Fakultät Informatik
Institut für Theoretische Informatik
Lehrstuhl für Automatentheorie

Diplomarbeit

**zur Erlangung des akademischen Grades
Diplom-Informatiker**

Untersuchung von konjunktiven Abfragen über verteilte Ontologien mit dem Fokus auf $\mathcal{E}$-Connections

Jörg Eichhorn

März 2009

betreut durch
Prof. Dr.-Ing. Franz Baader
Dipl.-Ing. Martin Knechtel

in Kooperation mit
SAP Research, CEC Dresden

Technische Universität Dresden/Fakultät Informatik

Aufgabenstellung für die Diplomarbeit

Name, Vorname:	Eichhorn, Jörg		
Studiengang:	Informatik	Matr.Nr.:	2943374

Thema: Untersuchung von konjunktiven Anfragen über verteilte Ontologien mit dem Fokus auf E-Connections.

Zielstellung:

Zunächst soll ein Überblick über die in der Literatur vorgestellten Ansätze zur Verteilung von Ontologien erarbeitet werden. Für E-Connections soll ein modulares Verfahren zur Beantwortung von konjunktiven Anfragen (conjunctive queries, CQ) erarbeitet werden, welches die Beantwortung einer CQ in einer E-Connection auf die Beantwortung von CQs in den Komponentenlogiken reduziert. Um dieses Verfahren zu validieren, soll ein Prototyp entwickelt und an Anwendungsbeispielen evaluiert werden.

Betreuer:	Dipl.-Inf. Martin Knechtel
Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. Franz Baader
Institut:	Theoretische Informatik
Beginn am:	13.11.2008
Einzureichen am:	13.05.2009

Dresden, 13.11.2008

Prof. Dr.-Ing. Franz Baader
Technische Universität Dresden
Fakultät Informatik
Institut für Theoretische Informatik
01062 Dresden

Unterschrift des verantwortlichen Hochschullehrers

Verteiler: 1x Prüfungsamt, 1x HSL, 1x Betreuer, 1x Student

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen **5**
2.1 Beschreibungslogiken 5
2.2 Konjunktive Abfragen 10
2.3 Konjunktive Abfragen in Beschreibungslogiken 12
2.3.1 Logische Konsequenz variablenfreier Abfragen 13
2.3.2 Boolesche Abfragen mit Variablen 16
2.3.3 Beantwortung von Abfragen 19

3 Ansätze für modulare Wissensbasen **23**
3.1 Modularisierung per Definition 23
3.1.1 $\mathcal{E}$-Connections 24
3.1.2 Distributed Description Logics 31
3.1.3 Package-based Description Logics 33
3.2 Modularisierung durch Analyse 35
3.3 Abgrenzung 36

4 Verwandte Arbeiten **39**

5 Konjunktive Abfragen auf modularen Ontologien **43**
5.1 $\mathcal{E}$-Connections 43
5.1.1 Variablenfreie Boolesche Abfragen 46
5.1.2 Boolesche Abfragen mit Variablen 49
5.1.3 Beantwortung konjunktiver Abfragen 60
5.1.4 Erweiterungen 63
5.1.5 Verteilung des Verfahrens 64
5.1.6 Implementierungskonzept 64
5.2 Distributed Description Logics 66
5.3 Analytische Modularisierung 67

6 Zusammenfassung **69**

Literaturverzeichnis **73**

Abbildungs- und Algorithmenverzeichnis **79**

Abkürzungsverzeichnis **80**

Selbstständigkeitserklärung **82**

1 Einleitung

Der Begriff Ontologie wird übersetzt mit „Lehre vom Sein“[1] und meint damit ursprünglich eine Disziplin der Philosophie, welche sich mit der grundlegenden Struktur des Seins und der Wirklichkeit auseinandersetzt [Bro07]. Nur vage angelehnt an die philosophische Disziplin wird später der Begriff Ontologie auch als Menge von explizit formalisierten Aussagen über das Existierende, speziell über einen Teilbereich einer Realität verstanden. Davon ausgehend übernimmt die Fachrichtung „Künstliche Intelligenz“ den Begriff als Synonym für eine Menge logischer Formeln, welche der Wissensrepräsentation dienen (z.B. [NFF+91]), und lässt damit ein weiteres Mal die klassische Philosophie auf das digitale Zeitalter Einfluss nehmen. Diese Interpretation von Ontologie liegt daher auch der vorliegenden Arbeit zugrunde und wird auf ihre Potenziale zur Anwendung in verteilten Informationssystemen untersucht.

Als das Internet zum festen Bestandteil des modernen Lebens und Arbeiten wurde und erneut die Frage aufkam, in welche Richtung es sich fortentwickeln würde, führte der Artikel „The Semantic Web“ [BLHL01] von Tim Berners-Lee und der damit geprägte Begriff „Semantic Web“ zur Beschleunigung einer Bewegung rund um formale Wissensrepräsentation und ihrer Anwendung in realen Szenarien. Mit der Web Ontology Language (OWL) hat sich dazu inzwischen ein umfassend anerkannter Standard auf Basis der gut verstandenen Beschreibungslogiken etabliert, welcher zudem in mehr und mehr Anwendungen zum Einsatz kommt [BCM+03, FHH+07]. Diese sind jedoch noch weitestgehend abgeschlossene Systeme, welche eine Vernetzung im Sinne des „Semantic Webs“ vermissen lassen. Das wesentlichste Problem ist dabei eher technischer Natur: Die Wiederverwendung von Wissen und die dazu in OWL vorgesehene Import-Relation lässt einen wesentlichen Vorteil von Beschreibungslogiken – die automatische Schlussfolgerung von implizitem Wissen – zur unkalkulierbaren Größe werden, da die komplette transitive Hülle vollständig lokal geladen und verarbeitet werden müsste. Für die Realisierung großer Informationssysteme, von denen man zudem annehmen kann, dass sie auf einer verteilten Architektur aufbauen, wünscht man sich eine geeignetere Möglichkeit, Wissensbasen miteinander in Beziehung zu setzen [WBHQ07].

[1] Aus dem Griechischen, *on* als Partizip zu *einai* - „sein“ und aus *logos* - „Lehre“ bzw. „Wort“.

Derart verteilte Ontologien erlaubten es, neue Formen der automatischen Kooperation zwischen Partnern zu etablieren, welche aufgrund hoher Aktualität der Daten, deren Interoperabilität und inhaltlichen Verknüpfungen untereinander die gemeinschaftlichen Geschäftsprozesse optimieren würden. Ebenso vorstellbar ist ein virtueller Marktplatz von Produkten, der wegen des Logik-basierten Fundaments nicht nur mächtige Suchen ermöglichen, sondern darüber hinaus Mehrwert aus zunächst nur impliziten Informationen ziehen könnte. Setzte sich das Wissen innerhalb dieses Marktplatzes nun direkt aus Wissensbasen unterschiedlicher Kooperationspartner zusammen und wären diese zudem noch auf unterschiedlichen Hosts gelagert, würde dies der Essenz des Begriffs Semantic Web bereits sehr nahe kommen.

Die Verwirklichungen solcher Visionen besitzen ein hohes wirtschaftliches Potenzial, dessen sich große IT-Unternehmen ebenso wie private und staatliche Forschungsinstitutionen bewusst sind. Dies führt zu einer intensiven Auseinandersetzung mit diesem Forschungsfeld, wie beispielsweise das vom deutschen Bundesministerium für Wirtschaft und Technologie[2] initiierte THESEUS-Programm[3] zeigt, in dessen Rahmen auch das Thema zu dieser Arbeit entstand.

Diese Arbeit diskutiert die in den den letzten Jahren vorgeschlagenen Ansätze, welche das existierende theoretische Fundament in einen neuen, modularen Kontext versetzen. Dabei stehen Ansätze mit rein syntaktisch-technischer Verteilung jenen mit solidem theoretischen Fundament gegenüber. Ersteren mangelt es dabei an der Fähigkeit, den verbundenen Wissensbasen eine globale Semantik zuzuweisen und in Folge dessen belegbare Aussagen über die Verbindung zu machen. Stattdessen legen sie den Schwerpunkt auf die bloße Abfrage eines losen Verbunds von Wissensbasen. Dies ist nur eine unbefriedigende Weiterentwicklung, da gerade aus der semantischen Verknüpfung unterschiedlicher Wissensbasen interessantes Wissen geschlussfolgert werden könnte. Aus diesem Grund werden in dieser Arbeit vordergründig die theoretisch fundierten Ansätze betrachtet. Dabei wird sich eine weitere Teilung des Forschungsfeldes zeigen. Auf der einen Seite stehen Ansätze, welche den Verbund global interpretieren, auf der anderen Seite jene, welche lokale Interpretationen nur miteinander verknüpfen. Gerade in diesen Ansätzen – zu denen auch Distributed Description Logics, Package-based Description Logics und $\mathcal{E}$-Connections zählen – wurde jedoch die Untersuchung von Abfragen zunächst vernachlässigt, obwohl diese in praktischen

[2] BMWi

[3] Siehe `http://www.theseus-programm.de/home/`.

Szenarien kaum entbehrlich sind.

Einer der vorgestellten Ansätze mit verknüpften Interpretationen sind die von Oliver Kutz et al. vorgeschlagenen $\mathcal{E}$-Connections. Sie ermöglichen eine Verknüpfung nicht nur mehrerer Wissensbasen, sondern auch mehrerer, unterschiedlicher Logiken. So lassen sich viele modale, spatiale, temporale und beschreibende Logiken, welche in einer bestimmten Abstraktion dargestellt werden können, miteinander verknüpfen. Die wahre Stärke liegt dabei in der Eigenschaft, dass die entstehende Logik der jeweiligen $\mathcal{E}$-Connection genau dann entscheidbar ist, wenn die verknüpften Logiken entscheidbar sind. Schnell wurde dieser Ansatz als mögliche Lösung des Problems modularer Wissensbasen erkannt und von Bernardo Cuenca Grau et al. zur intuitiven Verknüpfung von OWL-Wissensbasen weiterentwickelt. Im Wettbewerb der Lösungsfindung stehen jedoch noch weitere Formalismen zur Auswahl, so sind auch Distributed Description Logics (DDL), Package-based Description Logics (P-DL) sowie auf konservativen Erweiterungen basierende Ansätze wichtige Kandidaten, die in dieser Arbeit vorgestellt und abgegrenzt werden.

Mittels konjunktiver Abfragen gezielt Daten aus Wissensbasen zu ermitteln, ist bereits Basis solch erfolgreicher Sprachen wie SQL und SPARQL. In der Erforschung von klassischen Beschreibungslogiken haben konjunktive Abfragen daher schon länger einen festen Platz. Die dabei entstandenen Erkenntnisse sind prototypisch für modulare Ansätze und wurden daher in diese Arbeit aufgenommen. Beim Übergang zu modularisierten Ontologien wird schließlich die Frage nach der generellen Beantwortbarkeit und nach der Möglichkeit, die Modularität dabei aufrecht erhalten zu können, gestellt und aufgrund der Komplexität der Thematik der Fokus auf $\mathcal{E}$-Connections gelegt. Im Besonderen gilt es ein Verfahren für die Beantwortung konjunktiver Abfragen über $\mathcal{E}$-Connections zu entwickeln, welches ein Ergebnis über die Reduktion auf eine Beantwortung in den Komponenten ermittelt.

Das nächste Kapitel enthält zunächst eine Einführung klassischer Beschreibungslogiken, welche meist als Grundlage der modularen Ansätze dienen. Es werden anschließend konjunktive Abfragen allgemein eingeführt und im Rahmen klassischer Beschreibungslogiken exemplarisch analysiert, wobei ein einfaches Verfahren zu deren Beantwortung vorgestellt wird. Dabei werden Probleme aufgezeigt, die sich auch in $\mathcal{E}$-Connections wiederfinden werden. Kapitel 3 enthält eine Einführung in die theoretischen Grundlagen der genannten Ansätze zur Modularisierung und führt die entsprechende Terminologie ein. Für ein tiefgreifendes Verständnis werden dabei die formalen Definition der originalen Veröffentlichungen wiedergegeben, wobei diese vor allem bei

$\mathcal{E}$-Connections relevant für spätere Kapitel sind. Nach einem Überblick über verwandte Studien in Kapitel 4, welche diese Arbeit beeinflusst haben, werden im Kapitel 5 konjunktive Abfragen über modulare Ontologien mit dem Fokus auf $\mathcal{E}$-Connections ausführlich analysiert.

2 Grundlagen

Für eine Bearbeitung der Aufgabenstellung müssen zuerst die Begriffe Ontologien bzw. Beschreibungslogiken und konjunktive Abfragen geklärt werden. Dieses Kapitel hat daher eine vollständige, verständliche wie auch formale Erläuterung dieser Begriffe zum Ziel.

2.1 Beschreibungslogiken

Das Hauptziel der Forschung im Gebiet der Beschreibungslogiken ist es, Theorien und Systeme zur Repräsentation von strukturiertem Wissen verfügbar zu machen, so dass dieses universell abgefragt und für (automatische) logische Schlussfolgerungen verwendet werden kann [Lut08]. Beschreibungslogiken sind Teilmengen der Prädikatenlogik erster Stufe, welche mit dem Ziel, Entscheidbarkeit zu garantieren, gestaltet wurden[1]. Sie erlauben es, Wissen in Form von Konzepten, Individuen, die diesen Konzepten zugeordnet werden, und Rollen, welche binären Relationen zwischen Konzepten entsprechen, zu formalisieren. Bei einer Übersetzung in Prädikatenlogik entsprechen Konzepte einstelligen Prädikaten, Rollen zweistelligen Prädikaten und Individuen Konstanten[2]. Die Ausdruckskraft einer Beschreibungslogik (Description Logic, DL) korreliert mit den erlaubten Konstruktoren der Sprache, die als Akronyme auch den jeweiligen Namen der DL-Variante konstituieren (Übersicht siehe Tabelle 2.1). So werden unterschiedliche Umsetzungen der Abwägung zwischen Berechnungskomplexität und Ausdrucksstärke möglich. Eine der einfachsten Beschreibungslogiken ist beispielsweise $\mathcal{ALC}$ (*Attributive Language with Complement*), welche die als Minimum für praktische Belange identifizierten Attribut-Logik bereits erweitert, und deren Komplexität für Erfüllbarkeit und Konsistenz PSPACE-complete ist [Zol07].

In jüngerer Vergangenheit sind Beschreibungslogiken vor allem durch die vom World Wide Web Consortium (W3C) empfohlene Web Ontology Language OWL populär geworden [PSHH04]. Diese existiert in drei aufeinander aufbauenden Dialekten: OWL Li-

[1] Im Gegensatz zur unentscheidbaren Prädikatenlogik.
[2] Bzw. Null-stelligen Funktionssymbolen

Akronym	Konstruktoren	Erklärung
$\mathcal{ALC}$	$\top$, $\bot$, A, $\neg C$, $(C \sqcap D)$, $(C \sqcup D)$, $\exists r.C$, $\forall r.C$	Die grundlegenden Konstrukte erweitert um Negation, sowie existenziellen und universellen Quantoren.
$\mathcal{I}$	r^-	Inverse Rollen.
$_{R+}$	r^+	Transitive Rollen.
$\mathcal{S}$	$\mathcal{ALC}_{R+}$	Oft genutzte Abkürzung.
$\mathcal{F}$	$(\leq 1\ r)$	Funktionale Rollen.
$\mathcal{N}$	$(\geq n\ r)$, $(\leq n\ r)$	Kardinalitätsbeschränkungen.
$\mathcal{Q}$	$(\geq n\ r.C)$, $(\leq n\ r.C)$	Qualifizierte Kardinalitätsbeschränkungen.
$\mathcal{O}$	$\{a_1, \ldots, a_n\}$	Enumerierte Konzepte bzw. Nominale
$\mathcal{H}$	$r \sqsubseteq s$	Hierarchie für Rollen.
$\mathcal{R}$	$r \circ s \sqsubseteq t$	Komplexe Rolleninklusion.
(D)	Datentypen	Reasoning über z.B. Zahlen, Strings, usw.

Tabelle 2.1: Beschreibungslogik Konstruktoren

te, OWL DL und OWL Full; wobei aufgrund der Entscheidbarkeit der zugrunde liegenden Logik hier nur die ersten beiden relevant sind. Deren logisches Fundament entspricht $\mathcal{SHIF}(D)$ bzw. $\mathcal{SHOIN}(D)$, was bereits einer sehr hohen Ausdruckskraft gleichkommt. So ist die Komplexität des Reasonings bei OWL DL bereits NExpTime-complete und scheinbar kaum für eine Implementierung geeignet. Es zeigte sich jedoch, dass „schwierige“ Fälle[3] in der Praxis selten auftauchen und daher optimierte Reasoner tatsächlich in der Lage sind, praktische Aufgaben zu lösen. Im Folgenden wird stellvertretend für weniger ausdrucksstarke Beschreibungslogiken die Syntax und Semantik von $\mathcal{SHOIQ}$ vorgestellt.

[3]Eine präzise Einordnung in „einfache“ und „schwierige“ Fälle fehlt in der Literatur. Stattdessen werden diese empirisch ermittelt.

Definition 1: Signatur, Rollen, Interpretation. *Seien N_C, N_r und N_I schnittfreie Mengen von Konzeptnamen, Rollennamen und Namen von Individuen. Die Menge N_r enthält eine Teilmenge $N_{rt} \subseteq N_r$ von transitiven Rollennamen. Dann heißt $S = (N_C, N_r, N_I)$ Signatur und die Menge der $\mathcal{SHOIQ}$-Rollen über S ist $N_r \cup \{r^- \mid r \in N_r\}$, wobei r^- Inverse von r genannt wird. Eine Rolleninklusion ist ein Axiom der Form $r \sqsubseteq s$ mit r, s Rollen. Eine Rollenhierarchie (RBox) ist eine Menge $\mathcal{R}$ solcher Axiome. Eine Interpretation $\mathcal{I} = (\Delta_\mathcal{I}, \cdot^\mathcal{I})$ besteht aus einer nicht leeren Menge $\Delta_\mathcal{I}$, der Domäne von $\mathcal{I}$, und einer Funkion $\cdot^\mathcal{I}$, welche jeden Konzeptnamen $A \in N_C$ auf eine Teilmenge $A^\mathcal{I} \subseteq \Delta_\mathcal{I}$, jeden Rollennamen $r \in N_r \setminus N_{rt}$ auf eine binäre Relation $r^\mathcal{I} \subseteq \Delta_\mathcal{I} \times \Delta_\mathcal{I}$, jeden Rollennamen $t \in N_{rt}$ auf eine transitive binäre Relation $r^\mathcal{I} \subseteq \Delta_\mathcal{I} \times \Delta_\mathcal{I}$ und jeden Individuumsnamen $a \in N_I$ auf ein Element $a^\mathcal{I} \in \Delta_\mathcal{I}$ abbildet. Für jeden Rollennamen $r \in N_r$ gilt für seine Inverse $r^{-\mathcal{I}} = \{(d, d') \mid (d', d) \in r^\mathcal{I}\}$. Eine Interpretation $\mathcal{I}$ erfüllt eine Rolleneinschließung $r \sqsubseteq s$, falls $r^\mathcal{I} \subseteq s^\mathcal{I}$ gilt und eine Rollenhierarchie, falls sie jedes darin enthaltene Axiom erfüllt. Desweiteren soll eine Rolle s „primitiv“ heißen, falls weder s, noch s^-, noch eine von s oder s^- durch $\sqsubseteq$ erreichbare Rolle transitiv ist.*

Definition 2: Konzepte. *Sei S eine Signatur, dann ist die Menge der $\mathcal{SHOIQ}$-Konzepte über S die kleinste Menge, die induktiv über die Symbole von S mit folgender Grammatik erzeugt werden kann:*

$$C ::= \top \mid \bot \mid \{o\} \mid A \mid \neg C \mid (C_1 \sqcap C_2) \mid (C_1 \sqcup C_2) \mid \forall r.C \mid \exists r.C \mid (\geq n\ s.C) \mid (\leq n\ s.C)$$

Dabei ist $o \in N_I$, $A \in N_C$, $n \in \mathbb{N}_0$, r eine Rolle und s eine primitive Rolle. Die Beschränkung auf primitive Rollen in den Kardinalitätsbeschränkungen dient der Erhaltung der Entscheidbarkeit.

Definition 3: Semantik von Konzepten, TBox. *Die Semantik von $\mathcal{SHOIQ}$-Konzepten wird mittels einer Interpretation $\mathcal{I} = (\Delta_\mathcal{I}, \cdot^\mathcal{I})$ definiert als:*

$$\begin{aligned}
\top^\mathcal{I} &= \Delta_\mathcal{I} \qquad \bot^\mathcal{I} = \emptyset \qquad \{o\}^\mathcal{I} = \{o^\mathcal{I}\} \qquad A^\mathcal{I} \subseteq \Delta_\mathcal{I} \\
(\neg C)^\mathcal{I} &= \Delta_\mathcal{I} \setminus C^\mathcal{I} \qquad (C_1 \sqcap C_2)^\mathcal{I} = (C_1^\mathcal{I} \cap C_2^\mathcal{I}) \qquad (C_1 \sqcup C_2)^\mathcal{I} = (C_1^\mathcal{I} \cup C_2^\mathcal{I}) \\
(\forall r.C)^\mathcal{I} &= \{d \in \Delta_\mathcal{I} \mid (d, d') \in r^\mathcal{I} \rightarrow d' \in C^\mathcal{I}\} \\
(\exists r.C)^\mathcal{I} &= \{d \in \Delta_\mathcal{I} \mid \exists d'.((d, d') \in r^\mathcal{I} \wedge d' \in C^\mathcal{I})\} \\
(\geq n\ s.C)^\mathcal{I} &= \{d \in \Delta_\mathcal{I} \mid \#\{d' \in \Delta_\mathcal{I} \mid (d, d') \in s^\mathcal{I} \wedge d' \in C^\mathcal{I}\} \geq n\} \\
(\leq n\ s.C)^\mathcal{I} &= \{d \in \Delta_\mathcal{I} \mid \#\{d' \in \Delta_\mathcal{I} \mid (d, d') \in s^\mathcal{I} \wedge d' \in C^\mathcal{I}\} \leq n\}
\end{aligned}$$

Dabei bezeichnet $\#M$ die Anzahl der Elemente der Menge M.

Schließlich heißt für Konzepte C und D ein Axiom der Form $C \sqsubseteq D$ „General Concept Inclusion“ (GCI) und eine endliche Menge von GCIs TBox. Eine Interpretation I erfüllt $C \sqsubseteq D$ falls $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ gilt und eine TBox $\mathcal{T}$ falls alle enthaltenen GCIs erfüllt werden.

Definition 4: ABox. *Ein ABox-Axiom über eine Signatur S ist ein Ausdruck der Form $C(a)$ oder $r(a, b)$, wobei C eine Konzept über S, r eine Rollenname und a, b Namen von Individuen aus S sind. Diese heißen Konzept- bzw. Rollenaxiome. Eine ABox ist eine endliche Menge solcher Axiome. Eine Interpretation $\mathcal{I}$ erfüllt $C(a)$ falls $a^{\mathcal{I}} \in C^{\mathcal{I}}$ gilt und $r(a, b)$ falls $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ gilt. Sie erfüllt eine ABox $\mathcal{A}$, falls alle darin enthaltenen Axiome erfüllt werden.*

Definition 5: Wissensbasis. *Bei einer gegebenen Signatur S ist eine Wissensbasis $KB = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ ein Tripel bestehend aus einer TBox $\mathcal{T}$, einer Rollenhierarchie $\mathcal{R}$ und einer ABox $\mathcal{A}$ über S. Eine Interpretation $\mathcal{I}$ erfüllt KB, falls sie $\mathcal{T}$, $\mathcal{R}$ und $\mathcal{A}$ erfüllt. In diesem Fall heißt $\mathcal{I}$ Modell von KB, geschrieben $\mathcal{I} \models KB$.Die in der Wissensbasis verwendeten Individuen sollen mit $Inds(KB)$, die verwendeten Konzeptnamen mit $Cons(KB)$ bezeichnet werden.*

Innerhalb einer Wissensbasis sind demzufolge die RBox und TBox für die Definition einer Art Schema zuständig, welches von Elementen der ABox (teilweise) instanziiert wird. Natürlich nutzt man das Potenzial einer Logik-basierten Sprache erst dann, wenn man Dienste beschreibt, welche (mitunter nur implizit vorhandene) Informationen zu extrahieren vermögen. Diese „Reasoning-Dienste“ genannten Verfahren umfassen:

Konsistenz der Wissensbasis: Bezogen auf eine Wissensbasis KB stellt dieser Dienst die Konsistenz der Wissensbasis fest, falls eine Interpretation $\mathcal{I}$ existiert, so dass $\mathcal{I} \models KB$ gilt, d.h. falls die Wissensbasis ein Modell besitzt. Ansonsten lautet das Ergebnis „KB ist inkonsistent“. In manchen Fällen wird Konsistenz einer Wissensbasis auch als deren Erfüllbarkeit bezeichnet.

Erfüllbarkeit eines Konzepts: Ein Konzept C wird im Kontext einer Wissensbasis KB als erfüllbar erkannt, falls ein Modell $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ von KB gefunden werden kann, für das ein Element $d \in \Delta_{\mathcal{I}}$ existiert $d \in C^{\mathcal{I}}$. Ansonsten lautet die Antwort „C ist unter Berücksichtigung von KB unerfüllbar“.

Dies ist äquivalent zur Frage, ob $KB' = (\mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{C(a)\})$ mit einem „neuen“ Individuum $a \in N_I,\ a \notin Inds(KB)$ konsistent ist.

Subsumierung von Konzepten: Bei zwei gegebenen Konzepten C und D einer Wissensbasis KB liefert eine Entscheidungsprozedur für Konzeptsubsumierung „D subsumiert C“ zurück, falls kein Modell $\mathcal{I}$ für KB existiert, für das ein Element $d \in \Delta_{\mathcal{I}}$ existiert, für das $d \in C^{\mathcal{I}}$ und $d \in (\neg D)^{\mathcal{I}}$ gilt. Existiert jedoch eine solche Interpretation lautet die Antwort „C wird nicht von D subsumiert“.

Dies ist äquivalent zur Frage, das Konzept $(C \sqcap \neg D)$ im Rahmen von KB unerfüllbar ist (siehe oben).

Prüfung auf Instantiierung: Für ein Individuum a und ein Konzept C einer Wissensbasis KB liefert dieser Dienst die Antwort „a ist Instanz von C unter Berücksichtigung von KB“ falls für jedes Modell von KB gilt $a^{\mathcal{I}} \in C^{\mathcal{I}}$. Ansonsten lautet die Antwort „a ist keine Instanz von C unter Berücksichtigung von KB“.

Dies ist äquivalent zur Frage, ob $KB' = (\mathcal{T}, \mathcal{R}, \mathcal{A} \cup \{(\neg C)(a)\})$ inkonsistent ist.

Es ist zu bemerken, dass sich damit auch der hier nicht gesondert erwähnte Reasoning-Dienst *Instanzabfrage* realisieren lässt, welcher alle Instanzen eines gegebenen Konzeptes zurückliefert.

Logische Konsequenz: Ein beliebiges TBox, RBox oder ABox Axiom φ über der Signatur einer Wissensbasis KB ist eine logische Konsequenz der Wissensbasis, wenn für jede Interpretation $\mathcal{I}$ folgende Implikation gilt: $\mathcal{I} \models KB \rightarrow \mathcal{I} \models \varphi$.

Dies ist erneut äquivalent zu der Frage, ob $KB \cup \{\neg\varphi\}$ inkonsistent ist. Dabei ist die Negation von φ derart zu berücksichtigen, dass eine Aussage im Sinne der verwendeten Sprache entsteht.

Es wird deutlich, dass es genügt, eine Entscheidungsprozedur für die Konsistenz einer Wissensbasis zu finden.

Mit diesen Definitionen und Diensten als Basis lassen sich bereits viele Anwendungen realisieren, die von den Vorzügen automatischer Schlussfolgerungen profitieren. Durch das Vorhandensein von ausgereiften Werkzeuge, die die Entwicklung der notwendigen Ontologien unterstützen, umfangreichen Programmierbibliotheken, welche

den Zugriff darauf sehr einfach machen, und effizienter Reasoner, die als „Black-Box“ in Systemen verwendet werden können, ist der Einsatz von Beschreibungslogiken auf Basis von OWL mit geringen Hürden verbunden.

2.2 Konjunktive Abfragen

Die Daten aus Ontologien sollten mit flexiblen und allgemein verstandenen Mechanismen abgefragt werden können [OCE06]. So erreicht man eine neue Qualität von Wissensextraktion mit der Verknüpfung diverser Fragestellungen innerhalb von konjunktiven Abfragen. Schon in einfachen Varianten lassen sich damit alle oben Reasoning-Dienste intuitiv darstellen und kombinieren. Sie stellen damit einen akzeptablen Mittelweg zwischen der möglichen Komplexität der Abfragen und ihrer Berechenbarkeit dar. Dabei trifft dies nicht nur auf die gerade vorgestellten Beschreibungslogiken zu, da konjunktive Abfragen auch in allgemeinen Szenarien zur Datenintegration [Len02] und der verbreiteten Datenbankabfragesprache SQL [Dat86] zum Einsatz kommen, um nur zwei Beispiele zu nennen.

Allgemein ermöglicht es eine Sprache für konjunktive Abfragen, die Existenz von Elementen beliebig-stelliger Relationen R zwischen bestimmten Konstanten c oder Variablen x und y innerhalb eines relationalen Systems zu bestimmen. Dabei heißen die Variablen in $\vec{x}$ „Antwortvariablen“ und die Variablen in $\vec{y}$ „existenziell quantifizierte Variablen“. Im folgenden Beispiel existiert nur eine Antwortvariable x_1:

$$\vec{x} \leftarrow R_1(c_1, x_1) \wedge R_2(x_1) \wedge R_3(x_1, y_1, y_2) \wedge R_4(y_1, c_2)$$

In diesem Falle enthält eine Antwort Konstanten für x_1. Für jedes Element der Antwortmenge muss jedoch auch eine Abbildung der existenziell quantifizierten Variablen $\vec{y}$ auf Elemente des Systems existieren. Dabei müssen die durch die Ersetzung auf der rechten Seite entstehenden Aussagen in einer gegebenen Menge von Aussagen wahr sein, d.h. die entsprechenden Tupel müssen in den Interpretationen der Relationen enthalten sein.

Eine Abfrage, bei der nur Variablen y_i vorkommen, heißt Boolesche Abfrage. Als Antwort sind keine expliziten Variablenzuordnungen notwendig, so dass als Ergebnis ein `wahr` oder `falsch` genügt. Dabei steht ersteres für die notwendige Existenz von passenden Elementen für die verbliebenen Variablen, während `falsch` hier für die mögliche oder unmögliche Existenz steht.

Ein qualitativer Unterschied der Beantwortung von Abfragen ist zwischen einem Datenbank-Kontext und vielen Logik-basierten Kontexten gegeben. In Datenbanken wird üblicherweise die „Closed World Assumption“ (Annahme der Abgeschlossenheit) gemacht, welche dazu führt, dass alle Aussagen, die nicht explizit gemacht werden, als falsch angenommen werden. In Systemen, die auf Beschreibungslogiken oder Prädikatenlogik erster Stufe basieren, wird üblicherweise die gegenteilige Annahme, die „Open World Assumption“ gemacht. Dies führt dazu, dass auch unvollständiges Wissen abgefragt werden kann. In der Konsequenz gibt es für derartige Wissensbasen normalerweise viele zu berücksichtigende Modelle während für eine Datenbank immer genau ein Modell existiert. Dort genügt es daher dieses eine Modell zu überprüfen, wo bei fehlender Abgeschlossenheit die logische Konsequenz der Abfrage überprüft werden muss [Gli07]. In dieser Arbeit wird zu keinem Zeitpunkt die Abgeschlossenheit angenommen.

Im obigen Beispiel müsste nur die Variable x_1 an konkrete Werte gebunden werden, da die anderen Variablen existenziell quantifiziert sind. Ist jedoch aufgrund unvollständigen Wissens keine konkrete Konstante bekannt, sondern nur dass ein auf die Abfrage passendes Element für x_1 existieren muss, wird dennoch kein Ergebnis zurückgeliefert. Dies führt dazu, dass in diesen Fällen Antworten fehlen, sobald die entsprechende Variable als Antwortvariable deklariert wird. Erst bei existenzieller Quantifizierung werden alle gültigen Ersetzungen berücksichtigt. Um dieses Verhalten zu umgehen, wird im Laufe dieser Arbeit keine Unterscheidung zwischen den beiden Typen von Variablen gemacht. Stattdessen wird jede Variable sowohl als Antwortvariable, als auch als existenziell quantifizierte Variable verstanden. Ermöglicht wird dies durch ein spezielles, künstliches Element $\texttt{exists}_{x_i}$ welches immer dann als Ersetzung in einem Antworttupel auftauchen wird, wenn nur die Existenz, nicht aber die Identität eines passenden Elements für x_i bekannt ist. Für eine angenehmere Handhabung soll des Weiteren statt einer Konjunktion der Elemente eine Mengenschreibweise genutzt werden, welche in Folge der einheitlichen Betrachtung von Variablen eine konjunktive Abfrage vollständig beschreibt.

Die formale Definition konjunktiver Abfragen, Boolescher konjunktiver Abfragen und den gültigen Antworten darauf ist abhängig vom jeweiligen Anwendungskontext. Dabei werden die erlaubten Relationen sinnvoll eingeschränkt. So beschränkt man sich beispielsweise bei Abfragen über Beschreibungslogiken meist auf einstellige (Konzepte) und zweistellige Relationen (Rollen).

2.3 Konjunktive Abfragen in Beschreibungslogiken

Zur Illustration konjunktiver Abfragen und der Probleme, die vor allem bei der Verwendung existenziell quantifizierter Variablen entstehen, sollen Abfragen auf Beschreibungslogiken detailliert betrachtet werden. Je nach angestrebter Mächtigkeit lassen sich Abfragen über TBox, RBox und ABox miteinander kombinieren, doch wegen der dabei entstehenden Komplexität soll dies im Rahmen dieser Arbeit auf ABox-Abfragen begrenzt werden.

Ziel diesen Abschnitts soll es daher sein, die Reduktion von konjunktiven Abfragen auf ABoxen von Beschreibungslogik-Wissensbasen auf das Erfüllbarkeitsproblem von eben solchen Wissensbasen vorzustellen. Um die für eine Einführung unnötige Komplexität zu vermeiden, soll im Folgenden nur die Beschreibungslogik $\mathcal{ALC}$ betrachtet werden.

Definition 6: Konjunktive Abfrage. *Eine konjunktive Abfrage (CQ) auf einer Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$ einer Beschreibungslogik $\mathcal{S}$ ist eine endliche Menge $q = \{q_1, \ldots, q_n\}$ von Atomen q_i die von der Form $C(x)$ oder $r(x, y)$ sind, wobei $x, y \in \mathcal{A} \cup \mathcal{V}$, r eine Rolle und C ein Konzept aus $\mathcal{T}$ ist. Dabei ist $\mathcal{V}$ die Menge aller in der Abfrage verwendeten Variablen.*

Eine anschauliche Darstellung der Abfrage ist mittels des durch q induzierten, gerichteten Graphen $G(q) = (V, E, L)$ möglich (vgl. [GLHS08]).

Definition 7: Graph einer Abfrage. *Ein Graph $G(q) = (V, E, L)$ einer Abfrage q ist ein Tripel bestehend aus einer endlichen Knotenmenge V, einer endlichen Menge von Kanten E und einer Operation L, welche Knoten eine Menge von Konzepten aus KB als Beschriftung zuweist.*

$$\begin{aligned} V &= \{v \mid v \in \mathcal{A} \cup \mathcal{V} \text{ und } v \text{ kommt in } q \text{ vor.}\} \\ E &= \{(v_i, v_j, r) \mid r(v_i, v_j) \in q\} \\ L(v) &= \{C \mid C(v) \in q\} \end{aligned}$$

Zwischen zwei Knoten können beliebig viele Kanten existieren, die sich jedoch in r unterscheiden müssen. Die Blätter des Graphen B sind Knoten, von denen keine

Kanten ausgehen und welche Ziel genau einer Kante sind [HT00]:

$$B = \{v_i \mid \{v_j \mid (v_i, v_j, *) \in E\} = \emptyset \wedge \#\{v_k \mid (v_k, v_i, *) \in E\} = 1\}$$

Dabei ist $*$ als Platzhalter für eine beliebige Rolle zu verstehen.

Im folgenden wird ohne Beschränkung der Allgemeinheit davon ausgegangen, dass die betrachteten Abfragen verbunden sind, d.h. dass im Graph $G(q)$ für zwei beliebige Knoten v_1, v_n

$$\exists(v_1, \ldots, v_n) \bigwedge_{i=2}^{n} ((v_{i-1}, v_i) \in E \vee (v_i, v_{i-1}) \in E) \quad n \in \mathbb{N} \text{ erfüllt sein muss.}$$

2.3.1 Logische Konsequenz variablenfreier Abfragen

Als Vereinfachung werden zunächst nur Abfragen ohne Variablen betrachtet, welche demzufolge Boolesche Abfragen sind. Die Antwort $A(q)$ für eine solche Abfrage q ist genau dann `wahr`, wenn $KB \models \bigwedge_{i=1}^{n} q_i$ gilt, ansonsten lautet die Antwort `falsch`. Letzteres ist als Fehlschlag zu verstehen, dass heißt, die Abfrage ist keine logische Konsequenz **aller** die Wissensbasis erfüllenden Interpretationen. Daher kann es durchaus einige Interpretationen geben, welche die Abfrage erfüllen, so dass `falsch` gleichzeitig für `vielleicht` steht.

Im minimalen Fall, bei dem die Abfrage nur aus einem Atom der Form $C(a)$ besteht, lässt sich die logische Konsequenz bereits auf Erfüllbarkeit einer Wissensbasis reduzieren, da gilt:

$$(\mathcal{T}, \mathcal{A}) \models \text{Student(Jörg)} \iff (\mathcal{T}, \mathcal{A} \cup \{(\neg\text{Student})(\text{Jörg})\}) \text{ ist nicht erfüllbar.} \quad (2.1)$$

Bei mehreren Atomen dieser Form, kann dies aufgrund ihrer Unabhängigkeit für jedes Atom einzeln überprüft werden kann. Genauer gesagt, lässt sich in diesem Fall die Abfrage durch Ausnutzung folgender Äquivalenz beantworten:

$$KB \models \bigwedge_{i=1}^{n} q_i \iff \bigwedge_{i=1}^{n} KB \models q_i \text{ mit } q_i = C_i(a_i), a_i \in N_I \quad (2.2)$$

Diese Äquivalenz lässt sich nicht ohne weiteres bei Existenz von Variablen oder Rollenatomen ausnutzen. Im ersten Schritt soll dazu die Beantwortung von Rollenatomen betrachtet und weiterhin Variablenfreiheit angenommen werden.

In [HT00] wurde die Technik „Rolling Up“ beschrieben, welche zur Lösung des Problems angewendet werden kann. Es existieren nun innerhalb der Abfrage nur noch Atome der Form $q_i = C(a)$ (deren Wahrheitswert gemäß Formel 2.1 ermittelt werden kann) und Atome q_j der Form $r(a, b)$ mit a, b Namen von Individuen und r einem Rollennamen, für welche nun eine simple Äquivalenzumformung gezeigt wird. Einfach gesprochen besteht die Idee darin, mit Hilfe von Nominalen eine Konzeptbeschreibung zu generieren, welche bzgl. der Schlussfolgerungsrelation die gewünschte Semantik besitzt. Desweiteren lassen sich im Zuge dieser Umformung weitere Atome absorbieren:

$$\{r(a, b), C_1(b), C_2(b)\} \mapsto \{(\exists r.(\{b\} \sqcap C_1 \sqcap C_2))(a)\}$$

Wobei das Nominal $\{b\}$ genau das Konzept darstellt, welches ausschließlich b enthält. Damit ist das Rollenatom eliminiert und die Reduktion auf Unerfüllbarkeit ist wie oben gezeigt anwendbar. Dabei wird die ursprünglich beabsichtigte Semantik der Abfrage erhalten und die richtige Antwort ermittelt.

Da $\mathcal{ALC}$ jedoch keine Nominale enthält und dies auch im Allgemeinen nicht angenommen werden kann, müssen diese „simuliert“ werden. In einem Zwischenschritt werden dazu für alle in der Abfrage enthaltenen Individuennamen a primitive Stellvertreterkonzepte P_a eingeführt, deren Namen bisher nicht in der Wissensbasis enthalten waren. Die einzigen Axiome, welche nun Aussagen über diese Konzepte machen, sind in einer Menge $\mathcal{P} = \{P_a(a) \mid a \in Inds(q)\}$ enthalten und legen die Zugehörigkeit des stellvertretenen Individuums fest. Das folgende Theorem zeigt die mit „Rolling Up“ bezeichnete Transformation aus [HT00] in einer eigenen, angepassten Formalisierung.

Theorem 1: Rolling Up. *Seien $KB = (\mathcal{T}, \mathcal{A})$ eine Wissensbasis über eine Signatur S der Beschreibungslogik $\mathcal{ALC}$, C_i mit $i = 1 \ldots n, n \in \mathbb{N}$ Konzepte, r eine Rolle und a, b Individuen aus KB. Dann gilt:*

$$KB = (\mathcal{T}, \mathcal{A}) \models r(a, b) \wedge C_1(b) \wedge \ldots \wedge C_n(b) \tag{2.3}$$

genau dann, wenn

$$\begin{aligned} KB' = (\mathcal{T}, \mathcal{A} \cup \mathcal{P}) \models (\exists r.(P_b \sqcap C_1 \sqcap \ldots \sqcap C_n))(a), \\ \textit{mit } \mathcal{P} = \{P_b(b)\} \textit{ und } P_b \notin Cons(KB). \end{aligned} \tag{2.4}$$

Beweis. Dieser Beweis orientiert sich am Beweis von Lemma 3.2.2 aus [HSTT99]. Als Abkürzung sollen die rechten Seiten der Formeln 2.3 und 2.4 im Folgenden als

$\psi = r(a,b) \wedge C_1(b) \wedge \ldots \wedge C_n(b)$ bzw. $\varphi = (\exists r.(P_b \sqcap C_1 \sqcap \ldots \sqcap C_n))(a)$ bezeichnet werden. Sei $\mathcal{J}$ ein beliebiges Modell für KB für welches zusätzlich $b^{\mathcal{J}} \in P_b^{\mathcal{J}}$ gilt. Offenbar erfüllt $\mathcal{J}$ nun auch die zusätzlichen Aussagen aus KB' und ist damit auch dafür ein Modell. Bezüglich der anderen Richtung muss jedes Modell für KB' auch ein Modell für KB sein, da erstere alle Axiome aus letzterer enthält. Es gilt daher gemäß Abschnitt 2.1 und der Definition von logischer Konsequenz:

$$\mathcal{J} \models \varphi \iff a^{\mathcal{J}} \in \Big\{ d' \in \Delta_{\mathcal{J}} \mid \exists d \,.\, \Big((d', d) \in r^{\mathcal{J}} \wedge d \in (P_b^{\mathcal{J}} \cap C_1^{\mathcal{J}} \cap \ldots \cap C_n^{\mathcal{J}}) \Big) \Big\}$$

bzw. die äquivalente Vereinfachung

$$\mathcal{J} \models \varphi \iff \underbrace{\exists d \in \Delta_{\mathcal{J}} \,.\, \Big((a^{\mathcal{J}}, d) \in r^{\mathcal{J}} \wedge d \in (P_b^{\mathcal{J}} \cap C_1^{\mathcal{J}} \cap \ldots \cap C_n^{\mathcal{J}}) \Big)}_{\varphi(\mathcal{J})} \tag{2.5}$$

welche abgekürzt notiert wird als

$$\mathcal{J} \models \varphi \iff \varphi(\mathcal{J})$$

Damit verbleibt nur noch zu zeigen, dass für Modelle $\mathcal{J}$ von KB und KB' gilt:

$$\forall \mathcal{J} \; \mathcal{J} \models \psi \iff \forall \mathcal{J} \; \varphi(\mathcal{J}).$$

⇒: Wenn $\mathcal{J} \models \psi$ gilt, dann auch

$$(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{J}} \text{ und } b^{\mathcal{J}} \in C_i^{\mathcal{J}}, i = 1 \ldots n.$$

Wegen der Konstruktion von $\mathcal{J}$ ist damit $\varphi(\mathcal{J})$ erfüllt, d.h. die Implikation gilt bereits für jedes einzelne Modell.

⇐: In der Gegenrichtung ist zu zeigen, dass bei Existenz eines Modells von KB, welches ψ nicht erfüllt, auch ein Modell existiert, das $\varphi(\cdot)$ nicht erfüllt. Sei also $\mathcal{J}$ ein Modell, für welches $\mathcal{J} \models \psi$ **nicht** gilt. Außerdem kann davon ausgegangen werden, dass $P_b^{\mathcal{J}} = \{b^{\mathcal{J}}\}$ ist[4]. Es wird angenommen, dass sich $\mathcal{J}$ zu einer Interpretation $\mathcal{J}'$ erweitern lässt, für welche $\varphi(\mathcal{J}')$ gilt. Demzufolge muss ein Tupel $(a^{\mathcal{J}'}, d) \in r^{\mathcal{J}'}$ existieren, so dass $d \in (P_b^{\mathcal{J}'} \cap C_1^{\mathcal{J}'} \cap \ldots \cap C_n^{\mathcal{J}'})$ gilt. Es lassen sich wegen $\mathcal{J} \not\models \psi$ nun zwei Fälle unterscheiden.

1. Falls $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{J}}$, dann gilt dies auch für $\mathcal{J}'$. Aufgrund der Interpretation von P_b ist aber $P_b^{\mathcal{J}'} \cap C_1^{\mathcal{J}'} \cap \ldots \cap C_n^{\mathcal{J}'} = \{b^{\mathcal{J}'}\}$, woraus $\mathcal{J}' \not\models \varphi(\mathcal{J}')$ folgt.
2. Gilt $b^{\mathcal{J}} \notin C_i^{\mathcal{J}}$ für ein $i \in [1, n]$ so gilt dies auch für $\mathcal{J}'$ und es ist bereits

[4] Dies wurde in [HSTT99], Lemma 3.1.3 für $\mathcal{DLR}$ gezeigt und lässt sich auf $\mathcal{ALC}$ übertragen.

$$P_b^{\mathcal{J}'} \cap C_1^{\mathcal{J}'} \cap \ldots \cap C_n^{\mathcal{J}'} = \emptyset \text{ woraus ebenso } \mathcal{J}' \not\models \varphi(\mathcal{J}') \text{ folgt.}$$

Beide Fälle führen zu einem Widerspruch mit der Annahme, so dass $\mathcal{J} \not\models \psi \rightarrow \neg\varphi(\mathcal{J})$ geschlossen und damit auch die Gegenrichtung als erfüllt betrachtet werden kann.

□

Mit Hilfe von Theorem 1 lässt sich nun eine Boolesche Abfrage ohne Variablen beantworten. Dazu werden in der Abfrage enthaltene Rollenatome solange aufgerollt, bis ausschließlich Atome der Form $C(a)$ verbleiben, welche – wie in Formel 2.1 und 2.2 gezeigt – beantwortet werden können.

Im nächsten Schritt sollen nun existenziell quantifizierte Variablen erlaubt werden, d.h. jede Variable wird als ausschließlich existenziell quantifiziert angenommen.

2.3.2 Boolesche Abfragen mit Variablen

Die vorgestellte Technik ist bei Vorkommen von Variablen nicht ohne Weiteres anwendbar. Dazu soll zunächst der grundlegende Fall betrachtet werden, bei welchem die Abfrage der Form $\{C_1(x), \ldots, C_n(x)\}$, mit nur einer Variable x, beantwortet werden soll. Die Konzeptatome, welche Aussagen über nur eine Variable machen, lassen sich äquivalent zusammenfassen zu $\{(C_1 \sqcap \ldots \sqcap C_n)(x)\}$. Es muss `wahr` zurückgegeben werden, falls $(C_1 \sqcap \ldots \sqcap C_n)$ in jedem Modell der Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$ wenigstens ein Element enthält. Dies ist äquivalent zur Frage, ob

$$(\mathcal{T} \cup \{\top \sqsubseteq \neg(C_1 \sqcap \ldots \sqcap C_n)\}, \mathcal{A}) \text{ inkonsistent ist [HT00].}$$

Zur Veranschaulichung der Probleme bei Rollenatomen nehme man die Beispielabfrage

$$q_0 = \{\text{EnrolledIn}(x, y), \text{Student}(x), \text{Lecture}(y)\},$$

welche `wahr` ergibt, falls Studenten existieren, die in Vorlesungen eingeschrieben sind. Eine Transformation nach obigem Schema würde das Atom

$$(\text{Student} \sqcap \exists\text{EnrolledIn}.(P_y \sqcap \text{Lecture}))\,(x)$$

ergeben. Dies beschränkte y auf Elemente aus P_y was nicht im Sinne der Abfrage wäre. Stattdessen lässt sich P_y mit $\top$ ersetzen, bzw. im Beispiel ganz weglassen. Das

transformierte Atom

$$(\text{Student} \sqcap \exists\text{EnrolledIn}.(\text{Lecture}))\,(x)$$

ergibt die gewünschte Semantik. Problematisch wird dies jedoch, sobald y noch an anderer Stelle in der Abfrage genutzt wird. Die Beispielabfrage soll daher erweitert werden:

$$q_1 = q_0 \cup \{\text{GivenBy}(y, z), \text{Professor}(z)\}$$

Nun soll nur dann `wahr` zurückgegeben werden, wenn wenigstens ein Student in eine Vorlesung eingeschrieben ist, die von einem Professor gehalten wird. Eine gültige Ersetzung ist nun abhängig von der Reihenfolge. Betrachtet man zuerst das Rollenatom aus q_0 und erst danach das hinzugefügte, erhält man die transformierte Abfrage

$$\{(\text{Student} \sqcap \exists\text{EnrolledIn}.(\text{Lecture}))\,(x), (\exists\text{GivenBy}.(\text{Professor}))\,(y)\}$$

Diese Abfrage würde jedoch schon mit `wahr` beantwortet werden können, sobald ein Student existiert, welcher in eine Vorlesung eingeschrieben ist und ein Element der Domäne existiert, welches von einem Professor gehalten wird. Dies ist offensichtlich nicht mehr äquivalent zur beabsichtigten Abfrage. Das Beispiel lässt sich jedoch in anderer Reihenfolge gültig transformieren. Dabei wird am Graph $G(q_1)$ (siehe Abbildung 5.2) deutlich, dass für eine gültige Transformation immer an Blattknoten begonnen werden muss. Die korrekte Transformation ergibt schließlich

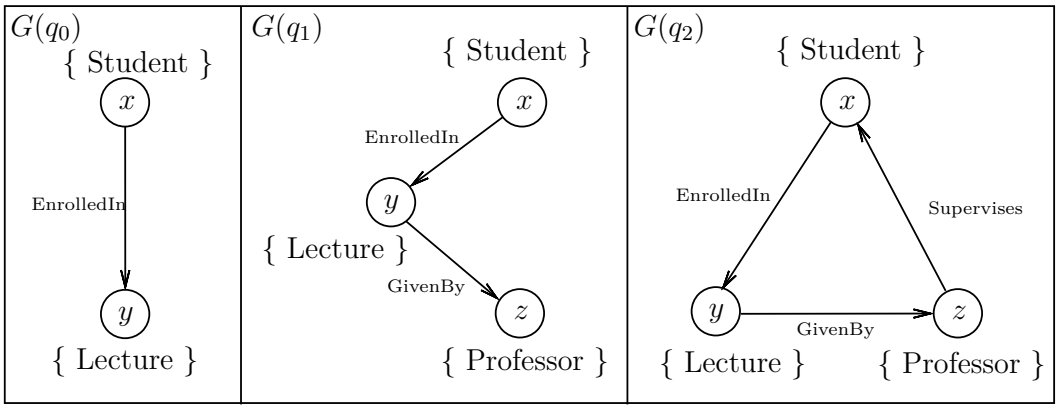
$$\{(\text{Student} \sqcap \exists\text{EnrolledIn}.\,(\text{Lecture} \sqcap \exists\text{GivenBy}.(\text{Professor})))\,(x)\}\,.$$

Dadurch ließen sich in diesem Fall zwei Variablen vollständig eliminieren.

Obwohl sich damit bereits sehr viele Fälle bearbeiten lassen, tritt ein weiteres Problem auf, sobald die Abfrage zyklisch wird. Folgendes Beispiel veranschaulicht dies:

$$q_2 = q_1 \cup \{\text{Supervises}(z, x)\}.$$

Der Graph der Abfrage enthält nun keine Blattknoten mehr, was ein Aufbrechen des Zyklus erforderlich macht. Dabei lässt sich auf die sogenannte „Tree Model"-Eigenschaft zurückgreifen, welche viele Beschreibungslogiken wie z.B. auch $\mathcal{ALCFI}_{reg}$ aufweisen [BCM+03] und welche von Ian Horrocks als Vorraussetzung für die Beantwortung zyklischer Abfragen identifiziert wurde [HT00]. Sie besagt, **wenn** eine Wissensbasis ein Modell besitzt, **dann** existiert ein Modell, welches eine (möglicherweise unendliche) Baumstruktur besitzt. Das heißt, dass allein mit TBox Aussagen unmög-



Jeder Knoten ist mit einer Menge von Konzepten beschriftet.

Abbildung 2.1: Graphen der Beispielabfragen.

lich sicherzustellen ist, dass alle Modelle zyklisch sind. Dies bedeutet jedoch, dass eine solche zyklische Abfrage nur dann logische Konsequenz aller Modelle sein kann, wenn der Zyklus explizit festgehalten worden ist, was wiederum nur mit benannten Individuen der Fall sein kann. Somit lässt sich die zyklische Abfrage aufbrechen, indem nicht-deterministisch eine Variable mit einem Individuum der ABox ersetzt wird und anschließend mit Stellvertreterkonzepten fortgefahren werden kann. Bezogen auf das Beispiel q_2 könnte man x mit einem Individuum[5] ersetzen und erhielte die transformierte und beantwortbare Abfrage:

$$\{(\text{Student} \sqcap \exists\text{EnrolledIn}.(\text{Lecture} \sqcap \exists\text{GivenBy}.(\text{Professor} \sqcap \exists\text{Supervises}.P_{\text{Jörg}}))) \\ (\text{Jörg})\}$$

oder äquivalent eine Ersetzung der Variable y

$$\{(\text{Lecture} \sqcap \exists\text{GivenBy}.(\text{Professor} \sqcap \exists\text{Supervises}.(\text{Student} \sqcap \exists\text{EnrolledIn}.P_{\text{LBKR}}))) \\ (\text{LBKR})\}.$$

Die gleiche Argumentation lässt sich auf Fälle anwenden, bei denen zwei oder mehr Rollenatome die gleiche Variable als zweites Argument enthalten [HT00]. Da die richtige Ersetzung nicht bekannt ist, müsste diese Ersetzung solange durchgeführt werden, bis entweder `wahr` zurückgegeben wird oder mit allen benannten Individuen getestet

[5] Die verwendeten Individuen entstammen einer hypothetischen Wissensbasis über den Lehrbetrieb der Fakultät Informatik der TU Dresden.

worden ist. Handelt es sich bei der abgefragten Logik nur um $\mathcal{ALC}$, genügt es, ein beliebiges Element des Zyklus' zu ersetzen, da jedes Element eines in jedem Modell enthaltenen Zyklus eine Interpretation eines Individuum sein muss. Dies führt zu soviel möglichen Ersetzungen wie es Individuen in der Wissensbasis gibt und daher im schlimmsten Fall zu ebenso vielen Verzweigungen, von denen eine jede ausreicht, die Abfrage mit `wahr` zu beantworten, eine Verneinung jedoch nur bei negativer Beantwortung aller Verzweigungen möglich ist.

Eine weitere Komplexitätssteigerung ergibt sich, sobald die Logik Nominale erlaubt. In diesem Fall ist es möglich, dass nur noch ein Zyklus-Element einem Individuum entspricht, weswegen jedes als Ersetzungskandidat probiert werden muss. Damit ergibt sich die Menge an zu verneinenden Verzweigungen vor einer endgültigen Beantwortung mit `falsch` als Produkt der Anzahl der Individuen der Wissensbasis und der im Zyklus enthaltenen Elemente.

Nach potentiell mehrfacher Anwendung oben stehender Methoden sind schließlich nur noch Atome der Form $C(d_i)$ in einer Abfrage $q = \{q_1, \ldots q_n\}$ enthalten, für welche nach Zusammenfassung $i \neq j \Rightarrow t_i \neq d_j$ und $d_i \in Inds(KB) \cup \mathcal{V}$ gilt und nun eine einfache Beantwortung von q ermöglichen:

$$A(q) = \begin{cases} \texttt{wahr}, & \text{falls } \bigwedge_{i=1}^{n}(KB \models q_i) \\ \texttt{falsch}, & \text{sonst.} \end{cases} \tag{2.6}$$

Einen prototypischen, aber vollständigen Algorithmus für die Transformation und anschließende Beantwortung einer Booleschen Abfrage q einer $\mathcal{ALC}$-Wissensbasis ist im Algorithmus 2.1 dargestellt. Ein anschauliches Beispiel dessen Arbeitsweise findet sich in Abbildung 2.2.

2.3.3 Beantwortung von Abfragen

Die Erweiterung auf Fälle, in denen die Variablen auch als Antwortvariablen interpretiert werden, ist unkompliziert, da es direkt auf die Beantwortung von Booleschen Abfragen reduziert werden kann.

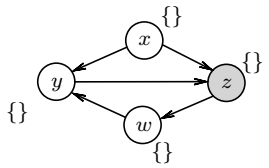
Gegeben sei also eine konjunktive Abfrage q für eine Wissensbasis $KB = (\mathcal{T}, \mathcal{A})$. Eine Antwort auf eine solche Abfrage ist eine Menge von Ersetzungen $\pi_j(q)$, welche jeder Antwortvariable ein Individuum aus $\mathcal{A}$ genau so zuweist, dass die Ersetzung $q[\pi_j(q)]$ logische Konsequenz von KB ist. Wie in Abschnitt 2.2 vorgestellt, sollen künstliche

Algorithmus 2.1 : Beantwortung von Booleschen Abfragen

Eingabe : Wissensbasis KB
Eingabe : Abfrage q
Daten : Menge $\mathcal{V}$ der Variablen aus q
Daten : Menge I der Individuen aus KB
Daten : Ersetzungsvektor $S = (S^1, \ldots, S^d)$, initial ist $d = 0$
Daten : Graph G von q
Daten : Neue Axiome $\mathcal{P}$
Ergebnis : `Wahr` oder `Falsch`
Beginn
  solange G *Kanten enthält* **tue**
3 **wenn** $\exists N \in G$ #`EingehendeKanten(`N`)` $= 1$ **dann**
    `Aufrollen( EingehendeKanten(`N`))` ;
5 **sonst wenn** $\exists N \in G$ #`EingehendeKanten(`N`)` > 1 **dann**
    wenn $N \in \mathcal{V}$ **dann**
      erweitere S um neues Element N
      `/* gegebenfalls werden neue Axiome in` $\mathcal{P}$ `abgelegt */`
    `Aufrollen( EingehendeKanten(`N`),` $\mathcal{P}$`)` ;
8 **sonst**
    Wähle beliebigen Knoten N ;
    wenn $N \in \mathcal{V}$ **dann**
      erweitere S um neues Element N
    `Aufrollen( EingehendeKanten(`N`),` $\mathcal{P}$`)` ;
  $KB' \leftarrow KB \cup \mathcal{P}$;
  Ermittle q' aus jetzt kantenlosem G ;
  $d \leftarrow$ Anzahl der Dimensionen von S ;
  für jedes *d-Tupel* $\hat{\imath}$ *aus* $I \otimes \ldots \otimes I$ **tue**
    für jedes $1 \leq l \leq d$ **tue**
      Ersetze in KB' und q' Variable S^l mit Individuum $\hat{\imath}^l$;
    wenn q' *in* KB' *mit* `wahr` *beantwortet wird* **dann**
      terminiere und liefere Ergebnis `wahr`
  liefere Ergebnis `falsch`;
Ende

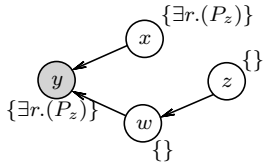
Für q_0 gilt $\mathcal{P} = \emptyset$ und $S = \emptyset$. Zur Übersichtlichkeit wird auf Kantenbeschriftungen verzichtet, welche in allen Fällen r lautete. Der markierte Knoten wird wegen Zeile 5 zuerst gewählt.

$$q_0 = \{r(x,z), r(z,w), r(w,y), r(y,z), r(x,y)\}.$$



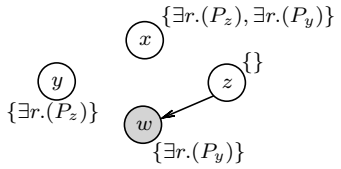
Es konnten zwei Kanten eliminiert werden und es gilt $\mathcal{P} = \{P_z(z)\}$ und $S = \{z\}$. Es wird nun Knoten y ausgewählt.

$$q_1 = \{(\exists r.(P_z))(x), r(z,w), r(w,y), (\exists r.(P_z))(y), r(x,y)\}.$$



Auch Variable y ist nun für Ersetzungen vorgemerkt und es gilt $\mathcal{P} = \{P_z(z), P_y(y)\}$ und $S = \{z, y\}$. Dadurch konnten erneut 2 Kanten eliminiert werden. Die letzte Kante wird gemäß Zeile 3 aufgerollt.

$$q_2 = \{(\exists r.(P_z))(x), r(z,w), (\exists r.(P_y))(w), (\exists r.(P_z))(y), (\exists r.(P_y))(x)\}.$$



Dies eliminiert die Variable w vollständig und führt keine neuen Ersetzungskandidaten ein. Die endgültige, transformierte Abfrage lautet nach Zusammenfassung:

$$q_3 = \{(\exists r.(P_z) \sqcap \exists r.(P_y))(x), (\exists r(\exists r.(P_y)))(z), (\exists r.(P_z))(y)\}.$$

Wobei $\mathcal{P} = \{P_z(z), P_y(y)\}$ und $S = \{z, y\}$ sind. Eine gültige Ersetzung mit dem Individuen-Tupel $\hat{\imath} = (a, b)$ würde zur leicht beantwortbaren Abfrage

$q' = \{(\exists r.(P_a) \sqcap \exists r.(P_b))(x), (\exists r(\exists r.(P_b)))(a), (\exists r.(P_a))(b)\}$ mit $\mathcal{P} = \{P_a(a), P_b(b)\}$ führen.

Abbildung 2.2: Transformation einer Abfrage nach Algorithmus 2.1

Antwortelemente der Form $\texttt{exists}_{v_i}$ zugelassen sein, welche nur die zwingende Existenz eines Elements der Interpretationsdomäne für die Variable v_i anzeigen.

1. Erstelle für jede Variable $v_i \in \mathcal{V}, i = 1 \ldots n, n \in \mathbb{N}$ eine Menge I_{v_i}, welche alle benannten Individuen der Wissensbasis, sowie zusätzlich eine neue, existenziell quantifizierte Variable $\texttt{exists}_{v_i}$ enthält.

2. Für jedes Tupel $c = (c_1, \ldots, c_n)$ der Menge von Antwortkandidaten $A_c = I_{v_1} \times \ldots \times I_{v_n}$ prüfe, ob die Boolesche Abfrage q', bei der die Antwortvariablen durch das entsprechende Element des n-Tupels ersetzt wurden, `wahr` zurückliefert. Ist dem so, füge c der Antwortmenge A hinzu.

3. Schließlich lässt sich Redundanz entfernen, indem wiederholt Tupel aus A entfernt werden, die sich von anderen Tupeln nur dadurch unterscheiden, dass sie an einigen Stellen statt eines Individuums das künstliche Element $\texttt{exists}_{v_i}$ enthalten. Ist dies nicht mehr möglich, gib A als Antwort zurück.

Diese Vorgehensweise ist offensichtlich recht ineffizient[6] und lässt sich einfach optimieren, z.B. indem die Mengen I_{v_i} bereits auf gewisse Kandidaten eingeschränkt und die Variablen $\texttt{exists}_{v_i}$ gesondert berücksichtigt werden. Solcherlei Optimierungen liegen jedoch nicht im Fokus dieser Arbeit.

Die gezeigte Vorgehensweise zur Beantwortung von Abfragen bezieht sich auf die Beschreibungslogik $\mathcal{ALC}$, ist jedoch mit unterschiedlichen Erweiterungen auch bei ausdrucksstärkeren Beschreibungslogiken anwendbar. Im Besonderen existieren Ergebnisse für wenigstens $\mathcal{SHIQ}$ und $\mathcal{SHOQ}$ [GLHS08].

[6] Tatsächlich handelt es sich bei der Beantwortung in dieser Form um einen „Brute Force“-Ansatz.

3 Ansätze für modulare Wissensbasen

Semantische Technologien haben sich zu einem ernstzunehmenden Werkzeug für die Lösung von praktischen Problemen entwickelt [PSHH04, MM08, BRM04, FHH+07]. Handelt es sich jedoch um die Realisierung eines großen, verteilten Informationssystems, stoßen die Sprachen aufgrund fehlender Fähigkeiten zur Aufteilung der Information unter Beibehaltung aller logischen Implikationen an ihre Grenzen. Beispielsweise ist, vom Standpunkt eines globalen Unternehmens aus betrachtet, eine Trennung der Informationen auf Basis der geografischen Verteilung seiner Niederlassungen oder analog seiner Abteilungen sinnvoll. Dies steht im Widerspruch zu dem monolithischen Ansatz, den die Semantik klassischer Beschreibungslogiken diktiert. Auch muss davon ausgegangen werden, dass viele Akteure gleichzeitig neue Informationen beisteuern werden, aber jeder die Kontrolle über einen gewissen Teil der Wissensbasis behalten möchte. Große, ungeteilte Wissensbasen sind demzufolge nicht wünschenswert, da nicht zuletzt die Verarbeitungsgeschwindigkeit bei ausdrucksstarken Beschreibungslogiken und noch recht einfachen Fragestellungen sehr schnell (z.B. doppelt exponentiell) mit der Größe der Wissensbasis ansteigen kann.

Die in der verbreiteten Sprache OWL vorgesehene Möglichkeit, mittels `owl:import` auf eine fremde Ontologie zu verweisen, ist dabei insoweit problematisch, als die Standard-Semantik einen vollständigen Import aller darin enthaltener Axiome erfordert. Dies führt praktisch wieder zu einer vollständig integrierten Wissensbasis. Um dieses Problem zu lösen, wurden viele Arbeiten veröffentlicht, aus denen vier ausgewählte Ansätze vorgestellt und diskutiert werden sollen [WBHQ07, CK07]. Dabei lassen sich diese grob in zwei Gruppen einteilen.

3.1 Modularisierung per Definition

Die erste Gruppe enthält Formalismen, die Syntax und Semantik um die Idee einer aus mehreren Komponenten bestehenden Wissensbasis erweitern. Dabei lässt sich die Berechenbarkeit der Reasoning-Probleme in einige Fällen durch eine Transformation in eine monolithische Wissensbasis zeigen. Dennoch ist in allen Fällen ein angepassten Entscheidungsverfahren notwendig, um Vorteile aus den gemachten Erweiterungen

ziehen zu können. Eine spezielle Zielstellung ist dabei, einen echt verteilten Algorithmus aufzuzeigen, welcher parallel und mit akzeptablem Kommunikationsaufwand implementiert werden kann.

3.1.1 $\mathcal{E}$-Connections

Ein Formalismus, konzipiert um unterschiedlichste, entscheidbare Logiken miteinander zu verbinden, solange sie nur als Abstract Description System (ADS) formulierbar sind wird von Oliver Kutz et al. unter dem Namen $\mathcal{E}$-Connections vorgestellt [KLWZ04]. Dabei werden die Interpretationsdomänen von n verbundenen Logiken über eine Menge von n-stelligen *Verbindungsrelationen* (Link Relations) realisiert. Die sehr kontrollierte Verknüpfung ermöglicht die Feststellung, dass die Entscheidbarkeit einer $\mathcal{E}$-Connection nur abhängig von der Entscheidbarkeit seiner Komponentenlogiken ist.

Die Idee, diesen Ansatz zur Realisierung von modularen Ontologien zu verwenden, wurde auch von Bernardo Cuenca Grau mit seinen Veröffentlichung zu $\mathcal{E}$-Connections von Beschreibungslogiken [CPS06] untersucht. Die ursprüngliche Intention wird dabei vereinfacht auf die Verknüpfung von Beschreibungslogiken unterschiedlicher Ausdrucksstärke, obwohl theoretisch Verknüpfungen nahezu beliebiger modaler, spatialer und temporaler sowie beschreibender Logiken möglich sind. Aufgrund dieser ursprünglichen Vielseitigkeit sollen in dieser Arbeit ganz allgemeine $\mathcal{E}$-Connections betrachtet werden, welche sich nur auf die Definitionen von ADS stützen.

Eine Übersetzungsfunktion $\cdot^{\#}$, welche eine Logik (z.B. $\mathcal{ALC}$) in das zugehörige ADS ($\mathcal{ALC}^{\#}$) transformiert, wurde bereits beschrieben [KLWZ04]. Hier sollen jedoch, so weit möglich, die üblichen Notationen der Komponentenlogiken direkt genutzt werden. Die Sprache der $\mathcal{E}$-Connection ist schließlich in gewisser Weise die Vereinigung der Sprachen der verknüpften Logiken, zuzüglich der Elemente, welche Aussagen über die Verbindungsrelationen machen können [CK07].

Zunächst sollen die Begriffe Abstract Description Language, Abstract Description Model und Abstract Description System analog [KLWZ04] formal eingeführt werden, da deren Verständnis grundlegend für den Großteil dieser Arbeit ist.

Definition 8: Abstract Description Language (ADL). *Eine* Abstract Description Language $\mathcal{L} = (\mathcal{V}, \mathcal{X}, \mathcal{R}, \mathcal{F})$ *ist ein Tupel bestehend aus einer Menge* $\mathcal{V}$ *von Mengenvariablen, einer Menge* $\mathcal{X}$ *von Objektvariablen, einer Menge* $\mathcal{R}$ *von Relationssymbolen* r_i *der Stelligkeit* m_{r_i} *und einer Menge* $\mathcal{F}$ *von Funktionssymbolen* f_i *der Stelligkeit* n_{f_i},

mit $\{\wedge, \neg\} \cap \mathcal{F} = \emptyset$. *Die Terme* t_j *von* $\mathcal{L}$ *werden induktiv konstruiert:*

$$t_j := x \mid \neg t_1 \mid t_1 \wedge t_2 \mid f(t_1, \ldots, t_{n_f}),$$

mit $x \in \mathcal{V}$ *und* $f \in \mathcal{F}$. *Die Termaussagen von* $\mathcal{L}$ *sind von der Form* $t_1 \sqsubseteq t_2$, *und die Objektaussagen sind von der Form* $(a_1, \ldots a_{m_r}) : r$ *oder* $a : t$, *mit* $a, a_1, \ldots, a_{m_r} \in \mathcal{X}$, $r \in \mathcal{R}$ *und einem Term* t.

Definition 9: Abstract Description Model (ADM). *Ein* Abstract Description Model *einer ADL* $\mathcal{L} = (\mathcal{V}, \mathcal{X}, \mathcal{R}, \mathcal{F})$ *ist eine Struktur*

$$\mathfrak{W} = (W, \mathcal{V}^{\mathfrak{W}}, \mathcal{X}^{\mathfrak{W}}, \mathcal{R}^{\mathfrak{W}}, \mathcal{F}^{\mathfrak{W}})$$

$$\mathcal{V}^{\mathfrak{W}} = \{x^{\mathfrak{W}} \mid x \in \mathcal{V}\}, \mathcal{X}^{\mathfrak{W}} = \{a^{\mathfrak{W}} \mid a \in \mathcal{X}\}, \mathcal{R}^{\mathfrak{W}} = \{r^{\mathfrak{W}} \mid r \in \mathcal{R}\}, \mathcal{F}^{\mathfrak{W}} = \{f^{\mathfrak{W}} \mid \in \mathcal{F}\}$$

wobei W *eine nicht-leere Menge,* $x^{\mathfrak{W}} \subseteq W$, $a^{\mathfrak{W}} \in W$, *jedes* $f^{\mathfrak{W}}$ *eine Funktion, welche* n_f*-Tupel von Teilmengen von* W *auf Teilmengen von* W *abbildet und jedes* $r^{\mathfrak{W}}$ *eine* m_r*-stellige Relation auf* W *ist. Die Semantik eines Terms* $t^{\mathfrak{W}}$ *wird nun induktiv definiert:*

$$(\neg t)^{\mathfrak{W}} = W \setminus (t^{\mathfrak{W}}), (t_1 \wedge t_2)^{\mathfrak{W}} = t_1^{\mathfrak{W}} \cap t_2^{\mathfrak{W}} \text{ und } f(t_1, \ldots, t_{n_f})^{\mathfrak{W}} = f^{\mathfrak{W}}(t_1^{\mathfrak{W}}, \ldots, t_{n_f}^{\mathfrak{W}})$$

Die Wahrheitsrelation $\mathfrak{W} \models \varphi$ für eine $\mathcal{L}$-Aussage φ wird wie folgt definiert

$$\begin{aligned}
\mathfrak{W} &\models (a_1, \ldots, a_{m_r}) : r && \iff (a_1^{\mathfrak{W}}, \ldots, a_{m_r}^{\mathfrak{W}}) \in r^{\mathfrak{W}} \\
\mathfrak{W} &\models a : t && \iff a^{\mathfrak{W}} \in t^{\mathfrak{W}} \\
\mathfrak{W} &\models t_1 \sqsubseteq t_2 && \iff t_1^{\mathfrak{W}} \subseteq t_2^{\mathfrak{W}} \qquad .
\end{aligned}$$

Gilt eine Aussage $\mathfrak{W} \models \varphi$, so **erfüllt** $\mathfrak{W}$ die Formel φ. Eine Menge Γ von Aussagen heißt erfüllt, wenn jede darin enthaltene Aussage erfüllt wird, was mit $\mathfrak{W} \models \Gamma$ notiert wird. Damit sind die Voraussetzungen für die formale Definition eines ADS geschaffen:

Definition 10: ADS. *Ein* Abstract Description System *ist ein Tupel* $(\mathcal{L}, \mathcal{M})$ *einer ADL* $\mathcal{L}$ *und einer Klasse* $\mathcal{M}$ *von ADMs. Dabei gilt für* $\mathcal{M}$, *dass für jedes Modell* $\mathfrak{W} \in \mathcal{M}$

1. *das nahezu identische Modell* $\mathfrak{W}'$, *welches nur den Variablen aus* $\mathcal{V}$ *andere Mengen zuweist, ebenso in* $\mathcal{M}$ *liegt.*

2. *das nahezu identische Modell $\mathfrak{W}'$, welches bis auf eine endliche Teilmenge $\mathcal{X}_\mathcal{G}$ die Objektvariablen aus $\mathcal{X}$ anders zuweist, ebenfalls in $\mathcal{M}$ liegt. Dabei wird $\mathcal{X}_\mathcal{G}$ determiniert durch eine entsprechenden Teilmenge $\mathcal{G}$ von Funktionen aus $\mathcal{F}$, so dass $a^\mathfrak{W} = a^{\mathfrak{W}'}$ für $a \in \mathcal{X}_\mathcal{G}$ und $f^\mathfrak{W} = f^{\mathfrak{W}'}$ für $f \in \mathcal{G}$ gilt.*

Die in der Definition genannten Bedingungen stellen einerseits sicher, dass die Variablen tatsächlich variabel interpretiert werden und andererseits, dass trotzdem Nominale dargestellt werden können[1].

Aufbauend auf diesen Begriffen für die Komponenten, kann nun deren Verknüpfung definiert werden. Seien $\mathcal{S}_1 \dots \mathcal{S}_n$ die zu verbindenden ADS, wobei $\mathcal{S}_i = (\mathcal{L}_i, \mathcal{M}_i)$ für $i \in [1, n]$ und $n \in \mathbb{N}$ ist. Es wird ohne Beschränkung der Allgemeinheit angenommen, das die Alphabete zweier ADS S_i und S_k mit $1 \leq i < k \leq n$, also deren Mengenvariablen, Objektvariablen, Funktionssymbole und Relationssymbole, disjunkt sind[2]. Eine Verbindung wird nun durch eine nicht-leere Menge $\mathcal{E} = \{E_j \mid j \in J\}$ von n-stelligen Relationssymbolen, sowie $(n-1)$-stelligen Funktionssymbolen $\langle E_j \rangle^i$ für jedes $j \in J$ und jedes $i \in [1, n]$ konstituiert, wobei J eine nicht-leere Indexmenge ist. Die Menge der Terme der elementaren $\mathcal{E}$-Connection Sprache $\mathcal{C}^\mathcal{E}(\mathcal{S}_1, \dots, \mathcal{S}_n)$ wird in n Mengen partitioniert. Dabei enthält die i-te Menge alle Terme aus $\mathcal{L}_i$ und heißt Menge der i-Terme. Zusätzlich gilt für i-Terme, falls $t_1, \dots, t_{i-1}, t_{i+1}, \dots t_n$ eine Sequenz von k-Termen t_k ist, mit $k \neq i$, so ist für alle $j \in J$

$$\langle E_j \rangle^i (t_1, \dots, t_{i-1}, t_{i+1}, \dots t_n) \text{ ebenfalls ein } i\text{-Term.}$$

Stellt man sich nun drei verbundene Logiken vor, welche jeweils die Grundlage für Aussagen über Firmen ($\mathcal{SROIQ}^\#$), Länder ($\mathcal{ALCO}^\#$) und Produkte ($\mathcal{ALC}^\#$) darstellen, ließen sich bereits interessante Aussagen formulieren, welche hier zunächst als Veranschaulichung von Termkonstruktionen dienen sollen:

- Alle Firmen verkaufen Produkte in Ländern

$$\texttt{Company} \sqsubseteq \langle \texttt{sells} \rangle^1 (\texttt{Country}, \texttt{Product})$$

[1] Für eine formale Definition dieser Abgeschlossenheitskriterien sei auf Definition 3 in [KLWZ04] verwiesen.

[2] Die Symbole werden ohnehin unabhängig interpretiert, so dass gleiche Namen nur zu unnötiger Verwirrung führen würden.

- Firmen können nichts in den USA verkaufen, wenn sie im Iran Waffen verkaufen

$$\langle \texttt{sells} \rangle^1(\{\texttt{Iran}\}, \texttt{Weapons}) \wedge \langle \texttt{sells} \rangle^1(\{\texttt{USA}\}, \top_3) \sqsubseteq \neg\top_1$$

- Produkte, die nur in den USA von „Start-Ups“ verkauft werden

$$\texttt{NextBigThing} \sqsubseteq \langle \texttt{sells} \rangle^3(\texttt{StartUp}, \{\texttt{USA}\}) \wedge \neg\langle \texttt{sells} \rangle^3(\top_2, \neg\{\texttt{USA}\})$$

Definition 11: Wissensbasis für $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$. *Die möglichen Aussagen in dieser Sprache sind einerseits i-Aussagen der Form $t_1 \sqsubseteq t_2$, $a : t$ und $(a_1, \ldots, a_{m_r}) : r$ über i-Terme t_1, t_2 und t, Objektvariablen $a, a_1, \ldots a_{m_r} \in \mathcal{V}_i$ und einer Relation $r \in \mathcal{R}_i$ und andererseits Aussagen über die Verbindungsrelationen aus $\mathcal{E}$ der Form $(a_1, \ldots, a_n) : E_j$ mit $a_i \in \mathcal{V}_i$ und $i \in [1, n]$. Eine Struktur $eKB = (\Gamma_1, \ldots, \Gamma_n, \Gamma_{\mathcal{E}})$ mit Γ_i einer Menge von i-Aussagen für $i \in [1, n]$ und $\Gamma_{\mathcal{E}}$ eine Menge von Aussagen über Verbindungsrelationen $E_j, j \in J$ heißt Wissensbasis für die $\mathcal{E}$-Connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$. Die Menge der in Γ_i vorkommenden Mengenvariablen aus $\mathcal{V}_i$ soll mit $\mathcal{V}(\Gamma_i)$ bezeichnet werden.*

Entgegen der Definition in [KLWZ04] wird hier eine Wissensbasis in $n+1$ Teilmengen zerlegt, welche einerseits den n Komponenten zugeordnet werden können und andererseits Aussagen über die Verbindungsrelationen enthalten. Damit wird der modulare Charakter auch innerhalb der Wissensbasis deutlich.

Definition 12: Sematik einer $\mathcal{E}$-Connection. *Für eine $\mathcal{E}$-Connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ heißt eine Struktur*

$$\mathfrak{M} = \langle \mathfrak{W}_1, \ldots, \mathfrak{W}_n, \mathcal{E}^{\mathfrak{M}} \rangle \text{ mit } \mathcal{E}^{\mathfrak{M}} = \left\{ E_j^{\mathfrak{M}} \mid j \in J \right\} \text{ und } \mathfrak{W}_i \in \mathcal{M}_i, 1 \leq i \leq n$$

Interpretation, wobei $E_j^{\mathfrak{M}} \subseteq W_1 \times \ldots \times W_n$ für jedes $j \in J$ ist. Für Mengen- und Objektvariablen X und a aus $\mathcal{L}_i$ gilt $X^{\mathfrak{M}} = X^{\mathfrak{W}_i}$ und $a^{\mathfrak{M}} = a^{\mathfrak{W}_i}$. Für eine Relation r aus $\mathcal{R}_i$ gilt $r^{\mathfrak{M}} = r^{\mathfrak{W}_i}$. Die Interpretation eines Terms $t^{\mathfrak{M}}$ wird induktiv definiert und ähnelt Definition 9:

$$(\neg t)^{\mathfrak{M}} = W_i \setminus (t^{\mathfrak{M}}),\ (t_1 \wedge t_2)^{\mathfrak{M}} = t_1^{\mathfrak{M}} \cap t_2^{\mathfrak{M}}$$
$$\text{und für } f \in \mathcal{F}_i \qquad f(t_1, \ldots, t_{n_f})^{\mathfrak{M}} = f^{\mathfrak{W}_i}(t_1^{\mathfrak{M}}, \ldots, t_{n_f}^{\mathfrak{M}})$$

Zusätzlich müssen die Aussagen über die Verbindungsrelationen interpretiert werden, dazu soll $\bar{t}_i = (t_1, \ldots, t_{i-1}, t_{i+1}, \ldots t_n)$ eine Sequenz von $(n-1)$ j-Termen t_j sein.

Dann gilt

$$(\langle E_j \rangle^i (\bar{t}_i))^{\mathfrak{M}} = \left\{ x \in W_i \mid \exists_{\substack{l=1 \\ l \neq i}}^{n} x_l \in t_l^{\mathfrak{M}} \quad (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n) \in E_j^{\mathfrak{M}} \right\}.$$

Die Wahrheitsrelation für eine Aussage φ wird nun definiert als:

$$\begin{aligned}
\mathfrak{M} &\models (a_1, \ldots, a_{m_r}) : r && \iff (a_1^{\mathfrak{M}}, \ldots, a_{m_r}^{\mathfrak{M}}) \in r^{\mathfrak{M}} \\
\mathfrak{M} &\models (a_1, \ldots, a_n) : E_j && \iff (a_1^{\mathfrak{M}}, \ldots, a_n^{\mathfrak{M}}) \in E_j^{\mathfrak{M}} \\
\mathfrak{M} &\models a : t && \iff a^{\mathfrak{M}} \in t^{\mathfrak{M}} \\
\mathfrak{M} &\models t_1 \sqsubseteq t_2 && \iff t_1^{\mathfrak{M}} \subseteq t_2^{\mathfrak{M}} \\
\mathfrak{M} &\models (\varphi_1 \wedge \varphi_2) && \iff (\mathfrak{M} \models \varphi_1) \wedge (\mathfrak{M} \models \varphi_1)
\end{aligned}$$

Wie auch schon bei ADS wird eine Aussage φ als erfüllt in $\mathfrak{M}$ bezeichnet, falls $\mathfrak{M} \models \varphi$ gilt. Eine Wissensbasis $eKB = (\Gamma_1, \ldots, \Gamma_n, \Gamma_{\mathcal{E}})$ für $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ heißt erfüllbar, wenn eine Interpretation $\mathfrak{M}$ existiert, welche jede Aussage $\varphi_i \in \Gamma_1 \cup \ldots \cup \Gamma_n \cup \Gamma_{\mathcal{E}}$ erfüllt und in diesem Fall **Modell** genannt wird. Dies wird als $\mathfrak{M} \models eKB$ notiert.

Der grundlegende Ansatz lässt sich erweitern, indem zusätzliche Konstruktoren für Aussagen über die Verbindungsrelationen zugelassen werden. Eine Erweiterung[3] soll hier exemplarisch vorgestellt werden.

Definition 13. *Gegeben sei eine $\mathcal{E}$-Connection $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ gemäß Definition 11 und 12. In der erweiterten $\mathcal{E}$-Connection $\mathcal{C}_{\mathcal{O}}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ wird die Definition von i-Termen um folgenden Punkt erweitert: Wenn $\bar{a}_i = (a_1, \ldots a_{i-1}, a_{i+1}, \ldots, a_n)$ eine Sequenz von Objektvariablen a_j aus $\mathcal{X}_j$ ist, dann ist $\langle E_k \rangle^i \bar{a}_i$ ebenfalls ein i-Term für alle $k \in J$. Die Semantik des neuen Termkonstruktors ist bei gegebenen Modell $\mathfrak{M} = \langle \mathfrak{W}_1, \ldots, \mathfrak{W}_n, \mathcal{E}^{\mathfrak{M}} \rangle$ und einer Sequenz $\bar{a}_i$ wie folgt:*

$$(\langle E_j \rangle^i \bar{a}_i)^{\mathfrak{M}} = \left\{ x \in W_i \mid \left(a_1^{\mathfrak{M}}, \ldots a_{i-1}^{\mathfrak{M}}, x, a_{i+1}^{\mathfrak{M}}, \ldots, a_n^{\mathfrak{M}} \right) \in E_j^{\mathfrak{M}} \right\}$$

Das Besondere hierbei ist die Verwendung von Objektvariablen bei der Termkonstruktion und somit die Möglichkeit, Nominale aus verbundenen Logiken verwenden zu können, *unabhängig davon*, ob diese Nominale unterstützen. Diese Erweiterung

[3] Für eine Vorstellung anderer Erweiterungen sei der interessierte Leser erneut auf [KLWZ04] verwiesen.

beinhaltet die grundlegende Variante von $\mathcal{E}$-Connections vollständig, so dass Ergebnisse für erstere auch unmittelbar für den Basisfall gelten.

Das wesentliche Resultat, welches nun formal wiedergegeben werden soll, bezieht sich auf das Erfüllbarkeitsproblem:

Satz 1: [KLWZ04]. *Seien $\mathcal{S}_1, \ldots, \mathcal{S}_n$ ADS mit entscheidbaren Erfüllbarkeitsproblemen. Dann ist das Erfüllbarkeitsproblem für jede $\mathcal{E}$-Connection $\mathcal{C}^{\mathcal{E}}_{\mathcal{O}}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ ebenfalls entscheidbar.*

Das Hauptziel von $\mathcal{E}$-Connections wird demzufolge erreicht: Die robuste Kombination unterschiedlicher, als ADS formulierbarer Logiken. Dabei sind der Anzahl der Komponenten sowie der Menge der nutzbaren Logiken in der Theorie keine Grenzen gesetzt. Gerade diese Vielfalt von Sprachen, welche jeweils durch konkrete Kombinationen von Logiken entstehen, macht es schwierig, ein allgemeines und praktikables Entscheidungsverfahren zu ermitteln. Dieses kann in der Praxis nicht völlig unabhängig von den Entscheidungsverfahren für die Komponentenlogiken sein. So zeigt sich erst in konkreten Fällen, ob ein verteiltes Entscheidungsverfahren möglich ist.

Aufgrund der Fokussierung dieser Arbeit auf $\mathcal{E}$-Connections und des dringenden Bedarfs für Modularisierung im Semantic Web Kontext, ist es naheliegend, deren Eignung dafür detaillierter zu bewerten. So ist es wünschenswert, über eine einfache Modularisierung die gemeinschaftliche Modellierung von Wissensbasen zu ermöglichen bzw. zu erleichtern [WBHQ07]. Zwischen den Modulen sollen Symbole wiederverwendet, aber auch spezialisiert und generalisiert werden können. Letztlich ist es wichtig, eine einfache, klare und möglichst unmissverständliche Semantik anbieten zu können [CP08].

Eine Anwendung des $\mathcal{E}$-Connection Ansatzes auf OWL wurde in [CPS06] vorgenommen. Dabei wird Syntax und Semantik von OWL modifiziert, um binäre[4] und gerichtete $\mathcal{E}$-Connections darstellen zu können. Diese können dann aus mehreren, die Komponentenlogiken repräsentierenden OWL-Dateien konstruiert werden. Es wird der Versuch deutlich, die Verbindungsrelationen sehr an die aus Beschreibungslogiken bekannten Rollen anzunähern. Ein Tableau-basiertes Entscheidungsverfahren für $\mathcal{E}$-Connections von Beschreibungslogiken wurde ebenfalls entwickelt.

Es lassen sich die folgenden drei Punkte als wesentlich für eine Einschätzung heranziehen:

[4] Dabei werden mit jeder Verbindungsrelation nur zwei Logiken verbunden.

Möglichkeit der Partitionierung Ein Vorteil von $\mathcal{E}$-Connections von Beschreibungslogiken ist, dass bei der Partitionierung einer Ontologie in den meisten Fällen Teilontologien geringerer Ausdrucksstärken entstehen, was Potenzial für Optimierungen birgt. Wie Bernardo Cuenca Grau zeigt, lassen sich bereits existierenden Ontologien über den $\mathcal{E}$-Connection Ansatz automatisiert partitionieren [CPSK05].

Disjunkte Alphabete Eine intuitive Realisierung modularer Ontologien ist nicht möglich. Es muss entschieden werden, wie eine insgesamt zu modellierende Domäne partitioniert werden kann. Dabei ist es schwierig, die möglichen Nutzungsarten und Erweiterungen einer Ontologie vorauszuahnen, da eine spätere Vermischung der Vokabulare nicht mehr möglich ist [CP08]. Sollen stattdessen existierende Ontologien wiederverwendet werden, speziell im Sinne einer Verfeinerung oder Generalisierung von Konzepten, so versagt der $\mathcal{E}$-Connection-Ansatz.

Ungenügende Werkzeugunterstützung Es fehlt im Vergleich zu anderen Ansätzen an Werkzeugen zur Arbeit mit $\mathcal{E}$-Connections. Aufgrund der Vielseitigkeit ist es sehr schwierig eine allgemeine Syntax zu entwerfen, so dass einzelne Lösungen wie die OWL-Erweiterung durch Bernardo Cuenca Grau nur ein kurzes Nischendasein führten[5]. Selbst der darin erwähnte Reasoner Pellet, welcher in Version 1.4 eine Erweiterung von Cuenca Grau enthält [CPS04], konnte erst durch mehrere Fehlerbehebungen erfolgreich zur Klassifizierung und Realisierung einiger Beispielontologien verwendet werden. Da diese Erweiterung aufgrund deutlich erhöhter Komplexität bereits wieder fallengelassen worden ist, existieren praktisch keinerlei Werkzeuge, welche die Entwicklung einer $\mathcal{E}$-Connection-Implementierung unterstützen könnten.

Dies führt dazu, dass $\mathcal{E}$-Connections wenig für einen Einsatz im Semantic Web geeignet sind. Die Möglichkeiten zur Verwendung fremder Symbole sind aufgrund der eingesetzten Sprache stark beschränkt. Trotz der Arbeiten von Bernardo Cuenca Grau, gelang es nicht, die Semantik intuitiv nachvollziehbar zu gestalten. Unerwartete Konsequenzen aufgrund unbedarfter Verknüpfung sind damit nicht auszuschließen. Schließlich wurde dieser Ansatz mit anderer Zielstellung entworfen, weswegen die ausschließliche Verknüpfung von Beschreibungslogiken schon an sich als Sonderfall betrachtet werden kann. Als problematisch kann desweiteren gelten, dass $\mathcal{E}$-Connections in gewissen Fällen inkompatibel mit `owl:import` sind [WBHQ07]. Damit würden die in vielen Szenarien üblichen Architekturen von mehrschichtigen Ontologien [CJB99]

[5] SWOOP wurde dabei als Editor genannt, in praktischen Szenarien ist er jedoch wegen des frühen Entwicklungsstadiums und der nicht fortgeführten Entwicklung unnütz.

potentielle Quelle unerwarteter Inkonsistenzen.

Ihre Stärken spielen $\mathcal{E}$-Connections aus, wenn multimodale Aussagen gemacht werden sollen, wie beispielsweise „was war wann und wo Instanz eines bestimmten Konzeptes“ oder „wo war ein Individuum bereits überall“. Die Formulierung solchen Wissens gelingt einfach und nachvollziehbar, solange die verwendeten Logiken bekannt sind. Für praktische Einsätze hemmend ist dabei jedoch die fehlende Standardisierung der Wissensrepräsentation außerhalb von Beschreibungslogiken, so dass eine Anwendung von $\mathcal{E}$-Connections erst in konkreten, eng abgegrenzten Szenarien empfehlenswert ist.

3.1.2 Distributed Description Logics

Einen weiteren Ansatz zur Aufteilung von Wissensbasen entwickelten Alex Borgida und Luciano Serafini mit dem Ziel, die Vorteile von Beschreibungslogiken auch bei lose gekoppelten Informationsquellen zu nutzen [BS02, BS03]. Das Ergebnis wird DDL genannt und im Folgenden vorgestellt. Ein Paar $\mathfrak{D} = (\mathcal{S}_1, \mathcal{S}_2)$ zweier gegebener Beschreibungslogiken $\mathcal{S}_1$ und $\mathcal{S}_2$ wird **Distributed Description Logics**[6] genannt. Die Verbindung der beiden Komponenten einer DDL wird über zweierlei Arten von Aussagen, *Bridge Rules* (Verbindungsregeln) und *Individual Correspondences* (Individuenentsprechungen), hergestellt.

Definition 14: Verbindungsregeln. *Seien C_i und C_j Konzepte aus $\mathcal{S}_i$ bzw. $\mathcal{S}_j$. Eine* Verbindungsregel *ist eine Aussage der Form*

$$C_i \xrightarrow{\sqsubseteq} C_j \qquad \text{(hinein-Regel)}$$

oder der Form

$$C_i \xrightarrow{\sqsupseteq} C_j. \qquad \text{(darüber-Regel)}$$

Definition 15: Individuenentsprechungen. *Sei a_i eine Bezeichnung für ein Objekt aus der Komponente $\mathcal{S}_i$ und $b_j, b_j^1, b_j^2, \ldots, b_j^n$ Bezeichnungen für Objekte aus $\mathcal{S}_j$. Dann ist eine* teilweise Individuenentsprechung *eine Aussage der Form*

$$a_i \longmapsto b_j \qquad \text{(TIE)}$$

[6] Vereinfachend wird hier nur die Verknüpfung zweier Komponenten betrachtet. Die Verbindung von n Komponenten ist ebenso möglich, geschieht jedoch auch paarweise.

und eine vollständige Individuenentsprechung *eine Aussage der Form*

$$a_i \overset{=}{\longmapsto} \{b_j^1, \ldots, b_j^n\}. \tag{VIE}$$

Definition 16: Verteilte Wissensbasis. *Eine* verteilte Wissensbasis $\mathcal{K}$ *ist ein Paar* $(\mathfrak{T}, \mathfrak{A})$. *Die verteilte TBox* $\mathfrak{T}$ *besteht aus den TBoxen* $\mathcal{T}_i$ *der beteiligten* $\mathcal{S}_i$ *sowie einer Menge aus Verbindungsregeln. Die verteilte ABox* $\mathfrak{A}$ *besteht aus den jeweiligen ABoxen* $\mathcal{A}_i$ *der* $\mathcal{S}_i$ *sowie einer Menge an Individuenentsprechungen.*

Definition 17: Semantik. *Eine* verteilte Interpretation $\mathfrak{I}$ *einer verteilten Wissensbasis* $\mathcal{K}$ *ist ein Paar* $(\{\mathcal{I}_i\}_{i \leq n}, \mathcal{R})$ *mit* $\mathcal{I}_i$ *einem Modell von* $\mathcal{S}_i$ *und* $\mathcal{R}$ *einer Funktion, welche jedem Paar* $(i, j), i \neq j$ *eine binäre Relation* $r_{ij} \in \Delta^i \times \Delta^j$ *zwischen den Domänen* Δ^i *und* Δ^j *von* $\mathcal{I}_i$ *bzw.* $\mathcal{I}_j$ *zuweist. Für ein Element* $u \in \Delta^i$ *und einer Teilmenge* $U \subseteq \Delta^i$ *wird definiert*

$$r_{ij}(u) = \{v \in \Delta^j \mid (u, v) \in r_{ij}\}, \qquad r_{ij}(U) = \bigcup_{u \in U} r_{ij}(u)$$

Die Wahrheitsrelation für Aussagen innerhalb der TBoxen und ABoxen der Komponenten gilt gemäß Abschnitt 2.1, für die zusätzlichen Verbindungsregeln und Individuenentsprechungen gilt:

$$\begin{aligned}
\mathfrak{I} \models C_i \overset{\sqsubseteq}{\longrightarrow} C_j \quad & gdw. \quad r_{ij}(C_i^{\mathfrak{I}}) \subseteq C_j^{\mathfrak{I}} \\
\mathfrak{I} \models C_i \overset{\sqsupseteq}{\longrightarrow} C_j \quad & gdw. \quad r_{ij}(C_i^{\mathfrak{I}}) \supseteq C_j^{\mathfrak{I}} \\
\mathfrak{I} \models a_i \longmapsto b_j \quad & gdw. \quad b_j^{\mathfrak{I}} \in r_{ij}(a_i^{\mathfrak{I}}) \\
\mathfrak{I} \models a_i \overset{=}{\longmapsto} \{b_j^1, \ldots, b_j^n\} \quad & gdw. \quad r_{ij}(a_i^{\mathfrak{I}}) = \{(b_j^1)^{\mathfrak{I}}, \ldots, (b_j^n)^{\mathfrak{I}}\}
\end{aligned}$$

Es wird deutlich, dass auch in DDL Relationen zwischen den Domänen der lokalen Interpretationen zum Einsatz kommen, um die Verknüpfung zu realisieren. Dabei handelt es sich jedoch um gerichtete und binäre Relationen, von denen genau eine für jedes Paar von Wissensbasen existiert. Ein verteilter Reasoningalgorithmus wurde implementiert und ist in [ST04] beschrieben. Das entwickelte Peer-To-Peer (P2P)-Reasoning System mit dem Namen „DRAGO“ ist ein Netzwerk sogenannter „DRAGO Reasoning Peers“ (DRP), welche jeweils eine Ontologie repräsentieren [ST05b, ST05a]. Es basiert auf der Idee eines verteilten Tableau-Algorithmus, in welchem durch die Verbindungsregeln determinierte Teiltableaus im Kontext der jeweils verbundenen Onto-

logie erstellt werden können. Diese Teilberechnungen werden von den DRPs an den zuständigen Knoten delegiert.

Als Erweiterung zu OWL realisiert, werden die Verbindungsregeln bzw. die Individuenentsprechungen in separaten Dateien abgelegt, was dazu führt, dass ein Ontologie-Entwickler zunächst nicht von fremden Konzepten profitieren kann. Andererseits lassen sich schon existierende Ontologien dadurch einfach wiederverwenden.

Jüngere Untersuchungen von DDL ergaben weitere Ergebnisse und im Besonderen den formalen Zusammenhang zu $\mathcal{E}$-Connections. So werden die Verbindungsregeln – die Notation legt eine Ähnlichkeit zu Subsumptionsbeziehungen nahe – in einigen Fällen nicht im erwarteten Maße propagiert. Eine mögliche Lösung wird in [Hom08] vorgeschlagen, indem die von *Package-based Description Logics* (siehe nächsten Abschnitt) bekannte *Compositional Consistency Condition*[7] auf DDL übertragen wird. In [CK07, Cue05, KLWZ04] wird schließlich festgestellt, dass bezüglich der Ausdruckskraft DDL selbst einer äußerst eingeschränkten Form von $\mathcal{E}$-Connections[8] unterlegen sind und daher von ihnen dargestellt werden können[9].

3.1.3 Package-based Description Logics

Den Kanon der drei populärsten modularen Ontologiesprachen komplettiert Jie Baos P-DL [BSH07], die eine partielle Import-Relation zur Verknüpfung mehrerer $\mathcal{SHOIQ}$-Ontologien verwendet. Dazu soll im folgenden eine kurze Definition analog [CK07] gegeben werden.

Es sei $\mathcal{S}$ eine Beschreibungslogik, $\mathcal{T}$ eine in $\mathcal{S}$ verfasste TBox und S eine Teilmenge der Signatur von $\mathcal{T}$. Ein Package wird durch $\mathcal{T}$ und S konstituiert, wobei S diejenigen Symbole enthält, deren Bedeutung „innerhalb“ von $\mathcal{T}$ definiert ist, während die Bedeutung der verbleibenden Symbole aus $Sig(\mathcal{T})$ von anderen Packages „geliehen“ wird. Formal werden Packages wie folgt definiert:

Definition 18: Package. *Ein* ***Package*** $\mathcal{P}_i = \langle Ho(\mathcal{P}_i), \mathcal{T}_i) \rangle$ *ist durch eine TBox* $\mathcal{T}_i$ *und einer Signatur* $Ho(\mathcal{P}_i) \subseteq Sig(\mathcal{T}_i)$ *gegeben. Ist ein Symbol* t *Element von* $Ho(\mathcal{P}_i)$,

[7] Transitivität der Domänen-Relation.

[8] Den sog. „negationsfreien, unär verknüpften, gerichteten $\mathcal{E}$-Connections“.

[9] Die vollständige Individuenentsprechung ist jedoch nur durch eine Erweiterung von $\mathcal{E}$-Connections modellierbar, führt dann aber nicht mehr zu einem allgemeinen Transfer der Entscheidbarkeit.

*so heißt $\mathcal{P}_i$ **Home Package** von t. Enthält ein (möglicherweise komplexes) Konzept C wenigstens ein Symbol aus $Ho(\mathcal{P}_i)$, so wird es P_i-Konzept genannt.*

Angenommen, es soll eine Menge von Packages $\mathcal{P}_1, \ldots, \mathcal{P}_n$ der Logik $\mathcal{S}$ integriert werden. Jedes Symbol wird nur in seinem Home Package $\mathcal{P}_i$ definiert und in anderen Packages $\mathcal{P}_j, j \neq i$ lediglich wiederverwendet, indem die Bedeutung importiert wird. Es soll angenommen werden, dass für jedes Symbol $s \in \bigcup_{i=1\ldots n} \mathcal{P}_i$ genau ein Home Package existiert.

Definition 19: Import-Deklaration. *Gegeben sind zwei Packages $\mathcal{P}_i, \mathcal{P}_j, i \neq j$ und ein Symbol $s \in Ho(\mathcal{P}_j)$. Eine **Import-Deklaration** für $\mathcal{P}_i$ ist ein Ausdruck der Form $\mathcal{P}_j \xrightarrow{s} \mathcal{P}_i$. Man sagt „$\mathcal{P}_i$ importiert s von $\mathcal{P}_j$“.*

Damit sind die Voraussetzungen für die Definition einer Package-basierten Ontologie geschaffen.

Definition 20: Package-basierte Ontologie. *Ein Paar $\mathcal{O} = (\mathfrak{P}, \mathfrak{I})$ ist eine Package-basierte Ontologie, wenn $\mathfrak{P} = (\mathcal{P}_1 \ldots \mathcal{P}_n)$ ein n-Tupel von Packages mit $Ho(\mathcal{P}_i) \cap Ho(\mathcal{P}_j) = \emptyset$ für $i \neq j$ und $n \in \mathbb{N}$, sowie $\mathfrak{I} = \{\alpha_k\}_{k \leq m}$ eine endliche Menge von Import-Deklarationen für Packages aus $\mathfrak{P}$ ist.*

Die Semantik für eine solche Ontologie weist jedem Package ein lokales Modell $\mathcal{I}_i$ zu [CK07]. Importiert ein Package $\mathcal{P}_i$ ein Symbol s aus $\mathcal{P}_j$, so müssen die beiden lokalen Modelle dieses Symbol identisch interpretieren. Formal wird dies durch bijektive Relationen $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ festgeschrieben, für welche zusätzlich $r_{ij} \circ r_{jk} = r_{ik}$ gilt. Dies weitet die Bedingungen auf mehrere Packages aus. Ein verteilt arbeitendes Entscheidungsverfahren auf Tableau-Basis und einer P2P-Vernetzung wurde beschrieben [BCH06], eine Implementierung ist noch nicht vorhanden.

Dieser Formalismus wurde direkt als Lösung für eine Anwendung von verteilten Ontologien im Semantic Web konzipiert und hält dafür vielversprechende Möglichkeiten bereit. So wurde in [CK07] gezeigt, dass auf die Import-Deklarationen o.B.d.A. verzichtet werden kann, womit die verwendbaren Sprachen klassischen Beschreibungslogiken bis hin zur Ausdruckskraft von $\mathcal{SHOIQ}$ entsprechen. Desweiteren lassen sich laut Bao sowohl DDLs als auch einige spezielle $\mathcal{E}$-Connections[10] auf P-DLs reduzieren [BSH07]. Dies zeigt erneut die Verwandtschaft der drei bis hier vorgestellten

[10] One-Way-$\mathcal{E}$-Connections der Sprache $C^{\mathcal{E}}_{\mathcal{IHQ}}(\mathcal{SHOIN})$.

Formalismen.

3.2 Modularisierung durch Analyse

Die zweite Möglichkeit zur Modularisierung versucht die standardisierte Syntax und Semantik beizubehalten, stellt jedoch neue Analysemethoden vor, die die verwendeten Ontologien auf eine modulare Struktur untersuchen. Im Besonderen wird eine einzige, globale Interpretation der beteiligten Wissensbasen beibehalten. Die Grundidee ist, dass der Import ganzer Ontologien durch den Import nur derjenigen Axiome ersetzt wird, welche für die daraus verwendeten Symbole von Bedeutung sind. Das heißt, man sucht nach der kleinsten Teilmenge der Axiome der verwendeten Ontologie, aus welcher alle Schlussfolgerungen bezüglich der verwendeten Symbole genau so gezogen werden können, als ob alle Axiome importiert würden. Dabei soll die Bedeutung der importierten Symbole nicht verändert werden, was formal durch den Begriff der konservativen Erweiterung (Conservative Extension) beschrieben wurde [LWW07].

Definition 21: Konservative Erweiterung. *Seien $\mathcal{T}_1$ und $\mathcal{T}_2$ TBoxen einer Beschreibungslogik $\mathcal{L}$ und $S \subseteq Sig(\mathcal{T}_1)$ eine Signatur. $\mathcal{T}_1 \cup \mathcal{T}_2$ ist eine S-konservative Erweiterung von $\mathcal{T}_1$, wenn für alle $C_1, C_2 \in \mathcal{L}(S)$ gilt:*

$$\mathcal{T}_1 \models C_1 \sqsubseteq C_2 \iff \mathcal{T}_1 \cup \mathcal{T}_2 \models C_1 \sqsubseteq C_2.$$

Es stellt sich heraus, dass die Frage nach der konservative Erweiterung schon in $\mathcal{ALCQIO}$ nicht mehr entscheidbar ist [LWW07]. Übrig bleiben beispielsweise Approximationen wie Locality [JRCS+08], welche syntaktisch schon mit polynomialen Aufwand überprüft werden kann und auf dem Begriff der Sicherheit von Ontologien beruht [JRCS+08].

Definition 22: Sicherheit bzgl. einer Signatur. *Sei $\mathcal{T}$ eine Ontologie und S eine Signatur. $\mathcal{T}$ heißt sicher für S, wenn für jede Ontologie $\mathcal{T}'$ mit $Sig(\mathcal{T}) \cap Sig(\mathcal{T}') \subseteq S$ die Vereinigung $\mathcal{T} \cup \mathcal{T}'$ eine konservative Erweiterung von $\mathcal{T}'$ ist.*

Dies konnte durch Locality approximiert werden. Intuitiv formuliert spricht man dann von $\top$-Locality ($\bot$-Locality), falls die importierten Konzepte ausschließlich generalisiert (wiederverwendet/spezialisiert) werden sollen [JRCS+08]. Dies führt zu einer stärkeren Einschränkung als für Sicherheit nötig ist, so das auch Ontologien sicher sein können, für welche Locality nicht nachgewiesen werden kann. Damit eine Modulex-

traktion für eine von $\mathcal{T}$ importierte Ontologie $\mathcal{T}_1$ ohne Nebeneffekte möglich ist, muss $\mathcal{T}$ sicher bzw. lokal bezüglich der importierten Symbole aus $\mathcal{T}_1$ sein. Die eigentliche Bestimmung eines auf Locality basierenden Moduls besteht aus der Konstruktion einer Menge von Symbolen, welche für die Schlussfolgerungen der importierten Symbole relevant sind. Schließlich werden alle Axiome, welche wenigsten ein Symbol aus dieser Menge enthalten, als Ergebnismodul zurückgegeben [JRCS+08].

Bedingt durch unveränderte Syntax und Semantik integriert sich dieser Ansatz elegant mit bestehenden Standards. So können für diese Art der Modularisierung Werkzeuge entwickelt werden, welche auf bereits vorhandene, Standard-konforme Infrastruktur aufsetzen und vollständig damit kompatibel sind [CP08].

3.3 Abgrenzung

Die vorgestellten Ansätze modularer Ontologien verknüpfen mehrere Wissensbasen auf semantischer Ebene. Dazu werden die Interpretationen der einzelnen Wissensbasen mit Bedingungen verknüpft, welche die Interpretationsdomänen in Beziehung setzen und eine globale Betrachtung des Verbunds erfordern. Damit muss der Verbund als Einheit verstanden werden, welchem unter Umständen nur dann eine Semantik zugewiesen werden kann, wenn alle beteiligten Wissensbasen verfügbar sind. Dies unterscheidet sie deutlich von lose gekoppelten Ansätzen, von denen einige im Abschnitt 4 vorgestellt werden.

Aber auch untereinander sind wesentliche Unterschiede festzustellen. So verändern $\mathcal{E}$-Connections die zugrundeliegende Semantik wohl am deutlichsten, da sie eine weitere Abstraktionsstufe oberhalb der zur Verknüpfung geeigneten Logiken darstellen. Dabei sind aufgrund der Vielseitigkeit nur sehr eingeschränkte Verknüpfungen mittels der vorgestellten Verbindungsrelationen möglich. Verknüpfungen mit Sprachelementen der Komponenten (z.B. Subsumption) sind nur möglich, sofern sie sich über Verbindungsrelationen darstellen lassen. Ähnlich lassen sich die der Semantik von DDL zugrundeliegenden Domänenrelationen auffassen. Diese sind zwar einerseits binär und andererseits implizit durch die Verbindungsregeln und Individuenentspechungen festgelegt, haben jedoch mit der Verknüpfung von Elementen der Interpretationsdomänen eine ganz ähnliche Aufgabe. Dabei ist die Syntax der Verbindungsregeln etwas irreführend, da es sich dabei nicht um echte Subsumptionen handelt – stattdessen lässt sich eine ähnliche Subsumptionspropagierung auch in $\mathcal{E}$-Connections erreichen. Eine vorhandene und auch durch den engen Fokus ermöglichte Implementierung gehört

zweifelsohne zu den Stärken von Distributed Description Logics. Schließlich finden sich in den Package-based Description Logics erneut Relationen, welche Elemente der Domänen verbinden. In diesem Ansatz ist eine nahezu transparente Verwendung fremder Symbole möglich, solange gewisse Regeln eingehalten werden. Da keine Implementierung existiert und noch keine belastbaren Ergebnisse bzgl. der Entscheidbarkeiten vorliegen [CK07] lassen sich jedoch noch keine Empfehlungen abgeben.

Als problematisch bezüglich der definierten Modularisierung kann jedoch gelten, dass auch in absehbarer Zukunft keine modulare Sprache Teil des OWL-Standards werden wird. Damit werden Werkzeuge und Reasoner einer stetig wachsenden Infrastruktur für klassische Beschreibungslogiken bzw. OWL immer unterlegen sein. Des Weiteren ist anzunehmen, dass Modellierer nur in seltenen Fällen über ein ausreichend fundiertes Logik-Wissen verfügen, um die teilweise deutlich modifizierten Semantiken verinnerlichen zu können. Um dieses Problem zu umschiffen wird beispielsweise von den Entwicklern des freien Reasoners Pellet inzwischen die oben beschriebene Modularisierung durch Analyse weiter verfolgt. Das geschah nach Aufgabe der Implementierung eines $\mathcal{E}$-Connection-basierten Ansatzes, auf welchen nur unbefriedigende Resonanz aus den Reihen der Wissens-Ingenieure zu spüren war [CP08].

Der vorgestellte Ansatz von analytischer Modularisierung ist aufgrund seiner Integration mit bestehenden Ontologiesprachen interessant. Es sind wegen der analytischen Natur keinerlei Nebenwirkungen zu erwarten, so dass bestehende Standards übernommen werden können. Die Weiterentwicklung zu einem praktisch einsetzbaren Produkt, welches konjunktive Abfragen auf solchen modular importierten Ontologien ermöglichen kann, ist neben anderen verwandten Arbeiten im folgenden Kapitel vorgestellt.

4 Verwandte Arbeiten

Verwandte Arbeiten lassen sich in zwei Gruppen einteilen. Auf der einen Seite stehen Arbeiten, welche sich mit konjunktiven Abfragen auf Ontologien beschäftigen und im Rahmen konkreter Logiken Verfahren zu deren Beantwortung aufzeigen. Gemessen an den Zeitpunkten der entsprechenden Veröffentlichungen entwickelte sich die Forschung in diesem Bereich erst, als bereits ein Bedarf zur bequemen Abfrage großer ABoxen entstanden war. Ausgehend von vergleichsweise einfachen Abfragemechanismen in Form der üblichen Reasoningdienste, begann die Auseinandersetzung mit mächtigeren Abfragesprachen außerdem signifikant später als im Datenbankbereich.

Für diese Arbeit äußerst relevant war die Beschreibung von konjunktiven Abfragen in der einfachen Beschreibungslogik $\mathcal{ALC}$ von Ian Horrocks und Sergio Tessaris, in welcher Rolling-Up (vgl. Abschnitt 2.3) beschrieben worden ist [HT00]. Diese grundlegende Technik wurde in nahezu allen folgenden Arbeiten aufgegriffen und verfeinert. So zeigt Birte Glimm später die Beantwortung konjunktiver Abfragen für $\mathcal{SHQ}$ [GHS06], $\mathcal{SHIQ}$ [GHLS07] sowie $\mathcal{SHOQ}$ [GHS07].

Auf der anderen Seite stehen Arbeiten, welche das Ziel verfolgen, verteilte Wissensbasen bei der Beantwortung von konjunktiven Abfragen zu berücksichtigen. Maßgebliche Resultate sind dabei Entwürfe von geeigneten Systemarchitekturen, Mechanismen zu Vorverarbeitung der Abfragen sowie heuristische Ansätze zur Identifikation relevanter Antwortontologien. Besonders zu berücksichtigen ist dabei die angewandte Semantik bezüglich einer kombinierten Wissensbasis, da oft eine allein technische Verteilung vorgestellt wird. Dabei werden die zentral gestellten Abfragen umgeschrieben und mit Hilfe eines Metadatenverzeichnises auf die verfügbaren Wissensbasen aufgeteilt. Durch die fehlende formale Definition einer kombinierten Wissensbasis lassen sich dabei jedoch mitunter keine belastbaren Aussagen über Vollständigkeit und Korrektheit im üblichen Sinne machen.

Theoretische Betrachtungen zur Integration verteilter Datenbasen werden von Maurizio Lenzerini in [Len02] vorgenommen. Im Speziellen werden die Kategorien Local-

As-View (LAV), Global-As-View (GAV) und Kombination von LAV und GAV (GLAV) vorgestellt, welche zur Charakterisierung von Integrationsformalismen dienen können. In diesem Rahmen wird ein allgemeines Framework für Datenintegration eingeführt. Bedingt durch die allgemeine Natur der Arbeit sowie dem ausschließlichen Fokus auf – in Beschreibungslogik-Terminologie: – ABoxen, eignen sich die vorgestellten Ergebnisse jedoch nicht ohne Weiteres zur Anwendung auf Ontologien.

Die vom W3C empfohlene Resource Description Framework (RDF)-Abfragesprache SPARQL Protocol and RDF Query Language (SPARQL) ist aufgrund der Möglichkeit, OWL-Wissensbasen auf RDF-Graphen reduzieren zu können, auch zur Abfrage von Ontologien naheliegend [PS08, PSHH04]. Darüber hinaus ist in der Definition von SPARQL die Möglichkeit einer Erweiterung durch die Einbeziehung einer komplexeren logischen Konsequenzbeziehung (entailment) beschrieben. Dies trifft im Besonderen auch auf OWL zu. Dabei entstehen jedoch Schwierigkeiten aufgrund der deutlich höheren Ausdruckskraft von OWL [SP07]. Die triviale Modularisierung von OWL selbst kann jedoch im hier betrachteten Rahmen vernachlässigt werden.

Von Quilitz et al. und Langegger et al. werden SPARQL-Abfragen unterstützt, die mehrere (physisch getrennte) Wissensbasen umfassen können [QL08, LBW07, LKB08] . Dazu werden in einem Vorverarbeitungsschritt auf Basis von bekannten Metainformationen die in Frage kommenden Wissensbasen ermittelt und für die Beantwortung von Sub-Anfragen herangezogen. Dafür wird vornehmlich das *Mediator*-Muster zur Verteilung der Anfragen genutzt. Offen bleibt dabei, inwieweit die verfügbaren Metadaten für eine hinreichend korrekte Aufteilung geeignet sind. Außerdem sind in keinem Fall logische Schlussfolgerungen über mehrere Wissensbasen hinweg zu erwarten.

In [HW07] und der dort vorgestellten prototypischen Realisierung *KAONp2p* wird die semantische Verknüpfung zweier Ontologien über unsymmetrische, disjunktive Abfragen bzw. erforderlichen Mengenrelationen bezüglich der Ergebnisse dieser Abfragen hergestellt. Damit entspricht der Abbildungsformalismus dem GLAV-Ansatz. Die Beantwortung von Abfragen geschieht dabei in zwei Phasen:

1. **Selektion** - Auf Basis eines Metadaten-Verzeichnis, in welchem Informationen über die verwendeten Ontologien mit Hilfe des Ontology Metadata Vocabulary (OMV) abgelegt sind, werden die für eine Abfrage relevanten Knoten identifiziert.

2. **Beantwortung** - Durch eine physische Integration der TBoxen kann auf Basis der sogenannten *Virtual Ontology* die Anfrage unter Berücksichtigung aller se-

lektierten Ontologien ausgewertet werden. Dabei verbleiben die ABoxen bis auf den für die Beantwortung relevanten Teil auf ihrem entfernten Host.

Die *Virtual Ontology* entspricht einer vollständigen Integration der berücksichtigten Ontologien, so dass von einem einzelnen, globalen Modell ausgegangen werde kann. Damit lassen sich beweisbare Aussagen über Vollständigkeit und Korrektheit machen.

Distributed Ontology Framework on Grid environment wie von Bui et. al. vorgeschlagen, nutzt zwar eine andere Sprache zur Abfrage mehrerer Ontologien, basiert jedoch auf ähnlichen Prinzipien wie KAONp2p [BLC07]. Im Besonderen wird hier zwar auf die Berücksichtigung globaler OWL-Schlussfolgerungen verzichtet, aber es wird ebenso ein Metadaten-Verzeichnis benötigt, um aus dem Satz verfügbarer Ontologien Antwortkandidaten abfragenabhängig auszuwählen. Die Architektur der Implementierung entspricht ebenfalls einem (hybriden) P2P-Netz.

Bei *ProSÉ* handelt es sich um eine Erweiterung[1] für den Ontologie-Editor Protégé[2]. Sie implementiert den Modularitätsansatz aus Abschnitt 3.2 und ist als Werkzeug für Ontologie-Entwickler konzipiert worden. Damit lässt sich die Sicherheit der erstellten Ontologien überprüfen und eine Modulextraktion aus normalerweise zu importierenden Ontologien vornehmen. Dies ermöglicht eine (momentan noch) statische Wiederverwendung von Fragmenten anderer Ontologien. Diese Fragmente werden letztlich in die bearbeitete Ontologie integriert, weswegen eine durch konjunktive Abfragen notwendige, automatische Neuberechnung der Fragmente (vgl. Abschnitt 5.3) noch nicht unterstützt wird. Sollte dies jedoch nicht notwendig sein – weil die Abfrage nur Symbole enthält, die in der Signatur der Wissensbasis enthalten sind – lassen sich konjunktive Abfragen mittels bereits verfügbarer Werkzeuge effizient durchführen. Die Implementierung ist noch recht jung, wird aber anhand der während einer ersten Evaluierung erhaltenen Rückmeldungen als vielversprechend bewertet [JRCS+08].

[1] A Protégé plugin for Reusing Ontologies: `http://krono.act.uji.es/people/Ernesto/safety-ontology-reuse`

[2] Ontology Editor Protégé: `http://www.co-ode.org/downloads/protege-x/`

5 Konjunktive Abfragen auf modularen Ontologien

Es wurde bereits deutlich, dass auf dem Gebiet der logikbasierten Wissensrepräsentation ein starker Trend zur Modularisierung existiert, welcher weitere Hürden für einen praktischen Einsatz zu überwinden sucht. Jedoch lassen die vorgestellten Ansätze modularer Sprachen Betrachtungen zur einfachen Abfrage der enthaltenen Informationen zunächst vermissen. Die verwandten Arbeiten offenbaren demgegenüber Bemühungen, bei Abfragen mehrere Wissensbasen gleichzeitig zu berücksichtigen, vernachlässigen dabei aber mitunter die eigentliche Stärke Logik-basierter Ansätze. Es ist daher nur naheliegend, diese beiden Bemühungen zusammen zu bringen und konjunktive Abfragen, die auch die Basis so mächtiger Sprachen wie der SQL und SPARQL stellen, im Kontext dreier Ansätze modularer Ontologien zu betrachten. Dazu wird zuerst eine umfangreiche Untersuchung im Kontext von $\mathcal{E}$-Connections durchgeführt, für welche noch keine Ergebnisse bezüglich dieses Problems existieren. Im Anschluss daran werden die vorhandenen Arbeiten zu konjunktiven Abfragen über DDL vorgestellt, sowie ein Lösungsansatz für die Realisierung bei analytischer Modularisierung gegeben.

5.1 $\mathcal{E}$-Connections

Die wichtigen Resultate für $\mathcal{E}$-Connections beziehen sich bisher vor allem auf deren Entscheidbarkeit [KLWZ04]. Dabei wird klar, dass $\mathcal{E}$-Connections ein eleganter Weg sind, ausdrucksstarke Logiken zu verbinden und dabei unerwünschte Nebeneffekte zu vermeiden.

Aufgrund ihrer Nähe zu Beschreibungslogiken wird der im Kapitel 2.3 gemäß [HT00] vorgestellte Ansatz auf konjunktive Abfragen über $\mathcal{E}$-Connections übertragen. Zur Klärung des Begriffs im Kontext von $\mathcal{E}$-Connections werden so auch Varianten der dort gemachten Definitionen verwendet. Das Ziel ist, die Beantwortung von Abfragen erneut auf das Erfüllbarkeitsproblem zu reduzieren, wie der in Abbildung 5.1 dargestellte Überblick zeigt. Eine besondere Herausforderung ist die Berücksichtigung von n-stelligen Relationen und die Unabhängigkeit von speziellen, in den Komponenten-

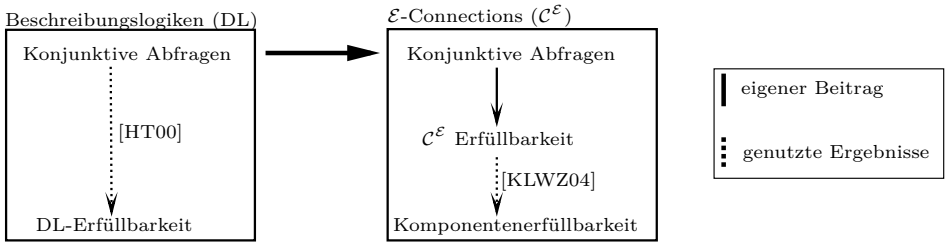


Abbildung 5.1: Inspiration und Ansatz für konjunktive Abfragen

logiken enthaltenen Funktionen.

Definition 23: Konjunktive Abfrage. *Eine konjunktive Abfrage auf einer Wissensbasis eKB einer $\mathcal{E}$-Connection $C^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_m)$ ist eine Menge $q = \{q_1, \ldots, q_n\}$ von Elementen q_i, die von der Form $(y : t)$ oder $((x_1, \ldots, x_m) : E_j)$ sind und Termatome bzw. Verbindungsrelationsatome genannt werden. Dabei gilt für $i = 1 \ldots m$:*

$$\begin{aligned} &\mathcal{S}_i = (\mathcal{L}_i, \mathcal{M}_i) = ((\mathcal{V}_i, \mathcal{X}_i, \mathcal{R}_i, \mathcal{F}_i), \mathcal{M}_i) \\ &\quad t \in \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_m \text{ und } t \in \mathcal{V}_i \rightarrow y \in \mathcal{X}_i \cup \mathfrak{V}^q \\ &x_i \in \mathcal{X}_i \cup \mathfrak{V}^q \text{ und } E_j \in \mathcal{E} \end{aligned}$$

Dabei ist $\mathfrak{V}^q$ die Menge der in der Abfrage q verwendeten Variablen.

Die Notation ist gegenüber den im Abschnitt 2.3 vorgestellten konjunktiven Abfragen derart angepasst, dass sie konjunktiv verknüpften Aussagen in $\mathcal{E}$-Connections gemäß Abschnitt 3.1.1 entsprechen. Es ist außerdem anzumerken, dass gewisse Anfragen bereits aus syntaktischen Gründen verneint werden können, beispielsweise bei Existenz zweier Atome $q_1 = ((x_1, x_2) : E_j)$ und $q_2 = ((x_2, x_1) : E_j)$[1]. Solche Abfragen werden nicht als im Sinne der Definition betrachtet und bleiben unberücksichtigt.

Offenbar wird hierbei auch auf Atome, welche ein Abfragen der Relationen der einzelnen ADS ermöglichen, verzichtet. Damit sind beispielsweise Abfragen von Rollen in auf Beschreibungslogiken basierenden Komponenten zunächst ausgeschlossen. Doch

[1] In diesem Fall müssten die Variablen jeweils Objektvariablen aus *unterschiedlichen* Sprachen bzw. Logiken entsprechen.

wird es dadurch möglich, die potentiell komplexen Interaktion zwischen Funktionen und Relationen innerhalb der Komponenten zu vernachlässigen. Beispielsweise wird aus Abschnitt 2.3 deutlich, dass zur Beantwortung von Rollen- bzw. Relationsatomen Funktionen $f_{\exists r}(X)$ für alle Relationen r innerhalb der ADS existieren müssten[2], deren Präsenz hier nicht angenommen werden soll.

Die Antwort auf eine Abfrage gemäß Definition 23 wird mittels einer Ersetzung definiert.

Definition 24: Antwort auf eine konjunktive Abfrage. *Gegeben sei eine konjunktive Abfrage q für eine Wissensbasis eKB, mit $\mathfrak{V}^q = \{x_1, \ldots, x_n\}, n \in \mathbb{N}$. Eine Antwort A für q ist eine endliche Menge von n-Tupeln $a_i = (a_i^1, \ldots, a_i^n)$. Mittels einer Ersetzungsfunktion $[\cdot]_{a_i}$ werden die Variablen x_l mit den entsprechenden Elementen a_i^l ersetzt. Ein jedes Element a_i einer gültigen Antwort A muss nun folgenden Bedingungen genügen:*

$$a_i^l \in \left(\bigcup_{k=1}^{m} \mathcal{X}_k\right) \cup \{\texttt{exists}_{x_l}\}\,,\ l \in [1,n] \text{ und } \texttt{exists}_{x_l} \text{ eine neue Variable.} \tag{5.1a}$$

$$\forall_{j \neq i} \forall_{k=1}^{n} \left(\forall_{\substack{l=1 \\ l \neq k}}^{n} a_j^l = a_i^l\right) \implies \left(a_j^k \neq \texttt{exists}_{x_k} \iff a_i^k \neq \texttt{exists}_{x_k}\right). \tag{5.1b}$$

$$eKB \models [q_1 \wedge \ldots \wedge q_n]_{a_i} \tag{5.1c}$$

Es ist anzumerken, dass der Begriff *Antwort* im Folgenden sowohl für A als auch für ein Element $a_i \in A$ verwendet wird, die beabsichtigte Bedeutung jedoch aus dem Kontext folgt.

Die erste Bedingungen grenzt den Antwortraum ab: Eine Variable kann entweder Antwortvariable sein und mit einer Objektvariable beantwortet werden, oder als existenzielle Variable verstanden werden. Die zweite Bedingung verhindert Redundanz, indem existenzielle Elemente in einer Antwort vermieden werden, wenn eine (nahezu) identische Antwort existiert, welche eine Objektvariable an gleicher Stelle enthält. Die dritte Bedingung gibt bereits einen Hinweis auf einen Algorithmus und reduziert die Beantwortung von konjunktiven Abfragen auf deren logische Konsequenz.

[2] Vgl. ADS $\mathcal{ALC}^{\#}$ aus [KLWZ04]

5.1.1 Variablenfreie Boolesche Abfragen

Es müssen daher prinzipiell nur Boolesche Abfragen betrachtet werden, hier zunächst noch ohne Variablen. Da es dabei nur noch um die Frage der logischen Konsequenz geht, existieren nur zwei Antwortmöglichkeiten: `wahr` oder `falsch`. Der einfachste Fall, für eine Menge von Atomen $q_1, \ldots, q_n$, der Form $(a : t_i)$ mit $1 \leq i \leq n$, lässt sich aufgrund der in jeder ADL enthaltenen Funktion $\wedge$ zusammenfassen:

$$\{(a : t_1), \ldots, (a : t_n)\} \mapsto \{(a : t_1 \wedge \ldots \wedge t_n)\}$$

Die Semantik wird dabei zweifelsohne erhalten, so dass im Folgenden ohne Beschränkung der Allgemeinheit davon ausgegangen werden kann, dass in einer Abfrage $q = \{(a_1 : t_1), \ldots, (a_n : t_n)\}$ für zwei beliebige Atome q_i, q_j $a_i \neq a_j$ gilt, falls $i \neq j$ gilt. Sind nun keine zusätzlichen Atome über Verbindungsrelationen enthalten, lässt sich aufgrund der Unabhängigkeit der Atome die Äquivalenz

$$eKB \models \bigwedge(a_i : t_i) \iff \bigwedge eKB \models (a_i : t_i)$$

ausnutzen. Daher können diese Abfragen beantworten werden, indem für jedes Atom $(a_i : t_i)$ die logische Konsequenz für eine $\mathcal{E}$-Connections-Wissensbasis gemäß [KLWZ04] geprüft wird:

$$\begin{gathered} eKB = (\Gamma_1, \ldots, \Gamma_m, \Gamma_{\mathcal{E}}) \models a_i : t_i \\ \text{genau dann, wenn} \\ (\Gamma_1, \ldots, \Gamma_i \cup \{a_i : \neg t_i\}, \ldots \Gamma_m, \Gamma_{\mathcal{E}}) \text{ unerfüllbar ist.} \end{gathered}$$

Ergibt diese Prüfung für jedes Atom `wahr`, so ist auch die Antwort auf q `wahr`.

Im nächsten Schritt werden Atome über Verbindungsrelationen betrachtet. Dazu sei eine Beispielabfrage definiert als

$$q_E = \{(a_1 : t_1), ((a_1, a_2, \ldots a_m) : E_1)\}.$$

Dies wirft das Problem auf, m-stellige Atome beantworten zu müssen. Adaptiert man Horrocks' Aufrollen („Rolling Up"), erhält man eine transformierte Abfrage:

$$q'_E = \left\{(a_1 : t_1 \wedge \langle E_1 \rangle^1 (a_2, \ldots, a_m))\right\}$$

Offenbar führt die Tatsache, dass Verbindungsrelationen nicht gerichtet sind, zu m

Möglichkeiten, dieses Aufrollen zu bewerkstelligen. So sind für alle $2 \leq i \leq m$ folgende Transformationen ebenso möglich:

$$q'_E = \Big\{(a_1 : t_1), (a_i : \langle E_1 \rangle^i (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_m))\Big\}$$

Diese Transformationsergebnisse ließen sich so jedoch nur im Kontext einer $\mathcal{E}$-Connection $C_O^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_m)$ beantworten, welche eine solche Verwendung von Objektvariablen innerhalb der Funktionssymbole $\langle E_j \rangle^i$ zulässt. Um ein allgemeingültiges Ergebnis zu erhalten, sollen Stellvertreterterme genutzt werden, welche im Rahmen von konjunktiven Abfragen ausreichend sind, obige Transformation zu adaptieren. Dazu wird vor dem Aufrollen eines Atoms $((a_1, a_2, \ldots a_m) : E_1)$ für jede Objektvariable $a_i \in \mathcal{X}_i$ eine neue Mengenvariable V_{a_i} eingeführt, mit $V_{a_i} \notin \mathcal{V}(\Gamma_i)$ und $\Gamma'_i = \Gamma_i \cup \{a_i : V_{a_i}\}$. Damit kann die allgemeine Umwandlung umgeschrieben werden in

$$q'_E = \Big\{(a_1 : t_1), (a_i : \langle E_1 \rangle^i (V_{a_i}, \ldots V_{a_{i-1}}, V_{a_{i+1}}, \ldots, V_{a_m}))\Big\},$$

welche nun im Kontext einer neuen Wissensbasis

$$eKB' = (\Gamma'_1, \ldots, \Gamma'_{i-1}, \Gamma_i, \Gamma'_{i+1}, \ldots \Gamma'_m, \Gamma_{\mathcal{E}})$$

beantwortet werden muss. Es ist zu bemerken, dass für jede Objektvariable nur genau ein Stellvertreterterm existieren soll, welcher gegebenenfalls erneut genutzt werden kann. Dass dies eine zulässige Umwandlung ist, zeigt folgende Adaption von Theorem 1:

Lemma 1: Aufrollen von $\mathcal{E}$-Connection-Abfragen. *Gegeben seien eine Wissensbasis $eKB = (\Gamma_1, \ldots, \Gamma_m, \Gamma_{\mathcal{E}})$ einer m-dimensionalen $\mathcal{E}$-Connection $C^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_m)$ mit $\mathcal{S}_i = ((\mathcal{V}_i, \mathcal{X}_i, \mathcal{R}_i, \mathcal{F}_i), \mathcal{M}_i)$, $a_1, \ldots, a_m \in \mathcal{X}_i$ und einer Verbindungsrelation $E_j \in \mathcal{E}$. Dann gilt:*

$$eKB \models (a_1, \ldots, a_m) : E_j \tag{5.2}$$

genau dann, wenn

$$\begin{aligned} eKB' \models a_i : \langle E_j \rangle^i (V_{a_1}, \ldots, V_{a_{i-1}}, V_{a_{i+1}}, \ldots, V_{a_m}) \ \textit{mit}\ i \in [1, m] \\ \textit{wobei}\ \Gamma'_l = \Gamma_l \cup \{a_l : V_{a_l}\} \ \textit{mit}\ V_{a_l} \notin \mathcal{V}(\Gamma_l) \\ \textit{und}\ eKB' = (\Gamma'_1, \ldots, \Gamma'_{i-1}, \Gamma_i, \Gamma'_{i+1}, \ldots, \Gamma'_m) \end{aligned} \tag{5.3}$$

Beweis. Zunächst soll die Korrespondenz von Modellen für eKB und eKB' deutlich gemacht werden. So existiert für jedes Modell $\mathfrak{M}$ von eKB ein erweitertes Modell

$\mathfrak{M}'$, welches zusätzlich die neuen Mengenvariablen V_{a_i} genau so interpretiert, dass $a_i^{\mathfrak{M}'} \in V_{a_i}^{\mathfrak{M}'}$ gilt. Damit ist $\mathfrak{M}'$ auch ein Modell von eKB'. Es ist zu sehen, dass jedes Modell für eKB' auch alle Aussagen aus eKB erfüllen muss und insofern ebenso ein Modell für eKB ist. Im Folgenden genügt es daher, Modelle für eKB' zu betrachten.

Sei $\bar{t}_i = (V_{a_1}, \ldots, V_{a_{i-1}}, V_{a_{i+1}}, \ldots, V_{a_m})$ die Termsequenz der neuen Mengenvariablen aus eKB'. Es ist zu zeigen, dass

$$eKB \models (a_1, \ldots, a_m) : E_j \iff eKB' \models a_i : \langle E_j \rangle^i (\bar{t}_i) \text{ gilt.}$$

$\Rightarrow$ Sei $\mathfrak{M}$ ein Modell von eKB und eKB' mit $\mathfrak{M} \models (a_1, \ldots, a_m) : E_j$. Wegen der Konstruktion von eKB' gilt für jedes $l \in [1, m]$ und $l \neq i$

$$a_l^{\mathfrak{M}} \in V_{a_l}^{\mathfrak{M}}. \tag{i}$$

Aufgrund der Annahme gilt des Weiteren

$$\left(a_1^{\mathfrak{M}}, \ldots, a_m^{\mathfrak{M}}\right) \in E_j^{\mathfrak{M}}. \tag{ii}$$

Nun ist zu sehen, dass wegen (i) und (ii) gemäß Definition 12 auch

$$a_i^{\mathfrak{M}} \in \left(\langle E_j \rangle^i (\bar{t}_i)\right)^{\mathfrak{M}} \qquad \text{gelten muss.}$$

Damit gilt die Implikation zunächst für jedes einzelne Modell und aufgrund der Korrespondenz der Modelle für eKB und eKB' auch in der oben dargestellten Form.

$\Leftarrow$ Sei $\bar{a}$ die abgekürzte Schreibweise der Sequenz von Objektvariablen $(a_1, \ldots, a_m)$. Folgende Implikation ist noch zu zeigen:

$$\neg (eKB \models \bar{a} : E_j) \implies \neg \left(eKB' \models a_i : \langle E_j \rangle^i (\bar{t}_i)\right).$$

Das heißt, es genügt zu zeigen, dass bei Existenz eines Modells $\mathfrak{M}$, mit $\bar{a}^{\mathfrak{M}} \notin E_j^{\mathfrak{M}}$, auch ein Modell $\mathfrak{M}'$ existiert, mit $a^{\mathfrak{M}'} \notin (\langle E_j \rangle^i (\bar{t}_i))^{\mathfrak{M}'}$. Gegeben sei also ein entsprechendes Modell $\mathfrak{M}$ von eKB. Es lässt sich zeigen, dass dann auch ein Modell $\mathfrak{M}^*$ existiert, mit gleichen Domänen W_i, welches für alle i alle $\mathcal{X}_i$, $\mathcal{F}_i$, $\mathcal{R}_i$, $\mathcal{X}_i$ und $\mathcal{V}_i \backslash \bigcup_{1 \leq j \leq m} \{V_{a_j}\}$ sowie $\mathcal{E}$ identisch interpretiert. Damit bleibt sowohl die Modelleigenschaft für eKB erhalten, als auch die Annahme $\bar{a}^{\mathfrak{M}} \notin E_j^{\mathfrak{M}}$ erfüllt. Für die neu eingeführten Mengenvariablen soll aber $V_{a_i}^{\mathfrak{M}^*} = \{a_i^{\mathfrak{M}^*}\}$ gelten. Da

damit auch die einzigen neu hinzugefügten Axiome in eKB' erfüllt werden, ist $\mathfrak{M}^*$ auch ein Modell für eKB'. Nun kann $\mathfrak{M}^* \models a_i : \langle E_j \rangle^i (\bar{t}_i)$ aber nicht mehr gelten, da dies nur bei der Existenz eines m-Tupels $\bar{d} \in E_j^{\mathfrak{M}^*}$ möglich wäre, für dessen Komponenten $d_i \in V_{a_i}^{\mathfrak{M}^*}$ gilt. Die einzige Möglichkeit eines solchen Tupels ist durch die angenommene Interpretation der V_{a_i} vorgegeben, führt aber gemeinsam mit der Annahme $\bar{a}^{\mathfrak{M}} \notin E_j^{\mathfrak{M}}$ zu einem Widerspruch. Somit ist die Existenz des Modells $\mathfrak{M}'$ in Form des hier konstruierten $\mathfrak{M}^*$ gezeigt und die Implikation erfüllt.

□

Ganz wie bei Theorem 1 ließe sich auch hier die Absorption von weiteren Termatomen integrieren, worauf zur Steigerung der Übersichtlichkeit verzichtet wurde. Solange nur Objektvariablen enthaltende Verbindungsrelationsatome aufgerollt werden, ist diese Absorption nicht notwendig, kann aber zur Reduktion der schließlich zu prüfenden Erfüllbarkeitsprobleme dienen[3]. Aufgrund der „Umwandlung“ der Objektvariablen in Mengenvariablen (bzw. Terme) ist klar, dass auch die ursprüngliche Vermengung von Objektvariablen und Termen als Elemente eines Verbindungsrelationsatoms problemlos möglich wäre.

Eine Beantwortung variablenfreier Abfragen ist nun mittels Eliminierung der Verbindungsrelationsatome und der anschließenden Transformation in endliche Mengen von Erfüllbarkeitsproblemen ermöglicht. Dies erfordert keine zusätzlichen Annahmen oder Einschränkungen der Art der $\mathcal{E}$-Connection und ist daher in jedem Fall durchführbar.

5.1.2 Boolesche Abfragen mit Variablen

Ein anderes Bild ergibt sich, sobald existenziell quantifizierte Variablen zugelassen werden.

Definition 25: Beantwortung von Booleschen Abfragen. *Derartige Abfragen q können mit* `wahr` *beantwortet werden, wenn für jedes Modell $\mathfrak{M}$ für die existenziell quantifizierten Variablen eine Ersetzung $[\cdot]^{\mathfrak{M}}$ mit Elementen der Interpretationsdo-*

[3] Im Allgemeinen trifft dies jedoch nicht zu, weswegen hier nicht von Absorption beim Aufrollen von Objektvariablen ausgegangen wird.

mänen existiert, so dass $\mathfrak{M} \models [q]^{\mathfrak{M}}$ *gilt. Dabei ist* $[\cdot]^{\mathfrak{M}}$ *formal definiert als:*

$$\begin{aligned}
[q]^{\mathfrak{M}} &= [q_1]^{\mathfrak{M}} \wedge \ldots \wedge [q_n]^{\mathfrak{M}} \\
[x : t]^{\mathfrak{M}} &= [x]^{\mathfrak{M}} : t \\
[(x_1, \ldots, x_m) : E]^{\mathfrak{M}} &= ([x_1]^{\mathfrak{M}}, \ldots, [x_m]^{\mathfrak{M}}) : E \\
[x]^{\mathfrak{M}} &= \begin{cases} x & \textit{falls } x \in \mathcal{X}_i \\ d \textit{ mit } d \in W_i & \textit{falls } x \in \mathfrak{V}^q \end{cases}
\end{aligned}$$

Damit die Wahrheitsrelation aus Definition 12 unverändert übernommen werden kann, soll $d \in W_i \rightarrow d^{\mathfrak{M}} = d$ *gelten.*

Für die einfache Abfrage $q_0 = \{(x_1 : t)\}$ mit einer Variable x_1 ist dies beispielsweise genau dann gegeben, wenn der Term t in jedem Modell der Wissensbasis wenigstens ein Element hat. Dazu wird ein Term $\top_i$ benötigt, welcher innerhalb einer Komponente $\mathcal{S}_i$ immer als vollständige Interpretationsdomäne interpretiert wird, d.h. $\top_i^{\mathfrak{M}_i} = W_i$. Ein solcher Term lässt sich mit Hilfe einer vorhandenen Mengenvariablen $V_i \in \mathcal{V}_i$ leicht definieren als $\top_i = \neg(V_i \wedge \neg V_i)$. Es soll angenommen werden, dass t der Komponente Γ_1 einer Wissensbasis $eKB = (\Gamma_1, \ldots, \Gamma_m, \Gamma_{\mathcal{E}})$ zugeordnet werden kann. Wie bereits in [HSTT99] und [HT00] vorgestellt, ist die Prüfung von $(\Gamma_1 \cup \{\top_1 \sqsubseteq \neg t\}, \ldots, \Gamma_m)$ auf Inkonsistenz äquivalent zu der Aussage, dass in jeder Interpretation der Term t wenigstens ein Element enthält, d.h. der Variable x_1 in jedem Modell $\mathfrak{M}$ ein Element d der Domäne zugewiesen werden kann, so dass $d \in t^{\mathfrak{M}}$ gilt. Formal:

$$eKB \models (x_1 : t) \iff (\Gamma_1 \cup \{\top_1 \sqsubseteq \neg t\}, \ldots, \Gamma_m, \Gamma_{\mathcal{E}}) \text{ ist unerfüllbar.} \tag{5.4}$$

Im nächsten Schritt wird ein zusätzliches Atom betrachtet:

$$q_1 = \{(x_1 : t), ((x_1, x_2, \ldots, x_m) : E)\}$$

Die Variablen $x_2, \ldots, x_m$ werden an keiner anderen Stelle der Abfrage verwendet und erfordern daher nur die Existenz eines beliebigen Elements der entsprechenden Interpretationsdomäne. Dies kann durch erneute Verwendung des jeweiligen $\top$-Terms zum Ausdruck gebracht werden. So ist die Antwort auf q_1 genau dann `wahr`, wenn die Antwort auf

$$q_1' = \left\{\left(x_1 : t \sqcap \langle E \rangle^1 (\top_2, \ldots, \top_m)\right)\right\} \text{ `wahr` ist.}$$

Hier wird deutlich, dass nunmehr die Vermengung von Variablen, Objektvariablen und Termen innerhalb der Verbindungsrelationsatome möglich wäre, da die ersten

beiden durch die Transformation in Terme umgewandelt werden.

Aufgrund der speziellen Eigenschaften von Verbindungsrelationen sind bei q_1 erneut m äquivalente Umformungen möglich. Für $2 \leq i \leq m$ lauten die restlichen Umformungen:

$$q_1' = \left\{ \left(x_i : \langle E \rangle^i \left(t, \top_2, \ldots, \top_{i-1}, \top_{i+1}, \ldots \top_m, \right) \right) \right\}$$

Dabei wird klar, dass die Absorption von Termatomen beim Aufrollen von Variablen notwendig ist, um diese nicht aus ihrem Zusammenhang zu reißen. Würde beispielsweise das Atom $(x_1 : t)$ hier nicht absorbiert und im transformierten Atom stattdessen $\top_1$ verwendet werden, resultierte daraus eine deutlich relaxierte Semantik, was zu falschen Ergebnissen führen könnte. Dabei tritt das Problem zu Tage, dass die eliminierten Variablen nicht mehr an anderer Stelle, d.h. in keinen anderen Atomen über Verbindungsrelationen erwähnt sein dürfen. Dass dies zweifellos vorkommen kann zeigt folgende Beispielabfrage[4]:

$$q_2 = \{(x_1 : t), ((x_1, x_2, x_3) : E), ((z_1, x_2, y_3) : E), ((z_1, z_2, z_3) : E)\}$$

Einen Graph einer Abfrage analog Definition 7 zu gestalten ist im Falle von n-stelligen Relationen ungünstig. Offensichtlicher wird die Abfragenstruktur, wenn man Knoten und Kanten vertauscht[5]:

Definition 26: Abhängigkeitsgraph einer Abfage. *Ein Graph $G(q) = (V, E)$ einer Abfrage q ist ein Tupel bestehend aus einer endlichen Knotenmenge V und einer endlichen Menge von ungerichteten Kanten E.*

$$V = \{v \mid v \in q \text{ und der Form } (x_1, \ldots, x_n) : E_j\}$$
$$E = \left\{ (e^1, e^2, e^x) \mid e^1, e^2 \in V \text{ und } e^1, e^2 \text{ haben eine Variable } e^x \in \mathfrak{V}^q \text{ gemeinsam.} \right\}$$

Dabei gilt aufgrund fehlender Richtung $(e^1, e^2, e^x) = (e^2, e^1, e^x)$. Das dritte Element e^x einer Kante wird als deren Beschriftung genutzt. Die Randknoten *des Graphen B sind Knoten, bei denen alle anliegenden Kanten die gleiche Beschriftung tragen.*

$$B = \left\{ v \mid \forall e_i, e_j \in E \ \left(v \in \{e_i^1, e_i^2\} \cap \{e_j^1, e_j^2\} \right) \rightarrow \left(e_i^x = e_j^x \right) \right\}$$

Randknoten umfassen so auch Knoten, welche an keiner Kante liegen.

[4] Zur Erhöhung der Übersichtlichkeit wird innerhalb der folgenden Beispiele $m = 3$ angenommen.
[5] Ein ähnlicher Graph ist bereits in [HSTT99] beschrieben

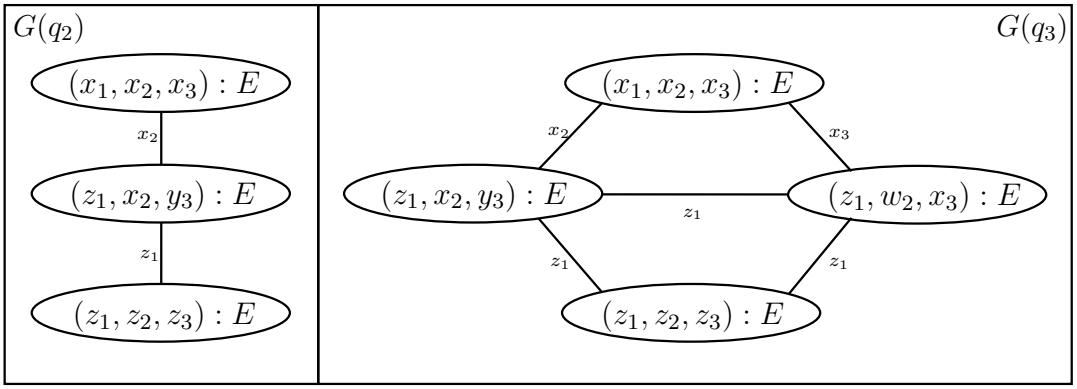


Abbildung 5.2: Graphen der Beispielabfrage q_2 und q_3.

Die etwas unkonventionelle Definition von Randknoten als verallgemeinerte Blätter wird zur Vermeidung trivialer Zyklen benötigt, welche bereits bei Erwähnung einer Variable in mehr als zwei Knoten entstehen. Der mittels dieser Definition konstruierte Graph für q_2 ist in Abbildung 5.2 zu sehen.

Eine gültige Umformung erhält man wieder nur dann, wenn man an den Rändern des Graphen beginnt. Aufgrund der veränderten Graphenstruktur ergibt sich eine leicht andere Arbeitsweise. Man kann annehmen, dass alle Abfragenatome der Form $(x : t_i)$ in einer gesonderten Menge $\mathfrak{T}$ vorliegen und dass diese Menge zu keinem Zeitpunkt zwei Atome $(x_1 : t_1)$ und $(x_2 : t_2)$ enthält, bei denen $x_1 = x_2$ gilt. Das Ziel einer Transformation einer Abfrage muss nun sein, alle Knoten des Graphen und damit auch alle Kanten zu eliminieren. Dafür sind jeweils zwei Teilschritte nötig:

1. Ein Randknoten wird ausgehend von der Beschriftung seiner Kanten[6] in ein Termatom umgeformt und absorbiert gegebenenfalls Termatome aus der Menge $\mathfrak{T}$. Im Beispiel für den ersten Knoten:

   $$(x_1, x_2, x_3) : E \mapsto x_2 : \langle E \rangle^2 (\top_1, \top_3)$$

   Dieses wird $\mathfrak{T}$ hinzugefügt und gegebenfalls mit bereits vorhandenen Atomen zusammengefasst.

2. Der behandelte Knoten sowie die daran angeschlossenen Kanten werden aus dem

[6] Handelt es sich um einen kantenlosen Knoten, so ist der Ausgangspunkt unerheblich.

Graphen entfernt.

Ein mögliches Endergebnis für q_2 ist eine Abfrage bestehend aus einem einzelnen Termatom:

$$q_2' = \left\{ \left(z_3 : \langle E \rangle^3 \left(\langle E \rangle^1 \left(\langle E \rangle^2 \left(\top_1, \top_3 \right), \top_3 \right), \top_2 \right) \right) \right\}$$

An dieser Stelle kann als Zwischenergebnis festgehalten werden, dass azyklische, Boolesche Abfragen auf beliebigen $\mathcal{E}$-Connections durch eine einfache Reduktion auf eine Menge von Erfüllbarkeitsproblemen beantwortet werden können. Die entstehende Komplexität ist somit direkt abhängig von der Komplexität des Entscheidungsverfahrens für die jeweilige $\mathcal{E}$-Connection. Die Anzahl der zu entscheidenden Erfüllbarkeiten ist dabei abhängig von der Anzahl der verbleibenden Termatome. Im schlimmsten Fall müssen dennoch nur soviele Erfüllbarkeiten geprüft werden, wie es Atome in der Abfrage gibt.

Wie sich zeigen wird, ist der Aufwand zur Beantwortung zyklischer Abfragen deutlich höher. Für eine detaillierte Betrachtung soll der Begriff „Zyklus“ jedoch zunächst formalisiert werden.

Definition 27: Zyklus im Graph G. *Existiert eine Sequenz* $(v_1, \ldots, v_n, v_{n+1} = v_1)$ *von Knoten aus* $G = (V, E)$*, so dass für jedes* $1 \leq i \leq n$ *und* $n \in \mathbb{N}$ *eine Kante* $e = (v_i, v_{i+1}, d_i)$ *existiert und für alle* $1 \leq i \neq j \leq n$ $d_i \neq d_j$ *und* $v_i \neq v_j$ *gilt, so heißt diese Sequenz Zyklus in* G *der Länge* n.

Es ist anzumerken, dass aufgrund dieser Definition kein Randknoten Element eines Zyklus sein kann. Eine Abfrage q soll nun zyklisch genannt werden, sobald $G(q)$ zyklisch ist. Das heißt, jeder Knoten ist über wenigstens zwei unterschiedlich beschriftete Kanten mit anderen Knoten verbunden. Verdeutlichen soll dies eine erweiterte Beispielabfrage q_3 deren Graph ebenfalls in Abbildung 5.2 gezeigt ist und welche diesen Bedingungen genügt.

$$q_3 = q_2 \cup \{((z_1, w_2, x_3) : E)\}$$

Bei einer Transformation würde nun der Knoten $(z_1, z_2, z_3) : E$ als einziger Randknoten identifiziert und in ein entsprechendes Termatom $(z_1 : \langle E \rangle^1 (\top_2, \top_3))$ umgewandelt werden können. Die verbleibenden Knoten ergeben den Zyklus und erfordern besondere Berücksichtigung.

Eine Möglichkeit zyklische Abfragen aufzulösen ist in Kapitel 2.3 vorgestellt, wo dies durch Ausnutzung der Baum-Modell-Eigenschaft von Beschreibungslogiken ge-

lang. Bisher sind jedoch derartige Eigenschaften bei $\mathcal{E}$-Connections nicht untersucht worden. Deshalb sollen $\mathcal{E}$-Connections erst noch auf diese Eigenschaft geprüft werden. Da jedoch aufgrund der gewählten Abfragesprache hier nur Zyklen bezüglich der Verbindungsrelationen interessant sind, genügt eine genauere Betrachtung von $\mathcal{E}^{\mathfrak{M}}$. Zunächst werden die Interpretationen der Verbindungsrelationen in eine anschaulichere Darstellung gebracht.

Definition 28: Graph für $\mathcal{E}^{\mathfrak{M}}$. *Eine alternative Darstellung einer Interpretation $\mathcal{E}^{\mathfrak{M}}$ von Verbindungsrelationen, welche Teil einer Interpretation $\mathfrak{M}$ einer $\mathcal{E}$-Connection Wissensbasis $eKB = (\Gamma_1, \ldots, \Gamma_m, \Gamma_{\mathcal{E}})$ für $\mathcal{C}^{\mathcal{E}}(\mathcal{S}_1, \ldots, \mathcal{S}_m)$ ist, sei ein Graph $G^{\mathfrak{M}} = (V, E)$. Dabei sind V und E die endlichen Mengen aller Knoten $v = (d_1, \ldots, d_m) : E_j$ und Kanten $e = (e^1, e^2, e^d)$. In V sollen alle Elemente aller Verbindungsrelationen liegen, während die Kantenmenge die Verbindungen zwischen Knoten enthält, die ein Element d_l gemeinsam haben, welches nicht die Interpretation einer Objektvariable ist:*

$$V = \left\{ v \mid E_j \in \mathcal{E} \wedge (d_1, \ldots, d_m) \in E_j^{\mathfrak{M}} \right\}$$
$$E = \left\{ e \mid e^1, e^2 \in V \wedge \exists i \in [1, m] \ \left(e^d = d_i^1 = d_i^2 \wedge e^d \notin \mathcal{X}_i^{\mathfrak{M}_i} \right) \right\}$$

Da es sich um einen ungerichteten Graphen handeln soll, gilt $(e^1, e^2, e^d) = (e^2, e^1, e^d)$. Desweiteren dient das dritte Element einer Kante erneut als deren Beschriftung. Die Menge der Randknoten B enthält wieder alle „frei schwebenden“ Knoten und solche, deren anliegende Kanten gleich beschriftet sind:

$$B = \left\{ v \mid \forall e_i, e_j \in E \ \left(v \in \{e_i^1, e_i^2\} \cap \{e_j^1, e_j^2\} \right) \rightarrow \left(e_i^d = e_j^d \right) \right\}$$

Diese Graphendarstellung ist einerseits verträglich mit Definition 12 und andererseits stark verwandt zu Graphen von Abfragen. Um davon profitieren zu können, muss die Ersetzung von Variablen zur Beantwortung einer Abfrage auf Graphen ausgedehnt werden.

Definition 29: Ersetzung von $G(q)$. *Der aus $G(q)$ entstehende Graph $[G(q)]^{\mathfrak{M}}$ enthält die Knoten $(d_1, \ldots, d_m) : E$, die wie folgt aus den Knoten von $G(q)$ entstanden sind:*

$$[(x_1, \ldots, x_m) : E]^{\mathfrak{M}} = ([x_1]^{\mathfrak{M}}, \ldots, [x_m]^{\mathfrak{M}}) : E, \ \textit{mit } [x_i]^{\mathfrak{M}} = \begin{cases} d \in W_i & \textit{falls } x_i \in \mathfrak{V} \\ x_i^{\mathfrak{M}} & \textit{falls } x_i \in \mathcal{X}_i \end{cases}$$

Es sollen außerdem nur die Kanten aus $G(q)$ erhalten bleiben, für deren neue Beschriftung d_j gilt:

$$d_j \notin \mathcal{X}_1^{\mathfrak{M}} \cup \ldots \cup \mathcal{X}_m^{\mathfrak{M}}.$$

Für einen solchen Graph gilt nun

$$\mathfrak{M} \models [G(q)]^{\mathfrak{M}} \iff \forall\, ((d_1, \ldots, d_m) : E \in V)\ \ (d_1, \ldots, d_m) \in E^{\mathfrak{M}}$$

Es ist anzumerken, dass die Ersetzung hier im Gegensatz zu Definition 25 bereits die Interpretation in $\mathfrak{M}$ enthält. Nun ist folgende Feststellung bezüglich des Wahrheitsgehaltes einer Abfrage möglich.

Lemma 2: Beantwortung einer Abfrage mittels $G^{\mathfrak{M}}$ und $G(q)$. *Eine Abfrage q kann immer genau dann mit* `wahr` *beantwortet werden, wenn für jedes Modell $\mathfrak{M}$ der abgefragten Wissensbasis eKB eine Ersetzung $[\cdot]^{\mathfrak{M}}$ aller Variablen $x_i \in \mathfrak{V}^q$ mit Elementen $d_j \in W_j$ existiert, so dass*

(i) alle Termatome q_i^T gemäß Definition 25 erfüllt sind: $\mathfrak{M} \models [q_i^T]^{\mathfrak{M}}$

(ii) $[G(q)]^{\mathfrak{M}}$ ein Teilgraph von $G^{\mathfrak{M}}$ ist.

Beweis. Gegeben sei die konjunktive Abfrage $q = \{q_1, \ldots, q_n\}$ mit den Variablen $\mathfrak{V}^q = \{x_1, \ldots, x_k\}$. Die Atome q_i werden näher bestimmt als Verbindungsrelationsatome q_i^E oder Termatome q_i^T. Für eine positive Antwort muss laut Definition 25 für jedes Modell $\mathfrak{M}$ der abgefragten Wissensbasis eKB eine Ersetzung $[\cdot]^{\mathfrak{M}}$ existieren, so dass $\mathfrak{M} \models [q]^{\mathfrak{M}}$ gilt. Es ist daher zu zeigen, dass dies äquivalent zu den Punkten (i) und (ii) aus Lemma 2 ist. (i) ist trivial, da es sich direkt auf Definition 25 bezieht, weswegen ausschließlich (ii) und daher die Atome q_i^E betrachtet werden müssen.

$\Rightarrow$ Zunächst wird angenommen dass $\mathfrak{M} \models [q]^{\mathfrak{M}}$ gilt. Es ist zu zeigen, dass (I) sowohl die Knoten , als auch (II) die Kanten von $[G(q)]^{\mathfrak{M}}$ eine Teilmenge von $G^{\mathfrak{M}}$ sind. Da die Knoten von $G(q)$ die Verbindungsrelationsatome aus q repräsentieren (vgl. Definition 26), gilt auch $\mathfrak{M} \models [G(q)]^{\mathfrak{M}}$. Dies impliziert, dass die Knoten auch in $G^{\mathfrak{M}}$ vorhanden sind (I). Es sei darauf hingewiesen, dass sich durch die Ersetzung möglicherweise die Anzahl der Knoten in $[G(q)]^{\mathfrak{M}}$ gegenüber $G(q)$ verringert, da Duplikate aufgrund der Betrachtung als Menge unberücksichtigt bleiben. Eine Kante in $G(q)$ zwischen zwei Knoten v_i und v_j kann wegen Definition 26 nur mit einer Variable beschriftet sein. Bei Ersetzung der Variablen sind zwei Fälle zu unterscheiden. (a) Die Kante entfällt und unterstützt damit das Teilmengenkriterium. (b) die Kante wird mit einem Element $d \in W_i \wedge d \notin \mathcal{X}_i^{\mathfrak{M}}$

beschriftet. In letzterem Fall wird die ersetzte Variable auch in den Knoten v_i und v_j zu d. Diese Knoten sind wie oben gezeigt jedoch auch in $G^{\mathfrak{M}}$, wobei aufgrund der Eigenschaften von d gemäß Definition 28 auch eine Kante in $G^{\mathfrak{M}}$ entsteht (II).

$\Leftarrow$ Für die Gegenrichtung soll angenommen werden, dass $[G(q)]^{\mathfrak{M}} \subseteq G^{\mathfrak{M}}$ gilt. Das heißt jeder Knoten $v_i \in G(q)$ wird durch $[\cdot]^{\mathfrak{M}}$ zu einem Element in $\mathcal{E}^{\mathfrak{M}}$. Da v_i jedoch einem Atom $q_i^E \in q$ entspricht, lässt sich $\mathfrak{M} \models [q_i^E]^{\mathfrak{M}}$ und zusammen mit (i) auch $\mathfrak{M} \models [q]^{\mathfrak{M}}$ schließen.

□

Die neue Darstellung der Interpretation von Verbindungsrelationen hat es ermöglicht, zyklische Interpretationen als Interpretationen mit zyklischem $G^{\mathfrak{M}}$ zu formalisieren, und wird im weiteren Verlauf die Vorteile seiner Ähnlichkeit zu Abhängigkeitsgraphen von Abfragen noch ausspielen können. Doch vorher soll die Basis für weitere Schlüsse durch folgenden Satz geschaffen werden:

Satz 2: $G^{\mathfrak{M}}$ ist azyklisch. *Eine jede erfüllbare Wissensbasis eKB einer $\mathcal{E}$-Connection Anzahl-toleranter Komponenten, besitzt wenigstens ein Modell $\mathfrak{M}$ für welches $\mathcal{E}^{\mathfrak{M}}$ keinen Zyklus enthält.*

Für allgemeine $\mathcal{E}$-Connections gilt dieser Satz nicht, was zunächst mittels eines Gegenbeispiels gezeigt wird. Gegeben sei die $\mathcal{E}$-Connection $C^{\mathcal{E}}(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ mit $\mathcal{S}_i = \mathcal{SHOIQ}^{\#}$. Für $i = 1 \ldots 3$ seien folgende Axiome in den Wissensbasen Γ_i enthalten:

$$\top_i \sqsubseteq \exists R_i.\{o_i\}$$
$$\{o_i\} \sqsubseteq \leq 1 R_i^-.F_i$$

Dies limitiert die Anzahl der in einem Modell $\mathfrak{M}$ in $F_i^{\mathfrak{M}}$ enthaltenen Elemente auf maximal eins. Desweiteren sollen folgende Axiome berücksichtigt werden:

$$F_1 \sqsubseteq \langle E \rangle^1(\neg F_2, F_3) \in \Gamma_1$$
$$F_2 \sqsubseteq \langle E \rangle^2(F_1, \neg F_3) \in \Gamma_2$$
$$F_3 \sqsubseteq \langle E \rangle^3(\neg F_1, F_2) \in \Gamma_3$$

Dies stellt sicher, dass das Element aus $F_1^{\mathfrak{M}}$ mit dem Element in $F_3^{\mathfrak{M}}$, dieses mit dem aus $F_2^{\mathfrak{M}}$ und dieses wiederum mit dem Element aus $F_1^{\mathfrak{M}}$ über die Relation E verbunden ist. Außerdem können die drei Axiome nicht über ein einzelnes Element der Relation

E erfüllt werden, welche alle drei F_i-Elemente verbindet. Um die Existenz dieser drei Elemente zu erzwingen sei zusätzlich noch das Axiom $a : \exists S.F_1 \in \Gamma_1$. Damit ist in jedem Modell $\mathfrak{M}$ der Wissensbasis Γ ein Zyklus in $G^{\mathfrak{M}}$ enthalten.

Dieses Ergebnis ist auf die Interaktion von beliebigen Funktionen innerhalb der ADS sowie der Möglichkeit von Nominalen zuzuschreiben. Werden jedoch die in der $\mathcal{E}$-Connection verbundenen Logiken auf Anzahl-tolerante ADS beschränkt, lässt sich die Azyklizität beweisen. Die Definition eines Anzahl-toleranten ADS nach [KLWZ04] lautet wie folgt:

Definition 30. *Ein ADS S heißt Anzahl-tolerant, wenn es ein $k \in \mathbb{N}$ gibt, so dass für jedes $k' \geq k$ und jede erfüllbare Menge Σ von Axiomen ein Modell $\mathfrak{M} \in \mathcal{M}$ existiert, welches Σ erfüllt und in welchem für jedes Element $d \in W$ k' weitere Elemente $d' \in W$ enthalten sind, so dass*

$$\left\{t \in term(\Sigma) \mid d \in t^{\mathfrak{M}}\right\} = \left\{t \in term(\Sigma) \mid d' \in t^{\mathfrak{M}}\right\} \text{ gilt.}$$

Intuitiv gesprochen existieren in den Modellen erfüllbarer Wissensbasen Anzahl-toleranter ADS für jedes Element d viele weitere Elemente d', welche in genau den gleichen Terminterpretationen liegen und hier Klone genannt werden sollen. Es ist zu bemerken, dass damit auch Nominale unterbunden werden, da deren Interpretationen immer einelementige Mengen sind.

Beweis von Satz 2. Die Beweisidee ist, jedes zyklische Modell $\mathfrak{M}$ in ein azyklisches umwandeln zu können[7]. Ein Zyklus gemäß Definition 27 kann nur über Kanten (v_k, v_l, d_i) entstehen, mit $d_i \in W_i$ und $d_i \notin \mathcal{X}_1^{\mathfrak{M}} \cup \ldots \cup \mathcal{X}_m^{\mathfrak{M}}$. Daraus folgt, dass Aussagen der Form $(a_1, \ldots, a_m) : E_j \in \Gamma_{\mathcal{E}}$ nicht zur Zyklenbildung geeignet sind, da diese zu Knoten in $G^{\mathfrak{M}}$ führen, an welchen keine Kante anliegt (vgl. Def. 28). Es verbleiben die Aussagen der Form $a : t \in \Gamma_1 \cup \ldots \cup \Gamma_m$, welche bei enthaltenen Teiltermen der Form $\langle E_j \rangle^i$ geeignete Knoten induzieren.

Die Menge der Axiome einer Wissensbasis eKB sind jedoch endlich, so dass davon ausgegangen werden kann, dass die in einem Modell vorkommenden Zyklen maximal der Länge l sind[8]. Nun sei $\mathfrak{M} = \langle \mathfrak{W}_1, \ldots, \mathfrak{W}_n, \mathcal{E}^{\mathfrak{M}} \rangle$ ein solches Modell, welches n Zy-

[7] Dies kann als Variation des in der Literatur unter „unravelling“ bekannten Verfahrens verstanden werden [BCM+03].

[8] Zyklen können ohnehin nicht unendlich sein.

klen der maximalen Länge l enthält, aber für welches zusätzlich für die enthaltenen Komponenteninterpretation $\mathfrak{W}_i$ gemäß Definition 30 gilt $k' \geq l \cdot n$. Nun wird in jedem existierenden Zyklus ein Teilstück $\ldots v_i, v_{i+1} \ldots$ ausgewählt. Das die beiden Knoten verbindende Element d wird nur in v_i durch den Klon d' ersetzt, welcher noch in keinem anderem Knoten des Zyklus vorkommt. Dessen Existenz ist durch die begrenzte Länge des Zyklus und die hohe Anzahl von Klonen gesichert. Der Zyklus wird damit aufgebrochen, da v_{i+1} weiterhin das alte Element enthält und somit keine Kante mehr zu v_i existiert. Desweiteren handelt es sich bei d um ein anonymes Element, d.h. nicht um die Interpretation einer Objektvariable. Damit ist aus „Sicht der Wissensbasis“ d nicht von d' zu unterscheiden, weswegen diese Ersetzung die Modelleigenschaft der Interpretation nicht beeinträchtigt. □

Die Konsequenz ist, dass für jeden zyklischen Abhängigkeitsgraph entweder eine Ersetzung existiert, welche den Zyklus eliminiert, oder die zugehörige Abfrage mit `falsch` beantwortet werden kann. Die Eliminierung des Zyklus bedeutet jedoch die Ersetzung einer der beteiligten Variablen mit der Interpretation einer Objektvariable. Nach Definition 29 ist es aber äquivalent, die Variable bereits vorher durch die entsprechende Objektvariable zu ersetzen. Einzig die Frage, welche Variable mit welcher Objektvariable ersetzt werden muss, bleibt indeterminiert, weswegen hierbei theoretisch alle Möglichkeiten probiert werden müssen. Wenn $\mathfrak{V}_i^{qZ}$ die Teilmenge von $\mathfrak{V}^q$ ist, welche nur die Variablen enthält, die einerseits im betrachteten Zyklus Z liegen und andererseits der i-ten Komponente zugeordnet werden können, sowie $\#M$ die Anzahl der Elemente einer Menge M bezeichnet, so wird eine obere Schranke für die notwendige Anzahl an Ersetzungen für einen Zyklus Z durch folgende Formel markiert:

$$\sum_{i=1}^{m} \left(\#\mathcal{X}_i \cdot \#\mathfrak{V}_i^{qZ} \right)$$

Wird die durch eine Ersetzung entstehende Abfrage mit `wahr` beantwortet, kann die Ursprungsabfrage ebenfalls mit `wahr` beantwortet werden. Führt die Ersetzung zu einem `falsch`, so müssen weitere Ersetzungen getestet werden. Erst wenn keine der möglichen Ersetzungen zu einem positiven Ergebnis für die Ursprungsabfrage führt, kann diese mit `falsch` beantwortet werden.

In jedem Fall führt die Ersetzung zur Eliminierung des Zyklus, woraufhin mit den bereits vorgestellten Methoden die Beantwortung der Abfrage fortgeführt werden kann. Zur Verdeutlichung soll das Beispiel erneut aufgegriffen werden. Für eine bessere Lesbarkeit und erhöhte Anschaulichkeit soll q_3 um ein Termatom erweitert

werden. Desweiteren werden die Variablen durch ein vorangestelltes ? identifiziert und umbenannt. Es resultiert die strukturell im Wesentlichen zu q_3 identische Abfrage:

$$q_4 = \{(\texttt{?F}_1, \texttt{?L}_1, \texttt{?P}_1 : E), (\texttt{?F}_2, \texttt{?L}_1, \texttt{?P}_2 : E), (\texttt{?F}_2, \texttt{?L}_2, \texttt{?P}_1 : E), \\ (\texttt{?F}_2, \texttt{USA}, \texttt{?P}_3 : E), (\texttt{?P}_3 : \texttt{Mp3Player})\}$$

Bei dieser Abfrage stößt man auf den bereits bekannten Zyklus, sobald der einzige Randknoten $(\texttt{?F}_2, \texttt{USA}, \texttt{?P}_3 : E)$ und das Termatom zu einem neuen Atom verschmolzen werden:

$$q_4' = \{(\texttt{?F}_1, \texttt{?L}_1, \texttt{?P}_1 : E), (\texttt{?F}_2, \texttt{?L}_1, \texttt{?P}_2 : E), (\texttt{?F}_2, \texttt{?L}_2, \texttt{?P}_1 : E), \\ \left(\texttt{?F}_2 : \langle E \rangle^1 (V_{\texttt{USA}}, \texttt{Mp3Player})\right)\}$$

Eine exemplarische Wissensbasis, auf welche q_4' angewendet werden soll, findet sich in Abbildung 5.3. Unter der Annahme E stünde für die ternäre Relation „verkaufen eines Produktes in einem Land durch eine Firma“ repräsentiert die Abfrage nun die Fragestellung: Welche Firmen $\texttt{?F}_2$ gibt es, für die folgende Bedingungen gelten:

- sie verkaufen in den USA ein Produkt der Klasse Mp3Player
- sie verkaufen in Land $\texttt{?L}_2$ ein Produkt $\texttt{?P}_1$, welches von einer Firma $\texttt{?F}_1$ im Land $\texttt{?L}_1$ verkauft wird
- sie verkaufen in $\texttt{?L}_1$ bereits ein Produkt $\texttt{?P}_2$

Anders gefragt, welche potentiellen Konkurrenzfirmen bezüglich welcher Produkte und Länder gibt es für welche Firmen, die in den USA Produkte der Klasse Mp3Player verkaufen, wobei beide Firmen bereits im gleichen Land $\texttt{?L}_1$ tätig sind?[9]

Zur Eliminierung des Zyklus wird nun nicht-deterministisch eine Objektvariable für eine im Zyklus enthaltene Variable eingesetzt. Im Beispiel soll für $\texttt{?F}_2$ die Objektvariable BlackPear ausgewählt werden. Wie Abbildung 5.4 zeigt, ist die Abfrage damit tatsächlich vollständig zyklenfrei. Die nun anwendbaren Transformationen führen schließlich zur nicht weiter umformbaren Abfrage:

$$q_4^* = \Big\{\left(\texttt{BlackPear} : \langle E \rangle^1 (V_{\texttt{USA}}, \texttt{Mp3Player})\right), \\ \left(\texttt{?F}_1 : \langle E \rangle^1 \left(\langle E \rangle^2 (V_{\texttt{BlackPear}}, \top_3), \langle E \rangle^3 (V_{\texttt{BlackPear}}, \top_2)\right)\right)\Big\}$$

[9] Dieses akademische Beispiel erhebt keinen Anspruch auf betriebswirtschaftliche Relevanz.

Γ_1

$\texttt{Company} \sqsubseteq (\langle E \rangle^1(\top_2, \top_3))$
$\texttt{GlobalCompany} \sqsubseteq \texttt{Company}$

$\texttt{GoMusicInc : GlobalCompany}$
$\texttt{BlackPear : Company}$

Γ_2

$\texttt{Country} \sqsubseteq \texttt{Region}$

$\texttt{USA : Country}$
$\texttt{Japan : Country}$
$\texttt{Germany : Country}$

Γ_3

$\texttt{Mp3Player} \sqsubseteq \texttt{Product}$

$\texttt{oPid : Mp3Player}$
$\texttt{oRover : Mp3Player}$
$\texttt{oSpeaker : Product}$

$\Gamma^{\mathcal{E}}$

$\texttt{(BlackPear, USA , oRover)} : E$
$\texttt{(BlackPear , Germany, oPid)} : E$
$\texttt{(GoMusicInc, Germany, oSpeaker)} : E$
$\texttt{(GoMusicInc , Japan, oPid)} : E$

Abbildung 5.3: $\mathcal{E}$-Connection Wissensbasis

welche im Kontext der Wissensbasis

$$(\Gamma_1 \cup \{\texttt{BlackPear} : V_{\texttt{BlackPear}}\}, \Gamma_2 \cup \{V_{\texttt{USA}}\}, \Gamma_3, \Gamma^{\mathcal{E}})$$

mit der Hilfe nur zweier Erfüllbarkeitstests mit `wahr` beantwortet werden kann. Bei genauer Betrachtung führt auch eine Ersetzung von $\texttt{?F}_1$ mit der Objektvariable `BlackPear` zum Erfolg, was gleichzeitig eine Schwäche des Beispiels offenbart: Eine Ersetzung, die zur Folge hat, dass $\texttt{?F}_1 = \texttt{?F}_2$, $\texttt{?P}_1 = \texttt{?P}_2$ und $\texttt{?L}_1 = \texttt{?L}_2$ gelten, führt hier zu einer positiven Antwort, wenngleich eine Adaption der Abfrage als Abfrage mit Antwortvariablen schließlich auch das interessante Ergebnis zu Tage fördern würde. Dieses ergibt, dass die beiden in `Germany` tätigen Firmen `BlackPear` und `GoMusicInc` das gleiche Produkt `oPid` in unterschiedlichen Ländern `Germany` bzw. `Japan` verkaufen. Damit würden sich diese noch ko-existierenden Firmen in direkte Konkurrenz begeben, sobald eine Expansion des Produktportfolio in die entsprechenden Länder vorgenommen wird.

Die Vorgehensweise zur Entscheidung der logischen Konsequenz ist in Algorithmus 5.1 zusammengefasst dargestellt.

5.1.3 Beantwortung konjunktiver Abfragen

Damit sind alle Voraussetzungen erfüllt, ein vollständiges Verfahren zur Beantwortung konjunktiver Abfragen auf $\mathcal{E}$-Connections beschreiben zu können. Im Kern des

Algorithmus 5.1 : Konsequenz konjunktiver Abfragen auf $\mathcal{E}$-Connections

Eingabe : Boolesche Abfrage q^B
Eingabe : Wissensbasis eKB der $\mathcal{E}$-Connection $C^{\mathcal{E}}(\mathcal{S}_1, \ldots \mathcal{S}_m)$
Daten : `S(`x`)` $= j$ falls x der Logik $\mathcal{S}_j$ zugeordnet werden kann
Daten : Menge $\mathfrak{V}^q = \{x_1, \ldots, x_k\}$ der (existenziell quantifizierten) Variablen aus q
Daten : Graph $G(q^B)$
Ergebnis : `Wahr` oder `Falsch`
Beginn
  solange $G(q^B)$ *Knoten enthält* **tue**
    wenn *Ein Randknoten* N *existiert* **dann**
      `Aufrollen(`q^B, N, eKB`)` ;
    sonst
      wähle beliebigen Zyklus $Z = (N_1, \ldots, N_l)$; `// vgl. Def. 27`
      für *jede Kantenbeschriftung (Variable)* x *in* Z **tue**
        für *jede Objektvariable* $a \in \mathcal{X}_{\mathtt{S}(x)}$ **tue**
          $q' \leftarrow q^B$ mit allen Vorkommen von x ersetzt mit a;
          `// Rekursiver Aufruf!`
          wenn `LogischeKonsequenz(`q', eKB`)` **dann**
            kehre zurück und liefere `Wahr` ;
      kehre zurück und liefere `Falsch` ;
  `/*` $G(q^B)$ `ist nun leer, d.h. es sind nur noch Termatome vorhanden.` `*/`
  für *jedes Termatom* $q_i^B \in q^B$ **tue**
    unterscheide *Form von* q_i^B **tue**
      Fall $(a : t)$, *mit* a *Objektvariable und* t *Term der gleichen Logik* j
        wenn $eKB \cup \{a : t\}$ *nicht erfüllbar* **dann**
          kehre zurück und liefere `Falsch` ;
      Fall $(x : t)$, *mit* x *Variable und* t *einem Term der Logik* j
        wenn $eKB \cup \{\top_j \sqsubseteq \neg t\}$ *erfüllbar* **dann**
          kehre zurück und liefere `Falsch` ;
  kehre zurück und liefere `Wahr` ;
Ende

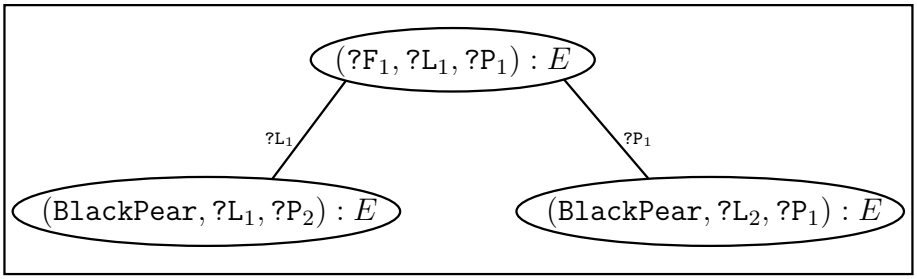


Abbildung 5.4: Graph einer aus q_4' erhaltenen Abfrage.

Verfahrens wird eine Abfrage in eine endliche Menge von Booleschen Abfragen umgewandelt, welche jede für ein mögliches Element der Antwortmenge steht. Ein Jeder dieser Kandidaten wird mittels unterschiedlicher Methoden umgeschrieben, um alle Verbindungsrelationsatome zu eliminieren. Die Menge wächst dabei eventuell weiter an, enthält aber nur aus Termatomen bestehende Abfragen. Kann von diesen auch nur eine mit `wahr` beantwortet werden, d.h. ist sie logische Konsequenz der abgefragten (und evtl. um einige Axiome erweiterten) Wissensbasis, wird das zum entsprechenden Antwortkandidaten gehörige Ersetzungstupel als Element der Antwortmenge übernommen. Eine formalere Darstellung des Verfahrens ist in Algorithmus 5.2 gegeben. Erneut wird ein nicht optimierter „Brute Force“-Ansatz zur Beantwortung herangezogen, da er das Grundprinzip verdeutlicht und mit einfachen Mitteln optimiert werden kann.

Es sei darauf hingewiesen, dass dies bereits in gewissem Maße die Reduktion der Beantwortung konjunktiver Abfragen über $\mathcal{E}$-Connections auf die Beantwortung in den Komponentenlogiken bedeutet, da die verbliebenen Termatome jeweils einer Komponentenlogik zugeordnet werden können. Aufgrund der durch die Verknüpfung der Logiken zugelassenen Funktion $\langle E_j \rangle$, besteht jedoch noch immer eine für Erfüllbarkeitsprüfungen relevante Verbindung zu den anderen Komponenten, weswegen nicht von einer vollständigen Reduktion gesprochen werden kann. Diese verbliebene Verbindung zu eliminieren macht ein spezielles Entscheidungsverfahren für die Erfüllbarkeit von Termaussagen der Form $x_i : \langle E_j \rangle^i(\ldots)$ notwendig, welches zwar außerhalb des Rahmens dieser Arbeit liegt, in zukünftigen Arbeiten aber unbedingt Gegenstand der Überprüfung sein sollte.

Algorithmus 5.2 : Beantwortung konjunktiver Abfragen über $\mathcal{E}$-Connections

Eingabe : Wissensbasis $eKB = (\Gamma_1, \ldots, \Gamma_m, \Gamma_{\mathcal{E}})$
Eingabe : konjunktive Abfrage q mit Variablen $\mathfrak{V} = \{x_1, \ldots, x_l\}, l \in \mathbb{N}$
Daten : $\mathtt{S}(x) = j$ falls x der Logik $\mathcal{S}_j, 1 \leq j \leq m$ zugeordnet werden kann
Daten : Antwortkandidaten

$$A_K = \mathcal{X}_{\mathtt{S}(x_1)} \cup \{\mathtt{exists}_{x_1}\} \times \ldots \times \mathcal{X}_{\mathtt{S}(x_l)} \cup \{\mathtt{exists}_{x_l}\}$$

Ausgabe : Antwort A auf q gemäß Definition 24
Beginn
  für *jedes Element* $a \in A_K$ **tue**
    wenn `LogischeKonsequenz` *(*$[q]_a$*,* eKB*)* **dann**
      $A \leftarrow A \cup \{a\}$;
  Entferne Redundanz aus A gemäß Definition 24 Bedingung 5.1b;
  liefere Ergebnis A;
Ende

Dieses grundlegende Verfahren zur Beantwortung konjunktiver Abfragen auf $\mathcal{E}$-Connections lässt sich nun als Basis für unterschiedliche Erweiterungen nutzen.

5.1.4 Erweiterungen

Bisher werden alle Variablen einerseits als Antwortvariablen, implizit jedoch auch als existenziell quantifizierte Variablen betrachtet. In einigen Fällen ist der Nutzer jedoch an den möglichen Ersetzungen für einige der verwendeten Variablen gar nicht interessiert. Dazu lässt sich die Syntax erweitern, indem Variablen, denen statt des `?` ein `_` vorgestellt ist, ausschließlich als existenzielle Variablen interpretiert werden. Dazu wird die entsprechende Dimension des Antwortraumes A_K auf ein einzelnes, entsprechendes Element `exists` beschränkt. In gleicher Weise ließen sich die Variablen durch syntaktische Mittel auf Antwortvariablen beschränken, falls nur gesichertes Wissen abgefragt werden soll.

Eine weitere und mächtigere Erweiterung erlaubt die Abfrage von Relationen $r \in \mathcal{R}_i$ einer Komponente $\mathcal{L}_i$ mittels Atomen der Form $(x_1, \ldots, x_{n_r}) : r$ für Variablen bzw. Objektvariablen x_i. Die Berücksichtigung solcher Atome führt zu einer erhöhten Komplexität und zu gewissen Annahmen bzgl. der vorhandenen Funktionen, wie sich bereits in Abschnitt 2.3 gezeigt hat. Unabhängig davon lassen sich Abfragen innerhalb der Komponenten in jedem Fall beantworten, wenn sie sich als einzelne Termatome

der Form $(x : t)$ im Rahmen der $\mathcal{E}$-Connection-Wissensbasis darstellen lassen. Dabei kann t ein beliebig komplexer Term sein, der jedoch keine Variablen enthalten darf. Dies erfordert keine gesonderte Berücksichtigung und ist mit den vorgestellten Verfahren handhabbar.

Wie aus dem Bereich der Datenbanken bekannt, lassen sich auch Möglichkeiten schaffen, die Antwortmenge nachträglich zu reduzieren. Es sei hierbei an die `FILTER`-Klausel von SPARQL erinnert [PS08].

5.1.5 Verteilung des Verfahrens

Das beschriebene Verfahren zur Beantwortung konjunktiver Abfragen über $\mathcal{E}$-Connections lässt sich nur mit Hilfe eines Entscheidungsverfahrens für die Erfüllbarkeit von Wissensbasen implementieren. Damit ist die vollständige Reduktion auf eine Beantwortung in den Komponenten gleichermaßen von einem entsprechenden Entscheidungsverfahren abhängig. Soll die Beantwortung zudem verteilt erfolgen, muss die noch unbeantwortete Frage geklärt werden, in welche Teile eine formale Wissensbasis zerlegt werden kann. Zudem müssen die interagierenden Komponenten beschrieben werden, welche unter Zuhilfenahme der jeweils lokalen Module Teilaufgaben lösen können.

Für allgemeine $\mathcal{E}$-Connections ist bisher kein Entscheidungsverfahren implementiert worden[10]. Es lässt sich jedoch aus dem Beweis zum Transfer der Entscheidbarkeit in [KLWZ04] eine allgemeine, nicht zur Implementierung geeignete Beschreibung eines solchen Verfahrens entnehmen. Das von Kutz et al. vorgestellte Verfahren ist hochgradig nicht-deterministisch, enthält jedoch die Reduktion der Erfüllbarkeit der Wissensbasis auf die Erfüllbarkeit von speziellen Wissensbasen nur der Komponentenlogiken. Diese enthalten keinen Bezug mehr zu $\mathcal{E}$-Connections und lassen sich somit durch bereits vorhandene Verfahren für die Komponentenlogiken entscheiden, welche als Black-Box genutzt werden können. Obwohl dadurch prinzipiell eine Verteilung möglich ist, lässt sich dies nur unter Berücksichtigung eines praktischen Verfahrens für spezielle Instanzen von $\mathcal{E}$-Connections endgültig feststellen.

5.1.6 Implementierungskonzept

Trotz der gezeigten Hürden, welche es auf einem Weg zu einer Implementierung zunächst zu überwinden gilt, soll dieser Abschnitt ein nachvollziehbares Konzept zur

[10] Wie Oliver Kutz auf eine Anfrage vom 12.01.2009 bestätigte.

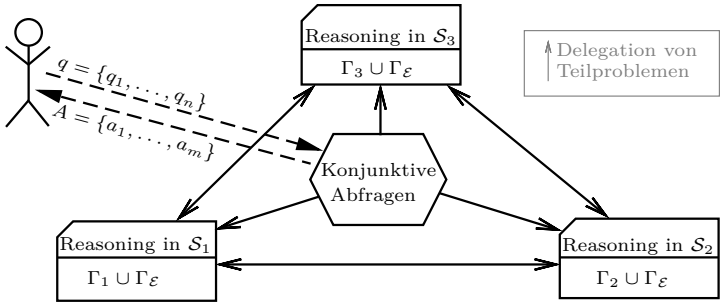


Abbildung 5.5: Verteiltes Reasoning-System

Realisierung verteilter Beantwortung von Abfragen über $\mathcal{E}$-Connections präsentieren. Als Voraussetzungen wurden die Existenz eines praktikablen Entscheidungsverfahrens für Wissensbasis-Konsistenz und dessen Eigenschaft, verteilt arbeiten zu können, genannt. Da eine Erfüllung dieser Voraussetzungen im Rahmen dieser Arbeit nicht geleistet werden kann, soll davon ausgegangen werden, dass ein beide Anforderungen erfüllendes Entscheidungsverfahren vorliegt. Dies ist eine realitätsnahe Annahme, wie das Beispiel von Distributed Description Logics zeigt. Diese können als eine spezielle Instanz von $\mathcal{E}$-Connections verstanden werden und wurden im Rahmen eines P2P-basierten Systems implementiert. Eine solche Netzwerkarchitektur ist naheliegend für verteilte Entscheidungsverfahren, so dass sie auch hier verwendet wird. In Anlehnung an DRAGO zeigt Abbildung 5.5 eine mögliche Ausprägung eines solchen Systems.

Dabei beherbergen physisch getrennte Knoten die entsprechenden Komponenten Γ_i einer Wissensbasis $eKB = (\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_\mathcal{E})$. Um einen weiteren Knoten zu vermeiden, soll in diesem Entwurf jede Komponente zusätzlich über eine Kopie von $\Gamma_\mathcal{E}$ verfügen. Eine typische Reasoningaufgabe kann nun an einen beliebigen Knoten gestellt werden, die Delegation der (Teil-) Aufgaben an den jeweils verantwortlichen Knoten wird automatisch abgewickelt.

Zur Einführung konjunktiver Abfragen empfiehlt sich nun eine weitere Komponente, welche die Umwandlung in atomare Reasoningaufgaben übernimmt. Während dieses Prozesses ist teilweise Wissen über die in den Komponenten enthaltenen Objektvariablen erforderlich, welches von den verantwortlichen Knoten bezogen werden kann. Ganz nach dem Vorbild des Mediator-Musters und seinem Einsatz in den im Abschnitt 4 genannten Ansätzen von Quilitz und Langegger [QL08, LBW07], dele-

giert dieser Knoten anschließend nur noch die atomaren Reasoningprobleme an die zuständigen Knoten, um aus den jeweiligen Antworten eine vollständige und korrekte Antwort auf die Abfrage zu konstruieren.

Es bleibt eine offene Frage, wie die Details einer solchen Implementierung zu gestalten sind. Dies ist abhängig von den Eigenschaften eines konkreten verteilten Entscheidungsalgorithmus und daher im Rahmen einer theoretisch ausgerichteten Arbeit nicht zu beantworten.

5.2 Distributed Description Logics

Distributed Description Logics können als eine spezielle $\mathcal{E}$-Connection-Instanz verstanden werden, wurden jedoch, wie Abschnitt 3.1.2 bereits zeigte, unabhängig davon entwickelt. Zur Abfrage eines solchen Verbunds von Ontologien beschäftigten sich die Autoren bereits mit dem Problem, fremde Individuen als Ergebnisse zu ermöglichen [ST05a]. Es wurden zunächst nur Instanzabfragen für ein gegebenes Konzept C untersucht, was der einfachen Abfrage $\{x : C\}$ entspricht. Um der Idee der sichtabhängigen Interpretation gerecht zu werden, wurde es ermöglicht, eine Transformationsfunktion für Individuen aus fremden Ontologien anzugeben. Werden beispielsweise unter `Kontakt` in Ontologie A URLs der Internetseiten von Personen verstanden, in Ontologie B jedoch deren E-Mail-Adressen, so ist es damit möglich, erstere in letztere zu transformieren, indem eine entsprechende Funktion die Webseite analysiert und die E-Mail-Adresse extrahiert [ST05b]. Es wird ein verteilter Algorithmus angegeben der unter gewissen Bedingungen korrekte und vollständige Ergebnisse liefert.

In Kombination mit den in diesem Kapitel vorgestellten Verfahren für $\mathcal{E}$-Connections sollten sich damit auch komplexere konjunktive Abfragen auf DDL realisieren lassen. Es sei daran erinnert, dass als Ergebnis der Umwandlung einer konjunktiven Abfrage eine Menge von Abfragen entsteht, deren Elemente nur noch voneinander unabhängige Konjunkte der Form $(x : C)$ enthalten. Eine genauere Untersuchung dieser Kombination kann jedoch als offenes Forschungsthema verstanden werden.

Ein weiterer Ansatz beschäftigte sich auf anderem Weg mit konjunktiven Abfragen über DDL [AZ07]. Als Resultat ist ein unter zwei Bedingungen korrektes, aber im Allgemeinen unvollständiges Verfahren zu verzeichnen. Dabei beschränken sich die Autoren ebenfalls auf zielgerichtete Abfragen, welche nur die Elemente der Signatur einer Ontologie enthalten dürfen. Dazu wird ausgehend von der anvisierten eine erwei-

terte Wissensbasis konstruiert, welche als abgeschlossen bezüglich der logischen Konsequenz im Rahmen der DDL-Semantik gilt. Dies führt zur ersten Bedingungen: Die Existenz eines korrekten und vollständigen Verfahrens zur Berechnung aller logischen Konsequenzen einer Menge von verbundenen Wissensbasen, welches nach endlicher Laufzeit terminiert. Die zweite Bedingung ist die Möglichkeit, lokale Abfragen beantworten zu können, was gleichzeitig die Intention der Autoren vervollständigt. Gerade für die erste und zweifellos schwerwiegendere Bedingung ist kein Verfahren vorgestellt oder auch nur ein Ansatzpunkt genannt worden, so dass diesem Ansatz noch keine Aussicht auf Verwirklichung zugeschrieben werden kann.

5.3 Analytische Modularisierung

Im Gegensatz zu den bisher untersuchten Formalismen sollen konjunktive Abfragen noch über einem weiteren, komplementären Ansatz zur Modularisierung von Ontologien untersucht werden. Bei analytischer Modularisierung ergibt sich aufgrund der beizubehaltenden Semantik der verwendeten Logik kaum ein Unterschied zu konjunktiven Abfragen über vollständig integrierte Ontologien.

Die grundlegende Idee bleibt die Extraktion eines vollständigen Moduls, d.h. einer Teilontologie aus der importierten Ontologie. Dabei werden die verwendeten, fremden Symbole als Signatur S zusammengefasst und davon ausgehend die kleinste Teilmenge der Axiome gesucht, welche bezüglich dieser Signatur zu identischen Schlussfolgerungen führt. Diese Teilmenge wird schließlich der importierenden Ontologie hinzugefügt und Modul genannt. Anschließend können die im Abschnitt 2.3 vorgestellten Verfahren für konjunktive Abfragen unverändert übernommen werden. Einzig die Verwendung von Symbolen innerhalb der Abfrage, welche nicht bereits in der ursprünglichen Ontologie erwähnt und somit für die Modulextraktion berücksichtigt wurden, erfordert eine spezielle Beachtung.

Zur Verdeutlichung sei folgender Fall gegeben. Eine Ontologie $\mathcal{T}$ verwendet Symbole $S_1 \subseteq Sig(\mathcal{T}_1)$ aus einer fremden Ontologie $\mathcal{T}_1$ und andere Symbole $S_2 \subseteq Sig(\mathcal{T}_2)$ aus einer weiteren Ontologie $\mathcal{T}_2$.

$$\begin{aligned} S_1 =& Sig(\mathcal{T}) \cap Sig(\mathcal{T}_1) \\ S_2 =& Sig(\mathcal{T}) \cap Sig(\mathcal{T}_2) \end{aligned}$$

Nun wird eine konjunktive Abfrage q an $\mathcal{T}$ gestellt. Normalerweise würden die Module M_i so ermittelt werden, dass alle Schlussfolgerungen bezüglich der Symbole in S_i gezogen werden können. Nun kann jedoch q weitere Symbole enthalten, welche zunächst nicht berücksichtigt wurden. Wenn man die Symbole, welche in der Abfrage vorkommen, in S_q zusammenfasst, können erweiterte Signaturen S'_i wie folgt ermittelt werden.

$$\begin{aligned}
S'_1 =& (Sig(\mathcal{T}) \cup S_q) \cap Sig(\mathcal{T}_1) = (Sig(\mathcal{T}) \cap Sig(\mathcal{T}_1)) \cup (S_q \cap Sig(\mathcal{T}_1)) \\
S'_2 =& (Sig(\mathcal{T}) \cup S_q) \cap Sig(\mathcal{T}_2) = (Sig(\mathcal{T}) \cap Sig(\mathcal{T}_2)) \cup (S_q \cap Sig(\mathcal{T}_2)) \\
S'_1 =& S_1 \cup (Sig(\mathcal{T}_1) \cap S_q) \\
S'_2 =& S_2 \cup (Sig(\mathcal{T}_2) \cap S_q)
\end{aligned}$$

Diese werden für die Berechnung neuer Module M'_i herangezogen. Mit Hilfe dieser erweiterten Module ist schließlich die Beantwortung von q vollständig möglich, selbst wenn darin beliebige Symbole aus den importierten Ontologien abgefragt werden. Die Erweiterungen der Wissensbasis im Rahmen der Beantwortung von konjunktiven Abfragen (vgl. Abschnitt 2.3) müssen auf ihre Verträglichkeit mit den Anforderungen dieses Ansatzes zur Modularisierung geprüft werden, was hier jedoch unberücksichtigt bleiben wird.

Es wird deutlich, dass zur Beantwortung konjunktiver Abfragen bei analytischer Modularisierung auf die Beantwortung konjunktiver Abfragen in allgemeinen Beschreibungslogiken zurückgegriffen werden kann. Aufgrund der Arbeitsweise von analytischer Modularisierung lässt sich jedoch nicht von einer Verteilung sprechen, da die fremden Symbole in die lokalen Wissensbasen importiert werden. Nimmt man eine große Anzahl von Individuen in der fremden ABox an, ist von diesem sehr einfachen Ansatz der Import großer Datenmengen zu erwarten.

6 Zusammenfassung

Zur Realisierung von Abfragen auf verteilten Ontologien wurden Formalismen untersucht, welche bestehende Ontologiesprachen in einen modularen Kontext einzubetten vermögen. Eine Verknüpfung der jeweiligen Ontologien wird dabei meist über eine übergeordnete Interpretation realisiert, welche die Elemente der lokalen Domänen in Relation setzt. Dabei werden weitere Bedingungen an die lokalen Modelle gestellt, deren Erfüllung für die Konsistenz des Verbundes notwendig sind. Es zeigte sich, dass die unterschiedlichen Ansätze für eine erweiterte Semantik in gewissen Teilen aufeinander abbildbar sind. Dem gegenüber wurde eine noch junge Variante zur Verknüpfung von Beschreibungslogik-Ontologien vorgestellt, welche Syntax und Semantik unberührt lässt. Allen Konzepten ist gemein, dass keine beliebig komplexen Verknüpfungen der Elemente unterschiedlicher Wissensbasen möglich sind.

Die Untersuchung konjunktiver Abfragen im Rahmen der vorgestellten Ansätze fokussierte auf $\mathcal{E}$-Connections und erzielte dabei das wesentliche Ergebnis dieser Arbeit: Konjunktive Abfragen über eine $\mathcal{E}$-Connection sind genau dann beantwortbar, wenn diese entscheidbar ist und eine der zwei folgenden Bedingungen erfüllt ist. Entweder ist die Abfrage azyklisch oder aber die $\mathcal{E}$-Connection umfasst nur Komponenten, welche der Definition von Anzahl-Toleranz gemäß Kutz et al. genügen.

Um dieses Ergebnis zu erzielen, wurde eine Abfragesprache auf Wissensbasen von $\mathcal{E}$-Connection eingeführt, welche zunächst nur Abfragen von Termzugehörigkeiten und Verbindungsrelationen zulässt. In Anlehnung an Arbeiten zur Abfrage von Wissensbasen von Beschreibungslogiken konnte dazu ein korrektes und vollständiges Verfahren ermittelt werden. Dieses reduziert in einem ersten Schritt die Beantwortung von Abfragen auf die logische Konsequenz Boolescher Abfragen. In einem weiteren Schritt werden die Abfragenelemente über Verbindungsrelationen eliminiert, wobei dabei die Berücksichtigung von Zyklen von Variablen eine besondere Herausforderung gewesen ist. Im letzten Schritt des Verfahrens entsteht eine endliche Menge von Erfüllbarkeitsprüfungen von $\mathcal{E}$-Connection-Wissensbasen, deren Ergebnisse unmittelbar zu den Ergebnissen für die ursprüngliche Abfrage führt.

Eine vollständige Reduktion auf die Beantwortung in den Komponentenlogiken ist aufgrund der Semantik prinzipiell nicht möglich. Es konnte jedoch gezeigt werden, dass die Möglichkeit einer verteilten Beantwortung von Abfragen von einem dafür geeigneten Algorithmus für die Konsistenzprüfung abhängig ist. Ein solches Verfahren ist für den allgemeinen Fall jedoch unbekannt, stattdessen sind praktische Entscheidungsverfahren immer im Rahmen konkreter Ausprägungen von $\mathcal{E}$-Connections, d.h. konkreter Kombinationen von Logiken zu entwickeln.

Daraus leitet sich auch die Erkenntnis ab, dass $\mathcal{E}$-Connections zwar einerseits wegen fehlender Standardisierung und Werkzeugunterstützung für praktische Belange sperrig zu handhaben sind, andererseits aber in definierten Anwendungsszenarien ob ihrer Fähigkeit, über mehrere Logiken schlussfolgern zu können, dennoch erwogen werden sollten. Die im Rahmen von gängigen Informationssystemen notwendigen konjunktiven Abfragen sind nur eine geringe Hürde, einzig das Finden eines praktikablen Entscheidungsverfahrens bleibt eine anspruchsvolle Aufgabe.

Inspiriert durch Distributed Description Logics und deren Implementierung DRAGO, sowie der Tatsache, dass sich DDL in gewissem Maße als $\mathcal{E}$-Connection verstehen lassen, konnte schließlich ein allgemeines Implementierungskonzept zur Beantwortung von konjunktiven Abfragen über $\mathcal{E}$-Connections vorgestellt werden. Dieses basiert auf einem P2P-Netz von Schlussfolgerungskomponenten, welche jeweils für Schlussfolgerungen in einer der Komponenten einer $\mathcal{E}$-Connection verantwortlich sind. Zusätzlich wird eine Mediator-Komponente eingeführt, welche Abfragen entgegen nimmt und mit Hilfe der restlichen Komponenten eine Antwort konstruiert.

Abgerundet wurde die Arbeit mit konkreten Betrachtungen zu konjunktiven Abfragen im Kontext anderer Ansätze für verteilte Ontologien, wie DDL und analytischer Modularisierung.

Zukünftige Forschungsarbeiten sollten sich mit der Ermittlung konkreter Entscheidungsverfahren für spezielle Klassen von $\mathcal{E}$-Connections beschäftigen, da dies zweifellos als größte Schwäche gewertet werden kann. Mit einem allgemeinen Verfahren und einer darauf aufbauenden, generischen Implementierung ließen sich komplexe Verknüpfungen erstellen, die auch in praktischen Szenarien von großem Nutzen sein würden. Eine Herausforderung wäre dabei die Kompatibilität mit in OWL formulierten Wissensbasen. Damit ließe sich das „Semantic Web“ in gewissen Grenzen an zusätzlichen Logiken „andocken“, wodurch einige Einschränkungen von Beschreibungslogiken, wie das unbefriedigende Schlussfolgern bzgl. Ort und Zeit, umgangen werden könnten.

Literaturverzeichnis

[AZ07] Alkhateeb, Faisal und Antoine Zimmermann: *Query Answering in Distributed Description Logics.* In: Labiod, Houda und Mohamed Badra (Herausgeber): *New Technologies, Mobility and Security - Proceedings of NTMS'2007 Conference*, Lecture Notes in Computer Science, Seiten 523–534. Springer, Mai 2007. [66]

[BCH06] Bao, Jie, Doina Caragea und Vasant Honavar: *A Tableau-Based Federated Reasoning Algorithm for Modular Ontologies.* In: *Web Intelligence*, Seiten 404–410. IEEE Computer Society, 2006. [34]

[BCM+03] Baader, Franz, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi und Peter F. Patel-Schneider (Herausgeber): *The description logic handbook: theory, implementation, and applications.* Cambridge University Press, New York, NY, USA, 2003. [1, 17, 57]

[BLC07] Bui, Ngot Phu, SeungGwan Lee und TaeChoong Chung: *Query Distributed Ontology over Grid Environment.* In: *ICCSA 2007, International Conference on Computational Science and Its Applications*, 2007. [41]

[BLHL01] Berners-Lee, Tim, James Hendler und Ora Lassila: *The Semantic Web.* Scientific American, May 2001. [1]

[BRM04] Brickley, Dan, R..Guha und Brian McBride: *RDF Vocabulary Description Language 1.0: RDF Schema.* Website, `http://www.w3.org/TR/2004/REC-rdf-schema-20040210/`, Februar 2004. [23]

[Bro07] *Der Brockhaus: in 15 Bänden.* Ständig aktualisierte Online-Ausgabe, `http://www.brockhaus.de/`, 2002-2007. [1]

[BS02] Borgida, Alex und Luciano Serafini: *Distributed description logics: Directed domain correspondences in federated information sources.* In: *On The Move to Meaningful Internet Systems 2002: CoopIS, Doa, and ODBase, volume 2519 of LNCS*, Seiten 36–53. Springer Verlag, 2002. [31]

[BS03] Borgida, Alex und Luciano Serafini: *Distributed description logics: Assimilating information from peer sources.* Journal of Data Semantics, 1:2003, 2003. [31]

[BSH07] Bao, Jie, Giora Slutzki und Vasant Honavar: *A Semantic Importing Approach to Knowledge Reuse from Multiple Ontologies.* In: *AAAI*, Seiten 1304–1309. AAAI Press, 2007. [33, 34]

[CJB99] Chandrasekaran, B., John R. Josephson und V. Richard Benjamins: *What Are Ontologies, and Why Do We Need Them?* IEEE Intelligent Systems, 14(1):20–26, 1999. [30]

[CK07] Cuenca Grau, Bernardo und Oliver Kutz: *Modular Ontology Languages Revisited.* In: *Proceedings of the IJCAI-07 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa)*, 2007. [23, 24, 33, 34, 37]

[CP08] Clark, Kendall und Bijan Parsia: *Modularity and OWL.* Bericht für das "National Cancer Institute", erhalten durch direkte Kommunikation mit den Autoren, März 2008. [29, 30, 36, 37]

[CPS04] Cuenca Grau, Bernardo, Bijan Parsia und Evren Sirin: *Tableau algorithms for econnections of description logics.* Technischer Bericht, Maryland Information and Network Dynamics Lab, 2004. [30]

[CPS06] Cuenca Grau, Bernardo, Bijan Parsia und Evren Sirin: *Combining OWL ontologies using E-Connections.* J. Web Sem., 4(1):40–59, 2006. [24, 29]

[CPSK05] Cuenca Grau, Bernardo, Bijan Parsia, Evren Sirin und Aditya Kalyanpur: *Automatic Partitioning of OWL Ontologies Using $\mathcal{E}$-connections.* In: *DL 2005, Proceedings of the 18th International Workshop on Description Logics, Edinburgh, UK, 2005*, 2005. [30]

[Cue05] Cuenca Grau, Bernardo: *COMBINACIÓN E INTEGRACIÓN DE ONTOLOGÍAS EN LA WEB SEMÁNTICA.* Doktorarbeit, Dpto. de Informática Universitat de València, 2005. [33]

[Dat86] Date, C J: *A guide to the SQL standard.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986. [10]

[FHH+07] Feigenbaum, Lee, Ivan Herman, Tonya Hongsermeier, Eric Neumann und Susie Stephens: *The Semantic Web In Action.* Scientific American, November 2007. [1, 23]

[GHLS07] Glimm, Birte, Ian Horrocks, Carsten Lutz und Uli Sattler: *Conjunctive Query Answering in the Description Logic SHIQ.* In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007. [39]

[GHS06] Glimm, Birte, Ian Horrocks und Uli Sattler: *Conjunctive Query Answering for Description Logics with Transitive Roles.* In: *CEUR workshop proceedings of the 2006 International Workshop on Description Logics (DL 06)*, 2006. [39]

[GHS07] Glimm, Birte, Ian Horrocks und Uli Sattler: *Conjunctive Query Entailment for SHOQ.* In: *Proceedings of the 2007 Description Logic Workshop (DL 2007)*, 2007. [39]

[GLHS08] Glimm, Birte, Carsten Lutz, Ian Horrocks und Ulrike Sattler: *Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic.* Journal of Artificial Intelligence Research, 31:150–197, 2008. [12, 22]

[Gli07] Glimm, Birte: *Querying Description Logic Knowledge Bases.* Doktorarbeit, The University of Manchester, Manchester, United Kingdom, 2007. [11]

[Hom08] Homola, Martin: *Subsumption Propagation between Remote Ontologies in Distributed Description Logic.* In: *Proceedings of the 21st International Workshop on Description Logics (DL2008)*, 2008. [33]

[HSTT99] Horrocks, Ian, Ulrike Sattler, Sergio Tessaris und Stephan Tobies: *Query Containment Using a DLR ABox.* Technischer Bericht, RWTH Aachen, 1999. [14, 15, 50, 51]

[HT00] Horrocks, Ian und Sergio Tessaris: *A Conjunctive Query Language for Description Logic Aboxes.* In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, Seiten 399–404. AAAI Press / The MIT Press, 2000. [13, 14, 16, 17, 18, 39, 43, 44, 50]

[HW07] Haase, Peter und Yimin Wang: *A Decentralized Infrastructure for Query Answering over Distributed Ontologies.* In: *Proceedings of The 22nd Annual ACM Symposium on Applied Computing (SAC)*, Seoul, Korea, 3 2007. ACM Press. [40]

[JRCS+08] Jimenez-Ruiz, Ernesto, Bernardo Cuenca Grau, Thomas Schneider, Ulrike Sattler und Rafael Berlanga: *Safe and Economic re-use of ontologies: a logic-based methodology and tool support.* In: *OWLEd 2008, Proceedings of the 4th International Workshop: OWL Experiences and Directions, April 1-2, 2008*, 2008. Another version of this paper will also appear in Procedings of the 21st Description Logics Workshop (DL-2008). [35, 36, 41]

[KLWZ04] Kutz, Oliver, Carsten Lutz, Frank Wolter und Michael Zakharyaschev: *$\mathcal{E}$-Connections of Abstract Description Systems.* Artif. Intell., 156(1):1–73, 2004. [24, 26, 27, 28, 29, 33, 43, 44, 45, 46, 57, 64]

[LBW07] Langegger, Andreas, Martin Blöchl und Wolfram Woss: *Sharing Data on the Grid using Ontologies and distributed SPARQL Queries.* Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on, Seiten 450–454, Sept. 2007. [40, 65]

[Len02] Lenzerini, Maurizio: *Data integration: A theoretical perspective.* In: *Proceedings of the 21st ACM SIGMOD-SIGART Symposium on Principles of Database*, 2002. [10, 39]

[LKB08] Langegger, Andreas, Wolfram Klöss und Martin Blöchl: *A Semantic Web Middleware for Virtual Data Integration on the Web.* In: *Proceedings of the 5th European Semantic Web Conference (ESCW08)*, Seiten 493–507, 2008. [40]

[Lut08] Lutz, Carsten: *Description Logics.* Website, `http://dl.kr.org/`, zuletzt besucht am 23. September 2008, September 2008. [5]

[LWW07] Lutz, Carsten, Dirk Walther und Frank Wolter: *Conservative Extensions in Expressive Description Logics.* In: Veloso, Manuela (Herausgeber): *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, Seiten 453–458. AAAI Press, 2007. [35]

[MM08] Manola, Frank und Eric Miller: *RDF Primer, W3C Recommendation 10 February 2004.* Website, `http://www.w3.org/TR/rdf-primer/` zuletzt besucht am 26. Juni 2008, Februar 2008. [23]

[NFF+91] Neches, Robert, Richard Fikes, Tim Finin, Thomas Gruber Ramesh Patil, Ted Senator und William R. Swartout: *Enabling Technology For Knowledge Sharing.* AI Magazine, Volume 12(No. 3):pp. 36–56 (1 p.), 1991. [1]

[OCE06] Ortiz, Magdalena, Diego Calvanese und Thomas Eiter: *Characterizing Data Complexity for Conjunctive Query Answering in Expressive Description Logics.* In: *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006)*, Seiten 275–280, 2006. [10]

[PS08] Prud'hommeaux, Eric und Andy Seaborne: *SPARQL Query Language for RDF, W3C Recommendation 15 January 2008.* Website, `http://www.w3.org/TR/rdf-sparql-query/` zuletzt besucht am 26. Juni 2008, Januar 2008. [40, 64]

[PSHH04] Patel-Schneider, Peter F., Patrick Hayes und Ian Horrocks: *OWL Web Ontology Language, Semantics and Abstract Syntax.* Website, `http://www.w3.org/TR/owl-semantics/` zuletzt besucht am 26. Juni 2008, Februar 2004. [5, 23, 40]

[QL08] Quilitz, Bastian und Ulf Leser: *Query Distributed RDF Data Sources with SPARQL.* In: *Proceedings of the 5th European Semantic Web Conference (ESCW08)*, Seiten 524–538, 2008. [40, 65]

[SP07] Sirin, Evren und Bijan Parsia: *SPARQL-DL: SPARQL Query for OWL-DL.* In: Golbreich, Christine, Aditya Kalyanpur und Bijan Parsia (Herausgeber): *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions*, Band Vol-258, 2007. [40]

[ST04] Serafini, Luciano und Andrei Tamilin: *Local tableaux for reasoning in distributed description logics.* In: *Proceedings of the 2004 Intl. Workshop on Description Logics (DL2004)*, Seiten 100–109, 2004. [32]

[ST05a] Serafini, Luciano und Andrei Tamilin: *Distributed instance retrieval in heterogeneous ontologies.* In: *In Proceedings of SWAP 2005, CEUR Workshop Vol 166*, 2005. [32, 66]

[ST05b] Serafini, Luciano und Andrei Tamilin: *DRAGO: Distributed Reasoning Architecture for the Semantic Web.* The Semantic Web: Research and Applications, Seiten 361–376, 2005. [32, 66]

[WBHQ07] Wang, Yimin, Jie Bao, Peter Haase und Guilin Qi: *Evaluating Formalisms for Modular Ontologies in Distributed Information Systems.* In: Marchiori, Massimo und Jeff Z. Pan (Herausgeber): *Proceedings of The First International Conference on Web Reasoning and Rule Systems (RR2007)*, LNCS 4524, Seiten 178–182, Innsbruck, Austria, JUN 2007. Springer. [1, 23, 29, 30]

[Zol07] Zolin, Evgeny: *Complexity of reasoning in Description Logics.* Website, `http://www.cs.man.ac.uk/~ezolin/dl/` zuletzt besucht am 22. September 2008, Oct 2007. [5]

Abbildungs- und Algorithmenverzeichnis

Abbildungsverzeichnis

2.1 Graphen der Beispielabfragen. 18
2.2 Transformation einer Abfrage nach Algorithmus 2.1 21

5.1 Inspiration und Ansatz für konjunktive Abfragen 44
5.2 Graphen der Beispielabfrage q_2 und q_3. 52
5.3 Einfache $\mathcal{E}$-Connection-Wissensbasis 60
5.4 Graph einer aus q_4' erhaltenen Abfrage. 62
5.5 Verteiltes Reasoning-System 65

Algorithmenverzeichnis

2.1 Beantwortung von Booleschen Abfragen 20
5.1 Konsequenz konjunktiver Abfragen auf $\mathcal{E}$-Connections 61
5.2 Beantwortung konjunktiver Abfragen über $\mathcal{E}$-Connections 63

Abkürzungsverzeichnis

ADL Abstract Description Language

ADM Abstract Description Model

ADS Abstract Description System

DDL Distributed Description Logics

DL Beschreibungslogik

GAV Global-As-View

GLAV Kombination von LAV und GAV

LAV Local-As-View

OMV Ontology Metadata Vocabulary

OWL Web Ontology Language

P2P Peer-To-Peer

P-DL Package-based Description Logics

RDF Resource Description Framework

RDFS RDF-Schema

SPARQL SPARQL Protocol and RDF Query Language

SQL Structured Query Language

W3C World Wide Web Consortium

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tage dem Prüfungsausschuss der Fakultät Informatik eingereichte Diplomarbeit zum Thema:

Untersuchung von konjunktiven Abfragen über verteilte Ontologien mit dem Fokus auf $\mathcal{E}$-Connections.

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 06.03.2009

Jörg Eichhorn