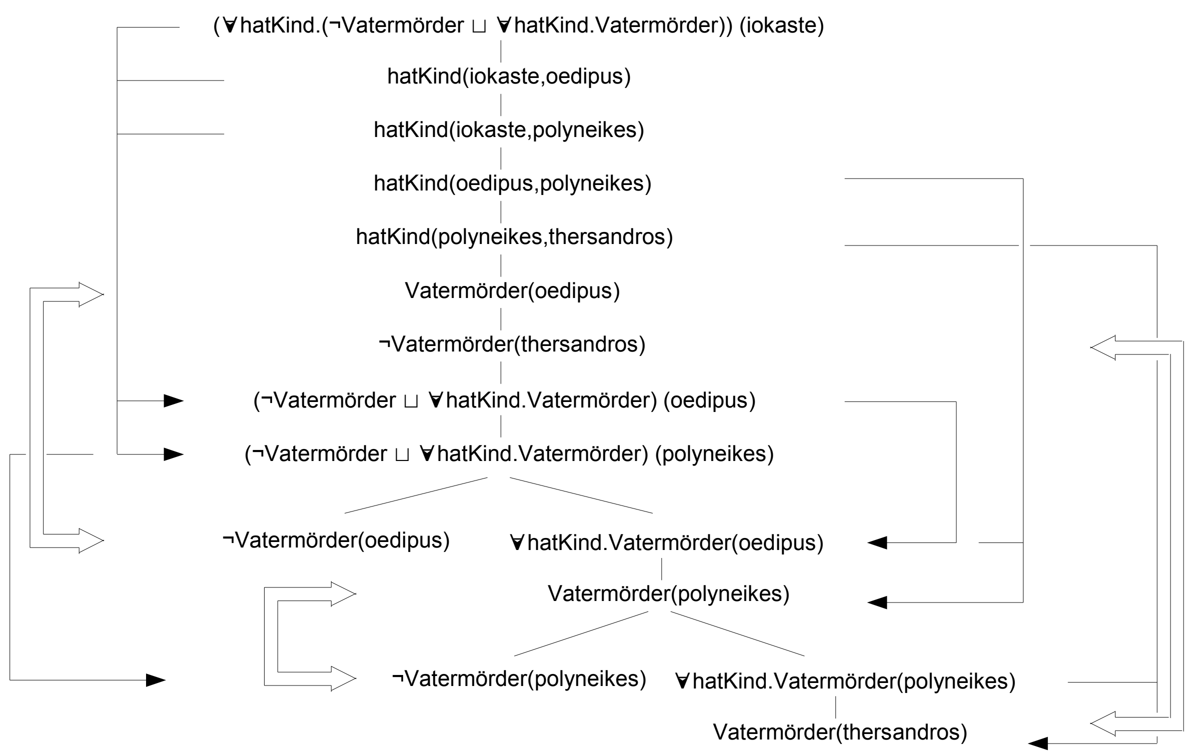


Abbildung 3.6: Beweis für den Instance-Check des Oedipus-Beispiels

Inferenzen durch die in diesem Abschnitt vorgestellten Fallunterscheidungen treten nur in Sprachen mit qualifizierten Existenzrestriktionen ($\mathcal{E}$) auf. In Sprachen mit unqualifizierten Existenzrestriktionen $\exists R.\top$ ergeben sich keine Bedingungen für Rollenfüller, die zu dieser Fallunterscheidung führen können. In [Donini et al., 1994] wurde bewiesen, dass Instance-Checks in Sprachen mit qualifizierten Existenzrestriktionen einen höheren Berechnungsaufwand erfordern als Reasoning Services, die keine Anfragen an eine ABox berücksichtigen. Während Erfüllbarkeits- und Subsumptionstests in $\mathcal{ALE}$ in coNP bzw. NP liegen, sind Instance-Checks in dieser Sprache PSPACE-vollständig.

Kapitel 4

Annahmen einer geschlossenen Welt

Ausgehend von einer Semantik, die auf der Open-World-Assumption basiert, besteht die Menge der Formeln, die inferiert werden kann, ausschließlich aus denjenigen Formeln, die logisch aus einer Wissensbasis folgen. Wird eine Wissensbasis unter dieser Semantik um zusätzliche Informationen ergänzt, ist ausgeschlossen, dass sich die Menge der inferierbaren Formeln verringern kann. Logiken mit dieser Eigenschaft werden monoton gennant und wurden im Kontext von beschreibungslogischen Sprachen in Kapitel 3 dargestellt.

In diesem Kapitel werden nicht-monotone Logiken untersucht, die auf Annahmen einer geschlossenen Welt basieren. Das sind Annahmen, die den aktuellen Wissensbestand als (möglichst) vollständig betrachten. Dazu werden der Wissensbasis – wenn bestimmte Kriterien erfüllt sind – negierte atomare Formeln hinzugefügt. Da sich diese Formeln ausschließlich auf Sprachelemente beziehen, die bereits in der Wissensbasis enthalten sind, wird die Menge der möglichen Modelle der Wissensbasis eingeschränkt. Das Ziel besteht darin, diese Menge zu minimieren, ohne die Konsistenz der Wissensbasis zu gefährden.

Die Ausführungen dieses Kapitels beziehen sich im Wesentlichen auf die Prädikatenlogik, gelten aber auch für Beschreibungslogiken, da diese Logiken entscheidbare Fragmente der Prädikatenlogik sind. Um den Begriff der Annahmen einer geschlossenen Welt eingrenzen zu können, wird in Abschnitt 4.1 zunächst der verallgemeinernde Begriff der Vervollständigung einer logischen Wissensbasis eingeführt. In Abschnitt 4.2 wird die *Closed-World-Assumption (CWA)* erläutert. Das ist die „klassische“ Annahme einer geschlossenen Welt nach Reiter [Reiter, 1978]. Es wird gezeigt, dass die CWA zu vollständigen Wissensbasen führt, so dass keine Anfrage mit „weiß ich nicht“ beantwortet werden kann. Allerdings führt die CWA in vielen Sprachen zu Inkonsistenzen. Daher werden in den Abschnitten 4.3 und 4.4 die *Generalized Closed-World-Assumption (GCWA)* und die *Weak Generalized Closed-World-Assumption (WGCWA)* für allgemeinere Wissensbasen beschrieben, die in den Fällen, in denen ihre Anwendung benötigt wird, weniger Inferenzen ermöglichen. Abschließend werden in Abschnitt 4.5 die Probleme angesprochen, die sich bei der Berücksichtigung von Annahmen einer geschlossenen Welt in den in dieser Untersuchung behandelten beschreibungslogischen Sprachen ergeben.

4.1 Vervollständigungen von logischen Wissensbasen

Annahmen einer geschlossenen Welt und viele andere Formen nicht-monotonen Schließens erweitern eine logische Wissensbasis. Eine solche Erweiterung kann auch als eine Vervollständigung dieser Wissensbasis bezeichnet werden. Das bedeutet aber nicht in allen Fällen, dass die erweiterte Wissensbasis vollständig ist. Vervollständigungen dienen in diesem Abschnitt der Darstellung grundlegender Eigenschaften aller in diesem Kapitel behandelten Annahmen. Für ein einführendes Beispiel wird eine Wissensbasis KB betrachtet, die der Tabelle aus Abschnitt 3.2.3 entspricht. Eine Vervollständigung $V(KB)$ dieser Wissensbasis ist beispielsweise die Assertion $\neg Gerinnungshemmer(paracetamol)$. Aus einer um diese Annahme vervollständigten Wissensbasis

$$\begin{aligned} KB^V = \ & \{Schmerzmittel(aspirin), Gerinnungshemmer(aspirin), \\ & Gerinnungshemmer(warfarin), Schmerzmittel(paracetamol), \\ & \neg Gerinnungshemmer(paracetamol)\} \end{aligned}$$

kann im Gegensatz zu dem in Abschnitt 3.2.3 dargestellten Beispiel die Konzeptassertion $Schmerzmittel \sqcap \neg Gerinnungshemmer(paracetamol)$ logisch gefolgert werden. Besteht Interesse daran, dass die hier verwendete Wissensbasis KB vollständig ist, kann KB mit der Menge $\{\neg Gerinnungshemmer(paracetamol), \neg Schmerzmittel(warfarin)\}$ erweitert werden.[1]

Definition 4.1.1 *Sei KB eine logische Wissensbasis und φ eine Formel. Eine Formelmenge $V(KB)$ ist eine Vervollständigung von KB, und für eine um diese Formelmenge vervollständigte Wissensbasis $KB^V = KB \cup V(KB)$ und eine Folgerungsrelation $\models_V$ zwischen KB und φ mit den Eigenschaften*

1. *Wenn $KB \models \varphi$, dann $KB \models_V \varphi$*
2. *Wenn $KB \models_V \varphi_i$ für $i = 1, ..., n$ und $\{\varphi_1, ..., \varphi_n\} \models \varphi$, dann $KB \models_V \varphi$*

gilt: $KB \models_V \varphi$ gdw. $KB^V \models \varphi$.

Dadurch, dass unter $\models_V$ dieselben Formeln folgen, die aus KB^V logisch folgen, berücksichtigt die Relation $\models_V$ eine um $V(KB)$ vervollständigte Wissensbasis. Die erste Eigenschaft von $\models_V$ besagt, dass diese Relation in jedem Fall eine „Oberrelation" der logischen Folgerung sein muss, so dass durch $\models_V$ keine Information verloren gehen kann [Brass und Lipeck, 1989]. Die zweite Eigenschaft sichert die Abgeschlossenheit gegenüber logischer Folgerung: Wenn zum Beispiel sowohl φ_1 als auch φ_2 mit $\models_V$ gefolgert werden können, dann muss das auch für $\varphi_1 \wedge \varphi_2$ gelten [Brass, 1989].

Eine wichtige Eigenschaft, die für alle in dieser Untersuchung verwendeten Vervollständigungen gilt, ist die *Äquivalenzerhaltung*. Das ist die Eigenschaft, dass äquivalente Wissensbasen äquivalent vervollständigt werden [Brass, 1989].

[1] Das entspricht der Anwendung der Closed-World-Assumption (siehe folgenden Abschnitt).

Durch vervollständigte logische Wissensbasen wird in vielen Fällen die Menge der inferierbaren Fakten vergrößert (es kann mehr bewiesen werden). Bei den zusätzlichen Fakten handelt es sich allerdings um Annahmen, die richtig oder falsch sein können. Ergeben sich durch neue Informationen (von einer verlässlichen Quelle) weitere Fakten, die einigen Annahmen widersprechen, müssen diese Annahmen zurückgenommen werden.

4.2 Closed-World-Assumption (CWA)

In diesem Abschnitt wird die für die Untersuchung zunächst relevante Form der Annahme einer geschlossenen Welt analysiert. Mit der Closed-World-Assumption wird angenommen, dass eine atomare Formel nicht gilt, wenn nicht bewiesen werden kann, dass sie gilt. Diese Vervollständigung hat die größte Wirkung, weil die Wissensbasis unter dieser Annahme als vollständig betrachtet werden kann.

Die Einführung einer CWA führt zu Schlussfolgerungen, die oft als natürlich angesehen werden: Entnimmt beispielsweise eine Person einem Zugfahrplan, dass Züge von einem bestimmten Gleis um 14.57 h, 16.57 h und 18.57 h abfahren, werden aufgrund der großen Menge an Information in dem Plan keine Uhrzeiten zu finden sein, zu denen keine Züge abfahren. Die Person wird sowohl annehmen, dass kein Zug um 15.57 h abfährt, als auch, dass es keinen gibt, der zu einer ungeraden Stunde abfährt.

Wie ausführlich in Kapitel 3 dargestellt wurde, gibt es viele Situationen, in denen man nicht an der Annahme einer offenen Welt interessiert ist oder sogar erwartet, dass der aktuelle Wissensbestand als vollständig betrachtet wird. Allerdings ist die CWA immer auf die Bereiche beschränkt, für die Informationen verfügbar sind. Die oben angesprochene Person wird zum Beispiel keine Schlussfolgerungen bzgl. der Abfahrtszeiten von anderen Bahnhöfen treffen, solange ihr keine Informationen über diese Abfahrtszeiten vorliegen. Schlussfolgerungen unter der CWA sind deswegen in vielen Fällen nicht erwünscht und können zu unabsehbaren Konsequenzen führen, sobald sich herausstellt, dass sie nicht korrekt sind. Aus diesem Grund ergibt sich die Forderung, die Entscheidung für die CWA stets abzuwägen und diese Annahme ausschließlich lokal zu verwenden.

Die CWA ist der *Negation-as-failure*-Semantik (NAF) sehr ähnlich: Wenn eine atomare Formel unter dieser Semantik logisch nicht abgeleitet werden kann, wird ebenfalls angenommen, dass die Negation der atomaren Formel gilt. Allerdings erfolgt die Einbeziehung dieser Semantik im Kontext von logischen Programmiersprachen, so dass die Negation-as-failure-Regel nur dann angewendet werden kann, wenn das Fehlschlagen der Ableitung terminiert [Brass, 1989].

In diesem Abschnitt wird zunächst die Definition der CWA vorgestellt (4.2.1). Anschließend (4.2.2) werden wichtige Eigenschaften der CWA erläutert. Dazu gehören die Vollständigkeit des repräsentierten Wissens und die Gefährdung der Konsistenz in komplexeren Wissensbasen. Der Unterabschnitt 4.2.3 zeigt auf, warum die CWA in vielen logischen Sprachen nur bei Verwendung weiterer Annahmen sinnvoll eingesetzt werden kann und der Unterabschnitt

4.2.4, unter welchen Bedingungen beliebige prädikatenlogische Anfragen unter der CWA durch rekursive Zerlegung auf atomare Anfragen unter der OWA zurückgeführt werden können.

4.2.1 Definition der CWA

In [Reiter, 1978] wird darauf hingewiesen, dass die Anzahl negativer Fakten eines betrachteten Weltausschnitts bei weitem die von positiven Fakten überschreitet. Die explizite Repräsentation aller negativen Fakten ist deswegen sehr aufwändig (zum Beispiel ist einer Person das Wissen um alle Zeiten, zu denen eine Zugabfahrt nicht geplant ist, nach Anfrage bekannt, aber alle diese Zeiten können nicht explizit im Gedächtnis festgehalten werden). Wenn zusätzlich eine Wissensbasis nicht nur aus atomaren Formeln besteht, ist es ohne eine Form der Inferenz nicht möglich, alle negativen Fakten zu erhalten. In [Reiter, 1978] wird eine Wissensbasis unter der Annahme der CWA deswegen für alle atomaren Formeln, die aus ihr nicht logisch folgen, um die entsprechenden negativen atomaren Formeln *implizit* vervollständigt:

Definition 4.2.1 *Seien KB eine Wissensbasis, AF die Menge aller atomaren Formeln ohne Variablen aus KB und φ eine beliebige Formel. Dann ist*

$$CWA(KB) = \{\neg p \mid p \in AF \text{ und } KB \not\models p\}$$

die Closed-World-Assumption von KB und für die Folgerungsrelation $\models_{cwa}$, die diese Annahme berücksichtigt, gilt:

$$KB \models_{cwa} \varphi \text{ gdw. } KB^+ \models \varphi, \text{ mit } KB^+ = KB \cup CWA(KB)$$

Eine Formel φ folgt nach dieser Definition aus einer Wissensbasis unter der CWA genau dann, wenn sie logisch aus der um die CWA vervollständigten Wissensbasis KB^+ folgt. In Wissensbasen, die nur aus atomaren Formeln bestehen, kann die Definition vereinfacht werden, so dass $CWA(KB) = \{\neg p \mid p \in AF \text{ und } p \notin KB\}$ gilt.

Im Kontext von Beschreibungslogiken steht die atomare Formel p für atomare Konzept- und Rollenassertionen $A(a)$ und $R(a,b)$. Eine Assertionstest für KB, der die CWA berücksichtigt, ist genau dann erfolgreich, wenn für KB^+ ein Assertionstest unter der OWA erfolgreich ist. In Wissensbasen, die nach einer Expansion bzgl. einer TBox $\mathcal{T}$ nur aus atomaren Formeln bestehen, kann die Definition analog zu dem prädikatenlogischen Fall vereinfacht werden, und es gilt $CWA(KB) = \{\neg p \mid p \in AF \text{ und } p \notin \mathcal{A}_{\mathcal{T}}\}$.

Definition 4.2.2 *Seien KB eine beschreibungslogische Wissensbasis und Konst(KB) die Menge aller in KB vorkommender Konstanten. Für jedes atomare Konzept A ist*

$$CWA(A) = \{\neg A(a) \mid a \in Konst(KB) \text{ und } KB \not\models A(a)\}$$

der Konzeptabschluss von A und für jede atomare Rolle R ist

$$CWA(R) = \{\neg R(a,b) \mid a, b \in Konst(KB) \text{ und } KB \not\models R(a,b)\}$$

der Rollenabschluss von R.

4.2.2 Eigenschaften der CWA

Neben den Eigenschaften, die für jede Vervollständigung gelten, werden die allgemeinen Eigenschaften der Vollständigkeit, der Äquivalenzerhaltung sowie die Eigenschaft, dass unter der CWA keine zusätzlichen atomaren Formeln bewiesen werden können, erläutert. Anschließend wird gezeigt, unter welchen Bedingungen die CWA zu Inkonsistenzen führt. Alle diese Eigenschaften werden in [Reiter, 1978] oder [Brass, 1989] definiert und bewiesen. Der Abschnitt schließt mit dem Bezug der CWA zu minimalen Herbrand-Modellen ab.

4.2.2.1 Allgemeine Eigenschaften

Theorem 4.2.3 (Vollständigkeit) *Für Formeln φ ohne Quantoren und ohne Identität gilt: $KB \models_{cwa} \varphi$ oder $KB \models_{cwa} \neg\varphi$.*

Wenn die CWA nicht berücksichtigt wird, können Wissensbasen im Allgemeinen unvollständig sein. Schon in dem Fall, in dem z.B. eine beschreibungslogische Wissensbasis ausschließlich die Formeln $A_1(a)$ und $A_2(b)$ enthält, gilt weder $KB \models A_1(b)$ noch $KB \models \neg A_1(b)$ (analog für $A_2(a)$). Vollständigkeit einer logischen Wissensbasis ist bei Erhaltung der Konsistenz die wünschenswerteste Eigenschaft einer Vervollständigung. Aus einer solchen Wissensbasis kann jede Formel oder die Negation dieser Formel gefolgert werden, es entsteht also in keinem Fall die Aussage: „Weiß ich nicht." Für Formeln mit Quantoren gilt ein entsprechendes Theorem nur, wenn zusätzlich der Domänenabschluss berücksichtigt wird (siehe Abschnitt 4.2.3.1).

Theorem 4.2.4 (Äquivalenzerhaltung) *Für jedes Paar von Wissensbasen KB_1 und KB_2 gilt: Wenn $KB_1 \equiv KB_2$, dann $KB_1^+ \equiv KB_2^+$.*

Die Äquivalenzerhaltung garantiert, dass die CWA ausschließlich die Semantik einer Wissensbasis berücksichtigt, also nicht zwischen syntaktisch unterschiedlichen, aber semantisch äquivalenten Wissensbasen unterscheidet.

Theorem 4.2.5 (keine zusätzlichen Beweise für atomare Formeln) *Sei KB eine Wissensbasis, p eine atomare Formel und KB^+ konsistent. Dann gilt: $KB \models_{cwa} p$ gdw. $KB \models p$.*

Wenn KB^+ konsistent ist, führen beliebige atomare Anfragen unter der CWA zu denselben Ergebnissen wie unter der OWA, können also zum Beispiel mit einem Tableau-Kalkül korrekt und vollständig auf Erfolg überprüft werden.

4.2.2.2 Konsistenzeigenschaften

Die Vervollständigung einer konsistenten Wissensbasis KB um die CWA führt in vielen Fällen zu einer inkonsistenten Wissensbasis KB^+. Sei $KB_I = \{(A_1 \sqcup A_2)(a)\}$. Dann gilt sowohl $KB_I \not\models A_1(a)$ als auch $KB_I \not\models A_2(a)$ und $CWA(KB_I) = \{\neg A_1(a), \neg A_2(a)\}$. Die um $CWA(KB_I)$

erweiterte Wissensbasis KB_I^+ ist in diesem Fall inkonsistent. In [Reiter, 1978] wird die Konsistenzerhaltung der in diesem Artikel konstruierten CWA untersucht. Dort wird bewiesen, dass die CWA in einer großen Klasse von logischen Wissensbasen die Konsistenz erhält. Bei diesen handelt es sich um Wissensbasen, die ausschließlich *Horn-Klauseln* enthalten.

Definition 4.2.6

- *Eine Klausel $L_1 \vee ... \vee L_n$ ist eine Disjunktion von Literalen L_i der Form p_i oder $\neg p_i$, wobei p_i atomare Formeln sind.*
- *Eine Grundklausel ist eine Klausel, die keine Variablen enthält.*
- *Eine Horn-Klausel ist eine Grundklausel mit höchstens einem positiven Literal.*
- *Eine indefinite Klausel ist eine positive Grundklausel $p_1 \vee ... \vee p_n$ mit $n > 1$.*

Die oben dargestellte beschreibungslogische Wissensbasis KB_I, die durch die Vervollständigung durch die CWA zu einer Inkonsistenz führt, besteht aus einer indefiniten Klausel $(A_1 \sqcup A_2)(a)$ mit zwei positiven Literalen.

Theorem 4.2.7 (Konsistenzerhaltung in Horn-Wissensbasen) *Angenommen, eine Wissensbasis KB ist konsistent und besteht ausschließlich aus Horn-Klauseln. Dann ist $KB^+ = KB \cup CWA(KB)$ ebenfalls konsistent [Reiter, 1978].*

Wissensbasen, die nicht ausschließlich aus Horn-Klauseln bestehen, sind unter der CWA nicht zwangsläufig inkonsistent. Das folgende Theorem, das die inkonsistenten Zustände einer Wissensbasis aufzeigt, bildet die Motivation für die Verwendung generalisierter Annahmen (Abschnitte 4.3 und 4.4).

Theorem 4.2.8 (Inkonsistenz) *Ist KB konsistent, dann ist KB^+ genau dann inkonsistent, wenn es positive Literale $L_1, ..., L_n$ ohne Variablen gibt, für die gilt: $KB \models L_1 \vee ... \vee L_n$ und $KB \not\models L_i$ für alle $i = 1, ..., n$.*

4.2.2.3 Bezug zu minimalen Herbrand-Modellen

Im Folgenden wird der Bezug der CWA zu minimalen Herbrand-Modellen dargestellt. Es werden dabei weder Terme mit Funktionssymbolen noch Terme mit Variablen berücksichtigt, so dass dieser Bezug auch im Kontext einer beschreibungslogischen ABox gilt.

Herbrand-Interpretationen $\mathcal{I}_H = (\triangle_H^{\mathcal{I}}, \cdot_H^{\mathcal{I}})$ sind Interpretationen mit folgenden Eigenschaften: $\triangle_H^{\mathcal{I}}$ ist das Herbrand-Universum und besteht aus allen in *KB* vorkommenden Konstanten, und $\cdot_H^{\mathcal{I}}$ ist die Herbrand-Interpretationsfunktion mit $c_H^{\mathcal{I}} = c$ für alle Konstanten c. Herbrand-Interpretationen unterscheiden sich nur durch die Interpretationsfunktion für Prädikate. Deswegen können sie mit der Menge der Grundatome (atomare Formeln ohne Variablen), die in ihnen wahr sind, beschrieben werden [Van Emden und Kowalski, 1976]. Dabei wird im Folgenden $p \in \mathcal{I}_H$ an Stelle von $\mathcal{I}_H \models p$ für atomare Formeln p verwendet. Herbrand-Interpretationen sind Modelle für ein Axiom bzw. für eine Wissensbasis, wenn sie das Axiom bzw. alle Axiome der Wissensbasis erfüllen. Ein Größenvergleich zweier Herbrand-Modelle kann durch den Vergleich der Mengen der Grundatome, die ihnen wahr sind, erfolgen. Ein Herbrand-Modell, für das kein weiteres Herbrand-Modell existiert, das weniger Elemente enthält, wird *minimales Herbrand-Modell* genannt.

Theorem 4.2.9 (Bezug zu minimalem Herbrand-Modell) *Sei KB eine konsistente Horn-Wissensbasis. Dann gibt es für KB genau ein minimales Herbrand-Modell* Γ_{min}*, das der Schnittmenge aller Herbrand-Modelle* Γ_i*,* $i = 1, .., n$*, von KB entspricht, und für jede atomare Formel p ohne Variablen gilt: KB* $\models_{cwa} p$ *gdw.* $p \in \Gamma_{min}$ *und KB* $\models_{cwa} \neg p$ *gdw.* $p \notin \Gamma_{min}$*.*

Die Menge der Herbrand-Modelle einer konsistenten Horn-Wissensbasis kann unter Berücksichtigung der CWA folglich auf ein minimales Herbrand-Modell reduziert werden. In [Van Emden und Kowalski, 1976] ist die Existenz eines minimalen Herbrand-Modells in konsistenten Horn-Wissensbasen zu finden und in [Eiter und Gottlob, 1993] der Bezug eines minimalen Herbrand-Modells zu Folgerungen unter der CWA.

4.2.3 Domänenabschluss (DCA) und Unique-Name-Assumption (UNA)

Das DCA und die UNA sind mögliche Erweiterungen der CWA (und aller anderen in diesem Kapitel vorgestellten Annahmen). Im Folgenden wird die Bedeutung dieser Annahmen für die CWA insbesondere in Beschreibungslogiken analysiert. Es ergibt sich, dass die Eigenschaft der Vollständigkeit einer Wissensbasis, die die CWA berücksichtigt und zusätzlich um das DCA und die UNA erweitert wird, für beliebige Anfragen gilt.

4.2.3.1 Der Domänenabschluss

Der Domänenabschluss ist die Annahme, dass die in einer Wissensbasis vorkommenden Konstanten alle Konstanten einer Domäne sind. Um diese Annahme zu treffen, wird einer Wissensbasis *KB* das folgende Domänenabschlussaxiom (engl.: domain closure axiom, DCA) hinzugefügt:

$$DCA(KB) = (\forall x)\ [x \doteq c_1 \vee ... \vee x \doteq c_m].$$

Dabei sind $c_1, ..., c_m$ alle in *KB* vorkommenden Konstanten. Die Domäne $\triangle^{\mathcal{I}}$ einer Interpretation $\mathcal{I}$ ist in diesem Fall also explizit bekannt, so dass es nur endlich viele Modelle geben kann. Im Kontext von Beschreibungslogiken ist dann $\triangle^{\mathcal{I}} = \top^{\mathcal{I}} = \{c_1^{\mathcal{I}}, ..., c_m^{\mathcal{I}}\}$. Diese Information bildet zunächst einen einfacheren Zugang für die Konstruktion einer Form der CWA in Beschreibungslogiken. Bei der Ausführung eines (erweiterten) Retrieval Services wird diese Annahme ebenfalls getroffen, um entsprechende Instance-Checks bzw. Relationentests auszuführen zu können. Die Erweiterung der Closed-World-Assumption um dieses Axiom wird definiert durch [Brachman und Levesque, 2004]:

$$KB \models_{cwa^\diamond} \varphi \;\; gdw. \;\; KB^\diamond \models \varphi, \; mit\, KB^\diamond = KB \cup CWA(KB) \cup DCA(KB).$$

Die Folgerungsrelation $\models_{cwa^\diamond}$ ist eine stärkere Folgerungsrelation als $\models_{cwa}$, weil die Anzahl der möglichen Modelle weiter eingeschränkt wird. Dadurch können zusätzliche Formeln bewiesen werden. Ist eine Wissensbasis beispielsweise gegeben durch

$$\begin{aligned} KB = \{ & wurdeVerschrieben(warfarin, patient), Patient(patient), \\ & Medikament(morphium)\} \end{aligned}$$

kann unter der CWA nicht bewiesen werden, dass keinem Patienten Morphium verschrieben wurde, d.h. $KB \not\models_{cwa} \neg\exists wurdeVerschrieben.Patient(morphium)$. Der Assertionstest ist nicht erfolgreich, weil die CWA nicht alle der Wissensbasis unbekannten Konstanten als nichtexistente Rollenfüller berücksichtigen kann. Wird die CWA allerdings um das DCA erweitert, gilt $KB \models_{cwa^\diamond} \neg\exists wurdeVerschrieben.Patient(morphium)$, weil für alle bekannten Konstanten c_i gezeigt werden kann, dass $KB \models_{cwa} \neg wurdeVerschrieben(morphium, c_i)$ gilt.

Aufgrund der Äquivalenz von negierten Existenzrestriktionen und Werterestriktionen ergibt sich diese Problematik ebenfalls bei Verwendung von Werterestriktionen. Scheitern Assertionstests mit diesen Konstruktoren, ist zusätzlich nicht gewährleistet, dass entsprechend negierte Anfragen $KB \models_{cwa} \exists R.C(a)$ bzw. $KB \models_{cwa} \neg\forall R.C(a)$ bewiesen werden können, und es folgt, dass die Wissensbasis bzgl. Anfragen mit beliebigen Quantoren unvollständig ist. Das Theorem 4.2.3 der Vollständigkeit kann deswegen nur bei Berücksichtigung des DCA auf Formeln, die möglicherweise Quantoren enthalten, erweitert werden:

Theorem 4.2.10 (Vollständigkeit$^\diamond$) *Für Formeln φ mit Quantoren und ohne Identität gilt: $KB \models_{cwa^\diamond} \varphi$ oder $KB \models_{cwa^\diamond} \neg\varphi$.*

Wenn das DCA verwendet wird, können sich Inkonsistenzen ergeben. Das ist für Formeln der Fall, in denen Existenzquantoren verwendet werden, durch die Skolem-Konstanten berücksichtigt werden müssen, die nicht mit den Konstanten des DCA identifiziert werden können (eine Behandlung von Skolem-Konstanten erfolgt in Abschnitt 7.2).

4.2.3.2 Die Unique-Name-Assumption

Die Unique-Name-Assumption ist die Annahme, dass unterschiedliche Konstanten unterschiedliche Objekte bezeichnen. In den Abschnitten 2.4.2 und 3.1 wurde bereits einerseits auf ihre mögliche Verwendung in beschreibungslogischen Tableau-Kalkülen für Konstanten, die in einer ABox explizit vorkommen, sowie andererseits auf den häufigen Verzicht dieser Annahme in offenen Welten hingewiesen. Dort wurde auch die Entstehung von Inkonsistenzen mit dieser Annahme dargestellt.

Die Unique-Name-Assumption einer Wissensbasis KB ist die Menge

$$UNA(KB) = \{(a_i \neq a_j) \mid a_i \text{ und } a_j \text{ sind unterschiedliche Konstanten}\}.$$

Wird eine Wissensbasis nicht um $UNA(KB)$ erweitert, ist die Beantwortung von Anfragen mit den Formeln $(a \doteq b)$ und $(a \neq b)$ nicht vollständig [Brachman und Levesque, 2004], unabhängig davon, ob die Wissensbasis um die CWA erweitert wurde oder nicht. Im Kontext von beschreibungslogischen Sprachen tritt diese Problematik zum Beispiel bei der Verwendung von Anfrageassertionen mit mindestens-Anzahlrestriktionen auf. In Abschnitt 3.2.4 wurde gezeigt, dass ein Abschluss für Anfrageassertionen mit diesem Konstruktor nicht ohne die Berücksichtigung der UNA erfolgen kann. Dabei war nicht entscheidend, ob die CWA verwendet wurde. Allerdings ist die UNA im Kontext von logischen Theorien, die an Stelle der mengentheoretischen Definition der CWA Vervollständigungsaxiome verwenden [Reiter, 1984] (in Beschreibungslogiken können das Sprachen mit dem one-of-Operator sein [Schaerf, 1994b]) von großer Bedeutung für die CWA.

Anfragen mit Formeln $(a \doteq b)$ und $(a \neq b)$ können vollständig beantwortet werden, wenn die Folgerungsrelation $\models_{una}$ mit

$$KB \models_{una} \varphi \;\; gdw. \;\; KB \cup UNA(KB) \models \varphi$$

verwendet wird. Wird $KB^{\diamond}$ um $UNA(KB)$ erweitert, gilt die Eigenschaft der Vollständigkeit für beliebige prädikatenlogische Anfragen.

4.2.4 Rekursive Anfragebearbeitung unter der CWA

In diesem Abschnitt wird vorgeführt, dass mit einer um die CWA, das DCA und die UNA erweiterten Wissensbasis beliebige Anfragen rekursiv auf Anfragen mit atomaren Formeln reduziert werden können, wenn die erweiterte Wissensbasis konsistent ist [Brachman und Levesque, 2004]. Die korrekte Beantwortung dieser atomaren Formeln führt dann zu einer korrekten Antwort der ursprünglichen Anfrage.

Für konjunktive Anfragen und doppelte Negationen gelten die folgenden Bedingungen:

$$\begin{array}{lll} KB \models (\varphi_1 \wedge \varphi_2) & gdw. & KB \models \varphi_1 \textit{ und } KB \models \varphi_2 \\ KB \models \neg\neg\varphi & gdw. & KB \models \varphi \end{array}$$

Wird eine Anfragesprache um Disjunktionen erweitert, gilt zusätzlich:

$$KB \models_{cwa} (\varphi_1 \vee \varphi_2) \;\; gdw. \;\; KB \models_{cwa} \varphi_1 \textit{ oder } KB \models_{cwa} \varphi_2$$

Bei Verwendung der logischen Folgerung ist die Reduktion von Disjunktionen nicht korrekt, wenn die Wissensbasis die Formel $(\varphi_1 \vee \varphi_2)$ selbst beinhaltet. Unter der CWA gilt diese Reduktion aber aufgrund der Vollständigkeit der Wissensbasis (Theorem 4.2.3).

Für Anfragen mit Quantoren muss die CWA um das DCA erweitert werden (siehe Abschnitt 4.2.3.1). Nach [Brachman und Levesque, 2004] gelten die folgenden Reduktionen:

$$\begin{array}{lll} KB \models_{cwa^\diamond} \forall x \varphi & gdw. & KB \models_{cwa^\diamond} \varphi_{[x/c_i]} \text{ für alle in } KB \text{ vorkommenden Konstanten } c_i \\ KB \models_{cwa^\diamond} \exists x \varphi & gdw. & KB \models_{cwa^\diamond} \varphi_{[x/c_i]} \text{ für eine in } KB \text{ vorkommende Konstante } c_i \end{array}$$

Bei Verwendung der UNA gelten die folgenden Reduktionen für Formeln mit $\doteq$:

$$KB \models_{una} (a \doteq b) \quad gdw. \quad a \text{ und } b \text{ sind dieselben Konstanten}$$
$$KB \models_{una} (a \not\doteq b) \quad gdw. \quad a \text{ und } b \text{ sind unterschiedliche Konstanten}$$

Theorem 4.2.11 *Ist KB eine Wissensbasis und die um die CWA erweiterte Wissensbasis KB^+ konsistent, dann gilt für Formeln φ ohne Quantoren und ohne Identität:*

$$KB \models_{cwa} \neg\varphi \;\; gdw. \;\; KB \not\models_{cwa} \varphi.$$

Beweis Angenommen $KB \models_{cwa} \neg\varphi$ und $KB \models_{cwa} \varphi$. Dann wäre KB^+ inkonsistent. Es wurde aber gefordert, dass KB^+ konsistent ist. Angenommen $KB \not\models_{cwa} \neg\varphi$ und $KB \not\models_{cwa} \varphi$. Dann wäre die Wissensbasis unvollständig. KB^+ ist aber vollständig für Formeln ohne Quantoren und ohne Identität (Theorem 4.2.3). □

In Abschnitt 4.2.3 wurde gezeigt, dass das Theorem der Vollständigkeit für beliebige Formeln gilt, wenn eine Wissensbasis um die CWA, das DCA und die UNA erweitert wurde, so dass Theorem 4.2.11 bei zusätzlicher Einbeziehung des DCA und der UNA ebenfalls für Formeln mit Quantoren und der Relation $\doteq$ gilt, wenn die Konsistenz weiterhin erhalten bleibt.

Die Annahmen des DCA und der UNA erweitern die CWA hinsichtlich einer geschlossenen Welt. Die Folgerungsrelation $\models_{CWA}$ berücksichtigt alle diese Annahmen und wird mit Großbuchstaben notiert, um von der Folgerungsrelation $\models_{cwa}$, die ausschließlich die CWA berücksichtigt, unterschieden werden zu können. Diese Relation ist definiert durch:

$$KB \models_{CWA} \varphi \;\; gdw. \;\; KB \cup CWA(KB) \cup DCA(KB) \cup UNA(KB) \models \varphi.$$

Alle in diesem Abschnitt vorgestellten Reduktionsregeln gelten für $\models_{CWA}$. Wird diese Folgerungsrelation verwendet, kann eine beliebige prädikatenlogische Anfrage rekursiv auf atomare Anfragen zurückgeführt werden. Aufgrund von Theorem 4.2.5 entsteht ein Rekursionsabschluss dann durch die Prüfung, ob $KB \models p$ oder $KB \not\models p$ für atomare Formeln p gilt.

Voraussetzung für die Korrektheit eines solchen rekursiven Verfahrens ist die Konsistenz der erweiterten Wissensbasis. Eine konsistente um die CWA, das DCA und die UNA vervollständigte Wissensbasis, wird in [Brachman und Levesque, 2004, Kapitel 16] auch *lebendig* (engl.: vivid) genannt. Diese Wissensbasen haben die Eigenschaft, dass es für sie nur ein einziges Modell gibt. Das rekursive Verfahren kann deswegen die ursprüngliche, nicht erweiterte Wissensbasis verwenden, als wenn es sich um eine Anfrage an eine relationale Datenbank handeln würde. Die Annahmen, um die diese Wissensbasis erweitert wird, werden für die Anfragebearbeitung nicht benötigt, garantieren aber die Korrektheit des Verfahrens.

In Abschnitt 7.1 wird eine rekursive Anfragebearbeitung unter der CWA für Beschreibungslogiken vorgestellt. Dort wird erläutert, aus welchen Gründen die Reduktionsregeln für entsprechende Konzeptkonstruktoren gelten, in welchen Sprachen dieses Verfahren korrekt ist und wie in diesen Sprachen Rekursionsabschlüsse durchgeführt werden können.

4.3 Generalized Closed-World-Assumption (GCWA)

In den Fällen, in denen bei Erweiterung einer konsistenten Wissensbasis um die CWA Inkonsistenzen entstehen (Theorem 4.2.8), kann eine generalisierte Form der CWA verwendet werden. Die Generalized Closed-World-Assumption (GCWA) [Minker, 1982] garantiert, dass die Konsistenz beliebiger Wissensbasen erhalten bleibt. Im Gegensatz zu der CWA führt die GCWA aber nicht in jedem Fall zu einer vollständigen Wissensbasis, so dass es weiterhin Anfragen geben kann, die mit „weiß ich nicht" beantwortet werden.

Generalisierte Annahmen sollten getroffen werden, wenn die verwendete Wissensbasis indefinit ist.[2] Eine Wissensbasis ist nicht zwangsläufig indefinit, wenn sie indefinite Klauseln enthält: Die Wissensbasis $KB = \{p_1 \vee p_2, \neg p_1\}$ zum Beispiel ist äquivalent zu der Horn-Wissensbasis $KB_{horn} = \{\neg p_1, p_2\}$. Es gilt $KB \models \neg p_1$ und $KB \models p_2$, so dass es für KB keine indefiniten Antworten gibt [Minker, 1982].

In Abschnitt 4.3.1 wird eine beweistheoretische Definition der GCWA vorgestellt. Es wird gezeigt, dass die Menge aller atomaren Formeln, die sich aus dem Vokabular einer Wissensbasis bilden lassen, aus der Menge der definiten atomaren Formeln, der Menge der atomaren Formeln, die unter der GCWA nicht gelten und aus der Menge der indefiniten atomaren Formeln besteht. Anschließend (Abschnitt 4.3.2) werden wichtige Eigenschaften der GCWA vorgestellt. Dazu gehören die Konsistenzerhaltung, der Bezug der GCWA zu der CWA und eine modelltheoretische Definition der GCWA, die äquivalent zu der beweistheoretischen Definition ist.

4.3.1 Definition der GCWA

Die hier verwendete Definition der GCWA verallgemeinert die Definition der CWA durch die zusätzliche Berücksichtigung von Klauseln, die aus einer Wissensbasis logisch folgen und basiert auf der syntaktischen (beweistheoretischen) Definition nach [Minker, 1982].

Definition 4.3.1 *Sei KB eine Wissensbasis, AF die Menge aller Grundatome aus KB, PK die Menge aller positiver Grundklauseln aus KB und φ eine beliebige Formel. Dann ist*

$$GCWA(KB) = \{\neg p \mid p \in AF,\ K \in PK,\ KB \not\models p, \text{ und wenn } KB \models p \vee K, \text{ dann } KB \models K\}$$

die Generalized Closed-World-Assumption von KB und für die Folgerungsrelation $\models_{gcwa}$, die diese Annahme berücksichtigt, gilt:

$$KB \models_{gcwa} \varphi \text{ gdw. } KB^* \models \varphi, \text{ mit } KB^* = KB \cup GCWA(KB).$$

Aus dieser Definition ergibt sich, dass unter der GCWA Literale indefiniter Klauseln konsistent vervollständigt werden können, wenn über andere Literale dieser Klauseln definite Information verfügbar ist. Die Wissensbasis $KB = \{p_1 \vee p_2, p_1\}$ zum Beispiel kann unter der GCWA mit $\{\neg p_2\}$ vervollständigt werden. Es gilt $KB \models_{gcwa} p_1$ und $KB \models_{gcwa} \neg p_2$.

[2] Eine Übersicht über indefinite Wissensbasen ist in [Fernández und Minker, 1992] zu finden.

In einer möglicherweise indefiniten Wissensbasis KB gibt es unter der Berücksichtigung der GCWA für jede atomare Formel p, die keine Variablen enthält, die folgenden vier Möglichkeiten [Henschen und Park, 1988]:

1. p ist nachweislich wahr, $KB \models p$
2. p ist nachweislich falsch, $KB \models \neg p$
3. p gilt nicht unter der GCWA, $KB \models_{gcwa} \neg p$
4. p ist indefinit, $KB \not\models p$, $KB \not\models \neg p$ und $KB \not\models_{gcwa} \neg p$

Dieser Sachverhalt läßt sich auf Formelmengen erweitern: Die Menge der nachweislich gültigen atomaren Formeln p ist dann $\Gamma_P(KB) = \{p \mid KB \models p\}$, die Menge der indefiniten atomaren Formeln $\Gamma_I(KB) = \{p \mid KB \models p \vee K \textit{ und } KB \not\models K\}$, die Menge der Formeln, die unter der GCWA als falsch angenommen werden, $\Gamma_{gcwa} = \{p \mid \neg p \in GCWA(KB)\}$, und für die Menge $\mathcal{F}_p(KB)$ aller atomaren Formeln p, die sich durch Prädikaten- und Konstantensymbole, die in KB vorkommen, bilden läßt, gilt dann:

$$\mathcal{F}_p(KB) = \Gamma_P(KB) \cup \Gamma_I(KB) \cup \Gamma_{gcwa}(KB)$$

Die Menge der atomaren Formeln, die nachweislich ungültig sind, kann dabei vernachlässigt werden, weil sie sich aus der Vervollständigung um die GCWA ergibt. Es besteht Interesse daran, dass $\Gamma_I(KB)$ möglichst wenig Elemente enthält, so dass die meisten Anfragen mit „ja“ oder „nein“ entschieden werden können.

Beispiel 4.3.2 Gegeben sei die beschreibungslogische Wissensbasis

$$KB = \{(A_1 \sqcup A_2)(a), (A_3 \sqcup \neg A_4)(a), A_5(a)\}.$$

Dann ist $\mathcal{F}_p(KB) = \{A_1(a), A_2(a), A_3(a), A_4(a), A_5(a)\}$, $\Gamma_P(KB) = \{A_5(a)\}$, $GCWA(KB) = \{\neg A_3(a), \neg A_4(a)\}$ und demzufolge $\Gamma_{gcwa}(KB) = \{A_3(a), A_4(a)\}$. Für die atomare Konzeptassertion $A_1(a)$ zum Beispiel gilt zwar $KB \not\models A_1(a)$, aber auch $KB \models (A_1 \sqcup A_2)(a)$, ohne dass $KB \models A_2(a)$ gilt, so dass $KB \not\models_{gcwa} \neg A_1(a)$. Da Entsprechendes für $A_2(a)$ gilt, ist $\Gamma_I(KB) = \{A_1(a), A_2(a)\}$.

4.3.2 Eigenschaften der GCWA

In [Minker, 1982] werden die Eigenschaften der Konsistenzerhaltung, dem Bezug der GCWA zu der CWA und zu minimalen Herbrand-Modellen, die Eigenschaft, dass es durch die GCWA keine zusätzlichen Beweise für positive Grundklauseln gibt sowie die Eigenschaft der maximalen Konsistenz vorgestellt und bewiesen.

Theorem 4.3.3 (Bezug zu der CWA) *Sei KB eine Wissensbasis und KB^+ konsistent. Dann gilt: $KB^* = KB^+$.*

In Wissensbasen, die bzgl. der Erweiterung um die CWA konsistent sind (z.B. Horn-Wissensbasen), entspricht die GCWA der CWA. Die Menge $\Gamma_I(KB)$ der indefiniten atomaren Formeln ist in diesem Fall leer.

Theorem 4.3.4 (Konsistenzerhaltung) *Sei KB eine konsistente Wissensbasis. Dann ist KB^* ebenfalls konsistent, so dass für keine Formel φ gilt: $KB \models_{gcwa} \varphi$ und $KB \models_{gcwa} \neg\varphi$.*

In den Fällen, in denen eine Wissensbasis bei der Vervollständigung einer negierten atomaren Formel durch die CWA inkonsistent wird (das gilt für Wissensbasen, die das Theorem 4.2.8 erfüllen), unterbleibt eine Vervollständigung um diese Formel durch die GCWA. Die GCWA kann deswegen für jede konsistente Wissensbasis verwendet werden.

Theorem 4.3.5 (keine zusätzlichen Beweise für positive Grundklauseln) *Sei KB eine konsistente Wissensbasis und K eine Grundklausel, die nur aus positiven Literalen besteht. Dann gilt: $KB \models_{gcwa} K$ gdw. $KB \models K$.*

Unter Berücksichtigung der GCWA können nur diejenigen positiven Grundklauseln gefolgert werden, die logisch gefolgert werden können. Klauseln, die mit $\models_{gcwa}$ zusätzlich gefolgert werden können, enthalten also mindestens ein negatives Literal. Das Theorem folgt direkt aus [Minker, 1982, Theorem 4].

Theorem 4.3.6 (maximale Konsistenz) *Seien KB und KB^* konsistente Wissensbasen und $\neg p$ eine negierte atomare Formel, die nicht in KB^* enthalten ist. Dann ist $KB^* \cup \{\neg p\}$ inkonsistent oder es gibt eine positive Grundklausel K, so dass $KB \not\models K$ und $KB^* \cup \{\neg p\} \models K$.*

Wenn KB^* um zusätzliche negierte atomare Formeln erweitert wird, entsteht eine inkonsistente Wissensbasis oder es wird die Eigenschaft, dass keine zusätzlichen positiven Grundklauseln bewiesen werden können, aufgehoben.

Theorem 4.3.7 (Bezug zu minimalen Herbrand-Modellen) *Seien KB eine Wissensbasis, p eine atomare Formel ohne Variablen und Γ_{min^i}, $i = 1, ..., n$, alle minimalen Herbrand-Modelle von KB. Dann gilt: $KB \models_{gcwa} \neg p$ gdw. $p \notin \Gamma_{min^i}$ für alle $i = 1, ..., n$.*

Dieses Theorem ist eine Erweiterung des Theorems 4.2.9 und entspricht der semantischen (modelltheoretischen) Definition der GCWA. Eine Wissensbasis wird unter der GCWA um eine negierte atomare Formel vervollständigt, wenn die entsprechende atomare Formel in keinem minimalen Herbrand-Modell enthalten ist. In [Minker, 1982] wird diese Definition ausführlich dargestellt. Dort wird ebenfalls die Äquivalenz dieser Definition zu der Definition 4.3.1 bewiesen.

4.4 Weak Generalized Closed-World-Assumption (WGCWA)

Die Weak Generalized Closed-World-Assumption (WGCWA) ist wie die GCWA eine generalisierte Annahme und kann in jeder konsistenten Wissensbasis verwendet werden, ohne die Konsistenz zu gefährden. Es handelt sich um eine Annahme, die in vielen Fällen weniger Inferenzen als die GCWA ermöglicht und deswegen „schwächer" als diese ist. Die WGCWA wird in [Rajasekar et al., 1989] definiert, um Inferenzen unter der Annahme einer geschlossenen Welt in möglicherweise indefiniten Wissensbasen mit geringerem Aufwand durchführen zu können.

Definition 4.4.1 *Sei KB eine Wissensbasis, AF die Menge aller Grundatome aus KB, PK die Menge aller positiver Grundklauseln aus KB und φ eine beliebige Formel. Dann ist*

$$WGCWA(KB) = \{\neg p \mid p \in AF,\ K \in PK,\ KB \not\models p \text{ und } KB \not\models p \vee K\}$$

die Weak Generalized Closed-World-Assumption von KB und für die Folgerungsrelation $\models_{wgcwa}$, die diese Annahme berücksichtigt, gilt:

$$KB \models_{wgcwa} \varphi \text{ gdw. } KB^{(*)} \models \varphi, \text{ mit } KB^{(*)} = KB \cup WGCWA(KB).$$

Die Definition ist angelehnt an die der GCWA, erlaubt aber in keinem Fall eine Vervollständigung um eine negierte atomare Formel, wenn die entsprechende nicht-negierte atomare Formel Bestandteil einer Klausel ist, die logisch aus der Wissenbasis folgt.

Aufgrund der Beziehung $WGCWA(KB) \subseteq GCWA(KB)$ gelten viele der Eigenschaften, die für die GCWA erläutert wurden, auch für die WGCWA. Das betrifft den Bezug zu der CWA (Theorem 4.3.3), die Konsistenzerhaltung (Theorem 4.3.4) und die Eigenschaft, dass es keine zusätzlichen Beweise für positive Klauseln gibt (Theorem 4.3.5). Die WGCWA ist aber nicht maximal konsistent (Theorem 4.3.6) und neben der syntaktischen Definition unterscheidet sich auch eine semantische Definition der WGCWA von der der GCWA [Rajasekar et al., 1989].

4.5 Problematik in beschreibungslogischen Wissensbasen

In Wissensbasen, die ausschließlich Horn-Klauseln enthalten, ist die Berücksichtigung einer Annahme einer geschlossenen Welt mit viel geringerem Aufwand verbunden als in indefiniten Wissensbasen. Dadurch, dass Horn-Wissensbasen bei Erweiterung um die CWA konsistent bleiben (Theorem 4.2.7), kann das rekursive Verfahren, das in Abschnitt 4.2.4 dargestellt wurde, verwendet werden, um Anfragen unter der CWA korrekt zu beantworten.

Im Kontext von Beschreibungslogiken enthalten zum Beispiel $\mathcal{AL}_0$-Wissensbasen keine indefiniten Assertionen, so dass Vervollständigungen dieser Wissensbasen um die CWA konsistenzerhaltend sind. Wissensbasen, die ausdrucksstärker als $\mathcal{AL}_0$-Wissensbasen sind, können allerdings zu unvollständig spezifiziertem Wissen führen: Für die $\mathcal{ALC}$-Konzeptassertion

$$(\exists hasPet.(Dog \sqcup Cat))(bill)$$

ist beispielsweise weder vollständig spezifiziert, welches Individuum ein Haustier von Bill ist, noch, ob es ein Hund oder eine Katze ist [Baader, 1999].

In dieser Arbeit untersuchte, bzgl. einer TBox $\mathcal{T}$ expandierte, beschreibungslogische Wissensbasen $\mathcal{A}_\mathcal{T}$ sind indefinit und deswegen unter Berücksichtigung der CWA möglicherweise inkonsistent, wenn sie Konzeptassertionen mit folgenden Konstruktoren enthalten:

- Konzeptdisjunktionen $C \sqcup D$
- Existenzrestriktionen $\exists R.C$
- mindestens-Anzahlrestriktionen $(\geq n\, R)$

In Abschnitt 4.5.1 wird anhand eines Beispiels gezeigt, dass sich durch Konzeptdisjunktionen in beschreibungslogischen Wissensbasen unter Berücksichtigung der CWA Inkonsistenzen ergeben können. Bei Interesse an einer Annahme einer geschlossenen Welt führt diese Problematik zu der Verwendung generalisierter Annahmen, die nicht in allen Fällen eine vollständige Wissensbasis zur Folge haben. Anschließend wird in Abschnitt 4.5.2 erläutert, aus welchen Gründen unvollständig spezifizierte Rollenfüller, die sich durch die Konstruktoren $\exists R.C$ und $(\geq n\, R)$ ergeben, unter Einbeziehung der CWA ebenfalls zu einer widersprüchlichen Wissensbasis führen können. Es ergibt sich das Problem, dass eine Vervollständigung um generalisierte Annahmen in Sprachen, die unvollständig spezifizierte Rollenfüller enthalten, nicht ohne Weiteres erfolgen kann.

4.5.1 Konzeptdisjunktionen

Konzeptdisjunktionen können bei Vervollständigung einer Wissensbasis um die CWA – wie bereits zu Beginn des Abschnitts 4.2.2.2 dargestellt wurde – zu Inkonsistenzen führen. Sei beispielsweise die folgende $\mathcal{ALU}$-Wissensbasis gegeben durch:

$$KB = \{(Intelligent \sqcup Reich)(john)\}$$

Dann gilt sowohl $KB \not\models Intelligent(john)$ als auch $KB \not\models Reich(john)$, und die Vervollständigung um die CWA ist $CWA(KB) = \{\neg Intelligent(john), \neg Reich(john)\}$. Die vervollständigte Wissensbasis $KB^+ = \{(Intelligent \sqcup Reich)(john), \neg Intelligent(john), \neg Reich(john)\}$ ist inkonsistent.

Um die Konsistenz dieser Wissensbasis zu erhalten, kann zum Beispiel die GCWA verwendet werden. Unter Berücksichtigung dieser Annahme gilt allerdings sowohl $KB \not\models_{gcwa} Intelligent(john)$ und $KB \not\models_{gcwa} Reich(john)$ als auch $KB \not\models_{gcwa} \neg Intelligent(john)$ und $KB \not\models_{gcwa} \neg Reich(john)$, und es ergeben sich die Mengen $GCWA(KB) = \{\}$ und $\Gamma_I(KB) = \{Intelligent(john), Reich(john)\}$. Da Vervollständigungen um die WGCWA Teilmengen von Vervollständigungen um die GCWA sind, ist ebenfalls $WGCWA(KB) = \{\}$, so dass keine konsistenzerhaltende Vervollständigung von KB um eine Annahme einer geschlossenen Welt eine Wirkung hat.

4.5.2 Unvollständig spezifizierte Rollenfüller

Enthält eine beschreibungslogische Wissensbasis die Konzeptassertion $\exists R.\top(a)$, ist bekannt, dass die Konstante a über die Rolle R zu einem Rollenfüller in Beziehung steht. Allerdings ist nicht spezifiziert, um welchen Rollenfüller es sich dabei handelt. Im Kontext von Datenbanken werden diese unvollständig spezifizierten Rollenfüller als Nullwerte bezeichnet. In dieser Untersuchung wird aber der Begriff *Skolem-Konstante* verwendet. Skolem-Konstanten *new* wurden in Kapitel 2 im Zusammenhang mit dem Tableau-Kalkül eingeführt. Sie können mit einer aus einer endlichen Menge bekannten Konstanten oder mit einer völlig neuen Konstanten identifiziert werden. In [Minker, 1982] wird der Fall analysiert, in dem Skolem-Konstanten in jeder Interpretation mit einer in einer Wissensbasis *KB* vorkommenden Konstanten identifiziert werden. Wenn $c_1, ..., c_m$ alle diese Konstanten sind, gilt bei Spezifikation von unqualifizierten Existenzrestriktionen $\exists R.\top(a)$ in *KB*:

$$KB \models R(a, c_1) \vee ... \vee R(a, c_m)$$

Durch eine solche implizite Disjunktion kann die Berücksichtigung von Skolem-Konstanten bei Vervollständigung um die CWA entsprechend der Disjunktion von Konzepten zu Inkonsistenzen führen. Angenommen, es ist bekannt, dass Aspirin und Warfarin Medikamente sind und dass ein Arzt irgendetwas verschreibt:

$$KB = \{Medikament(aspirin), Medikament(warfarin), \exists verschreibt.\top(arzt)\}$$

Es ergibt sich $KB \not\models verschreibt(arzt, aspirin)$, $KB \not\models verschreibt(arzt, warfarin)$ und $KB \not\models verschreibt(arzt, arzt)$. Bei Vervollständigung um $CWA(KB) = \{\neg verschreibt(arzt, aspirin), \neg verschreibt(arzt, warfarin), \neg verschreibt(arzt, arzt)\}$ entsteht bei zusätzlicher Berücksichtigung des Domänenabschlusses $DCA(KB) = (\forall x)[x \doteq aspirin \vee x \doteq warfarin \vee x \doteq arzt]$ eine inkonsistente Wissensbasis.

Um Inkonsistenzen zu vermeiden, kann *KB* um generalisierte Annahmen vervollständigt werden. Da durch $\exists verschreibt.\top(arzt)$ implizite Disjunktionen berücksichtigt werden müssen, gilt aber sowohl $KB \not\models_{gcwa} \neg verschreibt(arzt, c_i)$ als auch $KB \not\models_{wgcwa} \neg verschreibt(arzt, c_i)$ für alle in *KB* vorkommenden Konstanten c_i, so dass die Verwendung der GCWA und der WGCWA in dieser Wissensbasis keine Wirkung hat.

Weitere Konstruktoren, die zu der Berücksichtigung unvollständig spezifizierter Rollenfüller führen, sind qualifizierte Existenzrestriktionen und mindestens-Anzahlrestriktionen. Qualifizierte Existenzrestriktionen $\exists R.C$ erweitern unqualifizierte Existenzrestriktionen dahingehend, dass die zu berücksichtigende Skolem-Konstante in der Extension der Konzeptbeschreibung C enthalten sein muss, und mindestens-Anzahlrestriktionen $(\geq n\, R)$ spezifizieren implizit n paarweise disjunkte Skolem-Konstanten. Für diese Konstruktoren ergeben sich zunächst ähnliche Bedingungen bzgl. Annahmen einer geschlossenen Welt, die sich für unqualifizierte Existenzrestriktionen ergeben, so dass die CWA nicht verwendet werden kann. Für Wissensbasen, die

Konzeptassertionen mit diesen Konstruktoren enthalten, besteht die Problematik allerdings darin, diese um generalisierte Annahmen zu vervollständigen. Aus der $\mathcal{ALE}$-Wissensbasis

$$KB = \{\exists R.\forall R.A(a), R(a, b)\}$$

beispielsweise können die Mengen $GCWA(KB)$ und $WGCWA(KB)$ nicht ohne Weiteres bestimmt werden. Es ist nicht ersichtlich, welche negierten Rollenassertionen und welche negierten atomaren Konzeptassertionen unter der GCWA bzw. der WGCWA folgen und welche nicht, da die Folgerung unter diesen Annahmen mit Klauseln definiert ist.

Kapitel 5

Epistemische Assertionstests

Die Verwendung eines epistemischen Operators **K** (engl. knows) erweitert in vielerlei Hinsicht die Ausdrucksstärke einer beschreibungslogischen Sprache. Mit diesem Operator ist die Spezifikation von Default-Regeln, Integritätsbedingungen und epistemischen Anfragen möglich [Grimm und Motik, 2005]. Im Zusammenhang mit der in dieser Arbeit untersuchten Problematik werden in diesem Kapitel allerdings ausschließlich epistemische Anfragen (ohne freie Variablen) behandelt, die ausführlich in [Schaerf, 1994a] dargestellt sind. Diese Anfragen werden im Folgenden *epistemische Assertionstests* genannt. Epistemische Assertionstests ermöglichen Inferenzen, die nicht nur die in einer Wissensbasis repräsentierten Aspekte einer externen Welt betreffen, sondern auch das, was eine Wissensbasis über diese Aspekte weiß. Dazu wird ausschließlich die Anfragesprache um den epistemischen Operator **K** erweitert. In Abschnitt 5.1 werden die Syntax und die Semantik der Sprache $\mathcal{ALC}/\mathcal{ALCK}$ sowie epistemische Assertionstests in dieser Sprache definiert. Anschließend wird in Abschnitt 5.2 dargestellt, in welchen Fällen die Ergänzung von Anfrageassertionen um epistemische Operatoren zu der Berücksichtigung einer lokalen CWA führt. Es ergibt sich, dass die in Abschnitt 5.1 definierten epistemischen Assertionstests keine generalisierten Annahmen einer geschlossenen Welt berücksichtigen können.

5.1 Die Sprache $\mathcal{ALC}/\mathcal{ALCK}$

In diesem Abschnitt ist die Definition epistemischer Assertionstests $KB \models \alpha_{_K}$ mit einer $\mathcal{ALC}$-Wissensbasis KB und einer $\mathcal{ALCK}$-Anfrageassertion $\alpha_{_K}$ von Interesse. Da die Sprache $\mathcal{ALC}$ bereits in Kapitel 2 vorgestellt wurde, erfolgt zunächst die Definiton der Syntax und der Semantik von $\mathcal{ALCK}$, der Erweiterung von $\mathcal{ALC}$ um den epistemischen Operator **K**.

Der Ausdruck $\mathbf{K}C$ bezeichnet die Menge der Konstanten, von denen bekannt ist (von denen die Wissensbasis weiß), dass sie Instanzen des Konzepts C sind. Das entspricht der Menge der Konstanten, die in jedem Modell der Wissensbasis in der Extension von C enthalten sind [Schaerf, 1994a]. Entsprechendes gilt für eine atomare Rolle P und die Notation $\mathbf{K}P$.

Konzeptbeschreibungen C, D und Rollen R (negierte Rollen $\neg R$ sind für negierte epistemische Relationentests zusätzlich erlaubt) werden in $\mathcal{ALCK}$ durch die folgenden Syntaxregeln geformt:

$$
\begin{array}{rcl}
C, D & \longrightarrow & \top \mid \bot \mid A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C \mid \mathbf{K}C \\
R & \longrightarrow & P \mid \mathbf{K}P
\end{array}
$$

Eine *epistemische Interpretation* ist ein Tupel $(\mathcal{I}, \mathcal{W})$ mit einer Interpretation $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$ und einer Menge von Interpretationen $\mathcal{W}$. Wenn sowohl die Domäne als auch die Abbildung von Konstanten auf Elemente der Domäne unabhängig von einer Interpretation $\mathcal{I}$ festgelegt sind, kann die Domäne mit $\triangle$ bezeichnet werden und die epistemische Interpretation der Sprachausdrücke von $\mathcal{ALCK}$ ist in diesem Fall definiert durch:

$$
\begin{array}{rcl}
\top^{\mathcal{I},\mathcal{W}} & = & \triangle \\
\bot^{\mathcal{I},\mathcal{W}} & = & \emptyset \\
A^{\mathcal{I},\mathcal{W}} & = & A^{\mathcal{I}} \\
P^{\mathcal{I},\mathcal{W}} & = & P^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I},\mathcal{W}} & = & C^{\mathcal{I},\mathcal{W}} \cap D^{\mathcal{I},\mathcal{W}} \\
(C \sqcup D)^{\mathcal{I},\mathcal{W}} & = & C^{\mathcal{I},\mathcal{W}} \cup D^{\mathcal{I},\mathcal{W}} \\
(\neg C)^{\mathcal{I},\mathcal{W}} & = & \triangle \setminus C^{\mathcal{I},\mathcal{W}} \\
(\neg R)^{\mathcal{I},\mathcal{W}} & = & \triangle \times \triangle \setminus R^{\mathcal{I},\mathcal{W}} \\
(\forall R.C)^{\mathcal{I},\mathcal{W}} & = & \{a \in \triangle \mid (\forall b)\,[(a,b) \in R^{\mathcal{I},\mathcal{W}} \rightarrow b \in C^{\mathcal{I},\mathcal{W}}]\} \\
(\exists R.C)^{\mathcal{I},\mathcal{W}} & = & \{a \in \triangle \mid (\exists b)\,[(a,b) \in R^{\mathcal{I},\mathcal{W}} \wedge b \in C^{\mathcal{I},\mathcal{W}}]\} \\
(\mathbf{K}C)^{\mathcal{I},\mathcal{W}} & = & \bigcap_{\mathcal{J} \in \mathcal{W}} (C^{\mathcal{J},\mathcal{W}}) \\
(\mathbf{K}P)^{\mathcal{I},\mathcal{W}} & = & \bigcap_{\mathcal{J} \in \mathcal{W}} (P^{\mathcal{J},\mathcal{W}})
\end{array}
$$

Die Extension von $\mathbf{K}C$ bzw. $\mathbf{K}P$ besteht aus den Elementen, von denen bekannt ist, dass sie Instanzen von C bzw. P in allen Interpretationen der Menge $\mathcal{W}$ sind. Die Interpretation dieser Ausdrücke mit der Schnittmenge der entsprechenden Extensionen ist korrekt, da die Domäne unabhängig von einer Interpretation $\mathcal{I}$ festgelegt ist [Schaerf, 1994a]. Für Sprachausdrücke, die keine epistemischen Operatoren enthalten, entsprechen epistemische Interpretationen für $\mathcal{ALCK}$ Interpretationen für $\mathcal{ALC}$.

Definition 5.1.1 *Sei KB_K eine $\mathcal{ALCK}$-Wissensbasis. Ein epistemisches Modell für KB_K ist eine epistemische Interpretation $(\mathcal{I}, \mathcal{W})$ mit den folgenden Eigenschaften:*

1. *$\mathcal{I} \in \mathcal{W}$*
2. *$\mathcal{W}$ ist eine maximale Menge von Interpretationen $\mathcal{J}$, so dass für jedes $\mathcal{J} \in \mathcal{W}$ jedes Axiom aus KB_K in $(\mathcal{J}, \mathcal{W})$ erfüllt ist.*

Für eine $\mathcal{ALC}$-Wissensbasis KB entspricht die maximale Menge von Interpretationen, für die jedes Axiom in KB erfüllt ist, der Menge der Modelle dieser Wissensbasis, so dass $\mathcal{W} = \mathcal{M}(KB)$ gilt. Ein epistemisches Modell für KB ist deswegen analog für $(\mathcal{I}, \mathcal{M}(KB))$ definiert.

Definition 5.1.2 *Sei KB eine $\mathcal{ALC}$-Wissensbasis und α_K eine $\mathcal{ALCK}$-Assertion. α_K folgt logisch aus KB, notiert mit $KB \models \alpha_K$, wenn α_K in allen epistemischen Modellen von KB gilt. Ein epistemischer Assertionstest für die Sprache $\mathcal{ALC}/\mathcal{ALCK}$ ist ein Inferenzprozess, der prüft, ob $KB \models \alpha_K$ oder $KB \not\models \alpha_K$. Im ersten Fall ist der epistemische Assertionstest erfolgreich, im zweiten Fall nicht.*

Beispiel 5.1.3 In Abschnitt 3.2.1 wurde die Wissensbasis

$$KB_2 = \{hatKind(vater, kind), Weiblich(kind)\}$$

spezifiziert. Der dort angegebene Assertionstest $KB_2 \models \forall hatKind.Weiblich(vater)$ ist nicht erfolgreich. Der epistemische Assertionstest

$$KB_2 \models \forall \mathbf{K} hatKind.Weiblich(vater)$$

dagegen ist erfolgreich: Für alle epistemischen Modelle $(\mathcal{I}, \mathcal{M}(KB_2))$ der Wissensbasis KB_2 ist $vater^{\mathcal{I}} \in (\forall \mathbf{K} hatKind.Weiblich)^{\mathcal{I},\mathcal{M}(KB_2)} = \{a \in \triangle \mid (\forall b)\, [(a, b) \in (\mathbf{K} hatKind)^{\mathcal{I},\mathcal{M}(KB_2)} \rightarrow b \in Weiblich^{\mathcal{I},\mathcal{M}(KB_2)}]\}$. Die Extension von $(\mathbf{K} hatKind^{\mathcal{I},\mathcal{M}(KB_2)})$ ist in allen epistemischen Modellen $\{(vater, kind)\}$, da nur die Rollenassertion $hatKind(vater, kind)$ in allen Interpretationen $\mathcal{J} \in \mathcal{M}(KB_2)$ von $hatKind$ erfüllt ist.[1] Der epistemische Assertionstest ist erfolgreich, weil für die Konstante $kind$, dem einzigen Rollenfüller von $vater$ bzgl. der Rolle $hatKind$, der der Wissensbasis bekannt ist, $kind^{\mathcal{I}} \in Weiblich^{\mathcal{I}} = Weiblich^{\mathcal{I},\mathcal{M}(KB_2)}$ für alle nicht-epistemischen Interpretationen $\mathcal{I}$ gilt.

In [Schaerf, 1994a] und [Donini et al., 1992] sowie ausführlich in [Donini et al., 1998] wird ein $\mathcal{ALC}/\mathcal{ALCK}$-Tableau-Kalkül vorgestellt, mit dem epistemische Assertionstests mit exponentieller Zeit- und polynomieller Speicherkomplexität auf Erfolg überprüft werden können. Der Kalkül besteht im Wesentlichen aus einer Erweiterung des $\mathcal{ALC}$-Tableau-Kalküls um zusätzliche Bedingungen für den Abschluss von Zweigen, die erfüllt sind, wenn bestimmte andere Zweige abgeschlossen oder nicht abgeschlossen sind. Wie in Abschnitt 5.2.2 aufgezeigt wird, entsprechen epistemische Assertionstests allerdings nicht in jedem Fall der logischen Folgerung aus einer vervollständigten Wissensbasis, so dass Anfrageassertionen mit dem $\mathcal{ALC}$-Kalkül aus Abschnitt 2.4.2 bewiesen werden können, deren Ergänzung um epistemische Operatoren nicht mit dem $\mathcal{ALC}/\mathcal{ALCK}$-Kalkül bewiesen werden können. Der Kalkül wird hier nicht näher beschrieben, weil der Schwerpunkt dieser Untersuchung nicht auf einer Erweiterung von Beschreibungslogiken um epistemische Operatoren liegt.

5.2 Lokale CWA durch den epistemischen Operator K

Es gibt Situationen, in denen die Semantik einer offenen Welt erwünscht ist, und andere, in denen eine Form der Closed-World-Assumption berücksichtigt werden soll. In vielen Applikationen (insbesondere im Kontext des Semantic Web) besteht deswegen Interesse an der

[1] Es gilt $hatKind^{\mathcal{J},\mathcal{M}(KB_2)} = hatKind^{\mathcal{J}}$.

Annahme einer offenen Welt, mit der zusätzlichen Möglichkeit, das Wissen über spezielle atomare Konzepte oder Rollen als vollständig zu betrachten. In diesem Abschnitt wird gezeigt, dass epistemische Operatoren in Anfragen eine solche lokale CWA ermöglichen.

Zunächst wird in Abschnitt 5.2.1 anhand der Sprache $\mathcal{AL}_0/\mathcal{ALCK}$ vorgeführt, dass die Ergänzung aller Teilkonzepte und Rollen einer Anfrageassertion um epistemische Operatoren in $\mathcal{AL}_0$-Wissensbasen zu denselben Inferenzen führt, die sich aus einer um die CWA nach [Reiter, 1978] erweiterten $\mathcal{AL}_0$-Wissensbasis ergeben. Anschließend wird in Abschnitt 5.2.2 anhand der Sprache $\mathcal{ALC}/\mathcal{ALCK}$ erläutert, dass das nicht für komplexere Wissensbasen gilt. Es wird gezeigt, dass eine lokale CWA in dieser Sprache (und in vielen anderen Sprachen mit $\mathcal{ALCK}$-Anfrageassertionen) nur erfolgen kann, wenn ausschließlich negierte atomare Assertionen sowie Rollen einer Werterestriktion um einen epistemischen Operator ergänzt werden.

5.2.1 CWA in $\mathcal{AL}_0/\mathcal{ALCK}$

Die Sprache $\mathcal{AL}_0$ stellt die Konzeptkonstruktoren $\top$, $\bot$, A, $\neg A$, $C \sqcap D$ und $\forall R.C$ zur Verfügung. Dadurch, dass in dieser Sprache weder Konzeptdisjunktionen enthalten sind noch Skolem-Konstanten berücksichtigt werden müssen, sind $\mathcal{AL}_0$-Wissensbasen äquivalent zu einer endlichen Menge von Horn-Klauseln. Ist eine Horn-Wissensbasis KB_{horn} erfüllbar, gibt es genau ein minimales Herbrand-Modell Γ_{min} für diese Wissensbasis. Dieses Modell wird durch die Erweiterung von KB_{horn} um die CWA berücksichtigt (siehe Theorem 4.2.9). Das gilt auch für Anfragen mit Quantoren, wenn die CWA um das Domänenabschlussaxiom (Abschnitt 4.2.3.1) erweitert wird.

Die Berücksichtigung dieses minimalen Herbrand-Modells kann aber auch durch eine $\mathcal{ALCK}$-Anfrage erfolgen. Bedingt durch die Interpretation epistemischer Sprachausdrücke mit der Schnittmenge entsprechender Extensionen wird Γ_{min} genau dann berücksichtigt, wenn der Operator $\mathbf{K}$ jedem atomaren Konzept A und jeder Rolle P eines Anfrageausdrucks E vorangestellt wird. Handelt es sich bei dem Ausdruck E um einen $\mathcal{ALC}$-Ausdruck, dann ist der $\mathcal{ALCK}$-Ausdruck $\overline{E}$ definiert durch (vgl. [Schaerf, 1994a]):

$$\begin{array}{rcl@{\qquad}rcl}
\overline{A} & = & \mathbf{K}A & \overline{C \sqcap D} & = & \overline{C} \sqcap \overline{D} \\
\overline{\neg A} & = & \neg\mathbf{K}A & \overline{C \sqcup D} & = & \overline{C} \sqcup \overline{D} \\
\overline{P} & = & \mathbf{K}P & \overline{\exists P.C} & = & \exists \mathbf{K}P.\overline{C} \\
\overline{\neg P} & = & \neg\mathbf{K}P & \overline{\forall P.C} & = & \forall \mathbf{K}P.\overline{C}
\end{array}$$

Aus den oben dargestellten Gründen entspricht ein epistemischer Assertionstest $KB \models \overline{E}(a)$ mit einer $\mathcal{AL}_0$-Wissensbasis KB und einer $\mathcal{ALCK}$-Anfrageassertion $\overline{E}(a)$ einem um die CWA und das DCA erweiterten Assertionstest, so dass gilt:

$$KB \models_{cwa^\diamond} E(a) \quad gdw. \quad KB \models \overline{E}(a)$$

Im Gegensatz zu Assertionstests in $\mathcal{AL}_0/\mathcal{ALC}$ können Assertionstests in $\mathcal{AL}_0/\mathcal{ALCK}$ mit einer Anfrageassertion $\overline{E}(a)$ mit polynomiellem Berechnungsaufwand auf Erfolg überprüft werden.

5.2.2 Lokale CWA in $\mathcal{ALC}/\mathcal{ALCK}$

Die Inferenzen, die mit epistemischen Operatoren in Anfrageassertionen getroffen werden können, entsprechen in vielen Fällen nicht denen einer vervollständigten Wissensbasis. Es gibt $\mathcal{ALCK}$-Assertionen, die nicht logisch aus einer $\mathcal{ALC}$-Wissensbasis folgen, obwohl entsprechende $\mathcal{ALC}$-Assertionen logisch aus dieser Wissensbasis folgen. Beispielsweise gelten für die in Abbildung 5.1 dargestellte $\mathcal{ALC}$-Wissensbasis $\mathcal{U}$ einer Universitätsdomäne die beiden folgenden Bedingungen [Schaerf, 1994a]:

$$\begin{aligned} \mathcal{U} &\models \exists enrolled.Grad(ee282), \\ \mathcal{U} &\not\models \exists \mathbf{K} enrolled.Grad(ee282). \end{aligned}$$

Der epistemische Assertionstest ist nicht erfolgreich, da der Wissensbasis kein Rollenfüller von $ee282$ bzgl. $enrolled$ bekannt ist, der in allen Modellen in der Extension von $Grad$ enthalten ist. Aus diesem Beispiel ist ersichtlich, dass durch die Ergänzung von Rollen um einen epistemischen Operator keine Skolem-Konstanten berücksichtigt werden können. Die Erweiterung von Anfrageassertionen, in denen Existenzrestriktionen (oder mindestens-Anzahlrestriktionen) vorkommen, um epistemische Operatoren kann also dazu führen, dass Assertionstests nicht mehr erfolgreich sind. Dasselbe ergibt sich für Anfrageassertionen mit Konzeptdisjunktionen:

$$\begin{aligned} \mathcal{U} &\models (Grad \sqcup Professor)(john) \\ \mathcal{U} &\not\models \mathbf{K} Grad \sqcup \mathbf{K} Professor(john), \end{aligned}$$

Die um epistemische Operatoren erweiterte Anfrageassertion folgt nicht logisch aus $\mathcal{U}$, weil weder $Grad(john)$ noch $Professor(john)$ in allen Modellen von $\mathcal{U}$ erfüllt ist.

Mit Anfrageassertionen, die verschachtelte Existenzrestriktionen $\exists P_1.(\exists P_2.D \sqcap ...)$ enthalten, können sich unter der OWA Fallunterscheidungen ergeben (siehe Abschnitt 3.3). Das Hinzufügen epistemischer Operatoren zu diesen Assertionen kann aufgrund der lokalen Berücksichtigung aller Modelle ebenfalls zu einer geringeren Anzahl an Inferenzen führen, auch wenn keine Skolem-Konstanten berücksichtigt werden müssen:

$$\begin{aligned} \mathcal{U} &\models \exists teaches.(Course \sqcap \exists enrolled.Grad \sqcap \exists enrolled.\neg Grad)(john) \\ \mathcal{U} &\not\models \exists teaches.\mathbf{K}(Course \sqcap \exists enrolled.Grad \sqcap \exists enrolled.\neg Grad)(john) \end{aligned}$$

Die in der ersten Anfrage berücksichtigte Fallunterscheidung wird durch das Hinzufügen des epistemischen Operators unterbunden. Da John keinen Kurs unterrichtet, für den in allen Modellen die aufgeführten Bedingungen gelten, schlägt der epistemische Assertionstest fehl. Der epistemische Assertionstest ist allerdings erfolgreich, wenn der letzten Bedingung ein zusätzlicher epistemischer Operator hinzugefügt wird ($\exists teaches.\mathbf{K}(Course \sqcap \exists enrolled.Grad \sqcap \exists enrolled.\neg \mathbf{K} Grad)$), weil in diesem Fall $\neg \mathbf{K} Grad(susan)$ in allen Modellen gilt und dadurch für den Rollenfüller $cs221$ von John bzgl. der Rolle $teaches$ die aufgeführten Bedingungen in allen Modellen gelten.

$Grad(mary)$.	$\neg Grad(peter)$.	$(Grad \sqcup Professor)(john)$.
$Course(cs221)$.	$Course(cs324)$.	$Course(ee282)$.
$\exists enrolled.Grad(ee282)$.	$\exists enrolled.\neg Grad(ee282)$.	$enrolled(cs221, mary)$.
$enrolled(cs221, susan)$.	$enrolled(cs324, susan)$.	$enrolled(cs324, peter)$.
$enrolled(ee282, peter)$.	$teaches(john, cs221)$.	$teaches(john, cs324)$.

Abbildung 5.1: Assertionale Universitätswissensbasis $\mathcal{U}$

Inferenzen, die denen einer vervollständigten Wissensbasis entsprechen, ergeben sich in $\mathcal{ALC}/\mathcal{ALCK}$, wenn epistemische Operatoren bei den bisher nicht angesprochenen Konzeptkonstruktoren verwendet werden. Während epistemische Operatoren vor Konzeptkonjunktionen keine Wirkung haben,[2] ergeben sich in vielen Fällen weitere Inferenzen durch Verwendung von $\neg\mathbf{K}A$ an Stelle von $\neg A$, $\neg\mathbf{K}P$ an Stelle von $\neg P$ und $\forall\mathbf{K}P.C$ an Stelle von $\forall P.C$.[3] Diese Sprachausdrücke führen zu der Berücksichtigung einer lokalen CWA. Es gelten die Bedingungen:

$$KB \models \neg\mathbf{K}A(a) \quad gdw. \quad KB \not\models A(a) \tag{5.2.2.1}$$

$$KB \models \neg\mathbf{K}P(a,b) \quad gdw. \quad KB \not\models P(a,b) \tag{5.2.2.2}$$

Es folgt, dass die in Abschnitt 5.1 definierten epistemischen Assertionstests keine generalisierten Annahmen berücksichtigen können. Der Ausdruck $\neg\mathbf{K}A$ führt zu der Einbeziehung eines Konzeptabschlusses für A und der Ausdruck $\neg\mathbf{K}P$ zu der Einbeziehung eines Rollenabschlusses für P, weil nur die Konstanten a_i, b_i berücksichtigt werden, für die $\neg A(a_i) \in CWA(A)$ bzw. $\neg P(a_i, b_i) \in CWA(P)$ gilt (vgl. Definition 4.2.2).[4] Konzeptbeschreibungen $\forall\mathbf{K}P.C$ beziehen ebenfalls einen Rollenabschluss ein, da für jede Konstante a die folgende Beziehung gilt:

$$KB \models \forall\mathbf{K}P.C(a) \quad gdw. \quad KB \models \neg\mathbf{K}P(a, c_i) \text{ oder } KB \models C(c_i) \text{ für alle } c_i \text{ aus } KB$$

Die Bedingung, dass alle bekannten Rollenfüller der Konstanten a bzgl. P Instanz des Konzeptes C sind, ist genau dann erfüllt, wenn für alle Konstanten c_i, die in KB vorkommen, nicht nachgewiesen werden kann, dass sie ein Rollenfüller von a bzgl. P sind oder bewiesen werden kann, dass sie Instanz des Konzeptes C sind.

Es besteht allerdings ein großer Unterschied zwischen der Verwendung negierter epistemischer Operatoren und der Erweiterung einer Wissensbasis um negierte atomare Formeln: Konzept- und Rollenabschlüsse werden einer Wissensbasis bei der Ausführung eines epistemischen Assertionstests nicht hinzugefügt, so dass durch die Verwendung epistemischer Operatoren keine Inkonsistenzen entstehen können. Beispielsweise sind die epistemischen Assertionstests

[2] Es gelten die Beziehungen $(\mathbf{K}(C \sqcap D))^{\mathcal{I}} = (\mathbf{K}C)^{\mathcal{I}} \cap (\mathbf{K}D)^{\mathcal{I}}$ und $KB \models \mathbf{K}C(a)$ $gdw.$ $KB \models C(a)$

[3] Negierte epistemische Operatoren vor komplexen Konzepten, $\neg\mathbf{K}C$, müssen nicht berücksichtigt werden, wenn sich das um epistemische Operatoren zu ergänzende Konzept C in Negationsnormalform befindet.

[4] Die Ausdrücke $\mathbf{K}A(a)$ und $\mathbf{K}P(a)$ sind komplementär zu den Ausdrücken $\neg\mathbf{K}A(a)$ und $\neg\mathbf{K}P(a)$. Es werden nur die Konstanten a_i, b_i berücksichtigt, für die $\neg A(a_i) \notin CWA(A)$ bzw. $\neg P(a_i, b_i) \notin CWA(P)$ gilt.

$$\mathcal{U} \models \neg\mathbf{K}Grad(john) \qquad \mathcal{U} \models \neg\mathbf{K}Professor(john)$$

erfolgreich. Die in $\mathcal{U}$ spezifizierte Konzeptassertion $(Grad \sqcup Professor)(john)$ führt nicht zu einer inkonsistenten Wissensbasis, da lediglich ausgesagt wird, dass die Assertionen $Grad(john)$ und $Professor(john)$ nicht in allen Modellen der Wissensbasis erfüllt sind.

Wird der **K**-Operator in einer $\mathcal{ALC}$-Anfrageassertion jeder Rolle innerhalb einer Werterestriktion und jeder negierten atomaren Assertion hinzugefügt und für alle weiteren Sprachausdrücke nicht berücksichtigt, dann wird eine um das DCA erweiterte CWA einbezogen, ohne zu Inkonsistenzen zu führen. Beispielsweise bezieht der folgende epistemische Assertionstest die CWA und das DCA ein:

$$\mathcal{U} \models \forall\mathbf{K}teaches.\neg\mathbf{K}Professor(john)$$

Für alle bekannten Rollenfüller von $john$ bzgl. $teaches$ kann nicht bewiesen werden, dass sie Professoren sind. Die Erweiterung von $\mathcal{U}$ um die CWA ist inkonsistent, so dass trivialerweise $\mathcal{U} \models_{cwa^\diamond} \forall teaches.\neg Professor(john)$ gilt. Der letztgenannte Assertionstest ist allerdings auch erfolgreich, wenn alle indefiniten Konzeptassertionen aus $\mathcal{U}$, die zu dieser Inkonsistenz führen, entfernt werden. Für die Ausführung des folgenden Assertionstests müssen Skolem-Konstanten berücksichtigen werden:

$$\mathcal{U} \models \forall\mathbf{K}enrolled.\neg\mathbf{K}Grad(ee282)$$

Alle der Wissensbasis $\mathcal{U}$ bekannten Rollenfüller von $ee282$ bzgl. $enrolled$ sind nicht in allen Modellen von $\mathcal{U}$ Absolventen. Der epistemische Assertionstest ist erfolgreich, obwohl aus $\mathcal{U}$ ersichtlich ist, dass in dem Kurs $ee282$ zumindest ein Student eingetragen ist, der ein Absolvent ist, da bei seiner Ausführung die Skolem-Konstante, die aus $\exists enrolled.Grad(ee282) \in \mathcal{U}$ hervorgeht, ignoriert wird.

Kapitel 6

Negation-as-failure in der new Racer Query Language

RacerPro (Renamed ABox and Concept Expression Reasoner Professional) ist ein Wissensrepräsentationssystem, das einen hoch optimierten Tableau-Kalkül für die beschreibungslogische Sprache $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$ implementiert. Die Erweiterung von $\mathcal{ALC}$ um qualifizierte Anzahlrestriktionen $(\geq n\, R.C)$ und $(\leq n\, R.C)$ (bezeichnet durch $\mathcal{Q}$), Rolleninklusionen $R_1 \sqsubseteq R_2$ $(\mathcal{H})$, inverse Rollen R^- $(\mathcal{I})$ und transitive Rollen $({}_{\mathcal{R}^+})$ wird in [Horrocks et al., 2000] erläutert. Die zusätzliche Erweiterung um Ausdrücke, die unter Beibehaltung der Entscheidbarkeit der Sprache einen Bezug zu konkreten Domänen (z.B. ganze Zahlen) ermöglichen, wird in [Haarslev et al., 2001] vorgestellt.

Die hervorstechendste Eigenschaft von RacerPro ist die Unterstützung von ABox-Inferenzdiensten in $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$. Die Benutzer von RacerPro haben dennoch eine Erweiterung der Anfragesprache vorgeschlagen. Ihre Anliegen haben zu der Implementation der *new Racer Query Language (nRQL)* [Racer Systems, 2007] [Wessel und Möller, 2005] geführt.

Die für diese Arbeit relevanteste Erweiterung von RacerPro-Anfragen durch nRQL ist die Einführung des Negation-as-failure-Operators „`neg`“. Es wird gezeigt, dass insbesondere durch diesen Operator die Formulierung spezieller epistemischer Assertionstests ermöglicht wird.

Testanfragen an den RacerPro-Server der Version 1.9.2 beta wurden mit der graphischen Benutzungsoberfläche *RacerPorter* durchgeführt. Da in dieser Untersuchung Sprachen, die $\mathcal{ALC}$ um qualifizierte Anzahlrestriktionen, Rollendeklarationen oder konkrete Domänen erweitern, nicht behandelt werden, beziehen sich alle Ausführungen in diesem Kapitel auf $\mathcal{ALCN}$, einer Teilsprache von $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$. Außerdem werden viele Aspekte der new Racer Query Language in diesem Kapitel nicht berücksichtigt. Eine einführende und ausführliche Darstellung aller dieser Aspekte ist in [Racer Systems, 2007] zu finden.

6.1 Formulierung von nRQL-Anfragen

Für Konzeptbeschreibungen, die in einer nRQL-Anfrage spezifiziert werden sollen, wird die RacerPro-Syntax [Racer Systems, 2007, Kapitel 3] verwendet. Sie ergibt sich für $\mathcal{ALCN}$-Konzeptkonstruktoren wie folgt aus der bisher verwendeten abstrakten Syntax:

	abstrakte Syntax	RacerPro-Syntax
logische Konstante „top"	$\top$	`*top*`
logische Konstante „bottom"	$\bot$	`*bottom*`
negierte Konzepte	$\neg C$	(`not` C)
Konzeptkonjunktionen	$C_1 \sqcap ... \sqcap C_p$	(`and` $C_1 ... C_p$)
Konzeptdisjunktionen	$C_1 \sqcup ... \sqcup C_p$	(`or` $C_1 ... C_p$)
Werterestriktionen	$\forall R.C$	(`all` R C)
Existenzrestriktionen	$\exists R.C$	(`some` R C)
mindestens-Anzahlrestriktionen	$(\geq n\, R)$	(`at-least` n R)
höchstens-Anzahlrestriktionen	$(\leq n\, R)$	(`at-most` n R)

Zusätzlich sind in nRQL negierte atomare Rollen $\neg R$ mit der RacerPro-Syntax (`not` R) erlaubt.

Die Formulierung von nRQL-Anfragen erfolgt mit der Funktion `retrieve`:

(`retrieve` (*query-head*) (*query-body*))

Im Anfragekörper (*query-body*) werden die Anfragebedingungen spezifiziert und im Anfragekopf (*query-head*) das Format der Ausgabe. Sowohl (*query-head*) als auch (*query-body*) enthalten Objekte. Ein Objekt ist entweder eine der in der verwendeten ABox vorkommenden Konstanten $c_1, .., c_m$, eine nicht-injektive Variable `?x,?y,?z` oder eine injektive Variable `$?x,$?y,$?z`. Unterschiedliche injektive Variablen können im Gegensatz zu nicht-injektiven Variablen nicht mit derselben Konstanten gebunden werden.

Es gilt die Bedingung, dass Objekte, die im Anfragekopf spezifiziert werden, im Anfragekörper vorkommen müssen. Eine Anfrage, die diese Bedingung erfüllt, ist zum Beispiel:

```
? (retrieve (?x) (?x ?y hasChild))
```

Wenn diese Anfrage an die assertionale Wissensbasis $KB = \{hasChild(salvatore,francesca), hasChild(francesca, pietro)\}$ gestellt wird, antwortet RacerPorter mit

```
> (((?x salvatore)) ((?x francesca))).
```

Wird dieselbe Anfrage mit dem Anfragekopf `(?x ?y)` formuliert, ergibt sich die Antwort

```
> (((?x salvatore) (?y francesca)) ((?x francesca) (?y pietro))).
```

Antworten werden in nRQL im Gegensatz zu der RacerPro-Retrieval-Funktion `concept-instances` nicht mit der Menge der Individuen angegeben, für die die entsprechende logische Folgerung gilt, sondern mit bindenden Listen. In den innersten Klammern werden die Antwortelemente in Form von Variablen-Werte-Paaren angegeben, die darauf folgenden Klammern grenzen die Tupel der Antwort ein, und die äußerste Klammer umschließt die gesamte Antwort. Ist die Antwort `NIL`, gibt es kein Tupel, für das die Bedingungen der Anfrage erfüllt sind.

In der Web-Ontology-Query-Language (OWL-QL), dem Nachfolger der Anfragesprache DQL, werden Variablen, die gebunden werden müssen (*must-bind-variables*) von Variablen, die nicht gebunden werden dürfen (*don't bind variables*) unterschieden [Glimm und Horrocks, 2004]. Für *don't-bind-variables* wird nur die Existenz eines geeigneten Rollenfüllers gefordert, und eine Bindung wird nicht durchgeführt. In nRQL wird eine Unterscheidung dieser beiden Variablen-Typen nicht vorgenommen. Alle Variablen sind *must-bind-variables* und werden ausschließlich mit den in der verwendeten Wissensbasis vorkommenden Konstanten gebunden. Wird die oben aufgeführte Wissensbasis um $\{\exists hasChild.Female(pietro)\}$ erweitert, ergeben sich deswegen bei Wiederholung der zuerst gestellten Anfrage keine weiteren Antworttupel:

```
? (retrieve (?x) (?x ?y hasChild))
> (((?x salvatore)) ((?x francesca)))
```

Allerdings können *don't-bind-variables* durch Konzeptbeschreibungen mit erzeugenden Konstruktoren in einer Anfrage berücksichtigt werden:

```
? (retrieve (?x) (?x (some hasChild *top*)))
> (((?x salvatore)) ((?x francesca)) ((?x pietro)))
```

Assertionstests in nRQL Die bisher vorgestellten nRQL-Anfragen mit einer Variablen im Anfragekopf entsprechen der Ausführung eines (erweiterten) Retrieval Services: Es wird die Menge der Konstanten a berücksichtigt, für die $KB \models R(a,b)$ bzw. $KB \models C(a)$ gilt. Durch Angabe eines leeren Anfragekopfes und der Verwendung von Konstanten im Anfragekörper sind Assertionstests (spezielle ja/nein-Anfragen) möglich. Es können Instance-Checks, Relationentests, aber auch negierte Relationentests formuliert werden:

```
? (retrieve () (francesca female))
? (retrieve () (salvatore francesca hasChild))
? (retrieve () (salvatore giovanni (not hasChild)))
```

Für die Wissensbasis

$$\begin{aligned} KB = \quad & \{\forall hasChild.Female(salvatore), hasChild(salvatore,francesca), \\ & \neg Female(giovanni)\} \end{aligned}$$

sind alle drei Assertionstests erfolgreich. Für jeden dieser Tests erfolgt die Ausgabe

```
> t.
```

6.2 Epistemische Anfragen durch komplexe Anfragekörper

Die bisher vorgestellten Anfragekörper enthalten Konzeptbeschreibungen C, atomare Rollen R oder negierte atomare Rollen $\neg R$ und werden in nRQL *Anfrageatome* genannt. Im Gegensatz zu der vollen Ausdrucksstärke von nRQL [Racer Systems, 2007] werden keine weiteren Anfrageatome einbezogen. *Komplexe Anfragekörper* setzen sich in nRQL aus Anfrageatomen zusammen: Sind $b_1, ..., b_n$ Anfragekörper, dann sind ebenfalls

$$\texttt{and}\ (b_1 \ldots b_n), \quad \texttt{union}\ (b_1 \ldots b_n) \ \text{ und } \ \texttt{neg}\ (b_i)$$

Anfragekörper. Weitere Anfragekörper ergeben sich mit `project-to` durch die Projektion auf Objekte, die für die Beantwortung einer Anfrage berücksichtigt werden sollen.

6.2.1 Anfragen mit `and` und `union`

Anfragekörper, die ausschließlich mit dem `and`-Operator aus Anfrageatomen zusammengesetzt sind, führen zu der Formulierung *konjunktiver Anfragen* [Abiteboul et al., 1995]. Viele konjunktive Anfragen können in korrekten und vollständigen beschreibungslogischen Systemen wie RacerPro oder FaCT nicht direkt spezifiziert werden. Zum Beispiel ist es in diesen Systemen nicht ohne Weiteres möglich, alle Paare von Individuen mit einem gemeinsamen Elternteil zu finden [Haarslev et al., 2004b]. Unter Verwendung des `and`-Operators können entsprechende Paare von Individuen ermittelt werden: Für die assertionale Wissensbasis $KB = \{hasParent(susy, john), hasParent(charles, john)\}$ ergibt sich zum Beispiel

```
? (retrieve ($?x $?y) (and ($?x $?z hasParent) ($?y $?z hasParent)))
> ((($?x susy) ($?y charles)) (($?x charles) ($?y susy))).
```

Da in diesem Fall kein Interesse daran besteht, dass unterschiedliche Variablen mit denselben Konstanten gebunden werden, sind alle Variablen injektiv. Die Schnittmenge aller dreistelligen Tupel, die sich aus den Anfragekörper erfüllenden Bindungen der Variablen `$?x`, `$?y` und `$?z` mit in KB vorkommenden Konstanten ergibt, wird anschließend auf die ersten beiden Komponenten projiziert. Die Verwendung von `union` in Anfragen erfolgt entsprechend durch die Bildung der Vereinigungsmenge aller Tupel, die den jeweiligen Anfragekörper erfüllen.

nRQL-Anfragen mit `and` und `union` können zu der Berücksichtigung spezieller epistemischer Assertionstests führen. In Abschnitt 5.2.2 wurden für die assertionale Wissensbasis $\mathcal{U}$ (Abbildung 5.1) nicht-epistemische Assertionstests mit epistemischen Assertionstests verglichen. Viele der dort aufgeführten epistemischen Assertionstests können auch mit nRQL durchgeführt werden. Durch die ausschließliche Verwendung von *must-bind-variables* entspricht beispielsweise die Prüfung, ob $\mathcal{U} \models \exists \mathbf{K} enrolled.Grad(ee282)$ gilt, der konjunktiven Anfrage

```
? (retrieve () (and (ee282 ?y enrolled) (?y grad))),
```

und es ergibt sich die Antwort `NIL`.

Ebenfalls kann die Überprüfung der Gültigkeit sowohl von $\mathcal{U} \models (Grad \sqcup Professor)(john)$ als auch von $\mathcal{U} \models \mathbf{K}Grad \sqcup \mathbf{K}Professor(john)$ mit nRQL durchgeführt werden:

```
? (retrieve () (john (or grad professor)))
? (retrieve () (union (john grad) (john professor)))
```

Wie in Abschnitt 5.2.2 dargestellt, ist der erste Assertionstest erfolgreich und der zweite nicht. Es ergeben sich die Antworten `t` und `NIL`.

6.2.2 Negation-as-failure-Semantik durch den `neg`-Operator

Im Folgenden wird untersucht, inwiefern der `neg`-Operator Konzept- und Rollenabschlüsse (vgl. Definition 4.2.2) ermöglicht. Es besteht Interesse daran, epistemische Anfragen in nRQL zu formulieren, die eine um das DCA erweiterte CWA einbeziehen (siehe Abschnitt 5.2.2).

Die Verwendung des `neg`-Operators führt in nRQL zu der Berücksichtigung einer Negation-as-failure-Semantik. Die Autoren von [Kaplunova et al., 2007] konnten keine weiteren Anfragesprachen für ausdrucksstarke Beschreibungslogiken finden, die diese Form der Negation bereitstellen. Anfragekörper, die sich durch das Hinzufügen dieses Operators vor Anfrageatomen ergeben, können durch `and` und `union` mit anderen Anfragekörpern zu komplexen Anfragekörpern zusammengesetzt werden. Aus der Struktur komplexer Anfragekörper in nRQL ist ersichtlich, dass der `neg`-Operator nicht innerhalb von Anfrageatomen verwendet werden kann.

6.2.2.1 Atomare Konzepte

Sollen alle Personen ausgegeben werden, für die bzgl. der Universitätswissensbasis $\mathcal{U}$ (Abbildung 5.1) nachgewiesen werden kann, dass sie Absolventen sind, kann $\mathcal{U}$ um $\{Person(mary), Person(john), Person(peter), Person(susan)\}$ erweitert und anschließend an die resultierende Wissensbasis $\mathcal{U}_P$ die folgende nRQL-Anfrage gestellt werden:

```
? (retrieve (?x) (and (?x person) (?x grad)))
> (((?x mary)))
```

Die zusätzliche Verwendung des `neg`-Operators ermöglicht die Ausgabe aller Personen, für die *nicht* nachgewiesen werden kann, dass sie Absolventen sind:

```
? (retrieve (?x) (and (?x person) (neg (?x grad))))
> (((?x john)) ((?x peter)) ((?x susan)))
```

Die Anfrage berücksichtigt einen Konzeptabschluss für *Grad* und ist äquivalent zu dem epistemischen Retrieval Service $\{a \mid \mathcal{U}_P \models (Person \sqcap \neg\mathbf{K}Grad)(a)\}$. Die Einbeziehung einer solchen Negation-as-failure-Semantik führt zu der Ausgabe der Komplementärmenge bzgl. aller Personen der zuvor gestellten Anfrage. Das ist möglich, weil in nRQL die *active-domain-semantics* gilt (die Konstanten, die in einer Wissensbasis vorkommen, sind alle Konstanten des betrachteten Weltausschnitts (vgl. Domänenabschluss, Abschnitt 4.2.3.1)).

In nRQL können ebenfalls die bzgl. aller Personen komplementären Anfragen

```
? (retrieve (?x) (and (?x person) (?x (not grad))))
? (retrieve (?x) (and (?x person) (neg (?x (not grad)))))
```

formuliert werden. Diese Anfragen entsprechen der Ausführung der epistemischen Retrieval Services $\{a \mid \mathcal{U}_P \models (Person \sqcap \mathbf{K}\neg Grad)(a)\}$ und $\{a \mid \mathcal{U}_P \models (Person \sqcap \neg\mathbf{K}\neg Grad)(a)\}$. Es ergeben sich die Antworten `(((?x peter)))` und `(((?x mary)) ((?x john)) ((?x susan)))`. Durch den Anfragekörper `(neg (?x (not grad)))` wird eine entgegengesetzte CWA berücksichtigt: Es werden die Individuen ausgegeben, für die nicht nachgewiesen werden kann, dass sie keine Absolventen sind.

6.2.2.2 Atomare Rollen

Durch den Negation-as-failure-Operator `neg` können in nRQL ebenfalls die Individuen ausgegeben werden, für die kein bekannter Rollenfüller bzgl. einer Rolle R existiert. Das entspricht der Ausführung einer epistemischen Anfrage $\{a \mid (\forall b)\,[KB \models \neg\mathbf{K}R(a,b)]\}$ an eine Wissensbasis KB unter Berücksichtigung aller in KB vorkommenden Konstanten b. Die Bildung der Komplementärmenge und anschließende Projektion auf die erste Komponente führt allerdings nicht zu dem gewünschten Ergebnis. Zum Beispiel gilt für die folgende Anfrage an $\mathcal{U}_P$:

```
? (retrieve (?x) (and (?x person) (neg (?x ?y teaches))))
> (((?x mary)) ((?x peter)) ((?x susan)) ((?x john)))
```

Es ist offensichtlich, dass es für die Konstante *john* bekannte Rollenfüller bzgl. der Rolle *teaches* in $\mathcal{U}_P$ gibt. `((?x john))` ist dennoch Bestandteil der Ausgabe, weil für den Kurs *ee*282 nicht nachgewiesen werden kann, dass er von John unterrichtet wird, so dass nach Bildung der Komplementärmenge weiterhin Tupel mit *john* als erster Komponente existieren.

Um das gewünschte Ergebnis zu erzielen, gibt es in nRQL die Möglichkeit, zuerst auf die erste Komponente zu projizieren und anschließend das Komplement zu bilden. Das erfolgt mit dem Operator `project-to`:[1]

```
? (retrieve (?x) (and (?x person) (neg (project-to (?x) (?x ?y teaches)))))
> (((?x mary)) ((?x peter)) ((?x susan)))
```

Entsprechend können mit `(and (?x person) (neg (project-to (?x) (?x ?y (not teaches)))))` die Personen ermittelt werden, für die sich nicht beweisen läßt, dass sie keinen bekannten Rollenfüller haben. Bezogen auf $\mathcal{U}_P$ sind das alle Personen.

[1] An Stelle dieses Operators kann auch `has-known-successor` verwendet werden [Racer Systems, 2007].

6.2.2.3 Werterestriktionen

In Abschnitt 6.2.2.2 wurden epistemische Anfragen nach Individuen, für die kein bekannter Rollenfüller bzgl. einer Rolle R existiert, dargestellt. Soll dieser Rollenfüller zusätzlich bestimmte Bedingungen erfüllen, ergeben sich epistemische Anfragen mit Werterestriktionen, die einen Rollenabschluss berücksichtigen. Beispielsweise werden durch die Anfrage

```
? (retrieve (?x) (and (?x course)
                      (neg (project-to (?x) (and (?x ?y enrolled)
                                                 (?y grad))))))
```

alle Kurse ermittelt, für die nicht bewiesen werden kann, dass in ihnen der Wissensbasis bekannte Absolventen eingeschrieben sind. Diese Anfrage entspricht dem epistemischen Retrieval Service $\{a \mid \mathcal{U} \models \neg\mathbf{K}(\exists\mathbf{K}enrolled.Grad)(a)\}$. Aufgrund der Bedingung

$$KB \models \forall\mathbf{K}R.C(a) \quad gdw. \quad KB \models \neg\mathbf{K}(\exists\mathbf{K}R.\neg\mathbf{K}C)(a) \tag{6.2.2.1}$$

entspricht die Anfrage aber ebenfalls dem Retrieval Service $\{a \mid \mathcal{U} \models \forall\mathbf{K}enrolled.\neg\mathbf{K}Grad(a)\}$. RacerPorter antwortet mit `(((?x cs324)) ((?x ee282)))`. Der Kurs $ee282$ ist in der Antwort enthalten, weil der Rollenfüller der in $\mathcal{U}$ enthaltenen Konzeptassertion $\exists enrolled.Grad(ee282)$ unvollständig spezifiziert ist und deswegen nicht berücksichtigt wird.

Sollen alle Kurse ermittelt werden, in denen nur der Wissensbasis bekannte Individuen eingetragen sind, die nachweislich Absolventen sind, $\{a \mid \mathcal{U} \models \forall\mathbf{K}enrolled.Grad(a)\}$, kann aufgrund von (6.2.2.1) in nRQL die komplexe Anfrage

```
? (retrieve (?x) (and (?x course) (neg (project-to (?x)
                                    (and (?x ?y enrolled) (neg (?y grad)))))))
```

gestellt werden, und es ergibt sich wie erwartet die Antwort `NIL`.

In nRQL können auch epistemische Anfragen formuliert werden, die Anfragen mit Werterestriktionen entsprechen, deren Skopus aus komplexeren Konzepten besteht. Gegeben sei zum Beispiel $KB = \{R_1(a,b), R_2(b,c), A(c), R_1(c,a), R_2(a,a)\}$ und es bestehe Interesse an der Durchführung des epistemischen Retrieval Services $\{d \mid KB \models \forall\mathbf{K}R_1.\forall\mathbf{K}R_2.A(d)\}$. Aufgrund von (6.2.2.1) ergeben sich die Umformungen

$$\begin{array}{ll} KB \models \forall\mathbf{K}R_1.\forall\mathbf{K}R_2.A(d) & \text{gdw.} \\ KB \models \neg\mathbf{K}(\exists\mathbf{K}R_1.\neg\mathbf{K}(\forall\mathbf{K}R_2.A))(d) & \text{gdw.} \\ KB \models \neg\mathbf{K}(\exists\mathbf{K}R_1.\neg\mathbf{K}(\neg\mathbf{K}(\exists\mathbf{K}R_2.\neg\mathbf{K}A)))(d) & \text{gdw.} \\ KB \models \neg\mathbf{K}(\exists\mathbf{K}R_1.\exists\mathbf{K}R_2.\neg\mathbf{K}A)(d), & \end{array}$$

so dass die folgende nRQL-Anfrage formuliert werden kann:

```
? (retrieve (?x) (neg (project-to (?x) (and (?x ?y R1) (?y ?z R2)
                                            (neg (?z A))))))
```

Die Antwort `(((?x a)) ((?x b)))` ist unter Berücksichtigung der um das DCA erweiterten CWA korrekt.

6.2.2.4 Anzahlrestriktionen

Um häufig unerwünschte Fehlschläge von Assertionstests mit mindestens-Anzahlrestriktionen $(\geq n\,R)$ zu vermeiden, muss die UNA berücksichtigt werden. In RacerPro erfolgt das durch `(set-unique-name-assumption t)`. Gilt die UNA, ist bei Verwendung der Universitätswissensbasis $\mathcal{U}$ beispielsweise der folgende Assertionstest erfolgreich:

```
? (retrieve () (john (at-least 2 teaches)))
```

Dagegen können Assertionstests mit höchstens-Anzahlrestriktionen $(\leq n\,R)$ aus den in Abschnitt 3.2.2 dargestellten Gründen nicht erfolgreich sein. Eine Lösung dieser Problematik ergibt sich durch die Überprüfung, ob das Komplement der betreffenden höchstens-Anzahlrestriktion nicht logisch aus der verwendeten Wissensbasis folgt [Garcia und Gil, 2006]. Sollen zum Beispiel alle Kurse aus $\mathcal{U}$ ausgegeben werden, in denen höchstens zwei Individuen eingetragen sind, kann die Anfrage

```
? (retrieve (?x) (and (?x course) (neg (?x (at-least 3 enrolled)))))
```

gestellt werden, die dem epistemischen Retrieval Service $\{a \mid \mathcal{U} \models (Course \sqcap \neg\mathbf{K}(\geq 3\,R))(a)\}$ entspricht. Die Antwort `(((?x cs221)) ((?x cs324)) ((?x ee282))` berücksichtigt eine um das DCA und die UNA erweiterte CWA, ohne zu Inkonsistenzen zu führen.

Für die Anfrage

```
? (retrieve (?x) (and (?x course) (neg (?x (at-least 2 enrolled)))))
```

ergibt sich entsprechend die Antwort `NIL`. Der Kurs $ee282$ hat mindestens zwei eingetragene Individuen, weil die Skolem-Konstante, die durch die in $\mathcal{U}$ spezifizierte Konzeptassertion $\exists enrolled.Grad(ee282)$ eingeführt wird, aufgrund von $enrolled(ee282, peter) \in \mathcal{U}$ und $\neg Grad(peter) \in \mathcal{U}$ nicht mit $peter$ identifiziert werden kann. Wird diese Konzeptassertion entfernt, führt die wiederholte Ausführung der Anfrage zu der Antwort `(((?x ee282)))`.

Höchstens-Anzahlrestriktion können in nRQL-Anfragen auch im Skopus einer Rollenrestriktion formuliert werden. Die epistemische Anfrage $\{d \mid KB \models \exists\mathbf{K}R_1.\neg\mathbf{K}(\geq 3\,R_2)(d)\}$ an die abstrakte Wissensbasis $KB = \{R_1(a,b), R_1(c,a), (\geq 3\,R_2)(a)\}$ zum Beispiel kann (unter Einbeziehung der UNA) mit nRQL formuliert und erwartungsgemäß beantwortet werden:

```
? (retrieve (?x) (and (?x ?y R1) (neg (?y (at-least 3 R2)))))
> (((?x a))).
```

Kapitel 7

Konstruktion einer Form der Closed-World-Assumption in ausgewählten Sprachen

In diesem Kapitel werden für ausgewählte beschreibungslogische Sprachen Verfahren formalisiert, die entscheiden, ob eine Assertion unter Berücksichtigung einer Form der Closed-World-Assumption aus einer Wissensbasis folgt oder nicht.

In Abschnitt 7.1 werden in Anlehnung an die in Abschnitt 4.2.4 dargestellte rekursive Anfragebearbeitung Verfahren für $\mathcal{ALCN}$-Anfragen an atomare beschreibungslogische Wissensbasen und $\mathcal{AL}_0$-Wissensbasen vorgestellt, die die CWA nach [Reiter, 1978] unter zusätzlicher Berücksichtigung des DCA und der UNA einbeziehen. Für diese Sprachen gibt es demnach eine vollständige Lösung der in Abschnitt 3.2 dargestellten Problematik.

Für alle weiteren in dieser Arbeit untersuchten Sprachen wird die GCWA verwendet. In Abschnitt 4.5 wurde gezeigt, dass für die Berücksichtigung der Definition dieser Annahme neben Konzeptdisjunktionen insbesondere implizite Klauseln aufgrund unvollständig spezifizierter Rollenfüller *new* einbezogen werden müssen. Nach [Minker, 1982] resultieren diese Klauseln unter Berücksichtigung des Domänenabschlusses aus der Identifikation einer Skolem-Konstanten *new* mit allen in einer Wissensbasis vorkommenden Konstanten. In Abschnitt 7.2 wird gezeigt, dass diese Auslegung von Skolem-Konstanten zu nicht intendierten Inferenzen und Inkonsistenzen führen kann. Um diese Problematik zu umgehen, wird dort die Auslegung einer Skolem-Konstante unter zusätzlicher Berücksichtigung einer neuen Konstante η vorgestellt. Die Einbeziehung aller dieser neuen Konstanten führt zu der Definition eines erweiterten Domänenabschlusses. Zuletzt wird in Abschnitt 7.2 dargestellt, dass die Auslegung spezieller Skolem-Konstanten unter der GCWA so eingeschränkt werden kann, dass die Konzeptassertionen, durch die diese Skolem-Konstanten berücksichtigt werden müssen, redundant sind.

Anschließend wird in Abschnitt 7.3 ein polynomielles Verfahren vorgestellt, dass in den Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ unter Einbeziehung vorvervollständigter Tableau-Zweige entscheidet, ob eine Assertion unter der GCWA aus einer Wissensbasis folgt oder nicht. Da in diesen Sprachen bereitgestellte erzeugende Rollenrestriktionen $\exists R.\top$ und $(\geq n\, R)$ unqualifiziert sind, erfolgt die Formalisierung dieses Verfahrens durch die unterschiedliche Behandlung möglicher $\mathcal{AL}$ bzw. $\mathcal{ALN}$-Anfragen ohne die direkte Berücksichtigung von impliziten Klauseln.

Schließlich wird in Abschnitt 7.4 die Möglichkeit aufgezeigt, eine lokale GCWA für die Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ zu berücksichtigen. Nach Bestimmung aller einzubeziehenden neuen Konstanten werden alle Assertionen einer $\mathcal{ALCN}$-Wissensbasis sowie eine $\mathcal{ALCN}$-Anfrageassertion in eine prädikatenlogische Klauselform überführt. Für jedes negierte Literal der Klauselform der Anfrage kann dabei lokal entschieden werden, ob die GCWA einbezogen werden soll oder nicht.

7.1 Sprachen mit einfach strukturierter Wissensbasis

Einfach strukturierte Wissensbasen sind Wissensbasen, die unter der Erweiterung um die CWA, das DCA und die UNA konsistent bleiben. Solcherart erweiterte Wissensbasen wurden in Abschnitt 4.2.4 als lebendig bezeichnet, und es wurde gezeigt, dass die dort vorgestellte rekursive Reduktion prädikatenlogischer Anfragen auf atomare Anfragen für diese Wissensbasen korrekt ist.

Im Folgenden werden Beschreibungslogiken mit lebendigen Wissensbasen behandelt. Dazu wird zunächst gezeigt, dass $\mathcal{ALCN}$-Anfrageassertionen bei Verwendung dieser Wissensbasen korrekt auf atomare Anfrageassertionen reduziert werden können (Abschnitt 7.1.1). Anschließend werden zwei Formen einfach strukturierter beschreibungslogischer Wissensbasen besprochen: atomare Wissensbasen und $\mathcal{AL}_0$-Wissensbasen. Durch geeignete Rekursionsabschlüsse für atomare Assertionen ergeben sich zwei Top-down-Algorithmen, die den Assertionstest für die Anfragesprache $\mathcal{ALCN}$ um die CWA nach [Reiter, 1978], das DCA und die UNA erweitern (Abschnitte 7.1.2 und 7.1.3). Für atomare Wissensbasen und $\mathcal{AL}_0$-Wissensbasen existiert demnach eine vollständige Lösung der in dieser Untersuchung geschilderten Problematik, so dass keine der in Abschnitt 3.2 dargestellten unerwünschten Fehlschläge von Beweisen auftreten können.

7.1.1 Reduktion von $\mathcal{ALCN}$-Anfrageassertionen

Reduktionen von $\mathcal{ALCN}$-Anfrageassertionen auf Teilformeln ergeben sich aus den in Abschnitt 4.2.4 dargestellten Reduktionen und werden in diesem Abschnitt ausschließlich mit der Folgerungsrelation $\models_{CWA}$ dargestellt. An Stelle einer prädikatenlogischen Formel φ werden beschreibungslogische Konzeptassertionen $C(a)$ verwendet. Rollenassertionen sind in $\mathcal{ALCN}$ atomar und werden nur für Rollenrestriktionen innerhalb von $C(a)$ berücksichtigt.

Für Konzeptbeschreibungen mit doppelten Negationen, Konzeptkonjunktionen und Konzeptdisjunktionen gelten bei Verwendung einer einfach strukturierten beschreibungslogischen Wissensbasis KB ähnliche Bedingungen wie im prädikatenlogischen Fall:

$$\begin{array}{lll} KB \models_{CWA} \neg\neg C(a) & gdw. & KB \models_{CWA} C(a) \\ KB \models_{CWA} C_1 \sqcap ... \sqcap C_n(a) & gdw. & KB \models_{CWA} C_i(a) \text{ für alle } i \geq 1 \\ KB \models_{CWA} C_1 \sqcup ... \sqcup C_n(a) & gdw. & KB \models_{CWA} C_i(a) \text{ für ein } i \geq 1 \end{array}$$

Die beiden letztgenannten Reduktionsregeln berücksichtigen allerdings auch verallgemeinerte Konzeptkonjunktionen und verallgemeinerte Konzeptdisjunktionen. Die Ausnahmefälle für $n = 0$ verhalten sich tautologisch bzw. kontradiktorisch.

Die Interpretationen von qualifizierten Werte- und Existenzrestriktionen sind:

$$\begin{array}{lll} (\forall R.C)^{\mathcal{I}} & = & \{a \in \triangle^{\mathcal{I}} \mid (\forall x)\,[(a,x) \in R^{\mathcal{I}} \rightarrow x \in C^{\mathcal{I}}]\} \\ (\exists R.C)^{\mathcal{I}} & = & \{a \in \triangle^{\mathcal{I}} \mid (\exists x)\,[(a,x) \in R^{\mathcal{I}} \wedge x \in C^{\mathcal{I}}]\} \end{array}$$

Werden an Stelle der Quantifikation alle in KB vorkommenden Konstanten c_i, $i = 1, ..., m$ einbezogen, ergeben sich aus den Reduktionen von Formeln mit Quantoren aus Abschnitt 4.2.4 die folgenden Reduktionen auf aussagenlogische Anfragen:

$$\begin{array}{llllll} KB \models_{CWA} \forall R.C(a) & gdw. & KB \models_{CWA} & (R(a,c_1) & \rightarrow C(c_1)) & \wedge \\ & & & \vdots & \vdots & \wedge \\ & & & (R(a,c_m) & \rightarrow C(c_m)) & \end{array}$$

$$\begin{array}{llllll} KB \models_{CWA} \exists R.C(a) & gdw. & KB \models_{CWA} & (R(a,c_1) & \wedge\ C(c_1)) & \vee \\ & & & \vdots & \vdots & \vee \\ & & & (R(a,c_m) & \wedge\ C(c_m)) & \end{array}$$

Eine Werterestriktionsassertion $\forall R.C(a)$ folgt also genau dann unter der Relation $\models_{CWA}$ aus einer einfach strukturierten Wissensbasis KB, wenn für alle c_i mit $KB \models_{CWA} R(a, c_i)$ der Instance-Check $KB \models_{CWA} C(c_i)$ erfolgreich ist, und eine Existenzrestriktionsassertion $\exists R.C(a)$, wenn es einen Rollenfüller c_i von a bzgl. R gibt, für den der Instance-Check $KB \models_{CWA} C(c_i)$ erfolgreich ist.

Die folgenden Reduktionsregeln für Anzahlrestriktionen gelten, weil die Folgerungsrelation $\models_{CWA}$ auch die UNA berücksichtigt und weil in einfach strukturierten Wissensbasen keine unvollständig spezifizierten Rollenfüller berücksichtigt werden müssen.

$$\begin{array}{lll} KB \models_{CWA} (\geq n\, R)(a) & gdw. & \text{es gibt mindestens } n \text{ Rollenfüller } f_i \\ & & \text{mit } KB \models_{CWA} R(a, f_i) \\ KB \models_{CWA} (\leq n\, R)(a) & gdw. & \text{es gibt höchstens } n \text{ Rollenfüller } f_i \\ & & \text{mit } KB \models_{CWA} R(a, f_i) \end{array}$$

Für die Reduktion von $\mathcal{ALCN}$-Anfrageassertionen auf atomare Anfrageassertionen muss nur noch die Negation von Konzepten berücksichtigt werden. Die entsprechende Reduktionsregel für prädikatenlogische Anfragen gilt auch für $\mathcal{ALCN}$-Anfrageassertionen, weil garantiert ist, dass um die CWA, das DCA und die UNA erweiterte einfach strukturierte Wissensbasen konsistent und vollständig sind.

$$KB \models_{CWA} \neg C(a) \quad gdw. \quad KB \not\models_{CWA} C(a)$$

7.1.2 Formalisierung einer CWA in Sprachen mit atomarer Wissensbasis

Enthält eine bezüglich einer TBox $\mathcal{T}$ expandierte beschreibungslogische Wissensbasis $KB = (\{\}, \mathcal{A}_{\mathcal{T}})$ nur atomare Assertionen $A(a)$ und $R(a,b)$ und keine Assertionen $\bot(a)$, können Assertionstests $KB \models_{CWA} \alpha$ mit einer $\mathcal{ALCN}$-Anfrageassertion α korrekt auf atomare Assertionstests reduziert werden. Die in Abschnitt 7.1.1 vorgestellten Reduktionsregeln sind in Tabelle 7.1 zusammengefasst. Bei Auftreten einer doppelten Negation sollte die Reduktion (7.1.2.1) auf-

$KB \models_{CWA} \neg\neg C(a)$	$gdw.$	$KB \models_{CWA} C(a)$	(7.1.2.1)
$KB \models_{CWA} \neg C(a)$	$gdw.$	$KB \not\models_{CWA} C(a)$	(7.1.2.2)
$KB \models_{CWA} C_1 \sqcap ... \sqcap C_n(a)$	$gdw.$	$KB \models_{CWA} C_i(a)$ für alle $i \geq 1$	(7.1.2.3)
$KB \models_{CWA} C_1 \sqcup ... \sqcup C_n(a)$	$gdw.$	$KB \models_{CWA} C_i(a)$ für ein $i \geq 1$	(7.1.2.4)
$KB \models_{CWA} \forall R.C(a)$	$gdw.$	für alle c_i mit $KB \models_{CWA} R(a, c_i)$ gilt: $KB \models_{CWA} C(c_i)$	(7.1.2.5)
$KB \models_{CWA} \exists R.C(a)$	$gdw.$	es ein c_i mit $KB \models_{CWA} R(a, c_i)$ gibt, so dass gilt: $KB \models_{CWA} C(c_i)$	(7.1.2.6)
$KB \models_{CWA} (\geq n\, R)(a)$	$gdw.$	es mindestens n Rollenfüller f_i mit $KB \models_{CWA} R(a,f_i)$ gibt	(7.1.2.7)
$KB \models_{CWA} (\leq n\, R)(a)$	$gdw.$	es höchstens n Rollenfüller f_i mit $KB \models_{CWA} R(a,f_i)$ gibt	(7.1.2.8)

Tabelle 7.1: Rekursive Reduktion von $\mathcal{ALCN}$-Assertionstests auf atomare Assertionstests

grund des geringeren Aufwands der Reduktion (7.1.2.2) vorgezogen werden. Für die rekursiven Aufrufe (7.1.2.3)–(7.1.2.6) gilt, dass ein Konjunkt, ein Disjunkt oder ein Rollenfüller, für das bzw. für den die angegebenen Bedingungen erfüllt oder nicht erfüllt sind, genügen kann, um festzustellen, ob die entsprechende Folgerung gilt oder nicht. Wenn ein Instance-Check mit einer $\mathcal{ALCN}$-Konzeptassertion durch diese Regeln auf Assertionstests mit atomaren Konzept- und Rollenassertionen zurückgeführt wird oder ausschließlich ein (negierter) Relationentest ausgeführt werden soll, ergibt sich ein Rekursionsabschluss bzw. eine direkte Antwort durch die Überprüfung der Zugehörigkeit der entsprechenden Assertionen zu der betrachteten Wissensbasis:

$$\begin{array}{lll} KB \models_{CWA} A(a) & gdw. & A(a) \in KB \\ KB \models_{CWA} R(a,b) & gdw. & R(a,b) \in KB \\ KB \models_{CWA} \neg R(a,b) & gdw. & R(a,b) \notin KB \end{array}$$

Rekursionsabschlüsse für logische Konstanten ergeben sich direkt durch $KB \models_{CWA} \top(a)$ und $KB \not\models_{CWA} \bot(a)$. Die Konsistenz der Wissensbasis ist gegeben, weil für beliebige Konstanten a und b nicht sowohl $A(a)$ als auch $\neg A(a)$, nicht sowohl $R(a,b)$ als auch $\neg R(a,b)$ und nicht $\bot(a)$ in der Wissensbasis enthalten sind.

Bei der Ausführung dieses rekursiven Top-down-Algorithmus' kann die Wissensbasis als vollständig betrachtet werden. Der Algorithmus ist korrekt, weil die Konsistenz der Wissensbasis bei Erweiterung um die CWA, das DCA und die UNA aufgrund der Tatsache, dass in der Wissensbasis ausschließlich atomare Assertionen (und nicht z.B. Konzeptdisjunktionen) vorkommen, erhalten bleibt.

Beispiel 7.1.1 Gegeben sei die in der Einführung dargestellte Problematik von Links auf Internetseiten. Die ABox $\mathcal{A}$ dieses Szenarios ist atomar und enthält die Zusicherungen

$$\begin{array}{cc} Page(page1). \quad Page(page2). \quad Page(page3). \\ Page(page4). \quad Page(page5). \quad Page(page6). \\ isLinkedFrom(page2, page1). \quad isLinkedFrom(page2, page3). \\ isLinkedFrom(page5, page2). \quad isLinkedFrom(page1, page6). \end{array}$$

Internetseiten, auf die unter Einbeziehung der CWA, des DCA und der UNA nicht verwiesen wird, resultieren aus dem Retrieval Service $\{page_i \mid \mathcal{A} \models_{CWA} \neg\exists isLinkedFrom.Page(page_i)\}$. Im Gegensatz zu der Verwendung der logischen Folgerung ergibt sich durch das in diesem Abschnitt dargestellte Verfahren ein Beweis für die Seiten 3, 4 und 6. Der Retrieval Service kann wie erwünscht beantwortet werden, wenn nur die Internetseiten $page_i$ der Antwort hinzugefügt werden, für die nicht gezeigt werden kann, dass auf sie verwiesen wird (7.1.2.2):

$$\mathcal{A} \models_{CWA} \neg\exists isLinkedFrom.Page(page_i) \;\; gdw. \;\; \mathcal{A} \not\models_{CWA} \exists isLinkedFrom.Page(page_i)$$

Das sind die Seiten, für die kein entsprechender Rollenfüller existiert, für die die folgende Reduktion (7.1.2.6) also nicht gilt:

$$\begin{array}{lll} \mathcal{A} \models_{CWA} \exists isLinkedFrom.Page(page_i) & gdw. & \text{es ein } page_j \text{ mit} \\ & & \mathcal{A} \models_{CWA} isLinkedFrom(page_i, page_j) \\ & & \text{gibt, so dass gilt: } \mathcal{A} \models_{CWA} Page(page_j) \end{array}$$

Die Konstanten, für die es keinen in $\mathcal{A}$ enthaltenen Rollenfüller $page_j$ bzgl. $isLinkedFrom$ mit $Page(page_j)$ gibt, sind die Konstanten $page3$, $page4$ und $page6$.

7.1.3 Formalisierung einer CWA in $\mathcal{AL}_0/\mathcal{ALCN}$

Für die Sprache $\mathcal{AL}_0/\mathcal{ALCN}$ existiert eine dem vorangegangenen Abschnitt ähnliche algorithmische Lösung durch Verwendung der CWA nach [Reiter, 1978]. Bevor ein solches Verfahren sinnvoll angewendet werden kann, muss allerdings sichergestellt sein, dass die gegebene $\mathcal{AL}_0$-Wissensbasis konsistent ist. Das ist zum Beispiel in der $\mathcal{AL}_0$-Wissensbasis

$$\begin{aligned} KB = \ & \{(Vater \sqcap \forall hatKind.Weiblich)(john), hatKind(john, charles), \\ & \neg Weiblich(charles)\} \end{aligned}$$

nicht der Fall. Dadurch, dass in $\mathcal{AL}_0$-Wissensbasen weder Konzeptdisjunktionen noch Rollenrestriktionen, durch die Skolem-Konstanten berücksichtigt werden müssen, vorkommen, sind diese Wissensbasen äquivalent zu einer Horn-Wissensbasis. Nach Theorem 4.2.7 ergibt sich, dass eine konsistente $\mathcal{AL}_0$-Wissensbasis unter der CWA die Konsistenz erhält. Inkonsistenzen durch Verwendung des DCA (siehe Abschnitt 4.2.3.1) können nicht entstehen, weil durch das Abhandensein von Existenzrestriktionen und mindestens-Anzahlrestriktionen alle Konstanten einer Wissensbasis bei Beginn eines Beweises bekannt sind. Dadurch, dass höchstens-Anzahlrestriktionen nicht in $\mathcal{AL}_0$-Wissensbasen vorkommen, treten ebenfalls keine Inkonsistenzen auf, die durch die Verwendung der UNA entstehen können (siehe Abschnitt 3.1). Die Anwendung aller der in Tabelle 7.1 dargestellten Reduktionsregeln ist deswegen korrekt.

Rekursionsabschlüsse für atomare Konzeptassertionen können mit dem Tableau-Kalkül vollzogen werden, weil in diesem Fall die Folgerungsrelation $\models_{CWA}$ der logischen Folgerung $\models$ entspricht (Theorem 4.2.5):

$$KB \models_{CWA} A(a) \ \ gdw. \ \ KB \models A(a) \tag{7.1.3.1}$$

Relationentests und negierte Relationentests dagegen können wie in atomaren Wissensbasen durch Überprüfung der Zugehörigkeit entsprechender Rollenassertionen zu der betreffenden Wissensbasis ausgeführt werden, weil in $\mathcal{AL}_0$ alle Rollenassertionen explizit spezifiziert sind.

In [Brachman und Levesque, 2004] wird erläutert, dass Algorithmen wie der hier dargestellte ausschließlich mit der ursprünglichen Wissensbasis Inferenzen ziehen. Die Erweiterung dieser Wissensbasis um die CWA, das DCA und die UNA garantiert die Korrektheit des Verfahrens. Wird die Sprache $\mathcal{AL}_0$ um unqualifizierte Existenzrestriktionen $\exists R.\top$ erweitert, ist dieser Algorithmus nicht in jedem Fall korrekt. Besteht beispielsweise eine $\mathcal{AL}$-Wissensbasis aus einer Konzeptassertion $\exists R.\top(a)$, gilt $KB \models_{CWA} \exists R.\top(a)$. Aufgrund der Berücksichtigung des DCA gilt aber $KB \models_{CWA} \neg R(a, c_i)$ für alle in KB vorkommenden Konstanten c_i, $i = 1, ..., m$, so dass ebenfalls $KB \models_{CWA} \neg\exists R.\top(a)$ gilt und die erweiterte Wissensbasis widersprüchlich ist. Diese Inkonsistenz entsteht bei dem rekursiven Algorithmus aber erst gar nicht. Da in der Wissensbasis kein entsprechender Rollenfüller explizit vorkommt (wie in 7.1.2.6 gefordert), ist der Instance-Check $KB \models_{CWA} \exists R.\top(a)$ nicht erfolgreich. Der Algorithmus ist deswegen für $\mathcal{AL}$-Wissensbasen und alle Wissensbasen, die $\mathcal{AL}$ erweitern, möglicherweise nicht korrekt.

7.2 Auslegung von Skolem-Konstanten

In diesem Abschnitt wird die mögliche Identifikation von Skolem-Konstanten mit anderen Konstanten festgelegt, um für beschreibungslogische Wissensbasen, die nicht einfach strukturiert sind, eine möglicherweise intendierte Repräsentation von Weltausschnitten zu erhalten und eine Grundlage für die Konstruktion von Verfahren zu haben, die eine generalisierte Annahme einer geschlossenen Welt berücksichtigen sollen.

Nach [Reiter, 1984] handelt es sich bei Skolem-Konstanten um Nullwerte, deren Wert gegenwärtig nicht bekannt ist und die nicht notwendigerweise einer Konstanten aus einer bekannten endlichen Menge entsprechen. Um diese Auslegung berücksichtigen zu können, wird in Abschnitt 7.2.1 das Domänenabschlussaxiom um zusätzliche Konstanten erweitert. Es wird gezeigt, dass diese Erweiterung notwendig ist, um spezielle nicht intendierte Inferenzen sowie Inkonsistenzen zu vermeiden. Außerdem wird bewiesen, dass durch die Annahme eines solchen erweiterten Domänenabschlussaxioms Relationentests mit explizit spezifizierten Konstanten in den in dieser Arbeit untersuchten Sprachen durch eine einfache Überprüfung der Zugehörigkeit der entsprechenden Rollenassertion zu einer Wissensbasis ersetzt werden können.

Anschließend werden in Abschnitt 7.2.2 spezielle Skolem-Konstanten als redundant angenommen, um weitere intendierte Inferenzen ziehen zu können. Anhand der prädikatenlogischen Klauselform der Konzeptassertionen $\exists R.\top(a)$ und $(\geq n\, R)(a)$ wird aufgezeigt, dass diese Annahme durch die GCWA berücksichtigt wird.

7.2.1 Erweiterung des Domänenabschlusses

Dadurch, dass das DCA ausschließlich alle explizit in der Wissensbasis vorkommenden Konstanten $c_1, ..., c_m$ einbezieht, können Skolem-Konstanten new, die implizit in Konzeptassertionen $\exists R.C(a)$ oder $(\geq n\, R)(a)$ enthalten sind, nur mit diesen Konstanten identifiziert werden, so dass $new \doteq c_1 \vee ... \vee new \doteq c_m$. Aufgrund dieser Auslegung von Skolem-Konstanten wird die Menge der Modelle einer Wissensbasis eventuell so reduziert, dass ausschließlich nicht intendierte Modelle verbleiben und unerwünschte Inferenzen entstehen. Ist eine $\mathcal{AL}$-Wissensbasis KB_1 beispielsweise gegeben durch

$$KB_1 = \{Mensch(john), \exists hatKind.\top(john)\},$$

ergibt sich durch die Erweiterung von KB_1 um $DCA(KB_1) = (\forall x)[x \doteq john]$ ausschließlich ein absurdes Modell, in dem John ein Mensch ist und sich selbst als Kind hat, so dass beispielsweise für die Folgerungsrelation $\models_{dca}$, die eine um das DCA erweiterte Wissensbasis berücksichtigt, der Relationentest $KB_1 \models_{dca} hatKind(john, john)$ erfolgreich ist.

Eine noch schwerwiegendere Problematik ist die Entstehung von Inkonsistenzen durch die Erweiterung von Wissensbasen mit unvollständig spezifizierten Rollenfüllern um den Domänenabschluss. Wird beispielsweise die $\mathcal{ALN}$-Wissensbasis

$$KB_2 = \{Mensch(john), Mensch(susy), (\geq 3\, hatKind)(john)\}$$

um das DCA erweitert, wird angenommen, dass die Domäne nur die beiden Objekte enthält, die von $john$ und $susy$ bezeichnet werden. Die Bedingung $(\geq 3\, hatKind)(john)$ fordert aber, dass John drei unterschiedliche Kinder hat, so dass sich ein Widerspruch ergibt.

Die Berücksichtigung eines Domänenabschlusses ist aber für Anfragen mit Werterestriktionen und höchstens-Anzahlrestriktionen von großer Bedeutung. Um die angesprochene Problematik zu umgehen, kann eine Skolem-Konstante new neben den explizit in einer Wissensbasis vorkommenden Konstanten mit einer bisher nicht bekannten Konstante η identifiziert werden. Die zusätzliche Berücksichtigung einer solchen Konstante führt für jede Skolem-Konstante new_k einer Wissensbasis zu der Auslegung

$$new_k \doteq c_1 \vee ... \vee new_k \doteq c_m \vee new_k \doteq \eta_k. \tag{7.2.1.1}$$

Definition 7.2.1 (Erweiterung des DCA um neue Konstanten) *Seien KB eine beschreibungslogische Wissensbasis, $c_1, ..., c_m$ alle in KB explizit spezifizerten Konstanten, $new_1, ..., new_p$ alle in KB durch $\exists R.C$ und $(\geq n\, R)$ zu berücksichtigende Skolem-Konstanten und $\eta_1, ..., \eta_p$ durch diese Skolem-Konstanten einzubeziehende neue Konstanten. Dann ist*

$$DCA^{+}(KB) = (\forall x)[x \doteq c_1 \vee ... \vee x \doteq c_m \vee x \doteq \eta_1 \vee ... \vee x \doteq \eta_p]$$

das (bzgl. dieser neuen Konstanten) erweiterte Domänenabschlussaxiom.

Die Bestimmung von $new_1, ..., new_p$ erfolgt in Abschnitt 7.3.1 für die Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ und in Abschnitt 7.4.1 für $\mathcal{ALC}$ und $\mathcal{ALCN}$ unter Verwendung von Tableau-Kalkülen. Da die Durchführung eines $\mathcal{ALCN}$-Tableau-Beweises bei der in [Baader und Nutt, 2003] vorgestellten Expansionsstrategie terminiert, werden dabei nur endliche viele Skolem-Konstanten eingeführt.

Die Domäne ergibt sich nach Definition 7.2.1 zu $\triangle^{\mathcal{I}} = \top^{\mathcal{I}} = \{c_1^{\mathcal{I}}, ..., c_m^{\mathcal{I}}, \eta_1^{\mathcal{I}}, ..., \eta_p^{\mathcal{I}}\}$. Konstanten η_k bezeichnen in allen Modellen einer Wissensbasis ein anderes Objekt als alle $c_1, ..., c_m$, führen also zu der Berücksichtigung bisher nicht verwendeter Objekte. Die Auslegung von Skolem-Konstanten new_k (7.2.1.1) bezieht auch die Möglichkeit ein, dass $\eta_k \doteq \eta_l$ bei Existenz einer weiteren Skolem-Konstante new_l gilt. Für Konzeptassertionen mit mindestens-Anzahlrestriktionen $(\geq n\, R)$ besteht diese Möglichkeit nicht: Skolem-Konstanten, die sich aus derselben mindestens-Anzahlrestriktion ergeben, sind paarweise disjunkt.

Definition 7.2.2 (Erweiterung der UNA um neue Konstanten) *Seien KB eine beschreibungslogische Wissensbasis, $c_1, ..., c_m$ alle in KB explizit vorkommenden Konstanten, $\eta_1, ..., \eta_p$ alle durch $DCA^{+}(KB)$ zu berücksichtigenden neuen Konstanten und v eine Meta-Variable, die entweder für eine Konstante oder für eine Skolem-Konstante steht. Dann ist*

$$\begin{aligned} UNA^{+}(KB) = \{&(c_i \not\doteq c_j) \mid 1 \leq i < j \leq m\} \cup \\ \{&(c_i \not\doteq \eta_k) \mid 1 \leq i \leq m,\, 1 \leq k \leq p\} \cup \\ \{&(\eta_k \not\doteq \eta_l) \mid 1 \leq k < l \leq p,\, \eta_k \textit{ und } \eta_l \textit{ gehen aus derselben explizit oder} \\ &\qquad \textit{implizit spezifizierten Konzeptassertion } (\geq n\, R)(v) \textit{ hervor}\} \end{aligned}$$

die (bzgl. neuer Konstanten $\eta_1, ..., \eta_p$) erweiterte Unique-Name-Assumption.

Entsprechend zu der Erweiterung der CWA um das DCA und die UNA (Abschnitt 4.2.4) erfolgt nun eine Erweiterung generalisierter Annahmen um den erweiterten Domänenabschluss und die erweiterte Unique-Name-Assumption. Dazu werden die Relationen $\models_{GCWA}$ und $\models_{WGCWA}$ eingeführt, die an Stelle von $\models_{gcwa}$ bzw. $\models_{wgcwa}$ mit Großbuchstaben notiert werden:

$$\begin{array}{lll} KB \models_{GCWA} \alpha & gdw. & KB \cup GCWA(KB) \cup DCA^{+}(KB) \cup UNA^{+}(KB) \models \alpha \\ KB \models_{WGCWA} \alpha & gdw. & KB \cup WGCWA(KB) \cup DCA^{+}(KB) \cup UNA^{+}(KB) \models \alpha \end{array}$$

Reduktion von Relationentests auf Wissensbasiszugehörigkeit In einfach strukturierten Wissensbasen können Relationentests auf Überprüfung der Zugehörigkeit zu der betrachteten Wissensbasis reduziert werden (Abschnitt 7.1). In Wissensbasen KB, die nicht einfach strukturiert sind, gilt diese Eigenschaft nicht, wenn KB um $DCA(KB)$ erweitert wird. Beispielsweise ist für die oben aufgeführte $\mathcal{AL}$-Wissensbasis KB_1 der Relationentest $KB_1 \models_{dca} hatKind(john, john)$ erfolgreich, obwohl die Rollenassertion $hatKind(john, john)$ nicht in KB_1 enthalten ist. Für die Folgerungsrelation $\models_{dca^+}$, mit

$$KB \models_{dca^+} \alpha \;\; gdw. \;\; KB \cup DCA^{+}(KB) \models \alpha$$

gilt dagegen $KB_1 \models_{dca^+} (hatKind(john, john) \vee hatKind(john, \eta_1))$ unter Einbeziehung einer neuen Konstante η_1, aber nicht $KB_1 \models_{dca^+} hatKind(john, john)$.

Theorem 7.2.3 *Sei KB eine konsistente $\mathcal{ALCN}$-Wissensbasis. Dann gilt für alle explizit in KB spezifizierten Konstanten a und b und eine Rolle R:*

$$\begin{array}{lll} KB \models_{GCWA} R(a,b) & gdw. & R(a,b) \in KB \quad und \\ KB \models_{WGCWA} R(a,b) & gdw. & R(a,b) \in KB. \end{array}$$

Beweis Da beschreibungslogische Wissensbasen $KB = (\mathcal{T}, \mathcal{A})$ in dieser Untersuchung keine terminologischen Axiome für Rollen enthalten, entspricht die Überprüfung einer Rollenassertion auf Zugehörigkeit zu einer Wissensbasis KB einer Überprüfung auf Zugehörigkeit zu einer bzgl. $\mathcal{T}$ expandierten ABox $\mathcal{A}_{\mathcal{T}}$.

Anhand der Interpretation aller Konzeptkonstruktoren aus $\mathcal{ALCN}$ mit prädikatenlogischen Formeln ist ersichtlich, dass durch diese Konstruktoren keine weiteren Rollenassertionen $R(a, b)$ mit in KB vorkommenden Konstanten a, b berücksichtigt werden. Es werden ausschließlich Rollenassertionen mit Skolem-Konstanten new als Rollenfüller durch Konzeptassertionen mit den Konstruktoren $\exists R.C$ und $(\geq n\, R)$ eingeführt. Unter Berücksichtigung eines erweiterten Domänenabschlusses können diese Skolem-Konstanten zumindest mit einer neuen Konstante η identifiziert werden. Eine eindeutige Identifikation von new mit einer explizit spezifizierten Konstante b kann in diesem Fall also nicht erfolgen, so dass $KB \models_{dca^+} R(a, b)$ $gdw.$ $R(a, b) \in KB$ gilt. Aufgrund des Theorems 4.3.5, das besagt, dass durch die Vervollständigung einer beliebigen konsistenten Wissensbasis um die GCWA (oder die WGCWA) keine neuen positiven Klauseln bewiesen werden können, gelten die Ausführungen ebenfalls für die Folgerungsrelationen $\models_{GCWA}$ und $\models_{WGCWA}$. $\square$

7.2.2 Redundante Konzeptassertionen durch Verwendung der GCWA

Durch Spezifikation von Konzeptassertionen mit unqualifizierten Rollenrestriktionen $\exists R.\top$ und $(\geq n\, R)$ ergibt sich unter Verwendung der WGCWA, auch ohne eine Erweiterung des DCA um neue Konstanten, ein weiteres, in vielen Situationen nicht intendiertes Fehlschlagen von Beweisen. Seien beispielsweise die beschreibungslogischen Wissensbasen KB_3 und KB_4 gegeben durch:

$$\begin{aligned} KB_3 = \quad & \{hatKind(john, susy), hatKind(john, mary), Weiblich(susy), \\ & Weiblich(mary), \neg Weiblich(peter)\} \end{aligned}$$

$$\begin{aligned} KB_4 = \quad & \{hatKind(john, susy), hatKind(john, mary), Weiblich(susy), \\ & Weiblich(mary), \neg Weiblich(peter), \exists hatKind.\top(john)\} \end{aligned}$$

Da KB_3 keine indefinite Information enthält, ist der um die WGCWA erweiterte Assertionstest $KB_3 \models_{WGCWA} \forall hatKind.Weiblich(john)$ erfolgreich. Für die um $\exists hatKind.\top(john)$ erweiterte Wissensbasis KB_4 dagegen gilt $KB_4 \not\models_{WGCWA} \forall hatKind.Weiblich(john)$. Dadurch, dass das durch $\exists hatKind.\top(john)$ eingeführte, unbekannte Kind von John möglicherweise Peter ist, ist das Fehlschlagen dieses Beweises nachvollziehbar. Die Information, dass John irgendein Kind hat, ist bereits in KB_3 bekannt, da $KB_3 \models \exists hatKind.\top(john)$ gilt. Das Hinzufügen einer solchen Konzeptassertion soll nicht dazu führen, dass erfolgreiche Assertionstests nicht weiterhin erfolgreich sind. Diese Problematik ergibt sich auch für Konzeptassertionen mit mindestens-Anzahlrestriktionen. Im Folgenden wird zunächst die Überführung unqualifizierter Rollenrestriktionen $\exists R.\top$ und $(\geq n\, R)$ in eine prädikatenlogische Klauselform erläutert. Anschließend wird anhand dieser Klauselform gezeigt, dass die Annahme der hier vorgestellten Redundanz durch die GCWA berücksichtigt wird.

Unqualifizierte Existenzrestriktionsassertionen $\exists R.\top(a)$ entsprechen einer Rollenassertion mit einem unvollständig spezifiziertem Rollenfüller, $R(a, new)$. Wird angenommen, dass eine Skolem-Konstante new ausschließlich mit den in einer Wissensbasis vorkommenden Konstanten $c_1, ..., c_m$ identifiziert werden kann, können Konzeptassertionen $\exists R.\top(a)$ in die prädikatenlogische Klauselform $R(a, c_1) \vee ... \vee R(a, c_m)$ überführt werden (vgl. [Minker, 1982]). Aus den in Abschnitt 7.2.1 dargestellten Gründen besteht allerdings Interesse daran, das um neue Konstanten η erweiterte Domänenabschlussaxiom einzubeziehen. Es ergibt sich die folgende Überführung in eine prädikatenlogische Klauselform:

$$\exists R.\top(a) \quad \Rightarrow \quad R(a, c_1) \vee ... \vee R(a, c_m) \vee R(a, \eta_k) \tag{7.2.2.1}$$

Konzeptassertionen mit mindestens-Anzahlrestriktionen $(\geq n\, R)(a)$ können ebenfalls in eine prädikatenlogische Klauselform transformiert werden. Sie entsprechen der Formelmenge $\{R(a, new_1), ..., R(a, new_n)\} \cup \{new_k \not\approx new_l \mid 1 \leq k < l \leq n\}$. Durch mögliche Identifikationen von new_k und new_l mit anderen Konstanten ergeben sich drei verschiedene Formen von

Ungleichheitsaxiomen $new_k \not\doteq new_l$: Die erste Form enthält zwei ausschließlich in einer Wissensbasis explizit spezifizierte Konstanten c_i und c_j, $i \neq j$, die zweite sowohl eine dieser Konstanten als auch eine neue Konstante η_k und die dritte zwei neue Konstanten η_k und η_l, $k \neq l$. Da die in Abschnitt 7.2.1 definierte erweiterte Unique-Name-Assumption genau diese Formen von Ungleichheitsaxiomen enthält, müssen die Axiome $new_k \not\doteq new_l$ in Wissensbasen $KB \cup UNA^+(KB)$ bei einer Überführung in eine prädikatenlogische duale Klauselform nicht einbezogen werden, und für diese Wissensbasen gilt:

$$\begin{array}{lll} (\geq n\, R)(a) & \Rightarrow & [R(a, c_{i_1}) \wedge ... \wedge R(a, c_{i_n})] \;\vee \\ & & \quad\vdots \qquad\qquad\qquad \vdots \qquad \vee \\ & & [R(a, \eta_{k_1}) \wedge ... \wedge R(a, \eta_{k_n})] \end{array} \tag{7.2.2.2}$$

Die duale Klauselform besteht aus der Disjunktion aller n-stelligen Kombinationen von Konjunktionen von Rollenassertionen mit in KB vorkommenden Rollenfüllern c_i und bisher nicht bekannten Rollenfüllern η_k. Um generalisierte Annahmen berücksichtigen zu können, kann diese duale Klauselform mit dem Distributivgesetz in eine Klauselform transformiert werden.

Definition 7.2.4 *Seien KB eine beschreibungslogische Wissensbasis, a und f_i, $i = 1, ..., m$ Konstanten und R eine Rolle. Dann ist*

- *$Fillers(R, a, KB)$ die Menge der Rollenfüller f_i mit $R(a,f_i) \in KB$*
- *$\mathcal{S}kol_\top(R, a, KB)$ die Menge der aus Konzeptbeschreibungen mit $\exists R.\top$ und $(\geq n\, R)$ resultierenden Skolem-Konstanten, die bzgl. a implizit in KB enthalten sind*
- *$\mathcal{S}kol_{\not\doteq}(R, a, KB)$ eine maximale Teilmenge von $\mathcal{S}kol_\top(R, a, KB)$, die ausschließlich unterschiedliche Skolem-Konstanten new_k und new_l mit $new_k \not\doteq new_l$ enthält*

Beispiel 7.2.5 *Gegeben sei $KB = \{(\geq 3\, R)(a), (\geq 2\, R)(a), \exists R.\top(a), R(a, a), R(a, b)\}$. Es sei festgelegt, dass die Skolem-Konstanten new_1, new_2 und new_3 aus $(\geq 3\, R)(a)$, new_4 und new_5 aus $(\geq 2\, R)(a)$ und new_6 aus $\exists R.\top(a)$ resultieren. Dann ist $Fillers(R, a, KB) = \{a, b\}$, $\mathcal{S}kol_\top(R, a, KB) = \{new_i \mid 1 \leq i \leq 6\}$ und $\mathcal{S}kol_{\not\doteq}(R, a, KB) = \{new_i \mid 1 \leq i \leq 3\}$.*

Bezogen auf die Kardinalitäten der oben definierten Mengen werden zwei Fälle unterschieden:

1. $|\,Fillers(R, a, KB)\,| \;\geq\; |\,\mathcal{S}kol_{\not\doteq}(R, a, KB)\,|$
2. $|\,Fillers(R, a, KB)\,| \;<\; |\,\mathcal{S}kol_{\not\doteq}(R, a, KB)\,|$

Im ersten Fall können bei Verwendung der GCWA alle Skolem-Konstanten aus $\mathcal{S}kol_\top(R, a, KB)$ mit Konstanten aus $Fillers(R, a, KB)$ so identifiziert werden, dass diese Skolem-Konstanten redundant sind. Für den zweiten Fall gilt das aber nicht. Diese Tatsache soll anhand eines einfachen, abstrakten Beispiels mit der Wissensbasis

$$KB = \{(\geq 2\,R)(a), R(a,a), \top(b), \top(c)\}$$

und der Erweiterung dieser Wissensbasis zunächst um $R(a,b)$ und anschließend zusätzlich um $R(a,c)$ erläutert werden. In KB kommen die Konstanten a, b und c vor, und die Skolem-Konstanten new_1 und new_2 sind implizit in KB enthalten. Dadurch ist die Domäne mit $\triangle^{\mathcal{I}} = \top^{\mathcal{I}} = \{a^{\mathcal{I}}, b^{\mathcal{I}}, c^{\mathcal{I}}, \eta_1^{\mathcal{I}}, \eta_2^{\mathcal{I}}\}$ festgelegt. Da die beiden Skolem-Konstanten unterschiedlich sein müssen, ist ihre Auslegung zunächst durch $new_1 \doteq a \vee new_1 \doteq b \vee new_1 \doteq c \vee new_1 \doteq \eta_1$ bzw. $new_2 \doteq a \vee new_2 \doteq b \vee new_2 \doteq c \vee new_2 \doteq \eta_2$ gegeben. Durch Berücksichtigung der GCWA soll diese Auslegung möglichst so eingeschränkt werden können, dass die Konzeptassertion $(\geq 2\,R)(a)$ redundant ist. Das ist bezogen auf dieses Beispiel genau dann der Fall, wenn $KB \models R(a, new_1)$ und $KB \models R(a, new_2)$ für alle möglichen Identifikationen dieser Skolem-Konstanten mit anderen Konstanten gilt.

Die prädikatenlogische duale Klauselform von $(\geq 2\,R)(a)$ ist

$$\begin{array}{l}[R(a,a) \wedge R(a,b)] \vee [R(a,a) \wedge R(a,c)] \vee [R(a,a) \wedge R(a,\eta_1)] \vee [R(a,a) \wedge R(a,\eta_2)] \vee \\ [R(a,b) \wedge R(a,c)] \vee [R(a,b) \wedge R(a,\eta_1)] \vee [R(a,b) \wedge R(a,\eta_2)] \vee \\ [R(a,c) \wedge R(a,\eta_1)] \vee [R(a,c) \wedge R(a,\eta_2)] \vee [R(a,\eta_1) \wedge R(a,\eta_2)].\end{array}$$

Die Klauselmenge von KB ergibt sich durch Überführung dieser dualen Klauselform in eine Klauselform und unter Berücksichtigung von $R(a,a) \in KB$ zu

$$\begin{array}{l}R(a,a) \vee R(a,b) \vee R(a,c) \vee R(a,\eta_1), \\ R(a,a) \vee R(a,b) \vee R(a,c) \vee R(a,\eta_2), \\ R(a,a) \vee R(a,b) \vee R(a,\eta_1) \vee R(a,\eta_2), \\ R(a,a) \vee R(a,c) \vee R(a,\eta_1) \vee R(a,\eta_2), \\ R(a,b) \vee R(a,c) \vee R(a,\eta_1) \vee R(a,\eta_2), \\ R(a,a).\end{array}$$

Es tritt der zweite Fall $|Fillers(R,a,KB)| < |Skol_{\not\equiv}(R,a,KB)|$ ein, da $|\{a\}| = 1$ kleiner ist als $|\{new_1, new_2\}| = 2$. Aufgrund der vorletzten Klausel, die nicht die atomare Formel $R(a,a)$ enthält, können bei Berücksichtigung der Definition der GCWA (siehe Abschnitt 4.3.1) keine negierten Relationentests für R mit der Konstanten a als erstem Tupelelement erfolgreich sein. Es gilt zum Beispiel

$$KB \models R(a,b) \vee R(a,c) \vee R(a,\eta_1) \vee R(a,\eta_2), \quad \text{aber nicht} \quad KB \models R(a,c) \vee R(a,\eta_1) \vee R(a,\eta_2)$$

und dadurch $KB \not\models_{GCWA} \neg R(a,b)$. Die oben dargestellte Auslegung der beiden Skolem-Konstanten kann deswegen nicht eingeschränkt werden, und für die Identifikationen $new_1 \doteq c$ und $new_2 \doteq \eta_2$ gilt zum Beispiel $KB \not\models R(a,c)$ und $KB \not\models R(a,\eta_2)$, so dass $(\geq 2\,R)(a)$ erwartungsgemäß auch unter Annahme der GCWA nicht redundant ist.

Wird KB um $R(a,b)$ erweitert, ist $|Fillers(R,a,KB)| = |Skol_{\not\equiv}(R,a,KB)| = 2$, und alle Klauseln enthalten entweder $R(a,a)$ oder $R(a,b)$, so dass $KB \cup \{R(a,b)\} \models_{GCWA} \neg R(a,c)$,

$KB \cup \{R(a,b)\} \models_{GCWA} \neg R(a,\eta_1)$ und $KB \cup \{R(a,b)\} \models_{GCWA} \neg R(a,\eta_2)$ gilt. Die Identifikationen von new_1 und new_2 mit den Konstanten c, η_1 und η_2 sind aus diesem Grund nicht möglich. Es ergeben sich nur die Auslegungen $new_1 \doteq a \vee new_1 \doteq b$ und $new_2 \doteq a \vee new_2 \doteq b$ mit $KB \cup \{R(a,b)\} \models R(a,a)$ sowie $KB \cup \{R(a,b)\} \models R(a,b)$. Die Skolem-Konstanten new_1 und new_2 sowie die Konzeptassertion $(\geq 2\,R)(a)$ sind deswegen in der um $R(a,b)$ erweiterten Wissensbasis unter Berücksichtigung von $UNA^+(KB)$ redundant und brauchen bei einem Assertionstest unter der GCWA nicht einbezogen werden.

Wenn KB um $\{R(a,b), R(a,c)\}$ erweitert wird, ist $|Fillers(R,a,KB)| > |\mathcal{S}kol_{\not\equiv}(R,a,KB)|$, und es ergeben sich unter Einbeziehung der GCWA die gleichen Redundanzen wie im zuvor geschilderten Fall. Der Unterschied besteht in der größeren Anzahl an Identifikationsmöglichkeiten für new_1 und new_2.

Es wurde erläutert, aus welchen Gründen bestimmte Konzeptassertionen, die zu unvollständig spezifizierten Rollenfüllern führen, bei Verwendung der GCWA redundant sind. Algorithmen, die die Folgerungsrelation $\models_{GCWA}$ verwenden, können deswegen so verfahren, als wenn diese Konzeptassertionen nicht vorhanden wären. Diese Eigenschaft gilt auch für spezielle qualifizierte Existenzrestriktionsassertionen $\exists R.C(a)$. Sollen die hier vorgestellten Redundanzen nicht angenommen werden, und enthält die Sprache keine Konzeptdisjunktionen, eignet sich für viele beschreibungslogische Sprachen die Verwendung der WGCWA.

7.3 Polynomielle Beschreibungslogiken $\mathcal{AL}$ und $\mathcal{ALN}$

In den Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ (und in allen Teilsprachen dieser Beschreibungslogiken) kann die Ausführung eines Assertionstests sowie vieler anderer Reasoning Services bei Verwendung adäquater Verfahren in polynomieller Berechnungszeit erfolgen. Dadurch, dass diese Sprachen keine qualifizierten Existenz- und Anzahlrestriktionen und keine Konzeptdisjunktionen enthalten, kann jede Konzeptbeschreibung C aufgrund der Äquivalenz

$$\forall R.(D_1 \sqcap D_2) \;\equiv\; \forall R.D_1 \sqcap \forall R.D_2$$

in eine Konjunktion konjunktionsfreier Konzepte $C_1 \sqcap ... \sqcap C_p$ umgeformt werden [Donini et al., 1994]. In diesem Fall ist ein Instance-Check $KB \models C(a)$ genau dann erfolgreich, wenn $KB \models C_1(a) \wedge ... \wedge KB \models C_p(a)$ gilt. Jedes Konjunkt C_i, $i = 1, ..., p$ hat dabei die Form $\forall R_1. \cdots .\forall R_s.B$, mit *Basiskonzepten* $B = \top \mid \bot \mid A \mid \neg A \mid \exists R.\top$ in der Sprache $\mathcal{AL}$ und $B = \top \mid \bot \mid A \mid \neg A \mid (\geq n\,R) \mid (\leq n\,R)$ in der Sprache $\mathcal{ALN}$.[1] Im Falle von $s = 0$ besteht C_i ausschließlich aus einem Basiskonzept. Für jedes Basiskonzept B ist ebenfalls $\neg B$ in der Sprache[2], so dass $KB \models B(a)$ bei Verwendung eines geeigneten Tableau-Kalküls weiterhin mit

[1] Basiskonzepte $\exists R.\top$ werden aufgrund der Äquivalenz $\exists R.\top \equiv (\geq 1\,R)$ in dieser Untersuchung für die Sprache $\mathcal{ALN}$ nicht verwendet.

[2] Es gelten z.B die Äquivalenzen $\neg\exists R.\top \equiv \forall R.\bot$, $\neg(\geq n\,R) \equiv (\leq (n-1)\,R)$ und $\neg(\leq nR) \equiv (\geq (n+1)\,R)$.

polynomieller Berechnungskomplexität überprüft werden kann. Im Falle von $s > 0$ hat C_i die Form $\forall R.D$, und für jeden Rollenfüller f_i mit $R(a,f_i) \in KB$ kann rekursiv geprüft werden, ob $KB \models D(f_i)$. Das Konzept D ist entweder ein Basiskonzept oder eine weitere Werterestriktion.

Dieses Top-down-Verfahren bildet eine Grundlage für die Konstruktion von Annahmen einer geschlossenen Welt in polynomiellen Beschreibungslogiken. In Abschnitt 7.3.1 wird zunächst erläutert, wie alle Konzeptassertionen einer $\mathcal{AL}$- oder $\mathcal{ALN}$-Wissensbasis, die zu der Berücksichtigung von Skolem-Konstanten führen, durch die Berechnung eines vorvervollständigten Tableau-Zweiges ermittelt werden können. Anschließend wird in Abschnitt 7.3.2 unter Verwendung dieser vorvervollständigten Zweige sowie der Einbeziehung der in Abschnitt 7.2.2 aufgezeigten Redundanzen ein Verfahren für die Überprüfung von Assertionstest $KB \models_{GCWA} \alpha$ für die Sprache $\mathcal{ALN}$ (und somit auch für die Sprache $\mathcal{AL}$) vorgestellt. Die Einbeziehung der GCWA durch dieses Verfahren erfolgt ohne die Berücksichtigung von Klauselmengen.

7.3.1 Vorvervollständigte Tableau-Zweige in $\mathcal{AL}$ und $\mathcal{ALN}$

Ein Tableau-Zweig wird in dieser Untersuchung *vorvervollständigt* genannt und mit Σ_{prec} bezeichnet, wenn er aus einem initialen Zweig Σ, für den die UNA berücksichtigt wird, nur durch Anwendung nicht-erzeugender Expansionsregeln hervorgeht und keine dieser Regeln mehr angewendet werden kann (vgl. precompletion, [Donini et al., 1994] und [Baader und Hollunder, 1991, S.14ff]). In $\mathcal{AL}$ und $\mathcal{ALN}$ ergibt sich eine Vorvervollständigung bei Verwendung des $\mathcal{ALCN}$-Tableau-Kalküls durch die ausschließliche Anwendung aller möglichen $\forall$- und $\sqcap$- bzw. aller möglichen $\forall$-, $\sqcap$- und $\leq$-Regeln.

Beispiel 7.3.1 Sei eine $\mathcal{AL}$-Wissensbasis mit $KB_1 = \{\forall R.\exists R.\top(a), R(a,b), \forall R.(A_1 \sqcap A_2)(b)\}$ gegeben. Der initiale Zweig Σ geht aus der Erweiterung von KB_1 um $UNA(KB_1) = \{a \not\doteq b\}$ hervor und die Vorvervollständigung von Σ ergibt sich nach Anwendung der $\forall$-Regel:

$$\begin{array}{c} \forall R.\exists R.\top(a) \\ | \\ R(a,b) \\ | \\ \forall R.(A_1 \sqcap A_2)(b) \\ | \\ a \not\doteq b \\ | \\ \exists R.\top(b) \end{array}$$

Vorvervollständigte Zweige Σ_{prec} enthalten keine Skolem-Konstanten. Sie beinhalten alle Informationen über Assertionen bzgl. Konstanten, die in der zugehörigen Wissensbasis explizit vorkommen. Rollenassertionen $R(a,b) \in \Sigma_{prec}$ mit explizit spezifizierten Konstanten a, b wurden vollständig berücksichtigt und tragen zu keiner weiteren Expansion bei.

Für $\mathcal{AL}$- und $\mathcal{ALN}$-Wissensbasen gilt insbesondere:

- Zu jedem initialen Zweig Σ gibt es nur einen vorvervollständigten Zweig Σ_{prec}

Die $\forall$- und die $\sqcap$-Regel sind deterministisch. Die nicht-deterministische $\leq$-Regel kann aufgrund der Einbeziehung der UNA und der Tatsache, dass Σ_{prec} keine Skolem-Konstanten enthält, nicht ausgeführt werden. Aus diesem Grund gibt es nur einen vorvervollständigten Zweig in $\mathcal{AL}$ und $\mathcal{ALN}$. Außerdem gilt für Σ_{prec} in $\mathcal{AL}$- und $\mathcal{ALN}$-Wissensbasen KB:

- Σ_{prec} enthält alle explizit oder implizit spezifizierten Konzeptassertionen $\exists R.\top(a)$ und $(\geq n\, R)(a)$ für in KB vorkommende Konstanten a.

Alle Konzeptassertionen, die sich aus Σ_{prec} durch die zusätzliche Ausführung der $\exists$- und $\geq$-Regeln ergeben würden, beziehen sich ausschließlich auf Skolem-Konstanten new. Neben der Menge $Fillers(R, a, KB)$ lassen sich in $\mathcal{AL}$- und $\mathcal{ALN}$-Wissensbasen KB folglich auch die Mengen $\mathcal{S}kol_{\top}(R, a, KB)$ und $\mathcal{S}kol_{\not\equiv}(R, a, KB)$ für alle in KB vorkommenden Rollen R und Konstanten a aus Σ_{prec} bestimmen.

Beispiel 7.3.2 Gegeben sei $KB_2 = \{\forall R.(\geq 2\, P)(a), R(a, b), (\geq 1\, R)(a)\}$. Die Vorvervollständigung Σ_{prec} von KB_2 ist:

$$\begin{array}{c} \forall R.(\geq 2\, P)(a) \\ | \\ R(a, b) \\ | \\ (\geq 1\, R)(a) \\ | \\ a \not\equiv b \\ | \\ (\geq 2\, P)(b) \end{array}$$

Es ergibt sich $Fillers(R, a, KB_2) = \{b\}$. Werden die Skolem-Konstante new_1 für den unvollständig spezifizierten Rollenfüller der Konzeptassertion $(\geq 1\, R)(a)$ eingeführt und new_2, new_3 für die unvollständig spezifizierten Rollenfüller der Konzeptassertion $(\geq 2\, P)(b)$, können außerdem die Mengen $\mathcal{S}kol_{\not\equiv}(R, a, KB_2) = \mathcal{S}kol_{\top}(R, a, KB_2) = \{new_1\}$ und $\mathcal{S}kol_{\not\equiv}(P, b, KB_2) = \mathcal{S}kol_{\top}(P, b, KB_2) = \{new_2, new_3\}$ bestimmt werden.

Werden alle Konzeptassertionen $\forall R.(D_1 \sqcap ... \sqcap D_p)(a)$ aus einer Vorvervollständigung Σ_{prec} durch die Konzeptassertionen $\forall R.D_i(a)$, $i = 1, ..., p$ ersetzt und anschließend alle weiteren Assertionen mit Konzeptkonjunktionen aus Σ_{prec} entfernt, ergibt sich eine *konjunktionsfreie Vorvervollständigung* Σ_{pre}. Für diese Vorvervollständigung gilt neben den Eigenschaften, die für Σ_{prec} gelten, zusätzlich:

- Σ_{pre} enthält alle explizit oder implizit spezifizierten Assertionen $\forall R_1. \cdots .\forall R_s.B(a)$ für in *KB* vorkommende Konstanten a mit Basiskonzepten B und $s \geq 1$.

Folglich gilt $\forall R_1. \cdots .\forall R_s.B(a) \in \Sigma_{pre}$ *gdw.* $KB \models \forall R_1. \cdots .\forall R_s.B(a)$. Die konjunktionsfreie Vorvervollständigung Σ_{pre} der weiter oben dargestellten Wissensbasis KB_1 ergibt sich beispielsweise aus der Vorvervollständigung von KB_1 nach Ersetzung der Konzeptassertion $\forall R.(A_1 \sqcap A_2)(b)$ durch die Konzeptassertionen $\forall R.A_1(b)$ und $\forall R.A_2(b)$.

7.3.2 Konstruktion einer GCWA in $\mathcal{ALN}$

In diesem Abschnitt wird ein Verfahren für die Überprüfung eines um die GCWA erweiterten Assertionstests $KB \models_{GCWA} \alpha$ mit einer $\mathcal{ALN}$-Wissensbasis *KB* und einer $\mathcal{ALN}$-Assertion α vorgestellt. Im Gegensatz zu dem Verfahren aus Abschnitt 7.4.3 bezieht dieses Verfahren nicht die aufwändige Berechnung einer prädikatenlogischen Klauselform ein. Da erzeugende Rollenrestriktionen $(\geq n\, R)$ unqualifiziert sind, gibt es nur eingeschränkte Informationen über Skolem-Konstanten. Es wird gezeigt, in welchen Fällen diese Informationen zu einem Fehlschlagen von Assertionstests führen und in welchen Fällen sie durch Überprüfung der Zugehörigkeit bestimmter Konzeptassertionen zu der konjunktionsfreien Vorvervollständigung Σ_{pre} einer $\mathcal{ALN}$-Wissensbasis korrekt berücksichtigt werden können. Alle Ausführungen beziehen sich zwar auf die Sprache $\mathcal{ALN}$, gelten aber auch für $\mathcal{AL}$, wenn die Äquivalenz der Konzeptbeschreibungen $(\geq 1\, R)$ und $\exists R.\top$ beachtet wird, und alle weiteren Bedingungen mit Anzahlrestriktionen ignoriert werden.

Für jede Rolle R und für jede Konstante a einer $\mathcal{ALN}$-Wissensbasis *KB* wird unterschieden, ob $|Fillers(R, a, KB)| \geq |\mathcal{S}kol_{\not\equiv}(R, a, KB)|$ oder $|Fillers(R, a, KB)| < |\mathcal{S}kol_{\not\equiv}(R, a, KB)|$. Gilt der erste Fall, sind alle Skolem-Konstanten aus $\mathcal{S}kol_{\top}(R, a, KB)$ und dadurch alle Konzeptassertionen, die diese Skolem-Konstanten einführen, unter der GCWA redundant (siehe Abschnitt 7.2.2). Skolem-Konstanten werden deswegen für alle R und a nur berücksichtigt, wenn $|Fillers(R, a, KB)| < |\mathcal{S}kol_{\not\equiv}(R, a, KB)|$ gilt. Die Vollständigkeit des in diesem Abschnitt vorgestellten Verfahrens gilt nur, wenn die Wissensbasis folgender Einschränkung unterliegt:

Bedingung 7.3.3 *Ist* $|Fillers(R, a, KB)| < |\mathcal{S}kol_{\not\equiv}(R, a, KB)|$*, kommen Konzeptassertionen* $\forall R_1.\cdots.\forall R_s.(\geq n\, P)(a)$ *für alle* $n > 0$ *und* $\forall R_1.\cdots.\forall R_s.A(a)$ *für alle* $s > 1$ *nicht in* Σ_{pre} *vor.*

Diese Einschränkung einer $\mathcal{ALN}$-Wissensbasis wird für alle Sätze und Theoreme dieses Abschnitts vorausgesetzt.

Relationentests und negierte Relationentests Relationentests $KB \models_{GCWA} R(a, b)$ werden korrekt durchgeführt, indem überprüft wird, ob die entsprechende Rollenassertion in *KB* enthalten ist (Theorem 7.2.3). Die Beantwortung von negierten Relationentests ist unter Berücksichtigung der GCWA in $\mathcal{ALN}$-Wissensbasen *KB* unvollständig: Ist $R(a, b)$ nicht in *KB* enthalten und es gibt nicht-redundante Konzeptassertionen $(\geq n\, R)(a)$ in Σ_{pre}, so dass sich

$|Fillers(R, a, KB)| < |Skol_{\not\approx}(R, a, KB)|$ ergibt, dann gilt sowohl $KB \not\models_{GCWA} R(a, b)$ als auch $KB \not\models_{GCWA} \neg R(a, b)$. Unvollständige Antworten ergeben sich ebenfalls bei der Ausführung von Instance-Checks mit höchstens-Anzahlrestriktionen $KB \models_{GCWA} (\leq n\, R)(a)$ und Werterestriktionen $KB \models_{GCWA} \forall R.C(a)$, da durch die Überführung dieser Konstruktoren in eine prädikatenlogische Formel mit Implikationen [Baader und Nutt, 2003, S. 54 ff] negierte Relationentests berücksichtigt werden müssen. Allerdings können Aussagen über negierte Relationentests mit neuen Konstanten η getroffen werden:

Satz 7.3.4 *Seien KB eine $\mathcal{ALN}$-Wissensbasis und Σ_{pre} die konjunktionsfreie Vorvervollständigung von KB. Dann gilt für alle durch Konzeptassertionen $(\geq n\, R)(a) \in \Sigma_{pre}$ eingeführten neuen Konstanten η_a, für alle Konstanten w_i und für alle atomaren Rollen P:*

$$KB \models_{GCWA} \neg P(\eta_a, w_i)$$

Unter der Bedingung 7.3.3 kann es keine Informationen über Rollenfüller von neuen Konstanten η_a geben, so dass alle negierten Relationentests mit η_a als erstem Tupelelement bzgl. aller Rollen P unter der GCWA erfolgreich sind.

Instance-Checks mit negierten atomaren Konzepten Instance-Checks mit negierten atomaren Konzepten unter der GCWA, $KB \models_{GCWA} \neg A(a)$, sind in vielen Fällen erfolgreich, wenn $KB \not\models A(a)$ gilt. Eine Ausnahme bildet die Spezifikation von Werterestriktionen, in deren Skopus das atomare Konzept A vorkommt:

Beispiel 7.3.5 Sei die $\mathcal{ALN}$-Wissensbasis

$$KB = \{\forall hatKind.Weiblich(john), (\geq 1\, hatKind)(john), \top(charles)\}$$

gegeben. Dann gilt $KB \not\models_{GCWA} \neg Weiblich(charles)$, obwohl $KB \not\models Weiblich(charles)$. Würde angenommen werden, dass Charles nicht weiblich ist, dann müsste dass auch für John und für das Objekt, das durch eine neue Konstante η bezeichnet wird, gelten. Dann wäre KB inkonsistent, da die Bedingung, dass John mindestens ein weibliches Kind hat, nicht erfüllt ist.

Theorem 7.3.6 *Sei KB eine $\mathcal{ALN}$-Wissensbasis, Σ_{pre} die konjunktionsfreie Vorvervollständigung von KB, A ein atomares Konzept und a eine in KB vorkommende Konstante. Des Weiteren sei $\forall P.A(c_i) \notin \Sigma_{pre}$ oder $|Fillers(P, c_i, KB)| \geq |Skol_{\not\approx}(P, c_i, KB)|$ für alle in KB vorkommenden atomaren Rollen P und Konstanten c_i, $i = 1, ..., m$. Dann gilt:*

$$KB \models_{GCWA} \neg A(a) \;\; gdw. \;\; KB \not\models A(a) \qquad (7.3.2.1)$$

Beweis Eine negierte atomare Konzeptassertion $\neg A(a)$ folgt unter der GCWA aus einer Wissensbasis KB genau dann, wenn die Konzeptassertion $A(a)$ nicht logisch aus KB folgt und jede positive Klausel K mit $KB \models A(a) \vee K$ aus KB logisch folgt (vgl. Definition 4.3.1). Die erste Bedingung wird durch (7.3.2.1) berücksichtigt. Die Überführung von Werterestriktionsassertionen $\forall P.A(c_i)$ in eine prädikatenlogische Klauselform (vgl. [Baader und Nutt, 2003, S. 54]) ist unter Berücksichtigung aller in *KB* vorkommender Konstanten $c_1, ..., c_m$ sowie aller einzubeziehender neuer Konstanten $\eta_1, ..., \eta_p$ gegeben durch:

$$\begin{aligned} \forall P.A(c_i) \quad \Rightarrow \quad & [\neg P(c_i, c_1) \vee A(c_1)] \wedge ... \wedge [\neg P(c_i, c_m) \vee A(c_m)] \wedge \\ & [\neg P(c_i, \eta_1) \vee A(\eta_1)] \wedge ... \wedge [\neg P(c_i, \eta_p) \vee A(\eta_p)] \end{aligned}$$

Die resultierende Klauselform enthält keine positiven Klauseln K. Ist $|Fillers(P, c_i, KB)| \geq |\mathcal{S}kol_{\not\approx}(P, c_i, KB)|$ für eine Rolle P und eine Konstante c_i, müssen Klauseln, die aus der Bedingung (7.2.2.2) resultieren, nicht einbezogen werden. Ist $|Fillers(P, c_i, KB)| < |\mathcal{S}kol_{\not\approx}(P, c_i, KB)|$ für eine Rolle P und eine Konstante c_i, dann müssen diese Klauseln berücksichtigt werden. Da sie ausschließlich Formeln mit der Relation P enthalten, können unter der Voraussetzung, dass $\forall P.A(c_i) \notin \Sigma_{pre}$ unter der Bedingung 7.3.3 keine Resolventen mit $A(a)$ gebildet werden. Aufgrund der Tatsache, dass in $\mathcal{ALN}$ weder Konzeptdisjunktionen noch qualifizierte erzeugende Rollenrestriktionen spezifiziert werden können, kann es in beiden Fällen für alle in *KB* vorkommenden Konstanten c_i keine zu berücksichtigenden positiven Klauseln mit $A(a)$ und weiteren Literalen geben, und es gilt $KB \models_{GCWA} \neg A(a)$. □

Da die GCWA eine Vervollständigung ist, gilt $KB \models_{GCWA} \neg A(a)$, wenn $KB \models \neg A(a)$ gelten sollte. Ist sowohl $\forall P.A(c_i) \in \Sigma_{pre}$ als auch $|Fillers(P, c_i, KB)| < |\mathcal{S}kol_{\not\approx}(P, c_i, KB)|$ für eine Rolle P und eine Konstante c_i, ist die Beantwortung von Instance-Checks mit negierten atomaren Konzepten $\neg A(a)$ für alle in *KB* vorkommenden Konstanten a unter der GCWA unvollständig, wenn $KB \not\models \neg A(a)$.

Ist die Bedingung 7.3.3 nicht erfüllt, können sich durch (7.3.2.1) inkorrekte Antworten ergeben, wenn zum Beispiel $\forall P.\forall R.A(c_i) \in \Sigma_{pre}$ und $|Fillers(P, c_i, KB)| < |\mathcal{S}kol_{\not\approx}(P, c_i, KB)|$ für eine Konstante c_i gilt. Aus diesem Grund wird für die Anwendung von (7.3.2.1) für jede Konstante c_i, für die die Bedingung $|Fillers(P, c_i, KB)| \geq |\mathcal{S}kol_{\not\approx}(P, c_i, KB)|$ nicht erfüllt ist, geprüft, ob es in Σ_{pre} eine Konzeptassertion $\forall P.R_1.\cdots.\forall R_s.A(c_i)$ mit $s \geq 0$ gibt. Gibt es eine solche Konzeptassertion nicht, gilt die Bedingung (7.3.2.1). Der Beweis eines entsprechenden Theorems ergibt sich aus dem Beweis für Theorem 7.3.6.

Wie im Falle negierter Relationentests können zusätzliche Aussagen über Instance-Checks mit negierten atomaren Konzepten und neuen Konstanten η getroffen werden:

Satz 7.3.7 *Seien KB eine $\mathcal{ALN}$-Wissensbasis, Σ_{pre} die konjunktionsfreie Vorvervollständigung von KB, A ein atomares Konzept, a eine in KB vorkommende Konstante und es sei $\forall R.A(a) \notin \Sigma_{pre}$. Dann gilt für alle durch Konzeptassertionen $(\geq n\, R)(a) \in \Sigma_{pre}$ eingeführten neuen Konstanten η_a:*

$$KB \models_{GCWA} \neg A(\eta_a)$$

Ist $\forall R.A(a) \notin \Sigma_{pre}$, gibt es unter der Bedingung 7.3.3 keine positive Information über atomare Konzeptassertionen $A(\eta_a)$ mit neuen Konstanten η_a, da erzeugende Rollenrestriktionen in $\mathcal{ALN}$ unqualifiziert sind und durch Konzeptassertionen $\forall R_1.\cdots.\forall R_s.B(a) \in \Sigma_{pre}$ mit $B = \top \mid \bot \mid \neg A \mid (\geq n\, R) \mid (\leq n\, R)$ keine (positiven) Aussagen über A getroffen werden.

Instance-Checks mit anderen Basiskonzepten Für Instance-Checks mit den Basiskonzepten $B = \top \mid \bot \mid A \mid (\geq n\, R)$ gelten die folgenden Bedingungen:

$$\begin{array}{lll} KB \models_{GCWA} A(a) & gdw. & KB \models A(a) \\ KB \models_{GCWA} (\geq n\, R)(a) & gdw. & KB \models_{una} (\geq n\, R)(a) \end{array}$$

Instance-Checks mit atomaren Konzepten (und den logischen Konstanten $\top$ und $\bot$) können unter der GCWA aufgrund von Theorem 4.3.5 auf logische Folgerung zurückgeführt werden. Während die zusätzliche Berücksichtigung der UNA auf Instance-Checks mit atomaren Konzepten keine Auswirkungen hat, ist sie für erfolgreiche Instance-Checks mit mindestens-Anzahlrestriktionen erforderlich (siehe Abschnitt 3.2.4). Werden Assertionen $(\geq n\, R)(a)$ in eine duale Klauselform überführt (7.2.2.2), enthält die Klauselmenge, die sich aus dieser dualen Klauselform ergibt, ausschließlich Klauseln mit positiven Literalen. Die Gültigkeit der oben aufgeführten Bedingung für mindestens-Anzahlrestriktionen ergibt sich deswegen ebenfalls aus Theorem 4.3.5.

Für Instance-Checks mit höchstens-Anzahlrestriktionen unter der GCWA ist die Bedingung

$$KB \models_{GCWA} (\leq n\, R)(a) \;\; gdw. \;\; KB \not\models_{una} (\geq (n+1)\, R)(a)$$

nur erfüllt, wenn $|Fillers(R, a, KB)| \geq |\mathcal{S}kol_{\not\approx}(R, a, KB)|$ oder $(\leq m\, R)(a) \in \Sigma_{pre}$, mit $m \leq n$.[3] Anderenfalls ist die Beantwortung dieser Instance-Checks unvollständig:

Beispiel 7.3.8 Für die $\mathcal{ALN}$-Wissensbasis $KB = \{(\geq 1\, R)(a)\}$ gilt $KB \not\models_{GCWA} (\leq 1\, R)(a)$. Unter Berücksichtigung von $DCA^+(KB)$ kann KB in die prädikatenlogische Klauselform $R(a, a) \vee R(a, \eta_a)$ überführt werden. Da nach Definiton der GCWA weder $KB \models_{GCWA} \neg R(a, a)$ noch $KB \models_{GCWA} \neg R(a, \eta_a)$ gilt, kann für KB nicht angenommen werden, dass es für die Konstante a bzgl. der Rolle R höchstens einen Rollenfüller gibt.

Instance-Checks mit Werterestriktionen Gilt $\forall R.D(a) \in \Sigma_{pre}$, ist der Instance-Check $KB \models_{GCWA} \forall R.D(a)$ unter der GCWA erfolgreich. Sollte $\forall R.D(a) \notin \Sigma_{pre}$ gelten, wird die GCWA für Instance-Checks mit Werterestriktionen wie folgt berücksichtigt:

Ist $|Fillers(R, a, KB)| \geq |\mathcal{S}kol_{\not\approx}(R, a, KB)|$, werden Instance-Checks $KB \models_{GCWA} \forall R.D(a)$ korrekt durchgeführt, indem für jeden Rollenfüller c_i mit $R(a, c_i) \in KB$ rekursiv geprüft wird, ob $KB \models_{GCWA} D(c_i)$.

[3] Unter der letztgenannten Bedingung sind Instance-Checks $KB \models (\leq n\, R)(a)$ erfolgreich.

Ist dagegen $|Fillers(R, a, KB)| < |\mathcal{S}kol_{\not\approx}(R, a, KB)|$, können Instance-Checks mit erzeugenden Rollenrestriktionen im Skopus einer Werterestriktion, $KB \models_{GCWA} \forall R.(\geq n\, P)(a)$ nicht erfolgreich sein, weil es keine Informationen über Rollenfüller von neuen Konstanten η_a gibt (Satz 7.3.4). Entsprechend gilt $KB \not\models_{GCWA} \forall R.A(a)$ aufgrund von Satz 7.3.7, wenn nicht-redundante Skolem-Konstanten berücksichtigt werden müssen.[4] Für Instance-Checks mit negierten atomaren Konzepten oder mit höchstens-Anzahlrestriktionen im Skopus einer Werterestriktion, $KB \models_{GCWA} \forall R.\neg A(a)$ bzw. $KB \models_{GCWA} \forall R.(\leq n\, P)(a)$ sowie für Instance-Checks mit verschachtelten Werterestriktionen, $KB \models_{GCWA} \forall R.\forall P.C(a)$ gelten andere Bedingungen: Da der Erfolg impliziter negierter Relationentests im Falle $|Fillers(R, a, KB)| < |\mathcal{S}kol_{\not\approx}(R, a, KB)|$ nicht entschieden werden kann, muss für alle in KB vorkommenden Konstanten c_i, unabhängig davon, ob $R(a, c_i) \in KB$ oder $R(a, c_i) \notin KB$ gilt, geprüft werden, ob $KB \models_{GCWA} D(c_i)$. Durch Skolem-Konstanten zu berücksichtigende neue Konstanten η_a müssen dabei aufgrund der Sätze 7.3.4 und 7.3.7 nicht einbezogen werden.

Das Verfahren Die Durchführung (negierter) Relationentests in $\mathcal{ALN}$ unter der GCWA wurde bereits ausreichend erläutert. Instance-Checks mit Konzeptbeschreibungen unter Berücksichtigung der GCWA, $KB \models_{GCWA} C(a)$, werden in $\mathcal{ALN}$ grundlegend wie weiter oben dargestellt durchgeführt: Das Anfragekonzept C wird in eine Konjunktion konjunktionsfreier Konzepte C_i umgeformt. Jedes Konjunkt C_i, $i = 1, ..., p$ hat die Form $\forall R_1. \cdots .\forall R_s.B$, mit *Basiskonzepten* $B = \top \mid \bot \mid A \mid \neg A \mid (\geq n\, R) \mid (\leq n\, R)$, so dass für jedes C_i entweder Instance-Checks mit Werterestriktionen $KB \models_{GCWA} \forall R.D(a)$ oder Instance-Checks mit Basiskonzepten $KB \models_{GCWA} B(a)$ auf Erfolg geprüft werden. Es wird zusätzlich vorausgesetzt, dass $n > 0$ für Basiskonzepte mit Anzahlrestriktionen gilt. Besteht Interesse daran, die Konstruktoren $\exists R.\top$, $(\geq 0\, R)$ oder $(\leq 0\, R)$ zu verwenden, können diese bei Beginn des Verfahrens in die Sprachausdrücke $(\geq 1\, R)$ bzw. $\top$ bzw. $\forall R.\bot$ überführt werden.

Abbildung 7.1 stellt den Algorithmus $AssertGCWA_{\mathcal{ALN}}$ dar. Die Notationen $Fillers$ und $\mathcal{S}kol_{\not\approx}$ werden als Abkürzung für $Fillers(R, a, KB)$ bzw. $\mathcal{S}kol_{\not\approx}(R, a, KB)$ verwendet, wenn Rollenfüller einer Konstanten a bzgl. einer Rolle R in KB berücksichtigt werden sollen. Um eine exponentielle Anzahl von rekursiven Aufrufen der Funktion *InstGCWA* zu vermeiden, wird die dynamische Datenstruktur *INST_OF* zur Speicherung von Ergebnissen bereits ausgeführter Instance-Checks verwendet [Donini et al., 1994]. Dadurch, dass Tableau-Beweise für $KB \models C(a)$ ausschließlich für Basiskonzepte benötigt werden, entscheidet $AssertGCWA_{\mathcal{ALN}}$ in polynomieller Berechnungszeit, ob $KB \models_{GCWA} \alpha$. Die Gültigkeit der Bedingung 7.3.3 kann durch Überprüfung der Zugehörigkeit entsprechender Konzeptassertionen zu Σ_{pre} kontrolliert werden. Ist diese Bedingung nicht erfüllt, ist das Verfahren unvollständig.

[4] Die Ausnahme $\forall R.A(a) \in \Sigma_{pre}$ wurde bereits berücksichtigt.

Function $InstGCWA(KB, \Sigma_{pre}, a, C)$
begin
  if $(INST\_OF[a, C] = unknown)$ {
    if $(C = \forall R.D)$ {
      if $\forall R.D(a) \in \Sigma_{pre}$ $INST\_OF[a, C]$:= TRUE;
      else if $((D = (\geq n\,P) \vee D = A) \wedge (|Fillers| < |\mathcal{S}kol_{\not\approx}|))$ $INST\_OF[a, C]$:= FALSE;
      else {
        if $(|Fillers| \geq |\mathcal{S}kol_{\not\approx}|)$ dann entspricht die Menge $\{c_1, ..., c_m\}$ der Menge $Fillers$;
        else $\{c_1, ..., c_m\}$ ist die Menge aller in KB vorkommender Konstanten;
        if $(m = 0)$ $INST\_OF[a, C]$:= TRUE;
        else $INST\_OF[a, C] := InstGCWA(KB, \Sigma_{pre}, c_1, D) \wedge ... \wedge InstGCWA(KB, \Sigma_{pre}, c_m, D)$; }}

    else if $KB \models_{una} C(a)$ $INST\_OF[a, C]$:= TRUE;

    else if $(C = \neg A)$ {
      if $(|Fillers(P, c_i, KB)| \geq |\mathcal{S}kol_{\not\approx}(P, c_i, KB)| \vee \forall P.R_1.\cdots.\forall R_s.A(c_i) \notin \Sigma_{pre}$ mit $s \geq 0$
        für alle Rollen P und Konstanten c_i aus $KB)$ $INST\_OF[a, C] := KB \not\models \neg C(a)$;
      else $INST\_OF[a, C]$:= FALSE; }

    else if $(C = (\leq n\,R))$ {
      if $(|Fillers| \geq |\mathcal{S}kol_{\not\approx}|)$ $INST\_OF[a, C] := KB \not\models_{una} \neg C(a)$;
      else $INST\_OF[a, C]$:= FALSE; }

    else $INST\_OF[a, C]$:= FALSE; }
  return $INST\_OF[a, C]$;
end

Algorithmus $AssertGCWA_{\mathcal{ALN}}(KB, \alpha)$
Input $\mathcal{ALN}$-Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$, $\mathcal{ALN}$-Assertion α
Output TRUE wenn $KB \models_{GCWA} \alpha$, FALSE sonst
begin
    if $(KB \models_{una} \bot(a)$ für eine in KB vorkommende Konstante $a)$ **return** TRUE;
    Sei $\mathcal{A}_{\mathcal{T}}$ die bzgl. $\mathcal{T}$ expandierte ABox $\mathcal{A}$;
    Sei Σ_{pre} die konjunktionsfreie Vorvervollständigung von $\mathcal{A}_{\mathcal{T}}$;
    for each (R und a kommen in KB vor) bestimme $Fillers(R, a, KB), \mathcal{S}kol_{\not\approx}(R, a, KB)$ aus Σ_{pre}
    if $(\alpha = C(a))$ {
      Sei $C_1, ..., C_p$ die konjunktionsfreie Form von C;
      for each (c kommt in KB vor, D ist Teilkonzept von C_i) $INST\_OF[c, D] := unknown$;
      return $InstGCWA(KB, \Sigma_{pre}, a, C_1) \wedge ... \wedge InstGCWA(KB, \Sigma_{pre}, a, C_p)$; }
    else if $(\alpha = \neg R(a, b))$ {
      if $(R(a, b) \notin \mathcal{A}_{\mathcal{T}} \wedge (|Fillers| \geq |\mathcal{S}kol_{\not\approx}|))$ **return** TRUE;
      else return FALSE; }
    else return $R(a, b) \in \mathcal{A}_{\mathcal{T}}$;
end

Abbildung 7.1: der *Der* $AssertGCWA_{\mathcal{ALN}}$*-Algorithmus*

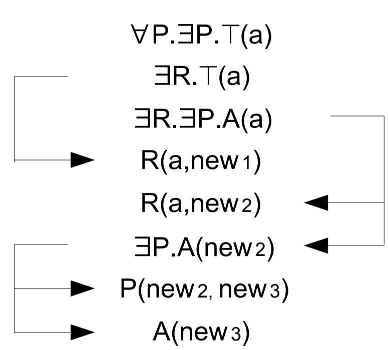
7.4 Die Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$

Die Formalisierung eines Assertionstests, der eine generalisierte Annahme einer geschlossenen Welt berücksichtigen soll, ist in aussagenlogisch abgeschlossenen beschreibungslogischen Sprachen wie $\mathcal{ALC}$ und $\mathcal{ALCN}$ äußerst aufwändig. Im Gegensatz zu den in Abschnitt 7.3 vorgestellten polynomiellen Beschreibungslogiken können in diesen Sprachen Konzeptassertionen mit Konzeptdisjunktionen und qualifizierten Existenzrestriktionen spezifiziert werden, so dass Konzepte nicht in eine Konjunkton konjunktionsfreier Konzepte umgeformt werden können.

Zunächst wird in Abschnitt 7.4.1 gezeigt, dass mit dem $\mathcal{ALCN}$-Tableau-Kalkül alle Skolem-Konstanten und durch diese zu berücksichtigenden neuen Konstanten η einer $\mathcal{ALCN}$-Wissensbasis ermittelt werden können. Im Anschluss daran wird in Abschnitt 7.4.2 unter Berücksichtigung dieser neuen Konstanten die Überführung von $\mathcal{ALCN}$-Konzeptassertionen in eine prädikatenlogische Klauselform erläutert. Abschließend wird in Abschnitt 7.4.3 ein Verfahren vorgestellt, das unter Verwendung dieser Überführung entscheidet, ob eine $\mathcal{ALCN}$-Assertion unter der GCWA aus einer $\mathcal{ALCN}$-Wissensbasis folgt oder nicht. Es wird gezeigt, dass die Berücksichtigung dieser GCWA lokal erfolgen kann.

7.4.1 Bestimmung neuer Konstanten η einer $\mathcal{ALCN}$-Wissensbasis

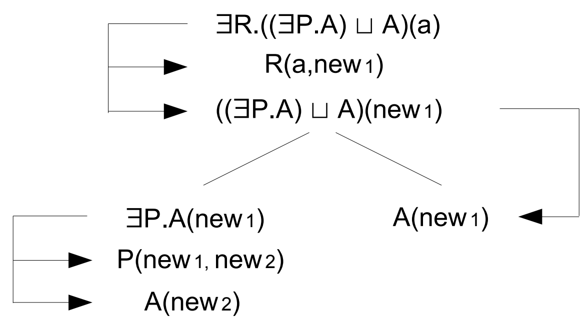
Unter der Voraussetzung, dass erzeugende Expansionsregeln nur dann ausgeführt werden, wenn keine anderen Expansionsregeln ausgeführt werden können, terminiert die Durchführung eines $\mathcal{ALCN}$-Tableau-Beweises [Baader und Nutt, 2003]. Die Menge $\mathcal{S}(KB)$, die aus den Assertionen aller vollständig expandierten Zweige $\Sigma_1, ..., \Sigma_r$ einer erfüllbaren $\mathcal{ALCN}$-Wissensbasis KB hervorgeht, enthält in diesem Fall alle aus KB einzubeziehenden Skolem-Konstanten new. Beispielsweise ist die $\mathcal{ALCN}$-Wissensbasis $KB_1 = \{\forall P.\exists P.\top(a), \exists R.\top(a), \exists R.\exists P.A(a)\}$ erfüllbar und der Zweig, der sich durch eine vollständige Expansion aller Assertionen aus KB_1 ergibt, enthält alle für KB_1 zu berücksichtigenden Skolem-Konstanten:



Für die im Skopus der Werterestriktion enthaltene Konzeptbeschreibung $\exists P.\top$ wird durch den Kalkül erwartungsgemäß keine Skolem-Konstante eingeführt, da KB_1 keine Information über einen Rollenfüller von a bzgl. P enthält.

In Abschnitt 7.2.1 wurde der Domänenabschluss um neue Konstanten η erweitert. Um mögliche Identifikationen aller in $\mathcal{S}(KB)$ enthaltenen Skolem-Konstanten new_k zumindest mit einer neuen Konstante η_k festzuhalten, kann für alle in $\mathcal{S}(KB)$ vorkommenden Rollen T und Konstanten v die Menge $NEU(T, v)$ erstellt werden, die für jede Rollenassertion $T(v, new_k) \in \mathcal{S}(KB)$ um η_k erweitert wird. Ist v selbst eine Skolem-Konstante, muss für alle Konstanten w, mit denen v nach (7.2.1.1) identifiziert werden kann, die Menge $NEU(T, w)$ erstellt werden. Für KB_1 ergeben sich zum Beispiel durch $R(a, new_1) \in \mathcal{S}(KB_1)$ und $R(a, new_2) \in \mathcal{S}(KB_1)$ die Menge $NEU(R, a) = \{\eta_1, \eta_2\}$ und durch $P(new_2, new_3) \in \mathcal{S}(KB_1)$ aufgrund der Auslegung $new_2 \doteq a \vee new_2 \doteq \eta_2$ die Mengen $NEU(P, a) = \{\eta_3\}$ und $NEU(P, \eta_2) = \{\eta_3\}$.

Ist eine Rollenassertion $T(v, new_k)$, die eine Skolem-Konstante new_k einführt, nicht im Hauptzweig enthalten, gibt es Modelle der betreffenden Wissensbasis, in denen new_k nicht berücksichtigt werden muss. Es besteht aber Interesse daran, alle Möglichkeiten einzubeziehen, so dass die Menge $NEU(T, v)$ wie bisher beschrieben erstellt wird. Für die vollständig expandierten Zweige der erfüllbaren $\mathcal{ALCN}$-Wissensbasis $KB_2 = \{\exists R.((\exists P.A) \sqcup A)(a)\}$



ergeben sich beispielsweise für die nicht im Hauptzweig vorkommende Rollenassertion $P(new_1, new_2)$ die Mengen $NEU(P, a) = \{\eta_2\}$ und $NEU(P, \eta_1) = \{\eta_2\}$.

7.4.2 Überführung von $\mathcal{ALCN}$-Konzeptassertionen in eine prädikatenlogische Klauselform

Beliebige $\mathcal{ALCN}$-Konzeptassertionen, die sich in Negationsnormalform (NNF) befinden, können in eine prädikatenlogische Klauselform transformiert werden. Für beschreibungslogische Konzeptassertionen ohne Rollenrestriktionen gelten die Transformationen

$$\begin{array}{lcl}
\top(a),\ \sqcap(a) & \Rightarrow & \top \\
\bot(a),\ \sqcup(a) & \Rightarrow & \bot \\
A(a) & \Rightarrow & A(a) \\
\neg A(a) & \Rightarrow & \neg A(a) \\
(C_1 \sqcap \ldots \sqcap C_n)(a) & \Rightarrow & C_1(a) \wedge \ldots \wedge C_n(a) \\
(C_1 \sqcup \ldots \sqcup C_n)(a) & \Rightarrow & C_1(a) \vee \ldots \vee C_n(a).
\end{array}$$

Für die Überführung von Konzeptassertionen mit Rollenrestriktionen in eine endliche prädikatenlogische (duale) Klauselform muss ein Domänenabschluss einbezogen werden. Die folgenden Transformationen beziehen den erweiterten Domänenabschluss $DCA^{+}(KB)$ und die erweiterte Unique-Name-Assumption $UNA^{+}(KB)$ für eine Wissensbasis KB ein:

$$
\begin{array}{lll}
\exists R.C(a) & \Rightarrow_{(7.2.1.1)} & [R(a,c_1) \wedge C(c_1)] \vee ... \vee [R(a,c_m) \wedge C(c_m)] \vee [R(a,\eta_k) \wedge C(\eta_k)] \\
(\geq n\,R)(a) & \Rightarrow_{(7.2.1.1)} & [R(a,c_{i_1}) \wedge ... \wedge R(a,c_{i_n})] \;\vee \\
 & & \quad\vdots \qquad\qquad\qquad \vdots \qquad \vee \\
 & & [R(a,\eta_{k_1}) \wedge ... \wedge R(a,\eta_{k_n})] \\
\forall R.C(a) & \Rightarrow_{DCA^{+}} & [\neg R(a,c_1) \vee C(c_1)] \wedge ... \wedge [\neg R(a,c_m) \vee C(c_m)] \;\wedge \\
 & & [\neg R(a,\eta_1) \vee C(\eta_1)] \wedge ... \wedge [\neg R(a,\eta_p) \vee C(\eta_p)] \\
(\leq n\,R)(a) & \Rightarrow_{DCA^{+}} & [\neg R(a,c_{i_1}) \vee ... \vee \neg R(a,c_{i_{n+1}})] \quad \wedge \\
 & & \quad\vdots \qquad\qquad\qquad\qquad \vdots \qquad \wedge \\
 & & [\neg R(a,\eta_{k_1}) \vee ... \vee \neg R(a,\eta_{k_{n+1}}) \vee \eta_{k_1} \doteq \eta_{k_2} \vee ... \vee \eta_{k_n} \doteq \eta_{k_{n+1}}]
\end{array}
$$

Die Konzeptassertionen $\exists R.C(a)$ und $(\geq n\,R)(a)$ können unter Berücksichtigung der erweiterten Unique-Name-Assumption in die prädikatenlogischen Formeln $(\exists y)\,[R(a,y) \wedge C(y)]$ bzw. $(\exists y_1,...,y_{n+1})\,[R(a,y_1) \wedge ... \wedge R(a,y_{n+1})]$ überführt werden. Aus den Skolemformen zu diesen Formeln, $R(a,new_k) \wedge C(new_k)$ bzw. $R(a,new_{k_1}) \wedge ... \wedge R(a,new_{k_n})$, ergeben sich die oben aufgeführten dualen Klauselformen durch Berücksichtigung der Auslegung (7.2.1.1). Sie können unter Verwendung des Distributivgesetzes in eine Klauselform überführt werden.

Die aus den Konzeptassertionen $\forall R.C(a)$ und $(\leq n\,R)(a)$ resultierenden Konjunkte von Klauseln $\neg R(a,w) \vee C(w)$ bzw. $\neg R(a,w_1) \vee ... \vee \neg R(a,w_{n+1}) \vee w_1 \doteq w_2 \vee ... \vee w_n \doteq w_{n+1}$ für alle Konstanten w einer um neue Konstanten $\eta_1,...,\eta_p$ erweiterten Domäne dagegen gehen aus der Überführung dieser Konzeptassertionen in prädikatenlogische Formeln mit allquantifizierten Implikationen hervor (vgl. [Baader und Nutt, 2003, S. 54]):

$$
\begin{array}{lll}
\forall R.C(a) & \Rightarrow & (\forall y)\,[R(a,y) \rightarrow C(y)] \\
(\leq n\,R)(a) & \Rightarrow & (\forall y_1,...,y_{n+1})\,[R(a,y_1) \wedge ... \wedge R(a,y_{n+1}) \rightarrow \bigvee_{i<j}(y_i \doteq y_j)]
\end{array}
$$

Unter Einbeziehung der erweiterten Unique-Name-Assumption kann weder $c_i \doteq c_j$ noch $c_i \doteq \eta_k$ für einer Wissensbasis bekannte Konstanten c_i, c_j und neue Konstanten η_k gelten. Die Formeln $y_i \doteq y_j$ werden bei der Überführung in eine Klauselform deswegen nur für Paare von neuen Konstanten η_k, η_l berücksichtigt.

Bei den Transformationen von $(\geq n\,R)(a)$ und $(\leq n\,R)(a)$ werden alle Kombinationen von in der verwendeten Wissensbasis vorkommenden Konstanten c_i, $i = 1,...,m$ und neuen Konstanten η_k, $k = 1,...,p$ berücksichtigt. Gibt es weniger als $n+1$ Konstanten in der um $\eta_1,...,\eta_p$ erweiterten Domäne, ist $(\leq n\,R)(a)$ direkt erfüllt und es gilt $(\leq n\,R)(a) \Rightarrow_{DCA^{+}} \top$.[5]

[5] Der Fall, dass ausschließlich die Anzahl neuer Konstanten $\eta_1,...,\eta_p$ kleiner ist als $n+1$ wird durch die hier verwendete Notation für höchstens-Anzahlrestriktionen nicht einbezogen.

In dem Augenblick, in dem die Überführung von Werterestriktionen und höchstens-Anzahlrestriktionen in eine prädikatenlogische Klauselform erfolgt, müssen bereits alle in der verwendeten Wissensbasis *KB* zu berücksichtigenden neuen Konstanten $\eta_1, ..., \eta_p$ bekannt sein. Das ist aber nicht möglich, wenn beispielsweise im Skopus einer in *KB* spezifizierten Werterestriktion eine Existenzrestriktion vorkommt, durch die weitere Konstanten einbezogen werden müssen.

Um diese Problematik zu lösen, kann die in Abschnitt 7.4.1 definierte Menge $NEU(T, v)$ für alle in $\mathcal{S}(KB)$ vorkommenden Rollen T und (Skolem-)Konstanten v bestimmt werden. Bei der Überführung von Konzeptassertionen mit Rollenrestriktionen bzgl. einer Rolle R und einer Konstanten a in eine prädikatenlogische (duale) Klauselform können dann neben allen in *KB* vorkommenden Konstanten $c_1, ..., c_m$ genau die in $NEU(R, a)$ enthaltenen neuen Konstanten $\eta_1, ..., \eta_r$ für die Bildung von Klauseln einbezogen werden:

$$
\begin{array}{lll}
\exists R.C(a) & \Rightarrow_{NEU(R,a)} & [R(a,c_1) \wedge C(c_1)] \vee ... \vee [R(a,c_m) \wedge C(c_m)] \vee \\
& & [R(a,\eta_1) \wedge C(\eta_1)] \vee ... \vee [R(a,\eta_r) \wedge C(\eta_r)] \\
(\geq n\, R)(a) & \Rightarrow_{NEU(R,a)} & [R(a,c_{i_1}) \wedge ... \wedge R(a,c_{i_n})] \vee \\
& & \quad \vdots \qquad\qquad\qquad \vdots \qquad \vee \\
& & [R(a,\eta_{k_1}) \wedge ... \wedge R(a,\eta_{k_n})] \\
\forall R.C(a) & \Rightarrow_{NEU(R,a)} & [\neg R(a,c_1) \vee C(c_1)] \wedge ... \wedge [\neg R(a,c_m) \vee C(c_m)] \wedge \\
& & [\neg R(a,\eta_1) \vee C(\eta_1)] \wedge ... \wedge [\neg R(a,\eta_r) \vee C(\eta_r)] \\
(\leq n\, R)(a) & \Rightarrow_{DCA^+} & [\neg R(a,c_{i_1}) \vee ... \vee \neg R(a,c_{i_{n+1}})] \wedge \\
& & \quad \vdots \qquad\qquad\qquad\qquad \vdots \qquad \wedge \\
& & [\neg R(a,\eta_{k_1}) \vee ... \vee \neg R(a,\eta_{k_{n+1}}) \vee \eta_{k_1} \doteq \eta_{k_2} \vee ... \vee \eta_{k_n} \doteq \eta_{k_{n+1}}]
\end{array}
$$

7.4.3 Konstruktion einer lokalen GCWA in $\mathcal{ALCN}$

Unter Verwendung der Mengen $NEU(T, v)$ sowie der in Abschnitt 7.4.2 vorgeschlagenen Transformation beliebiger $\mathcal{ALCN}$-Konzeptassertionen in eine prädikatenlogische Klauselform kann ein Verfahren formalisiert werden, das korrekt und vollständig entscheidet, ob $KB \models_{GCWA} \alpha$ für eine $\mathcal{ALCN}$-Wissensbasis *KB* und eine $\mathcal{ALCN}$-Anfrage α gilt. Die GCWA kann bei diesem Verfahren zusätzlich auch *lokal* berücksichtigt werden, wenn die Menge $\pi(KB)$ aller atomaren Konzepte A und Rollen R, für die diese Annahme einbezogen werden sollen, gegeben ist.

Definition 7.4.1 *Sei KB eine Wissensbasis, W die Menge aller aus $DCA^+(KB)$ zu berücksichtigenden Konstanten und F ein atomares Konzept A oder eine Rolle R. Dann ist die Folgerungsrelation $\models_{GCWA^\pi}$, die eine lokale GCWA bzgl. $\pi(KB)$ berücksichtigt, definiert durch*

$$KB \models_{GCWA^\pi} \alpha \;\; gdw. \;\; KB \cup \pi(GCWA(KB)) \cup DCA^+(KB) \cup UNA^+(KB) \models \alpha$$

mit $\pi(GCWA(KB)) = \{\neg F(w) \in GCWA(KB) \mid w \in W \text{ und } F \in \pi(KB)\}$.

Algorithmus $AssertGCWALocal_{\mathcal{ALCN}}(KB, \alpha, \pi(KB))$
Input $\mathcal{ALCN}$-Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$, $\mathcal{ALCN}$-Assertion α, Menge $\pi(KB)$
atomarer Konzepte und Rollen, für die die GCWA berücksichtigt werden soll
Output TRUE wenn $KB \models_{GCWA^\pi} \alpha$, FALSE sonst

1. Expandiere $\mathcal{A}$ um $\mathcal{T}$, so dass $KB = (\{\}, \mathcal{A}_\mathcal{T})$
2. Falls sich ein $\mathcal{ALCN}$-Tableau-Beweis für $KB \models_{una} \bot(a)$ für die Konstante a ergibt, gib TRUE aus und stoppe
3. Ermittle die Menge $NEU(T, v)$ zu berücksichtigender neuer Konstanten $\eta_1, ..., \eta_r$ für alle Rollen T und (Skolem-)Konstanten v, die in den vollständig expandierten Zweigen des fehlgeschlagenen Beweises von 2. vorkommen
4. Forme $NNF(KB)$ unter Berücksichtigung von $NEU(T, v)$ in eine prädikatenlogische Klauselform $\mathcal{K}l(KB)$ um
5. Berechne die Menge $\mathcal{R}es(\mathcal{K}l(KB))$, die sich durch das Hinzufügen aller Resolventen aus $\mathcal{K}l(KB)$ zu $\mathcal{K}l(KB)$ ergibt
6. Forme $NNF(\alpha)$ unter Berücksichtigung von $NEU(T, v)$ in eine prädikatenlogische Klauselform $\mathcal{K}l(\alpha)$ um
7. Wenn für alle Klauseln K aus $\mathcal{K}l(\alpha)$ mindestens eine der beiden Bedingungen
   - K oder eine Teilklausel von K ist in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten
   - K enthält ein negatives Literal $\neg A(a)$ bzw. $\neg R(a, b)$ mit $A \in \pi(KB)$ bzw. $R \in \pi(KB)$, so dass alle positiven Klauseln aus $\mathcal{R}es(\mathcal{K}l(KB))$, die $A(a)$ bzw. $R(a, b)$ enthalten, auch ohne $A(a)$ bzw. $R(a, b)$ in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten sind

   erfüllt ist, gib TRUE aus und stoppe. Anderenfalls gib FALSE aus und stoppe

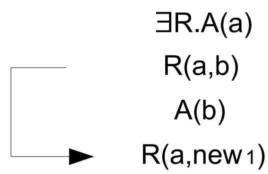
Abbildung 7.2: der *Der* $AssertGCWALocal_{\mathcal{ALCN}}$*-Algorithmus*

Das Verfahren ist in Abbildung 2 dargestellt. Zunächst wird die bzgl. $\mathcal{T}$ eliminierte Wissensbasis $KB = (\{\}, \mathcal{A}_\mathcal{T})$ mit dem $\mathcal{ALCN}$-Tableau-Kalkül auf Erfüllbarkeit überprüft: Wenn sich dabei ein Abschluss ergibt, ist KB unerfüllbar, und $KB \models_{GCWA^\pi} \alpha$ gilt trivialerweise (2.). Ergibt sich kein Abschluss, wurden alle Zweige $\Sigma_1, ..., \Sigma_r$ vollständig expandiert. Aus der Menge $\mathcal{S}(KB)$ aller in diesen Zweigen vorkommenden Assertionen können die Mengen $NEU(T, v)$ für alle Rollen T und (Skolem-)Konstanten v erstellt werden (3.). Unter Berücksichtigung dieser Mengen können alle in Negationsnormalform umgeformte Konzept- und Rollenassertionen aus KB in eine prädikatenlogische Klauselform transformiert werden (4.).[6] Für die resultierende Klauselmenge $\mathcal{K}l(KB)$ müssen anschließend alle subsummierenden Klauseln bestimmt werden (5.). Die Menge $\mathcal{R}es(\mathcal{K}l(KB))$, die sich durch das Hinzufügen aller dieser Resolventen ergibt, enthält alle Klauseln, die aus KB unter Berücksichtigung aller $NEU(T, v)$ logisch folgen.

[6] In KB enthaltende Rollenassertionen $R(a, b)$ werden dabei direkt in die Klausel $R(a, b)$ überführt.

Die Überführung von $NNF(\alpha)$ in die Klauselform $\mathcal{K}l(\alpha)$ (6.) erfolgt analog zu der Überführung von Assertionen aus KB in eine Klauselform, so dass keine weiteren η_k eingeführt werden. Ein Assertionstest $KB \models_{GCWA^\pi} \alpha$ ist erfolgreich, wenn für jede Klausel K aus $\mathcal{K}l(\alpha)$ K selbst oder eine Teilklausel von K in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten ist oder wenn die Bedingung erfüllt ist, dass K negative Literale $\neg A(a)$ bzw. $\neg R(a,b)$ mit $A \in \pi(KB)$ bzw. $R \in \pi(KB)$ enthält und alle positiven Klauseln aus $\mathcal{R}es(\mathcal{K}l(KB))$, die $A(a)$ bzw. $R(a,b)$ enthalten, auch ohne $A(a)$ bzw. $R(a,b)$ in $\mathcal{R}es(\mathcal{K}l(KB))$ vorkommen.[7] Neben in $NNF(\alpha)$ vorkommenden negierten atomaren Assertionen $\neg A(a), \neg R(a,b)$ ergeben sich diese negativen Literale durch Überführung von $\forall R.C(a)$ und $(\geq n\, R)(a)$ in eine Klauselform (siehe Abschnitt 7.4.2).

Beispiel 7.4.2 Gegeben sei die Wissensbasis $KB = \{\exists R.A(a), R(a,b), A(b)\}$ und es bestehe Interesse daran, ob $KB \models_{GCWA^\pi} \forall R.A(a)$ für $\pi(KB) = \{A, R\}$. Zunächst wird geprüft, ob $KB \models_{GCWA^\pi} \bot(a)$. Dabei hat die Tatsache, dass a in KB vorkommt, keine Auswirkungen. Der vollständig expandierte Zweig



enthält keinen Abschluss. Mit Ausnahme der Menge $NEU(R,a) = \{\eta_1\}$ gilt für alle Mengen $NEU(T,v) = \{\}$, so dass $DCA^+(KB) = (\forall x)[x \doteq a \vee x \doteq b \vee x \doteq \eta_1]$. Die Transformationen

$$\begin{array}{lll}
\exists R.A(a) & \Rightarrow_{NEU(R,a)} & [R(a,a) \wedge A(a)] \vee [R(a,b) \wedge A(b)] \vee [R(a,\eta_1) \wedge A(\eta_{n1})] \\
R(a,b) & \Rightarrow_{NEU(R,a)} & R(a,b) \\
A(b) & \Rightarrow_{NEU(R,a)} & A(b)
\end{array}$$

führen nach Umformung der durch $\exists R.A(a)$ zu berücksichtigenden dualen Klauselform in eine Klauselform zu der Klauselmenge

$$\begin{array}{rlll}
\mathcal{K}l(KB) = & \{R(a,a) \vee R(a,b) \vee R(a,\eta_1), & R(a,a) \vee R(a,b) \vee A(\eta_1), & R(a,a) \vee R(a,\eta_1) \vee A(b), \\
& R(a,b) \vee R(a,\eta_1) \vee A(a), & R(a,a) \vee A(b) \vee A(\eta_1), & R(a,b) \vee A(a) \vee A(\eta_1), \\
& R(a,\eta_1) \vee A(a) \vee A(b), & A(a) \vee A(b) \vee A(\eta_1), & R(a,b), \quad A(b)\}.
\end{array}$$

Es ergeben sich keine Resolventen, so dass $\mathcal{R}es(\mathcal{K}l(KB)) = \mathcal{K}l(KB)$. Die Anfrageassertion $\forall R.A(a)$ kann in $\mathcal{K}l(\alpha) = \{\neg R(a,a) \vee A(a), \neg R(a,b) \vee A(b), \neg R(a,\eta_1) \vee A(\eta_1)\}$ überführt werden. Die zweite Klausel aus $\mathcal{K}l(\alpha)$ folgt unter $\models_{GCWA^\pi}$ aus KB, da die Teilklausel $A(b)$ in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten ist, und die erste und dritte Klausel, da in $\mathcal{R}es(\mathcal{K}l(KB))$ für jede positive Klausel mit $R(a,a)$ oder $R(a,\eta_1)$ ebenfalls die Teilklausel $R(a,b)$ oder $A(b)$ enthalten ist. Es ergibt sich $KB \models_{GCWA^\pi} \forall R.A(a)$. Das Ergebnis kann einfacher nachvollzogen werden, wenn berücksichtigt wird, dass $\exists R.A(a)$ unter der GCWA redundant ist (Abschnitt 7.2.2).

[7] Das entspricht der Definition der GCWA (siehe Abschnitt 4.3.1)

Beispiel 7.4.3 Es soll geprüft werden, ob $KB \models_{GCWA^\pi} \neg A(a)$ mit $KB = \{\forall R.A(a), \exists R.\top(a)\}$ und $\pi(KB) = \{A, R\}$. Der aus KB resultierende Zweig wird nach Anwendung der $\exists$-Regel um $R(a, new_1)$ erweitert und ist vollständig expandiert, so dass $NEU(R, a) = \{\eta_1\}$. Die prädikatenlogische Klauselmenge von KB ergibt sich unter Berücksichtigung dieser Menge zu

$$\mathcal{K}l(KB) \;=\; \{\neg R(a,a) \vee A(a),\ \neg R(a,\eta_1) \vee A(\eta_1),\ R(a,a) \vee R(a,\eta_1)\}$$

Es können die Resolventen $A(a) \vee R(a,\eta_1)$ und $R(a,a) \vee A(\eta_1)$ gebildet werden. Der oben angegebene Assertionstest ist nicht erfolgreich, da in $\mathcal{R}es(\mathcal{K}l(KB))$, der Erweiterung von $\mathcal{K}l(KB)$ um alle Resolventen, die Klausel $A(a) \vee R(a,\eta_1)$ enthalten ist, aber nicht die Klausel $R(a,\eta_1)$.

Beispiel 7.4.4 Gegeben sei $KB = \{R(a,a), P(a,b)\}$ mit $\pi(KB) = \{R\}$, und es soll überprüft werden, ob $KB \models_{GCWA^\pi} \exists R.(\leq 1R)(a)$. Durch die vollständige Expansion des sich aus KB ergebenen Zweiges werden keine Skolem-Konstanten eingeführt, so dass sich $NEU(T, v) = \{\}$ für alle in $\mathcal{S}(KB)$ enthaltenen Rollen T und Konstanten v ergibt. Es gilt $\mathcal{R}es(\mathcal{K}l(KB)) = \mathcal{K}l(KB) = KB$. Die Überführung der Anfrage $\exists R.(\leq 1\, R)(a)$ in eine duale Klauselform ist

$$\exists R.(\leq 1R)(a) \;\Rightarrow\; [R(a,a) \wedge (\neg P(a,a) \vee \neg P(a,b))] \;\vee\; [R(a,b) \wedge (\neg P(b,a) \vee \neg P(b,b))],$$

und nach Umformung in eine Klauselform ergibt sich die Klauselmenge

$$\begin{aligned}\mathcal{K}l(\alpha) \;=\; \{&R(a,a) \vee R(a,b),\ \ R(a,b) \vee \neg P(a,a) \vee \neg P(a,b),\ \ R(a,a) \vee \neg P(b,a) \vee \neg P(b,b),\\ &\neg P(a,a) \vee \neg P(a,b) \vee \neg P(b,a) \vee \neg P(b,b)\}.\end{aligned}$$

Die erste und die dritte der vier Klauseln aus $\mathcal{K}l(\alpha)$ folgen unter $\models_{GCWA^\pi}$ aus KB, da die Teilklausel $R(a,a)$ in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten ist. Für die anderen Klauseln gilt das nicht, da P nicht in $\pi(KB)$ enthalten ist und deswegen nicht angenommen werden kann, dass negierte Literale mit P unter $\models_{GCWA^\pi}$ aus KB folgen und – bezogen auf die zweite Klausel – zusätzlich $R(a,b)$ nicht in $\mathcal{R}es(\mathcal{K}l(KB))$ enthalten ist.

Der Assertionstest $KB \models_{GCWA^\pi} \exists R.(\leq 2\, P)(a)$ dagegen ist erfolgreich: Die Überführung der Anfrage in eine Klauselform ist in diesem Fall

$$\begin{aligned}\exists R.(\leq 2\, P)(a) \quad\Rightarrow\quad &[R(a,a) \wedge \top] \vee [R(a,b) \wedge \top] \;\equiv\\ &[R(a,a) \vee R(a,b)] \wedge [R(a,a) \vee \top] \wedge [R(a,b) \vee \top] \wedge [\top],\end{aligned}$$

da bei der Transformation von $(\leq 2\, P)(a)$ und $(\leq 2\, P)(b)$ in eine Klauselform keine drei unterschiedlichen Konstanten berücksichtigt werden können, so dass diese Konzeptassertionen direkt erfüllt sind.[8] Alle resultierenden Klauseln, die $\top$ enthalten, folgen direkt unter $\models_{GCWA^\pi}$. Die Klausel $R(a,a) \vee R(a,b)$ folgt ebenfalls unter dieser Folgerungsrelation, da $\mathcal{R}es(\mathcal{K}l(KB))$ die Teilklausel $R(a,a)$ enthält.

[8] Im beschreibungslogischen Kontext gilt unter Berücksichtigung von $DCA^+(KB)$ die Äquivalenz $\exists R.(\leq 2\, P)(a) \;\equiv\; \exists R.\top(a)$.

Die Klauselformen, die sich durch Verwendung von Mengen $NEU(R,a)$ ergeben, führen zu denselben Inferenzen, die sich für Existenzrestriktionen und mindestens-Anzahlrestriktionen durch die Auslegung (7.2.1.1) sowie für Werterestriktionen und höchstens-Anzahlrestriktionen unter der Berücksichtigung von $DCA^{+}(KB)$ ergeben: Für Konzeptassertionen $\exists R.C(a)$ und $(\geq n\, R)(a)$ ist das der Fall, weil die in $NEU(R,a)$ enthaltenen neuen Konstanten, die durch die Auslegung (7.2.1.1) nicht berücksichtigt werden, für alle Rollenrestriktionen einbezogen werden, unabhängig davon, ob die entsprechende Klauselform aus der Wissensbasis oder der Anfrage resultiert. Für Konzeptassertionen $\forall R.C(a)$ und $(\leq n\, R)(a)$ können an Stelle aller durch $DCA^{+}(KB)$ zu berücksichtigender neuer Konstanten $\eta_1, ..., \eta_p$ die neuen Konstanten der Menge $NEU(R,a)$, $\eta 1, ..., \eta_r$, verwendet werden, da für alle neuen Konstanten η_k, die nicht in $NEU(R,a)$ enthalten sind, $KB \models_{GCWA} \neg R(a,\eta_k)$ gilt, so dass aus diesen Konzeptassertionen resultierende Klauseln $\neg R(a,\eta_k) \vee C(\eta_k)$ bzw. $\neg R(a,w_1) \vee ... \vee \neg R(a,\eta_k) \vee ... \vee \neg R(a,w_{n+1})$ in jedem Fall erfüllt sind und keine Auswirkungen auf Inferenzen haben. Da die Überführung von $\mathcal{ALCN}$-Konzeptassertionen in (quantifizierte) prädikatenlogische Formeln [Baader und Nutt, 2003, S. 54] durch die Interpretation aller Konzeptkonstruktoren aus $\mathcal{ALCN}$ definiert ist, ist der $AssertGCWALocal_{\mathcal{ALCN}}$-Algorithmus unter Berücksichtigung der Definition der GCWA (Abschnitt 4.3.1) und der Auslegung (7.2.1.1) korrekt.

Der Algorithmus ist allerdings äußerst komplex: Sowohl die Eliminierung der TBox als auch die Überprüfung einer $\mathcal{ALCN}$-Wissensbasis auf Erfüllbarkeit führen zu exponentiellen Berechnungszeiten. Klauselmengen, die sich aus der Überführung von Konzeptassertionen mit qualifizierten Existenzrestriktionen, $\exists R.C(a)$, ergeben, erfordern hochgradig exponentiellen Speicherplatz, da die Disjunktion von Konjunktionen $R(a,w) \wedge C(w)$ für alle möglichen Identifikationen einer einzubeziehenden Skolem-Konstante mit in KB vorkommenden oder in $NEU(R,a)$ enthaltenen Konstanten w berücksichtigt werden muss und die Konzeptbeschreibung C weitere qualifizierte (Existenz-)Restriktionen beinhalten kann. Unabhängig davon können sich exponentiell viele Resolventen aus der prädikatenlogischen Klauselmenge $\mathcal{K}l(KB)$ ergeben.

Kapitel 8

Schlussbetrachtungen und Ausblick

In dieser Arbeit wurde die Einbeziehung von Annahmen einer geschlossenen Welt in beschreibungslogischen Sprachen untersucht, um unerwünschte Fehlschläge beschreibungslogischer Assertionstests mit Werterestriktionen, höchstens-Anzahlrestriktionen und negierten atomaren Assertionen zu vermeiden. Es wurde gezeigt, dass die CWA nach [Reiter, 1978] durch die Spezifikation von Konzeptdisjunktionen, Existenzrestriktionen und mindestens-Anzahlrestriktionen in vielen Sprachen zu Inkonsistenzen führt und dass die Einbeziehung generalisierter Annahmen aufgrund zu berücksichtigender Klauseln in beschreibungslogischen Sprachen nicht direkt erfolgen kann.

Es wurde dargestellt, dass durch die Ergänzung negierter atomarer Assertionen und Werterestriktionsassertionen um einen epistemischen Operator eine lokale CWA berücksichtigt werden kann, ohne zu Inkonsistenzen zu führen. Anschließend wurde gezeigt, dass in der new Racer Query Language (nRQL) durch Verwendung eines Negation-as-failure-Operators viele dieser epistemischen Assertionstests formuliert werden können.

Die in Kapitel 7 vorgestellten Verfahren erweitern einen Assertionstest um Annahmen einer geschlossenen Welt in ausgewählten beschreibungslogischen Sprachen. Die Algorithmen für atomare Wissensbasen und $\mathcal{AL}_0$-Wissensbasen berücksichtigen die CWA nach [Reiter, 1978], den Domänenabschluss und die Unique-Name-Assumption und führen zu einer vollständigen Lösung der in dieser Arbeit untersuchten Problematik. Für die Sprachen $\mathcal{AL}$ und $\mathcal{ALN}$ wurde ein polynomielles Verfahren unter Berücksichtigung einer GCWA vorgestellt, dass allerdings nur unter Einschränkungen der Wissensbasis vollständig ist. Zuletzt wurde in den Sprachen $\mathcal{ALC}$ und $\mathcal{ALCN}$ eine lokale GCWA durch Überführung von $\mathcal{ALCN}$-Konzeptassertionen in eine prädikatenlogische Klauselform berücksichtigt.

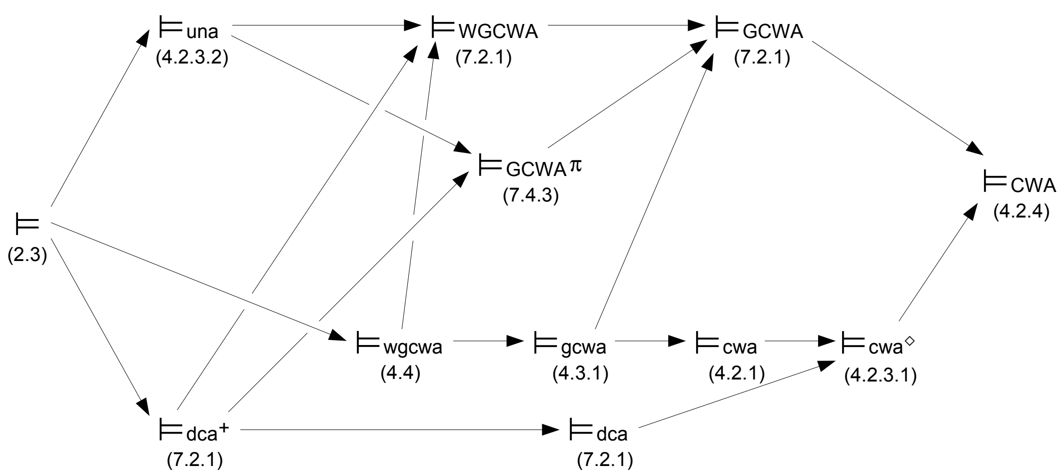
Das letztgenannte Verfahren kann ebenfalls für alle in dieser Arbeit untersuchten Teilsprachen von $\mathcal{ALCN}$ verwendet werden. Wie bereits geschildert, ist die Überführung von Konzeptassertionen in eine prädikatenlogische (duale) Klauselform aufwändig. Allerdings erfolgt diese Überführung unter einer bestimmten Systematik. Die Klauselform der Konzeptassertion $\exists R.A(a)$ aus Beispiel 7.4.2 beispielsweise ergibt sich unter Berücksichtigung von

$DCA^+(KB) = (\forall x)\,[x \doteq a \vee x \doteq b \vee x \doteq \eta_1]$ und enthält Klauseln aller möglichen dreistelligen Kombinationen von Literalen mit unterschiedlichen Konstanten. Eine ähnliche Systematik tritt bei der Überführung von mindestens-Anzahlrestriktionen in eine Klauselform auf. Das in Abschnitt 7.4.3 vorgestellte Verfahren kann deswegen möglicherweise optimiert werden.

Ein weiterer Ansatz für die Berücksichtigung von Konzeptabschlüssen ist die Verwendung des *one-of*-Operators $\{a_1, ..., a_n\}$. In [Schaerf, 1994a] wird gezeigt, dass Konzeptbeschreibungen $\mathbf{K}C$ bzgl. einer Wissensbasis KB äquivalent zu dem Konzept $\{a_1, ..., a_n\}$ sind, wenn $a_1, ..., a_n$ genau die Individuen sind, für die $KB \models C(a_i)$ gilt. Ein Tableau-Kalkül, mit dem Assertionstests in Sprachen mit dem one-of-Operator auf Erfolg überprüft werden können, wird in [Schaerf, 1994b] vorgestellt.

Anhang A

Verwendete Folgerungsrelationen



Die Darstellung enthält die logische Folgerungsrelation und alle in dieser Untersuchung verwendeten vervollständigenden Folgerungsrelationen. Gerichtete Kanten drücken Teilmengenbeziehungen aus. Für jede Relation ist in Klammern der Abschnitt vermerkt, an der sie eingeführt wurde.

Literaturverzeichnis

[Abiteboul et al., 1995] Serge Abiteboul, Richard Hull und Victor Vianu. Foundations of Databases. Addison-Wesley, 1995.

[Baader, 1999] Franz Baader. Logic-Based Knowledge Representation. In M. J. Wooldridge and M. Veloso, editors, Artificial Intelligence Today, Recent Trends and Developments. Springer-Verlag, 1999. http://citeseer.ist.psu.edu/110067.html

[Baader und Hollunder, 1991] Franz Baader und Bernhard Hollunder. A Terminological Knowledge Representation System with Complete Inference Algorithms. In Proc. of the Workshop on Processing Declarative Knowledge, PDK-91, Lecture Notes in Artificial Intelligence, pages 67–86, Springer-Verlag, 1991.
http://citeseer.ist.psu.edu/baader91terminological.html

[Baader und Nutt, 2003] Franz Baader und Werner Nutt. Basic description logics. In F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi und P. Patel-Schneider (eds.) The Description Logic Handbook. Theory, Implementation and Application, pages 47–100, 2003.

[Baader und Sattler, 2000] Franz Baader und Ulrike Sattler. An Overwiev of Tableau Algorithms for Description Logics. Studia Logica, 69(1), pages 5–40, October 2001. http://citeseer.ist.psu.edu/baader00overview.html

[Brachman und Levesque, 2004] Ronald J. Brachman und Hector J. Levesque. Knowledge Representation and Reasoning. San Francisco, CA: Morgan Kaufmann, 2004.

[Brass, 1989] Stefan Brass. Vervollständigung von Logikdatenbanken. Forschungsbericht 315, Fachbereich Informatik, Universität Dortmund, Dortmund, 1989.
http://www-b.informatik.uni-hannover.de/ftp/papers/1989/Bra89a.pdf

[Brass und Lipeck, 1989] Stefan Brass und Udo W. Lipeck. Specifying Closed World Assumptions for Logic Databases (1989), 2nd Symp. on Mathematical Fundamentals of Database Syst. (MFDBS'89) http://citeseer.ist.psu.edu/brass89specifying.html

[Cadoli et al., 1990] Marco Cadoli, Francesco M. Donini und Marco Schaerf. Closed world reasoning in hybrid systems. In Proceedings of the 5th International Symposium on

Methodologies for Intelligent Systems (ISMIS-90), pages 474–481. North Holland, 1990. http://www.dis.uniroma1.it/~ai/citations/CDS90.html

[Clark, 1978] Keith L. Clark. Negation as failure. In H. Gallaire & J. Minker (eds.) Logic and Data Bases, pages 293–322. New York: Plenum, 1978. http://www.doc.ic.ac.uk/ klc/neg.html

[Donini et al., 1992] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt und Andrea Schaerf. Adding Epistemic Operators to Concept Languages. Principles of Knowledge Represenation and Reasoning, pages 342–353, Cambridge, 1992. http://citeseer.ist.psu.edu/donini92adding.html

[Donini et al., 1994] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi und Andrea Schaerf. Deduction in concept languages: From subsumption to instance checking. J. Of Logic and Computation, 4(4): pages 423–452, 1994. http://citeseer.ist.psu.edu/donini94deduction.html

[Donini et al., 1998] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt und Andrea Schaerf. An Epistemic Operator for Description Logics. Artificial Intelligence, 100(1–2):225–274, 1998. http://www.macs.hw.ac.uk/~nutt/Publications/alck98.ps.gz

[Donini et al., 2002] Francesco M. Donini, Daniele Nardi und Riccardo Rosati. Description logics of minimal knowledge and Negation as Failure. ACM Transactions on Computational Logic, 3(2), pages 177–225, April 2002. http://tocl.acm.org/accepted/donini.ps

[Drummond und Shearer, 2006] Nick Drummond und Rob Shearer. The Open World Assumption. University of Manchester, 2006. http://www.cs.man.ac.uk/~drummond/presentations/OWA.pdf

[Eiter und Gottlob, 1993] Thomas Eiter und Georg Gottlob. Propositional Circumscription and Extended Closed World Reasoning are π_2^p-complete. Theoretical Computer Science 114, pages 231–245, 1993. http://citeseer.comp.nus.edu.sg/571846.html

[Fernández und Minker, 1992] José Alberto Fernández und Jack Minker. Disjunctive Deductive Databases. Proceedings of the Logic Programming and Automated Reasoning Conference, 1992. http://prism.cs.umd.edu/papers/FM92:dddbs/FM:LPAR92.html

[Garcia und Gil, 2006] Roberto García und Rosa Gil. An OWL Copyright Ontology for Semantic Digital Rights Management. OTM Workshops (2) 2006: pages 1745–1754 http://rhizomik.net/content/roberto/papers/rgrg-swws06.pdf

[Glimm und Horrocks, 2004] Birte Glimm und Ian Horrocks. Query Answering Systems in the Semantic Web. In Proc. of the KI-04 Workshop on Appl. of Description Logics, 2004. http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-115/03-glimm.pdf

[Grant und Minker, 1986] John Grant und Jack Minker. Answering queries in indefinite databases and the null value problem. In P. Kanellakis, editor, Advances in computing research Volume 3, pages 247–267, JAI Press, 1986.

[Grimm und Motik, 2005] Stephan Grimm und Boris Motik. Closed-World Reasoning in the Semantic Web through Epistemic Operators. In CEUR Proceedings of the OWL Experiences and Directions Workshop, Galway, Ireland, 2005.
http://dip.semanticweb.org/documents/Grimm-Motik-Closed-World-Reasoning-in-the-Semantic-Web-through-Epistemic-Operators.pdf

[Haarslev et al., 2001] Volker Haarslev, Ralf Möller und Michael Wessel. The description logic $\mathcal{ALCNH}_{R^+}$ extended with concrete domains: A practically motivated approach. In R. Goré, A. Leitsch, and T. Nipkow, editors, Automated Reasoning. First International Joint Conference (IJCAR'01), Siena, Italy, June 18–23, 2001, Proceedings, volume 2083 of Lecture Notes in Artificial Intelligence, pages 29–44, Berlin, 2001. Springer-Verlag. http://www.sts.tu-harburg.de/%7Er.f.moeller/papers/2001/HaMo01b.pdf

[Haarslev et al., 2004b] Volker Haarslev, Ralf Möller und Michael Wessel. Querying the Semantic Web with Racer + nRQL. In Proc. of the KI-2004 Intl. Workshop on Applications of Description Logics (ADL'04), 2004. http://citeseer.ist.psu.edu/727837.html

[Henschen und Park, 1988] Lawrence J. Henschen und Hyung-Sik Park. Compiling the GCWA in Indefinite Databases. In J. Minker, editor Foundations of Deductive Databases and Logic Programming, pages 395–438. Morgan Kaufmann Pub., Washington, D.C., 1988.

[Horrocks et al., 2000] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In David MacAllester, editor, Proceedings of the 17th International Conference on Automated Deduction (CADE-17), number 1831 in Lecture Notes in Computer Science, pages 482–496, Germany, 2000. Springer Verlag. http://citeseer.ist.psu.edu/273266.html

[Kaplunova et al., 2007] Alissa Kaplunova, Ralf Möller und Michael Wessel. Leveraging the Expressivity of Grounded Conjunctive Query Languages. In Proc. International Workshop on Scalable Semantic Web Systems, 2007.
http://www.sts.tu-harburg.de/~r.f.moeller/papers/2007/KaMW-07.pdf

[Levesque, 1984] Hector J. Levesque. Foundations of a functional approach to knowledge representation. Artificial Intelligence 23, pages 155–212, 1984.

[Minker, 1982] Jack Minker. On indefinite databases and the Closed World Assumption. In: Lecture Notes in Computer Science 138 (Springer, 1982) pages 292–308.

[Möller and Haarslev, 2003] Ralf Möller and Volker Haarslev. Description logics systems. In F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P. Patel-Schneider (eds.) (pages 289-312), 2003.

[Racer Systems, 2007] Racer Systems GmbH & Co. KG. RacerPro User's Guide Version 1.9.2. Technical report, Racer Systems GmbH & Co. KG, Oktober 2007. http://www.racer-systems.com/products/racerpro/users-guide-1-9-2-beta.pdf

[Rajasekar et al., 1989] Arcot Rajasekar, Jorge Lobo und Jack Minker: Weak Generalized Closed World Assumption. J. Autom. Reasoning Volume 5, Issue 3: pages 293–307, 1989. http://www.springerlink.com/index/V52747U262464N96.pdf

[Reiter, 1978] Raymond Reiter. On closed world data bases. In H. Gallaire & J. Minker (eds.) Logic and Data Bases, pages 55–76. New York: Plenum, 1978.

[Reiter, 1984] Raymond Reiter. Towards a logical reconstruction of relational database theory. In M. Brodie, J. Mylopoulos & J. Schmidt (eds.), On Conceptual Modelling, pages 191–233. New York: Springer, 1984.

[Reiter, 1990] Raymond Reiter. On asking what a database knows. In J. W. Lloyd, editor, Symposium on computational logics, pages 96–113, Springer Verlag, 1990.

[Schaerf, 1994a] Andrea Schaerf. Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity and Semantic Issues. PhD thesis, Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza" 1994. http://citeseer.ist.psu.edu/schaerf94query.html

[Schaerf, 1994b] Andrea Schaerf. Reasoning with individuals in concept languages. Data and Knowledge Engineering, 13(2): pages 141–176, 1994. http://citeseer.ist.psu.edu/schaerf94reasoning.html

[Schmidt-Schauß und Smolka, 1991] Manfred Schmidt-Schauß und Gert Smolka. Attributive concept descriptions with complements. Artificial Intelligence, 48(1): pages 1–26, 1991.

[Van Emden und Kowalski, 1976] Maarten H. Van Emden und Robert A. Kowalski. The semantics of predicate logic as a programming language. Journal of the ACM, 23(4), pages 733–742, 1976. http://www.doc.ic.ac.uk/˜rak/papers/kowalski-van_emden.pdf

[Wessel und Möller, 2005] Michael Wessel und Ralf Möller. A High Performance Semantic Web Query Answering Engine. In I. Horrocks, U. Sattler and F. Wolter (eds.) Proc. of the 2005 Description Logic Workshop, 2005. http://citeseer.ist.psu.edu/730034.html

Erklärung

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

Hamburg, den 15. Mai 2008

Oliver Gries