

Goethe Universität Frankfurt am Main
Fachbereich Informatik/Mathematik



Diplomarbeit

Grundlagen und Anwendungen von Wissensrepräsentationsformalismen in der medizinischen Ontologiedatenbank SNOMED CT: Eine Recherche

eingereicht von: Lütfi Tugrul <tugrul@rz.uni-frankfurt.de>

eingereicht am: 10. November 2009

Prüfer: Herr Prof. Schmidt-Schauss

Betreuer: Herr Prof. Schmidt-Schauss & Herr Sabel

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe und nur unter Verwendung der zulässigen Mittel sowie der angegebenen Literatur angefertigt habe.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Frankfurt am Main, den 10.11.2009

Unterschrift

Danksagung

Mein Dank gilt meiner Familie und allen Freunden für moralische und tatkräftige Unterstützung während der letzten Monate.

Ein besonderer Dank auch an Prof. Schmidt-Schauss, der mir die Erarbeitung dieser interessanten Thematik ermöglicht hat.

Mein Dank gilt auch Herrn Sabel für seine Anregungen und Unterstützung.

Zusammenfassung

Trotz intensiver Forschungsarbeiten ist eine automatisierte Verarbeitung von medizinischen Freitexten bis heute aufgrund begrifflicher Unschärfe, Ambiguität und unzähligen Umschreibungsvarianten nur mit großen Einschränkungen möglich.

Demnach werden seit Jahrzehnten medizinische Sachverhalte wie Befunde, Arzneimittel, Prozeduren mit Klassifikationssystemen wie die ICD standardisiert, die die interessierenden Sachverhalte in strengen Klassenhierarchien durch kodierte Schlüssel vorab festlegen und zu einer normierten Dokumentation beisteuern.

Für zukünftige Herausforderungen, wie das automatisierte Prüfen der Arzneimitteltherapiesicherheit als ein anvisiertes Ziel bei der Einführung der elektronischen Gesundheitskarte, erweisen sich die vorhandenen Klassifikationssysteme aufgrund ihrer fehlenden verarbeitbaren Begriffsebene zu starr und ausdruckschwach.

SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms) ist eine standardisierte medizinische Referenzterminologie, die im Gegensatz zu Klassifikationen eine Konzeptualisierung mit semantischer Datenintegration und die Anwendung von wissensbasierten Entscheidungsfunktionen erlaubt. Das formale Fundament der Terminologie wird durch die Beschreibungslogik \mathcal{EL}^+ festgelegt.

Im Vergleich zu ausdrucksstarken Beschreibungslogiken wie \mathcal{AL} bietet diese Variante der \mathcal{EL} -Sprachfamilie eine polynomielle Entscheidbarkeitseigenschaft beim Inferenzieren an und stellt eine effiziente Möglichkeit für ein standardisiertes Begriffssystem dar.

Ziel dieser Diplomarbeit war es, anhand einer Literaturrecherche zu erforschen, inwieweit der aktuelle Forschungs- bzw. Wissensstandpunkt von Wissensrepräsentationsformalismen genutzt wird, um konsistente medizinische Begriffssysteme basierend auf beschreibungslogischen Formalismen entwickeln zu können.

Im ersten Teil der Arbeit werden medizinische Wissensrepräsentationssysteme vorgestellt, die zur Dokumentation und Ordnung von medizinischen Sachverhalten erfolgreich eingesetzt werden, jedoch für das eGK-Projekt im Hinblick auf eine automatisierte Prüfung der Arzneimitteltherapiesicherheit unangemessen sind.

Der zweite Teil dieser Diplomarbeit beinhaltet eine ausführliche theoretische Ausarbeitung der Grundlagen und Anwendungen von Wissensrepräsentationsformalismen mit dem Schwerpunkt Beschreibungslogik \mathcal{EL} .

Der dritte Teil orientiert sich schlussendlich an einer möglichen praktischen Umsetzung der zuvor erarbeiteten Erkenntnisse. Hierbei werden der Grundaufbau der SNOMED CT Terminologie und die beschreibungslogischen Komponenten kollektiv vorgestellt. Anschließend wird eine praktische Anwendungsmöglichkeit präsentiert.

Im Rahmen dieser Diplomarbeit stellen die vorgestellten Themen trotz ihres Umfangs jeweils nur eine komprimierte Übersicht über den aktuellen Stand der Forschung im Wissenrepräsentationsumfeld dar, so dass bei theoretischen Betrachtungen aus Platzgründen öfters Beweisauszüge herangezogen wurden.

Nach wie vor bietet dieser Bereich sehr viele Forschungsfelder an, so dass auch in Zukunft mit vielen neuen und interessanten Forschungsergebnissen zu rechnen ist.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Einleitung und Motivation	1
1 Medizinische Dokumentation	4
1.1 Medizinische Ordnungssysteme	5
1.1.1 Klassifikation und Nomenklatur	6
1.1.2 ICD - Internationales Klassifikationssystem	8
1.1.3 Einsetzbare Ordnungssysteme für eine automatisierte Prüfung der Arzneimitteltherapiesicherheit	9
1.2 Wissensrepräsentation in der Medizin	11
1.2.1 Semantische Netze	12
1.2.2 Frames	15
1.2.3 Problematik bei inkonsistenten Wissensrepräsentationen	19
2 Beschreibungslogik (DL)	21
2.1 Allgemeines	21
2.2 Abstrakte Syntax und Semantik	21
2.2.1 Konzeptbildende Strukturen	25
2.2.2 Relationsbildende Strukturen	29
2.3 Inferenzdienste	31
2.3.1 Tableau-Methodik	35
2.4 \mathcal{AL} -Sprachfamilie	35
2.4.1 Auszug der Komplexitätsergebnisse für Subsumtion und Erfüllbar- keit in der \mathcal{AL} -Sprachfamilie	37
2.5 \mathcal{EL} -Sprachfamilie	38
2.5.1 Subsumtionsalgorithmus und -komplexität in \mathcal{EL}^{++}	39
2.5.2 Polynomielle Subsumtionskomplexitäten für diverse Extensionen in \mathcal{EL}	46
2.5.3 Non-polynomielle Subsumtionskomplexitäten für diverse Extensio- nen in \mathcal{EL}	49
2.6 Unifikation in \mathcal{EL}	53
2.6.1 \mathcal{EL} -Unifikationsproblem vom Typ 0	57
2.6.2 Entscheidungsprozedur für die \mathcal{EL} -Unifikation	59

3	SNOMED CT	64
3.1	Allgemeines	64
3.2	SNOMED CT - Historische Entwicklung	66
3.3	Grundaufbau von SNOMED CT	68
3.3.1	Konzepte und Bezeichnungen	69
3.3.2	Beziehungen und hierarchische Strukturen	70
3.3.3	SNOMED CT - Datenbank Tabellen	75
3.4	SNOMED CT aus beschreibungslogischer Sicht \mathcal{EL}^+	77
3.4.1	SEP-Tripel	80
3.4.2	Gruppierte Relationen	83
3.5	Anwendungsmöglichkeit der SNOMED CT-Terminologie	84
	Fazit und Ausblick	87
	Literaturverzeichnis	96

Abbildungsverzeichnis

1.1	Mächtigkeit bzw. semantische Reichhaltigkeit einzelner Wissensrepräsentationen. Graphische Anlehnung an [17]	12
1.2	Beispiel für ein semantisches Netz der Domäne „Wirbelsäule“	13
1.3	Graphische Darstellung eines Frames. Graphische Anlehnung an [25]	15
1.4	Graphische Darstellung eines generischen Frames und seiner Instanz. Graphische Anlehnung an [27]	16
1.5	Graphische Darstellung der Framestruktur zum Pseudocode	19
2.1	Architektur einer Wissensrepräsentation mittels Beschreibungslogik	33
2.2	Tabellarische Darstellung der zugelassenen Konstruktoren in der Basissprache $\mathcal{A}\mathcal{L}$ und dessen Extensionen. Graphik aus [36]	36
2.3	Graphische Darstellung der zugelassenen Konstruktoren in der Basissprache $\mathcal{A}\mathcal{L}$ und dessen Extensionen. Graphik aus [36]	37
3.1	Graphische Darstellung einer SCTID für das Konzept „Peribronchial pneumonia - 55679008“. Einer Graphik von [66] nachempfunden	69
3.2	Schematischer Aufbau eines Konzepts in SNOMED CT am Beispiel Bronchopneumonia. Einer Graphik von [66] nachempfunden	70
3.3	Schematischer Aufbau der Hierarchie in SNOMED CT am Beispiel Procedure. Einer Graphik von [67] nachempfunden	71
3.4	Beispiel Lungenentzündung: Pneumonia-FINDING SITE-Lung structure. Einer Graphik von [67] nachempfunden	74
3.5	Core tables in SNOMED CT. Einer Graphik von [67] nachempfunden.	75
3.6	Schematischer Aufbau des SEP-Tripels. Einer Graphik von [77] nachempfunden.	80
3.7	SEP-Tripel Ansatz zur Repräsentation anatomischer Konzepte am Beispiel „Amputation“. Einer Graphik von [77] nachempfunden.	81
3.8	SEP-Tripel Ansatz zur Repräsentation anatomischer Konzepte am Beispiel „Darmtrakt“, einer Graphik von [77] nachempfunden.	82
3.9	Software Architektur von SNOMED CT - Eine Implementierungsvariante	84
3.10	Bild- und Befunddokumentationssystem Clinic Win Data - Screenshot [82]	85

Tabellenverzeichnis

1.1	Übersicht diverser Klassifikationen, Nomenklaturen und Thesauri.	9
2.1	Identifikation von Konzepten, Relationen, Konstruktoren und Individuen .	22
2.2	Konzeptbildende Strukturen im Überblick	30
2.3	Relationsbildende Strukturen im Überblick	32
2.4	\mathcal{AL} -Sprachfamilie: Komplexitätsergebnisse für Subsumtion und Erfüllbarkeit	38
2.5	Syntax und Semantik von \mathcal{EL}	39
2.6	Die Extensionen von \mathcal{EL}^{++}	40
2.7	Weitere Extensionen von $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$	45
2.8	$\mathcal{EL}^{\cup, (-)}(\mathcal{D})$: Bedingungen für das Einfügen von (A, B) in S	48
2.9	$\mathcal{EL}^{\geq, \cup}$: Bedingungen für das Einfügen von (A, B) in S	49
2.10	Liste aller Subsumtionskomplexitäten für diverse Extensionsvarianten innerhalb der \mathcal{EL} -Sprachfamilie. Tabelle aus [47]	53
2.11	Nicht-deterministischer Algorithmus zur Entscheidung eines Unifikationsproblems Γ in \mathcal{EL}	61
3.1	Die historische Entwicklung von SNOMED CT [66]	67
3.2	Die 19 Grundkonzepte in SNOMED CT - Deutsche Übersetzung bzw. Erklärung in Klammern.	72
3.3	Beispiel für die Darstellung des Konzepts „Chronic disease (disorder)“ in der Konzepttabelle. Daten aus dem SNOMED Browser [71]	76
3.4	Die Beschreibungslogik \mathcal{EL}^+ in SNOMED CT. Notationshinweise und Beispiele sind in blauer Schriftfarbe.	79

Einleitung und Motivation

Hohe Gesundheitsausgaben und eine schlechte Versorgungsqualität: Die Gesundheitskarte als Antwort?!

Jährlich wird die Bundesrepublik Deutschland mit enorm zunehmenden Gesundheitskosten konfrontiert. Insgesamt geben die deutschen Bürger pro Jahr mehr als 250 Milliarden Euro für Gesundheitsleistungen aus, die zur Hälfte von den gesetzlichen Krankenversicherungen getragen werden, der übrige Teil wird von den privaten Krankenversicherungen oder von den Patienten selbst getragen [1].

Der Zuwachs der Gesundheitskosten übersteigt mit circa 4,6% jährlich deutlich den Zuwachs des Bruttoinlandsprodukts, so dass sich die Ausgaben im Gesundheitswesen seit 1985 mehr als verdoppelt haben [2].

Der Bedarf an Gesundheitsleistungen ist stark vom Umfang und von der Struktur der Bevölkerung geprägt. Es ist selbsterklärend, dass die Ausgaben für Gesundheitsleistungen mit dem Alter zunehmen. So verursacht ein 70-jähriger Patient im Durchschnitt nahezu die fünffachen Kosten eines 20-jährigen Patienten. Vergleicht man dazu die demographische Entwicklung in der Bundesrepublik Deutschland, so sind große Herausforderung für das Gesundheitswesen vorprogrammiert.

Laut statistischen Angaben ist jeder vierte Deutsche über 60 Jahre alt [3]. Die demographische Entwicklung zeigt, dass die Anzahl der Älteren in der Bevölkerung weiter zunehmen wird. Unabhängig von dieser demographischen Entwicklung, wird die Bevölkerung sogar jedes Jahr neuen Krankheiten gegenübergestellt, wie zur Zeit der Schweinegrippe. Die Ärzte werden mit Patientendaten überflutet und müssen trotzdem eine sichere Versorgungsqualität aufrechterhalten. Die optimale Versorgungsqualität kann nur durch zuverlässiges, sicheres und schnelles Austauschen von Informationen zwischen den beteiligten Ärzten erzielt werden. Moderne Informationstechnologien sollen dabei helfen, Daten systematisch zu erfassen, zu verwalten, zu ordnen, zu transportieren, aufzubereiten und zu interpretieren. Sie können den Fortschritt für eine verbesserte Patientenversorgung intensivieren und vorantreiben. Trotz der vorhandenen modernen Technik ist derzeit die Versorgungsqualität mangelhaft.

Aufgrund der hochgradigen Arbeitsteilung bei der Patientenversorgung ist der Kommunikationsbedarf zwischen den beteiligten Personen (Ärzten, Pflegekräften, Therapeuten usw.) von großer Bedeutung. Hinzu kommt sowohl die regionale Mobilität der Patienten und ihre Bereitwilligkeit zum Arztwechsel als auch die dabei entstehenden quantitativen Einzelbefunde, die gemeinsam bewertet werden müssen.

Patienten haben oft das Problem mehrere Ärzte aufsuchen zu müssen, weil ihre Diagnose nicht ausreichend oder sogar falsch war. Jede Praxis generiert ihre internen Patientenak-

ten meist ohne auf die individuelle Vorgeschichte des Patienten einzugehen, obwohl diese Informationen für den momentan aufgesuchten Arzt von großer Bedeutung sein könnten. Aus diesem Grund sind heutzutage mehrfache Diagnosen oder generell falsche Diagnosen nicht ausgeschlossen.

Oft sind Daten nicht interoperabel und können aus einem System nicht in ein anderes System übertragen werden. **Interoperabel** bedeutet, dass zwei heterogene Systeme möglichst reibungslos zusammenarbeiten sollen, damit Daten effizient bearbeitet und verwaltet werden können. Ein wichtiges Kriterium hierfür ist, dass gemeinsame Standards entwickelt und eingehalten werden müssen.

Die gegenwärtige Situation spiegelt eher den Kontrast der erwünschten Interoperabilität von medizinischen Systemen wider: Rezepte werden elektronisch erstellt und anschließend auf Papier ausgedruckt, dringend benötigte Daten stehen gar nicht oder zu spät zur Verfügung. Außerdem kommt noch die Vernachlässigung des Datenschutzes hinzu. Vertrauliche Daten werden häufig per E-Mail oder Fax ohne weiteren Datenschutz ausgetauscht.

Ebenso ist die Versorgungsqualität im Bereich der Arzneimittelsicherheit sehr fragwürdig. Noch immer sterben jedes Jahr mehr Menschen an den Folgen ungewollter Arzneimittelwirkungen als im Straßenverkehr.

Je besser das Krankheitsbild des Patienten protokolliert ist und zweckgemäß zur Verfügung gestellt wird, desto höher sind seine Genesungschancen, unabhängig davon wie viel Ärzte er schon besucht hat oder aufsuchen wird.

Solange es keine einheitlich organisierte und vernetzte, also interoperable Datenverwaltung der Patientendaten gibt, werden auch die Gesundheitskosten im Einklang mit der demographischen Entwicklung in Deutschland verstärkt zunehmen. Die Frage ist nun, wie man eine effiziente Patientenversorgung durch eine konsistente und vollständig vernetzte Dokumentation medizinischer Patientendaten realisieren kann.

Die Antwort liegt in der Einführung der **elektronischen Gesundheitskarte** (*kurz: eGK*), als integrales Element einer telematischen Infrastruktur zur Vernetzung des Gesundheitswesens. Die eGK realisiert die Basis für den Informationsaustausch aller Beteiligten und liefert den behandelnden Ärzten wichtige Informationen für Diagnose und Therapie, z.B. um vermeidbare Interaktionen von Arzneimitteln oder individuelle Unverträglichkeiten aufzuzeigen. Alle persönlichen Informationen wie Krankheitsverläufe, Medikamente, Diagnosen, Allergien usw. werden auf dieser Karte festgehalten. Die Einführung der eGK ist somit gesundheitlich als auch wirtschaftlich rational, weil unnötige Diagnosen und Arzneimitteltherapien verhindert werden können.

Das gilt bspw. für die Einführung des elektronischen Rezepts (e-Rezept). Durch die Einführung des e-Rezepts wird die verwaltungstechnische Abwicklung der Rezepte erleichtert und kann zusätzlich beim Generieren von Rezepten den Arzt auf persönliche Wechsel- und Nebenwirkungen von Arzneimitteln hinweisen. Das System ist somit für jeden Patienten individuell ausgerichtet und stellt einen großen Schritt in Richtung einer sicheren Versorgungsqualität dar.

Durch die Einführung der eGK werden nicht nur die Behandlungs- und Lebensqualität gestärkt, sondern auch die Patientenrechte geschützt. Dem Patienten wird die Möglichkeit gegeben, die gespeicherten Daten jederzeit vollständig auszudrucken und zu lesen,

damit er einen besseren Überblick über seinen eigenen Gesundheitsstatus (bzgl. Allergien, Impfstatus, Blutwerte, Vorsorgeuntersuchungen, Verlauf chronischer Erkrankungen etc.) erfahren und mehr Eigenverantwortung übernehmen kann.

Es ist klar, dass die Arzneimitteldokumentation allein nicht ihren Zweck erfüllen kann, wenn sie nicht in eine entsprechende Software eingebunden ist, die als Betreuungsprogramm für Ärzte fungiert. Diese Software soll mit einer benutzerfreundlichen Oberfläche die Ärzte bei ihrer Diagnose begleiten und unterstützen.

Jedoch ist nicht geklärt, wie die Patientendaten auf der eGK verwaltet und bereitgestellt werden, damit man aus all den Patienteninformationen medizinisches Wissen schlussfolgern und verarbeiten kann, so dass eine automatisierte Prüfung der Arzneimittel- und Therapiesicherheit mit einer geeigneten Software gewährleistet wird.

Zu den Möglichkeiten und Grenzen der Arzneimitteldokumentation sei auf folgenden Artikel verwiesen [4].

Ein erdenkliches Beispiel aus dem Alltag:

Eine Patientin nimmt regelmäßig die Anti-Baby-Pille ein, die sie vom Frauenarzt verschrieben bekommen hat. Sie erkrankt an einer bakteriellen Infektion. Ihr Hausarzt verschreibt ihr ein Antibiotikum. Diese medikamentöse Konstellation führt dazu, dass durch die Einnahme des Antibiotikums die Wirkung der Anti-Baby-Pille abgeschwächt oder vollständig unterbunden wird.

Um solche und ähnliche Ereignisse verhindern zu können, sollte ein intelligentes System vorhanden sein, das anhand der persönlichen eGK-Daten prüft und inferenziert, dass die Therapiesicherheit unter diesen Umständen gefährdet ist.

Diese automatisierte Prüfung der Arzneimitteltherapiesicherheit soll medizinische Entscheidungen effizient unterstützen und die Versorgungsqualität verbessern. Jedoch ist bei einer automatisierten Prüfung die maschinelle Verarbeitung von medizinischen Freitexten inbegriffen. Dazu gehören: Arztbriefe, Diagnosen, Rezepte, Prozeduren etc., die bis heute aufgrund von Mehrdeutigkeiten und begrifflicher Unschärfe nur mit großen Einschränkungen maschinell verarbeitet werden können.

Seit Jahrzehnten werden deshalb für solche Angelegenheiten **medizinische Dokumentationssysteme** zur Standardisierung verwendet.

Für konkrete Implementierungs- und Anwendungsziele der eGK siehe [5].

Ein geeignetes medizinisches Dokumentationssystem ist die Grundlage für eine problemfreie Datenverwaltung, die für eine automatisierte Prüfung der Arzneimittel- und Therapiesicherheit eingesetzt werden kann.

1 Medizinische Dokumentation

Die Tätigkeiten des Sammelns, Erschließens, Ordnen und Aufbewahrens von Information oder Wissen bezeichnet man als **Dokumentation**. Das Ziel der Dokumentation ist es, die Informationen oder das Wissen zu einem späteren Zeitpunkt nutzbar zu machen. Denn durch die inhaltliche Erschließung wird das in den Dokumenten enthaltene Wissen zugänglich und kann für weitere medizinische Zwecke von großer Wichtigkeit sein.

An einem durchschnittlichen Universitätsklinikum werden ca. 6 Millionen einzelne Dokumente pro Jahr verfasst. Dies entspricht etwa einer 1,5 Kilometer langen Papierrolle an Patientenakten [6].

Die dokumentierte Patientenakte gibt Auskunft über Feststellungen zum Behandlungsverlauf, die zu einem früheren Zeitpunkt, von anderen Personen oder an anderer Stelle getroffen wurden. Dieser Dokumentationsaufwand ist zum Teil rechtlich vorgeschrieben, damit eventuelle Rechtsstreitigkeiten anhand der Dokumente geklärt werden und als Nachweis dienen können. Dokumentiert wird nicht nur für die Verwaltung des Leistungs- und Kostenplans, sondern auch für die medizinische Forschung, damit neue Technologien entwickelt werden und die medizinische Versorgungsqualität gesichert wird.

Ein **Dokumentationssystem** hat die Aufgabe mit konventionellen informationsverarbeitenden Werkzeugen Dokumentationsaufgaben zu verwirklichen. Eine der größten Fehler mit dem Umgang der dokumentierten Daten ist die Tatsache, dass falsche Erkenntnisse aus den Dokumenten gewonnen werden, die auf unangemessene Dokumentationsmethoden zurückzuführen sind.

Wer dokumentierte Daten wissenschaftlich auswertet, ohne sich Gedanken über angemessene Dokumentationsmethoden gemacht zu haben, der riskiert eine substanzielle Fehlinterpretation der inferenzierten Resultate.

Beispielsweise wird unnötiger Aufwand betrieben, wenn dieselben Daten für mehrere Aufgaben mehrmals aufgezeichnet werden, anstatt die einmal festgehaltene Information multipel zu verwenden.

Um eine fehlerfreie Dokumentationsmethode realisieren zu können, ist folgender Grundwortschatz von existentieller Bedeutung:

- Ein **Objekt** entspricht einem Ausschnitt aus der wahrnehmbaren oder vorstellbaren Welt. Jedes Objekt weist eine Menge von Eigenschaften auf, durch die es sich von anderen Objekten differenzieren lässt. Gleichartige Objekte können zu einer Denkeinheit zusammengefasst und als **Begriff** oder **Objekttyp** bezeichnet werden. Ein Objekt wird durch seine **Merkmale** bzw. **Attribute** repräsentiert.
- Eine **Definition** ist kennzeichnend für die Bedeutung eines Begriffs und wird durch sprachliche oder formale Mittel beschrieben.

- Eine **Terminologie** ist der Gesamtbestand der Begriffe, die durch ihre Definitionen repräsentiert werden.
- **Information** ist die Kenntnis über bestimmte Sachverhalte oder Vorgänge.
- **Daten** sind Zeichenketten, die unter einheitlichen Abmachungen Information darstellen sollen.
- **Wissen** ist Information im weiteren Sinne und unterscheidet sich grundsätzlich von Daten. Daten können erst durch die semantische Betrachtung hinsichtlich einer Terminologie Wissen verkörpern.

Bsp.: Die Zeichenkette E10 stellt eine Datei dar und wird durch die semantische Betrachtung hinsichtlich der Klassifikation nach ICD-10 (siehe Kap.1.1.2) mit der Information „primär insulinabhängiger Diabetes mellitus“ eindeutig identifiziert.

Damit in einem Dokumentationssystem bestimmte Inhalte (wie Krankheiten, anatomische Strukturen, etc.) gezielt wiedergefunden werden können, sollte jede Dokumentation mit einer fundierten Ordnung ausgezeichnet sein, die durch eine Dokumentationssprache realisiert wird.

Die Dokumentationssprache besteht aus einer Menge von Deskriptoren (bzw. Schlagwörtern), die zur inhaltlichen Beschreibung eines Objekts ausgewählt wird und aus den spezifizierten Regeln, die für die Anwendung genutzt werden. Ihre Aufgabe besteht darin, dass der relevante Ausschnitt der medizinischen Inhalte abgedeckt werden kann. Dies wird nur erreicht, wenn die Deskriptoren einer systematischen Ordnung unterliegen, sprich einem **Begriffssystem**. Die Dokumentationssprache bildet somit ein **Ordnungssystem** und kann im Hinblick auf die Realisierung der eGK zur Patientendokumentation von Diagnosen und Therapien verwendet werden.

Da ein patientenzentriertes Ordnungssystem alle Informationen über den Gesundheitsstatus des Patienten, über alle beteiligten Versorgungseinrichtungen hinweg, zusammenführen und einordnen kann, stellt sie die Grundlage für eine problemfreie Verwaltung der Daten auf der eGK dar.

Medizinische Ordnungssysteme sollen zukünftig nicht nur die Dokumentationsaufgaben erleichtern, sondern auch für automatisierte Prüfungen der Arzneimittel- und Therapie-sicherheit Anwendung finden.

1.1 Medizinische Ordnungssysteme

Ein medizinisches Ordnungssystem dient zur Erleichterung und Standardisierung der Dokumentation. Sie ist in höchstem Maße benutzerfreundlich, wenn jede zugriffsberechtigte Person beim Dokumentieren von Diagnosen oder Therapien weitestgehend nicht eingeschränkt wird. Denn die Aufzeichnung medizinischer Information ist sehr komplex und erlaubt große Freiheiten. Dazu gehört die Auswahl aus mehreren synonymen Begrifflichkeiten, die Schreibweise sowie die Ausführlichkeit und die Konstruktion der Gesamtausage.

Damit Ärzte oder andere zugriffsberechtigte Personen beim Generieren von Dokumenten aus der Vielfältigkeit eines Ordnungssystems profitieren können, muss das Ordnungssystem auch über dementsprechende Methoden verfügen, die diese Strukturen fehlerfrei verarbeiten und einordnen.

Da die maschinelle Verarbeitung medizinischer Freitexte sehr viel Aufwand und Unsicherheit in der Implementierung mit sich bringt, wurden starre medizinische Ordnungssysteme eingeführt, die die Freiheiten von medizinischen Ausdrücken relativ stark einschränkten. Diese Einschränkung beruht auf der Tatsache, dass Aussagen nur aus Deskriptoren bestehen, die nur nach einfachen Prinzipien kombiniert werden können.

Meistens wird zur Vereinfachung statt des Deskriptors auch nur eine verkürzende Notation oder ein Schlüssel aufgezeichnet. Die Dokumentationssprache kann die Schlüssel verarbeiten, referenzieren und eine Dokumentation realisieren. Sie legt die Menge der zugelassenen Deskriptoren fest und bestimmt den hierarchischen Aufbau eines Ordnungssystems. Denn die Menge an Deskriptoren ist meistens sehr groß und wäre in Form einer Aufzählung unüberschaubar. Aus diesem Grund werden in einem medizinischen Ordnungssystem die von Deskriptoren bezeichneten Begriffe generell in eine hierarchische Ordnung gebracht, sprich die Dokumentationsprache wird mit einem Begriffssystem unterlegt, so dass ein Ordnungssystem resultiert.

Beispiel: In einem Ordnungssystem können operative Prozeduren in diverse Operationen, wie an Organen, Blutgefäßen, am Nervensystem usw. eingeteilt werden, indem ein dafür vorgesehener Deskriptor - wie **Operationen** im Ordnungssystem - vorhanden ist, der die einzelnen Varianten einer Operation auflistet.

Jedoch sind nicht alle Ordnungssysteme schlau genug, um auch Hinweise auf nicht zugelassene Benennungen aufmerksam zu machen. Es gibt Fälle, in denen man nach einem Deskriptor bspw. **Bandscheibenvorfall** sucht, der nicht im Ordnungssystem aufgeführt ist, jedoch einen synonymen Deskriptor bspw. **Discusprolaps** zur Verfügung stellt.

Deshalb wäre es von großem Vorteil, dass solche synonymen Begrifflichkeiten in einem Ordnungssystem automatisch ineinander referenziert werden. Diese erweiterte Variante eines Ordnungssystem bezeichnet man als **Thesaurus**.

Mehr zu Thesauri siehe [7].

Der nächste Abschnitt stellt zwei bekannte Ordnungssysteme vor.

1.1.1 Klassifikation und Nomenklatur

Klassifikation und Nomenklatur sind zwei mögliche Varianten eines Ordnungssystem, die man bei der Planung einer Dokumentation zu entscheiden hat. Je nachdem welchen Nutzen und Aufgabe das jeweilige Ordnungssystem hat, können beide Ordnungssysteme in Erwägung gezogen werden.

Klassifikationen, auch Klassifikationssysteme genannt, sind Ordnungssysteme, die auf dem Prinzip der Klassenbildung beruhen. Begriffe, die in mindestens einem klassenbildenden Merkmal übereinstimmen, werden in einer Klasse zusammengefasst.

Das wichtige Kriterium zum Klassifizieren ist, dass die Klassen einer Klassifikation das dokumentierte Gebiet vollständig abdecken müssen und ihre Inhalte sich nicht überschneiden sollten, damit jeder Begriff genau einer Klasse zugeordnet wird.

Jede Klasse wird mit einem Schlüssel versehen, welches einem strikten Muster unterliegt. Anhand der Notation wird die Bildung eines hierarchischen Systems realisiert, so dass zur Dokumentation eines Gegenstandes die Ermittlung und Aufzeichnung der zutreffenden Notation genügt. Diese Tätigkeit heißt notieren, verschlüsseln oder auch kodieren.

Hierarchische Klassifikationen kennzeichnen sich durch ihre generischen oder Teil-Ganzes-Beziehungen aus, d.h. die untergeordnete Klasse ist entweder eine Instanz oder ein Teil der übergeordneten Klasse. Wenn jede Klasse oder jeder Begriff höchstens eine übergeordnete Klasse (außer der obersten Klasse oder Wurzel der Hierarchie) hat, dann bezeichnet man diese Form der Ordnung als **Monohierarchie**.

Andernfalls kann eine Klasse mehr als eine übergeordnete Klasse aufweisen, was zu mehreren, aufeinandergelegten Hierarchien führt. Diese Form der Ordnung ist als **Polyhierarchie** bekannt.

Es ist auch möglich, innerhalb einer Klassifikation unabhängige Teilklassifikationen zu konstruieren, die achsenförmig bzw. facettenartig innerhalb der Klassifikation gegliedert sind. Jede Achse hat eine Menge von klassenbildenden Merkmalen und bildet genau ein semantisches Bezugssystem ab. Die Teilklassifikationen können auch in sich wiederum hierarchisch strukturiert sein, so dass die Aufteilung der gesamten Klassifikation eine detailliertere Hierarchisierung bewirkt.

Im Unterschied zu Klassifikationen stellen **Nomenklaturen** eine systematische Zusammenstellung von Deskriptoren dar, die für eine Dokumentationsaufgabe zugelassen sind. Eine Nomenklatur ist ein Spezialfall einer Terminologie, in der die Objekte aus einem bestimmten Wissens- und Themengebiet durch Richtlinien festgelegt sind, bspw. Medizinische Nomenklatur oder Nomenklatur chemischer Reaktionen etc. [8].

Die Nomenklatur dient nicht allein dem Einordnen, sondern vielmehr dem eindeutigen und möglichst genauen Beschreiben von Datenobjekten mit dem Ziel, sie später zuverlässig wiederzufinden und die Information weiterverarbeiten zu können. Die Nomenklatur nimmt die Form eines Thesaurus an, wenn die Deskriptoren durch Definitionen, Synonyme und andere terminologische Hinweise ergänzt sind.

Wie bei der Klassifikation werden zur vereinfachten Aufzeichnung die Deskriptoren einer Nomenklatur mit einem Schlüssel versehen. Jedoch können, im Gegensatz zur Klassifikation, die Begriffsinhalte der Deskriptoren sich überschneiden.

Die Nomenklatur hat die Aufgabe, Gegenstände mit allen zutreffenden Deskriptoren zu kennzeichnen, sprich der Gegenstand wird indexiert. Damit eine zuverlässige Suche realisiert werden kann, ist es wichtig, dass die zutreffenden Deskriptoren vollständig ermittelt werden.

Genau wie Klassifikationen können auch Nomenklaturen zur einfacheren Orientierung hierarchische Strukturen aufweisen und Deskriptoren in mehrere unabhängige Teilmengen aufteilen, damit verschiedene semantische Verhältnisse in mehreren Achse realisiert werden können. Jedoch erhöht die Mehrachsigkeit, wie bei einer Klassifikation, nicht die Ausdrucksmöglichkeiten, sondern erleichtert lediglich die Handhabung.

Beim Indexieren eines Begriffs werden die Achsen der Nomenklatur der Reihe nach auf zutreffende Deskriptoren geprüft.

Beispiel für eine Nomenklatur:

S1Kopf

S2Gelenke

S3Extremitaeten

S4Ruecken

Diese Deskriptoren könnten bspw. zur Schmerzlokalisierung eingesetzt werden, indem bei hierarchischem Aufbau auch folgende Unterteilung möglich wäre:

S1Kopf

S11Stirn

S12Kiefer

S13Schaedel

S14Schlaefe

Durch die Hinzunahme einer weiteren Achse, kann eine zweiachsige Nomenklatur zur Schmerzlokalisierung und -art realisiert werden:

A1ziehend

A2stechend

A3brennend

A4drueckend

Ein drückender, brennender Schmerz am Kiefer wäre als (*S12, A4, A3*) zu notieren.

Im Hinblick auf die Realisierung einer automatisierten Entscheidungsunterstützung anhand der eGK, stellen fundierte Ordnungssysteme die Basis für eine systematische Dokumentation dar und können durch gezieltes Einsetzen die erwünschte Arzneimittelsicherheit gewährleisten. Die bis heute international akzeptierten und einsatzfähigen Klassifikationen fassen Objekte wie Krankheiten, Arzneimittel oder Prozeduren in Klassen zusammen und ordnen diese meist in einer strengen Monohierarchie vom Allgemeinen zum Speziellen, siehe Tab.1.1.

Zurzeit ist die ICD-Klassifikation die weltweit gültige und anerkannte Methode zur Verschlüsselung von Diagnosen und wird für das eGK-Projekt eine wichtige Position übernehmen, weil hauptsächlich Krankheitsverläufe der Patienten dokumentiert werden.

Da eine ausführliche Ausarbeitung der einzelnen Ordnungssysteme den Rahmen der Arbeit sprengen würde, sind die entsprechenden Literaturverweise für zusätzliche Informationen vorgesehen.

1.1.2 ICD - Internationales Klassifikationssystem

Die ICD (*engl.*: International Statistical Classification of Diseases and Related Health Problems) ist ein Klassifikationssystem von Krankheiten und ermöglicht eine weltweite, sprachenunabhängige und exakte Suche nach Krankheiten über einen international gültigen normierten Kode. Durch die Standardisierung von Diagnoseverschlüsselungen, können alle Krankheiten klassifiziert und kodiert werden.

Die WHO verwendet die ICD für die weltweite Morbiditäts- und Mortalitätsstatistik. Nach den §295 und 301 SGB V sind in Deutschland Ärzte und Krankenhäuser zur Dia-

<i>ICD</i>	International Statistical Classification of Diseases and Related Health Problems - dient der Verschlüsselung von Diagnosen.
<i>ICF</i>	Internationale Klassifikation der Funktionsfähigkeit, Behinderung und Gesundheit der World Health Organization (<i>kurz:WHO</i>) - beschreibt gemeinsam mit der ICD den Gesundheitszustand [9].
<i>UMDNS</i>	Universal Medical Device Nomenclature System - dient der Verschlüsselung von Medizinprodukten [10].
<i>MeSH</i>	Medical Subject Headings - Thesaurus von Schlagwörtern zur Katalogisierung von Buch- und Medienbeständen und zur Indexierung von Datenbankbeständen [11].
<i>LOINC</i>	Logical Observation Identifiers Names and Codes - dient der Verschlüsselung im Laborbereich [12].
<i>ATC/DDD</i>	Anatomisch-Therapeutisch-Chemische Klassifikation - gruppiert Wirkstoffe nach Wirkort und Eigenschaften. DDD ist die definierte Tagesdosis [13].

Tabelle 1.1: Übersicht diverser Klassifikationen, Nomenklaturen und Thesauri.

gnosenverschlüsselung nach ICD verpflichtet. Die 10.Revision ist als ICD-10 bekannt und ist eine einachsige, monohierarchische Klassifikation. Die Notation ist alphanumerisch. An der ersten Stelle steht ein Buchstabe, an der zweiten bis fünften Stelle eine Ziffer, wobei die vierte und fünfte durch einen Punkt von den ersten drei Stellen abgetrennt werden.

Beispiel: *Paranoide Schizophrenie* ist eine psychische Störung, die mit dem Buchstaben F kodiert wird. Die Zahl 20 nach dem F steht für Schizophrenie und die darauf folgende Null beschreibt die paranoide Eigenschaft der psychologischen Störung, so dass man der paranoiden Schizophrenie den Schlüssel F20.0 zuordnet.

Die ICD-10 gliedert sich hierarchisch in 21 Krankheitskapitel, 261 Krankheitsgruppen mit insgesamt mehr als 12000 Krankheitsklassen. Ausführliche Bezeichnungen zu einzelnen Krankheitskapiteln und weitere detaillierte Informationen können aus [14] entnommen werden.

1.1.3 Einsetzbare Ordnungssysteme für eine automatisierte Prüfung der Arzneimitteltherapiesicherheit

Medizinische Ordnungssysteme ermöglichen eine übersichtliche und strukturierte Dokumentation, aber können sie für ambitionierte Projekte wie die eGK auch eine automatisierte Prüfung der Arzneimitteltherapiesicherheit realisieren und somit medizinische Entscheidungen unterstützen?

Für diese Herausforderungen sind Klassifikationssysteme wie ICD allein zu starr und grob, weil die Krankheiten nicht nur klassifikatorisch geordnet, sondern auch semantisch

miteinander verknüpft werden sollen, damit eine nicht ausdrücklich vorhandene Information aus implizit vorhandenen Informationen innerhalb des Ordnungssystems abgeleitet werden kann.

Klassifikationen können nur begrenzt erweitert werden und repräsentieren die interessierenden Begriffe in meist unsystematischer Art und Weise. Die Tatsache, dass mit einem ICD-Schlüssel bspw. T88.7 (*Nicht näher bezeichnete unerwünschte Nebenwirkung eines Arzneimittels oder einer Droge*) unterschiedliche Sachverhalte impliziert werden können, erweist sich für eine automatisierte Auswertung nicht als angemessen.

Bakterielle Infektionen eines Erregers werden mit verschiedenen Schlüsseln kodiert, die je nach anatomischer Manifestation und Erkrankungsart über die gesamte Klassifikation ausgebreitet sind, bspw. M00 (*eitrige Arthritis*), G00 (*Meningitis*), A04 (*Darminfektion*) etc. Diese Verstreuung der Begriffe wirkt sich für eine automatisierte Auswertung ebenso negativ aus.

Der größte Anteil der Deskriptoren orientiert sich nach den anatomischen Strukturen und den entsprechenden Krankheitsbeschreibungen. Der Hierarchie des Ordnungssystems liegt kein einheitliches semantisches Bezugssystem zugrunde. Aus diesem Grund sind semantische Referenzierungen innerhalb des Ordnungssystems nicht möglich.

Um eine automatisierte Prüfung der Arzneimitteltherapiesicherheit fehlerfrei realisieren zu können, ist der Einsatz von Klassifikationen allein aufgrund der genannten Ausdrucksschwäche nicht vernünftig und nicht erfolgversprechend.

Arzneimittel-Unverträglichkeitsprüfungen, im Zusammenhang mit der eGK, sind auf ausdrucksstarke Terminologien angewiesen. Denn im Gegensatz zu Klassifikationen erlauben Nomenklaturen bzw. Terminologien eine präzise Abbildung der Begriffe, damit Diagnosen möglich vollständig und detailliert beschrieben werden können.

Um Begriffe innerhalb einer Nomenklatur sinnvoll mit Relationen verknüpfen zu können, werden bei kompositionellen Ansätzen logische Relationen und Vererbungsmechanismen herangezogen, die aus der Theorie der formalen Logiken und der Objektorientierung bekannt sind.

Ziel ist es, anhand logischer Verknüpfungen den Begriffen einen semantischen Ausdruck zu verleihen, damit sie innerhalb des Systems als „Wissensbausteine“ aufgefasst werden können. Denn die massenhafte Verknüpfung von einzelnen Begriffen löst eine netzartige Struktur aus, die anhand von Regeln spezifiziert werden muss, damit im System keine Widersprüche auftreten und die Konsistenz des Systems nicht verletzt wird. Somit ist für das systematische Verknüpfen von Wissen die Einführung eines geeigneten Formalismus notwendig, damit aus dem vorhandenen Wissensbereich weiteres Wissen ermittelt und das System auch beliebig erweitert werden kann.

In der Informatik entspricht ein derartiger Formalisierungsansatz einer sogenannten **Ontologie** im Bereich der Wissensrepräsentation.

Der Begriff Ontologie kommt ursprünglich aus der Philosophie und tritt in der Informatik vor allem im Bereich der Wissensrepräsentation als formal definiertes System von Konzepten und zugehörigen Relationen auf. Dementsprechend bezeichnet Gruber eine Ontologie als eine „*formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung (Begriffsbildung)*“ [15].

Im Endeffekt stellen zwar alle vorgestellten Dokumentationssysteme jeweils Ontologieva-

rianten dar, jedoch reicht die semantische Reichhaltigkeit einiger Systeme nicht aus, um automatisierte Prüfungen vorzunehmen.

Letzendlich kommen nur die Ordnungssysteme für automatisiertes Prüfen der Arzneimitteltherapiesicherheit in Frage, die auch den vollständigen Charakter einer Ontologie haben, also Terminologien, die durch ihre hohe semantische Reichhaltigkeit gekennzeichnet sind und durch bestimmte Begriffsbildungs- und Relationsregeln gesteuert werden.

Der nächste Abschnitt ist eine kurze Einführung in die Wissensrepräsentation und stellt diverse Ontologien mit zunehmender Reichhaltigkeit vor.

In den darauffolgenden Abschnitten wird ebenso deutlich, warum nicht jede mögliche Ontologie für automatisches Schlussfolgern eingesetzt werden kann.

1.2 Wissensrepräsentation in der Medizin

Ein wichtiges Forschungsgebiet zur künstlichen Intelligenz (KI) ist die Wissensrepräsentation (KR) (*engl.*: knowledge representation). Die KR beschäftigt sich mit der Art und Weise von Formalismen mit dem Ziel, das Wissen aus einer bestimmten Domäne so in einer Form darzustellen, dass logische Schlussfolgerungen über das Wissen möglich sind. „*Entwicklung von Formalismen, mittels derer Wissen über die Welt in abstrakter Weise beschrieben werden kann und die effektiv verwendet werden können, um intelligente Anwendungen zu realisieren.*“ [16].

Semantische Modelle zur Wissensrepräsentation erlauben eine formale und interpretierbare Repräsentation von Wissen für Menschen und Maschinen. Da die Komplexität der Anwendung eines Systems auf die entsprechende Modellierung von Formalismen zurückzuführen ist, sind gut modellierte Formalismen das "A" und "O" einer KR. Aus diesem Grund gab es im Forschungsgebiet der KR über die Jahre mehrere tiefgreifende Studien über effiziente Modellierung und Implementierung von Formalismen.

Die folgende Abb.1.1 vermittelt in aufsteigender Reihenfolge die Mächtigkeit bzw. semantische Reichhaltigkeit einzelner Wissensrepräsentationen, die in den verschiedensten Wissenschaftsdisziplinen eingesetzt werden.

Je reichhaltiger eine Wissensrepräsentation wird, desto komplexer können die entsprechenden Modelle werden.

Ein **Glossar** ist eine Liste von Begriffen mit beiliegender Erklärung. Relationen zu anderen Begriffen sind nicht vorhanden [18].

Eine **Taxonomie** ist eine Hierarchie von Begriffen, die in Über- und Unterordnung dargestellt werden. Außer der hierarchischen Struktur lassen sich keine Beziehungen zwischen Elementen definieren. Sie werden auch als Klassifikationen bezeichnet, die eine monohierarchische Struktur aufweisen [19].

Ein **Thesaurus** stellt eine erweiterte Form einer Taxonomie dar, die zusätzlich über zwei fest definierte Relationen verfügt: die Ähnlichkeits- und Synonymrelation [20].

Im Bereich der Wissensrepräsentation wird ein formal definiertes System von Begriffen und Relationen als **Ontologie** bezeichnet.

Hierbei soll festgehalten werden, dass jede KR in der dargestellten hierarchischen Rei-

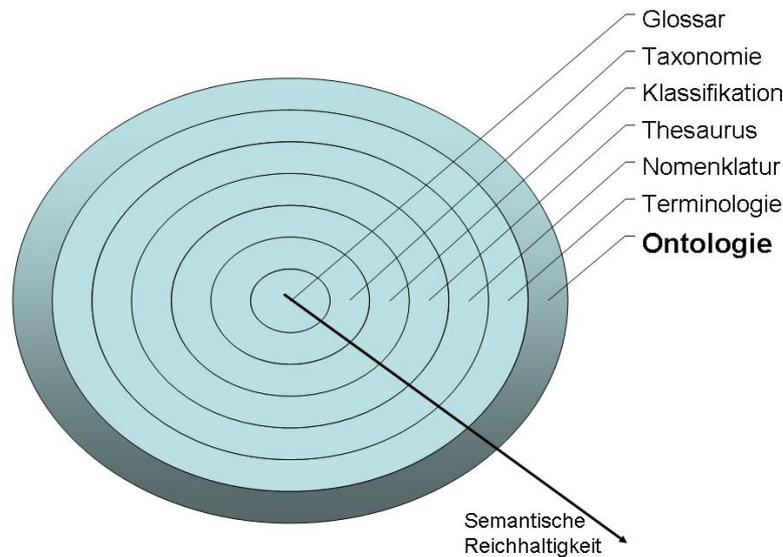


Abbildung 1.1: Mächtigkeit bzw. semantische Reichhaltigkeit einzelner Wissensrepräsentationen. Graphische Anlehnung an [17]

henfolge die unterhalb liegenden Wissensrepräsentationen beinhaltet. Aus diesem Grund stellt die Ontologie ein Oberbegriff für alle genannten Systeme dar.

Bsp.: Jede Taxonomie ist ein Thesaurus, wobei nicht jeder Thesaurus eine Taxonomie ist.

Das Ziel einer Ontologie ist ein Wissensbereich, die sogenannte Domäne, formal zu beschreiben, um ein einheitliches semantisches Verständnis zu schaffen. Um Fehlinterpretationen vorzubeugen, ist eine eindeutige Beschreibung der Domäne von großer Wichtigkeit. Aus diesem Grund sollte jede Ontologie über Inferenz- und Integritätsregeln verfügen, um Schlussfolgerungen zu regeln und ihre Gültigkeit zu gewährleisten [21].

Zum Verständnis kann eine Ontologie mit einem UML-Klassen-Diagramm verglichen werden, mit dem Unterschied, dass keine Softwareklassen verarbeitet, sondern einzelne Begriffe modelliert werden. Es gibt verschiedene Arten von Ontologien, die je nach Einsatzzweck auch unterschiedlich strukturiert und zunehmend komplex sein können.

Im Folgenden werden zwei wichtige Ontologievarianten vorgestellt, die aufgrund ihrer hohen semantischen Reichhaltigkeit zwar sehr bekannt sind, jedoch keine beweisbare Konsistenz aufzeigen, die für automatisiertes Prüfen von existentieller Bedeutung ist.

1.2.1 Semantische Netze

Es fing in den 1960ern mit der Idee der semantischen Netzen an, die der Sprachwissenschaftler Ross Quillian als mögliche Repräsentationsform von semantischem Wissen

eingeführt hat. Sein ursprüngliches Ziel war es, die Bedeutung von Wörtern in verschiedenen Satz- und Verwendungskontexten zu analysieren. Aus diesem Grund entwarf er einen gerichteten Graphen als ein formales Modell von Begriffen und ihren Relationen, so dass die Bedeutung von sprachlichen Ausdrücken in einer formalen Struktur wiedergegeben werden konnte.

Dies ermöglichte eine Modellierung und zugleich eine Visualisierung von Wissensstrukturen. Diese Art von Modell zur Wissensrepräsentation bezeichnet man in der KI als semantisches Netz.

„Ein semantisches Netz (SN) ist das mathematische Modell einer Menge von begrifflichen Entitäten und der zwischen diesen bestehenden kognitiven Beziehungen. Es wird in Form eines verallgemeinerten Graphen dargestellt.“ [22].

Erste rechnergestützte Implementierungen entstanden, anfangs vor allem mit dem Ziel, die natürliche Sprache zu verstehen und zu verarbeiten. Heute sind die Anwendungen semantischer Netze breit gefächert.

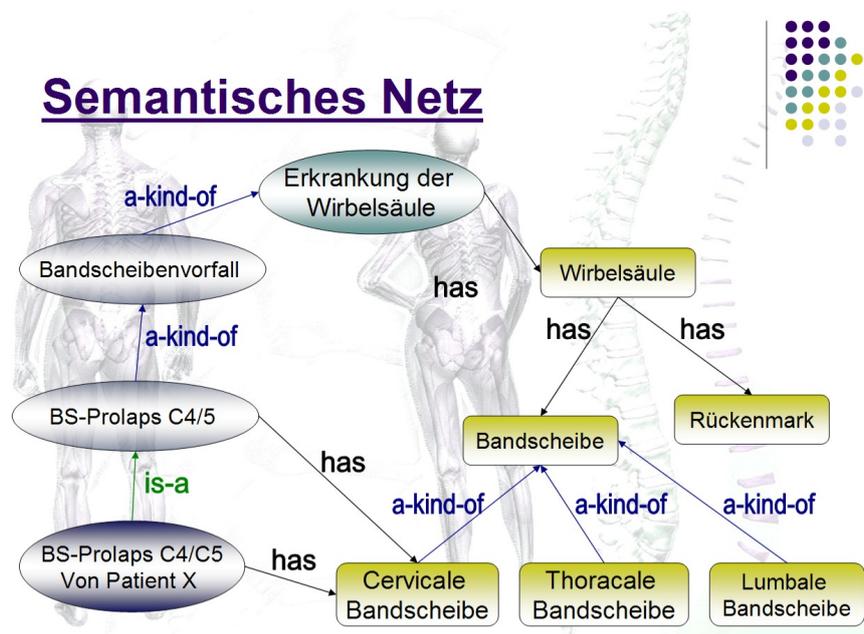


Abbildung 1.2: Beispiel für ein semantisches Netz der Domäne „Wirbelsäule“

Die Begriffe, auch Objekte oder Konzepte genannt, werden in einem semantischen Netz durch die Knoten des Graphen dargestellt, wobei die Beziehungen (Relationen) zwischen den Begriffen durch die gerichteten Kanten des Graphen realisiert werden.

Ein Knotenpunkt kann ein physischer Gegenstand, als auch ein gedankliches Element sein, welches Handlungen und Ereignisse beschreibt.

Durch gezielte Verbindungsrelationen zwischen mehreren Begriffen, ist es möglich, Schlüsse über diese Begriffe zu ziehen. Die Knotenpunkte schließen sich in ihren Relationen zu einem Netz zusammen, das eine regelbasierte Darstellung von Verknüpfungen erlaubt.

Folgende Assoziationen sind zulässig:¹

is-a: Diese Art von Relation wird als Element-Relation bezeichnet. Man benutzt sie, um eine Instanz der Klasse zu repräsentieren, die dann als individuelles Mitglied der Klasse fungiert. Der spezifischere Begriff bezeichnet quasi ein Exemplar bzw. Element aus dem allgemeineren Konzept.

Aus Abb.1.2 gilt:

Ein Bandscheibenvorfall C4/C5 vom Patienten X ist eine Instanz der Klasse Bandscheibenvorfall C4/C5.

is-a (BS-Prolaps C4/C5 von Patient X, BS-Prolaps C4/C5)

has: Diese Relation ist auch als Teil-Ganzes-Beziehung bekannt. Das spezifischere Konzept ist ein (Bestand-)Teil des allgemeineren Konzepts. Mithilfe von Has-Relationen ist es möglich, Untermengenbeziehungen zu definieren, weil die Knoten als Merkmale anderer Knoten bezeichnet werden.

Aus Abb.1.2 gilt:

Die Bandscheibe ist ein Teil der Wirbelsäule.

has (Wirbelsäule, Bandscheibe)

ako: Teilmengenbeziehungen werden durch a-kind-of-Relationen (*kurz:* ako) dargestellt. Jedes Element aus der Menge von spezifischen Begriffen, ist auch Element der Menge von allgemeinen Begriffen. Aus Abb.1.2 gilt:

Die cervicale Bandscheibe ist ein spezifischeres Konzept als die Bandscheibe, weil jede cervicale Bandscheibe eine Bandscheibe ist.

Diese Relation ermöglicht die Vererbung von Eigenschaften und das Aufbauen von Klassenhierarchien.

a-kind-of (cervicale Bandscheibe, Bandscheibe)

Beachte: Je nach Definition werden auch is-a-Relationen für Teilmengenbeziehungen verwendet! Aus [23] und [24], [26] ist klar zu erkennen, dass Autoren in der Namensgebung der Relationsformen nicht einig sind.

Man verwendet diese verschiedenen Arten von Relationen nicht nur um Begriffe und Sachverhalte formal miteinander verknüpfen zu können, sondern auch um sinnvolle Ergebnisse durch logisches Schließen zu erzielen. In der Lehre der Sprachwissenschaft und Logik wird dieser Vorgang des Schlussfolgerns als **Inferenz** bezeichnet.

Das Ziel der Inferenz ist, eine nicht ausdrücklich vorhandene Information aus implizit vorhandenen Informationen innerhalb des formalen Systems abzuleiten.

Beispielsweise lässt sich aus dem semantischen Netz aus Abb.1.2 mit Hilfe von Vererbung folgende Information schlussfolgern:

Damit überhaupt ein Patient an BS-Prolaps leiden kann, benötigt er grundsätzlich eine Wirbelsäule. Mithilfe der Inferenz lässt sich schließen, dass der Patient X, der an einem Bandscheibenvorfall C4/C5 leidet, eine Wirbelsäule besitzt. Wenn man nun das Netz

¹Je nach Tauglichkeit der Relationsarten, ist es möglich, eigene Assoziationen in einer spezifischen Domäne zu entwerfen und diese im semantischen Netz zu benutzen. Bsp.: operiert(Arzt, Patient)

erweitern würde, dass Patient X ein Mensch ist, könnte man schlussfolgern, dass jeder Mensch, der an BS-Prolaps C4/C5 leidet, eine Wirbelsäule besitzt. In Kap.2.3 werden verschiedene Arten von Inferenzdiensten ausgiebig vorgestellt.

1.2.2 Frames

Eine weitere Form der Wissensrepräsentation bietet die Ansicht mit **Frames**. Frames können als eine objektorientierte Repräsentation von semantischen Netzen gesehen werden. Ein Frame ist eine Datenstruktur, der alle Eigenschaften eines Objekts in einer tabellarischen Form zusammenfasst. Analog zu semantischen Netzen werden Frames ebenso vernetzt und stellen somit eine objektorientierte Wissensbasis dar. Ein Frame besteht allgemein aus:

- Framebezeichnung
- Slots mit den Facetten:
 - Slotbeschreibung
 - Slotwerten

Die Framebezeichnung dient zur Kennung des Frames innerhalb der Wissensbasis und symbolisiert einen paketförmigen Teil des Wissens aus der Domäne, siehe Abb.1.3. Ein Frame ist eine Liste von Slots und wird aufgrund seiner informationsbündelnden Eigenschaft auch als Paket bezeichnet. Durch diese zusammenfassende Eigenschaft der Datenstruktur wird die Übersichtlichkeit der Vernetzung gefördert und Referenzen, wie Instanzen oder Vererbungen, können in einer alternativ erweiterten Form gekennzeichnet werden. Die zugehörigen Informationen für einen bestimmten Slot werden facettenartig gegliedert und verwaltet.

Framebezeichnung		
Slot 1	<i>Slotbeschreibung 1</i>	Slotwert 1
Slot 2	<i>Slotbeschreibung 2</i>	Slotwert 2
Slot 3	<i>Slotbeschreibung 3</i>	Slotwert 3
Slot 4	<i>Slotbeschreibung 4</i>	Slotwert 4

Abbildung 1.3: Graphische Darstellung eines Frames. Graphische Anlehnung an [25]

Mit passenden Beschreibungen und Werten werden die Facetten der Slots sorgfältig mit Füllsel (*engl.*: fillers) gefüllt und geordnet, damit der Frame einer semantischen Einheit entspricht.

Ein Slot hat mindestens zwei Facetten und zwar den Slotnamen und den Slotwert. Jedoch

ist es bei großen Wissensdatenbanken üblich, mehrere Facetten zu benutzen, damit die objektzentrierende Eigenschaft besser zu Geltung kommt.

Die Facette mit den Slotbedingungen beschreibt den jeweiligen Slot und kann die Werte je nach Art der Brauchbarkeit einschränken. Slotbeschreibungen sind für eine korrekte Syntax sehr hilfreich, weil der Typ des Slotwertes festgesetzt und mit Min- und Maxima eingegrenzt wird. Durch eine vernünftige Slotbeschreibung können missverständliche Slotwerte somit syntaktisch ausgeschlossen werden.

Da mehrere unterschiedliche Objekte aus verschiedenen Objekttypen hergeleitet werden können, benutzt man generische Frames und deren Instanzen, um diesen Sachverhalt realisieren zu können. Der generische Frame hält für jedes Attribut mit dem ein Objekt beschrieben wird, einen Slot bereit. In einer Instanz des generischen Frames wird nun jedem Slot, entsprechend für das Attribut für das er steht, ein Wert zugeordnet.

Slotwerte sind entweder primitive Werte (z.B. Strings, Integer) oder spezielle Verweise auf andere Frames.

Mögliche Verweise sind:

- is-a
- ako
- has
- Parallelverweise (bspw. Muskel X wird innerviert von Nerv Y)

Je nach Implementierung können auch weitere Verweise, die von großem Nutzen sind, in die Wissensbasis eingebaut werden. Beispielsweise ein Verweis, der die Nachfolger in der Framehierarchie anspricht. Die Relation zwischen einem generischen Frame und einer Instanz wird mit Hilfe des is-a Slots hergestellt.

GENERISCHER FRAME			INSTANZ FRAME		
Auge			Auge_X		
AKO	<i>type: link</i>	Slotwert 1	IS-A	<i>type: link</i>	Auge
Größe	<i>type: float</i>	Slotwert 2	Größe	<i>type: float</i>	26
Farbe	<i>type: color</i>	Slotwert 3	Farbe	<i>type: color</i>	braun
Sehstärke	<i>type: dpt-value</i>	Slotwert 4	Sehstärke	<i>type: dpt-value</i>	-0,8

Abbildung 1.4: Graphische Darstellung eines generischen Frames und seiner Instanz. Graphische Anlehnung an [27]

Aus Abb.1.4 ist zu erkennen, dass das **Auge_X** eine Instanz von dem generischen Frame **Auge** ist, weil im Frame **Auge_X** mittels einem is-a Slot auf das Frame **Auge** verwiesen wird. Die Werte zu den Attributen (bspw. Größe, Farbe...) werden in den übrigen Slots gespeichert. In einem generischen Frame wird zu einem Slot nicht nur das Attribut gespeichert, für das der Slot steht, sondern auch zusätzliche Informationen und Programmkonstrukte,

die die Nutzung von Frames sehr flexibel gestalten lassen.
Folgende Slotwerte sind zulässig:

- Defaultwerte, die beim Lesezugriff zurückgegeben werden, wenn kein expliziter Wert zugeordnet wurde.
- Bedingungen, die der Wert erfüllen muss.
- Prozeduranhänge, die beim Lese-, Schreib- oder Löschzugriff auf den Slotwert ausgeführt werden.

Anders als bei semantischen Netzen erlauben alle realisierten Framesysteme einen Prozeduranhang (*engl.*: procedural attachment).

Je nach Anwendung können zwei Auswertungsarten für Prozeduranhänge in Frage kommen.

- **If-added:**

Die Prozedur eines Slots wird ausgewertet, sobald ein Wert für den Slot aufgesetzt wird. Da eine Veränderung in der Datenbank eine Verkettung von Modifikationen auslösen kann, werden Schreibprozeduren benutzt, um die Datenbank fehlerfrei zu verwalten. Diese Art der Auswertung ist für aktive Datenbanken geeignet, die regelmäßig verändert und angepasst werden müssen.

Zum Beispiel ist bei der Amputation des Armes selbstverständlich auch die Amputation der Hand und Finger inbegriffen. Solche plausiblen Informationen sind in einer Datenbank nicht immer selbstverständlich, da die Informationen zum Teil getrennt vorliegen und vernetzt worden sind. Aus diesem Grund sind die Strukturen der Vernetzungen für Schreibprozeduren äußerst wichtig, weil durch falsche Angaben oder Verlinkungen eine fehlerhafte medizinische Wissensdatenbank entsteht.

- **If-needed:**

Die Prozedur eines Slots wird ausgewertet, sobald auf den Slot zugegriffen wird. Diese zielorientierte Auswertung ist ideal für Lese-Prozeduren, die anhand der vorhandenen Daten neue errechnen, welches als Inferenz bekannt ist. Da bei jedem Zugriff auf die Slots Schlüsse gezogen werden, wird das Wissen nicht nur passiv gespeichert, sondern trägt aktiv dazu bei, andere Frames zu modifizieren und die Wissensbasis auf eine leichte Art konsistent zu halten.

Im Folgenden wird eine beispielhafte Implementierung einer Wissensdatenbank vorgestellt, die auf Frames basiert und in Pseudocode-Darstellung abgebildet ist.

[*Auge*

ako : *Sinnesorgan*

groesse :

type : *float*

minKardinalitaet & *maxKardinalitaet* : 1

farbe :

```

    type : color
    minKardinalitaet & maxKardinalitaet : 1
sehstaerke :
    type : dpt - value]

[Sehtest
  ako : Untersuchung
  patient :
    type : String
  sehfehler :
    type : [weitsichtig, kurzsichtig, none]
    proc : if patient.sehstaerke == 0
           then none
           else
             if patient.sehstaerke < 0
               then kurzsichtig
             else
               weitsichtig
    exec : if - needed]
sehstaerke :
  type : dpt - value]

[Auge_X
  is - a : Auge
  groesse : 26
  farbe : braun
  sehstaerke : -0, 8]

```

Dies stellt lediglich ein Kodefragment dar, welches sich nach der Sehkraft des Patienten richtet und anhand dieser Information die Datenbank aufstellt.

Zur medizinischen Sehkraftmessung wird der Dioptrie-Wert benutzt. Die Prozedur benutzt den **Dpt-Wert** (Dioptrie-Wert) und entscheidet, ob eine Weit- bzw. Kurzsichtigkeit des Patienten vorliegt. Diese zugeordneten Prozeduren können in einer Objekthierarchie ähnlich wie Werte vererbt werden.

Aufgrund der objekt-orientierten Beschreibung mit Frames können Vererbungseigenschaften ebenfalls zum Inferenzieren benutzt werden. Die generischen Frames können die Slots an weitere generische Frames vererben und letztendlich ein Netz aus Vererbungsreferenzen aufspannen, welches zu den allgemeinen Stärken von Framekonzepten zählt.

Um diese Relationen in die Slots einbauen zu können, werden ako-Verweise in die Slots eingebettet, so dass im ako-Slot des erbenden Frame vermerkt ist, von welchem Frame er die Slots erbt.

Abb.1.5 ist eine graphische Darstellung des Pseudokodes und dient zur Visualisierung möglicher Frame-Vererbungen innerhalb eines Netzes.

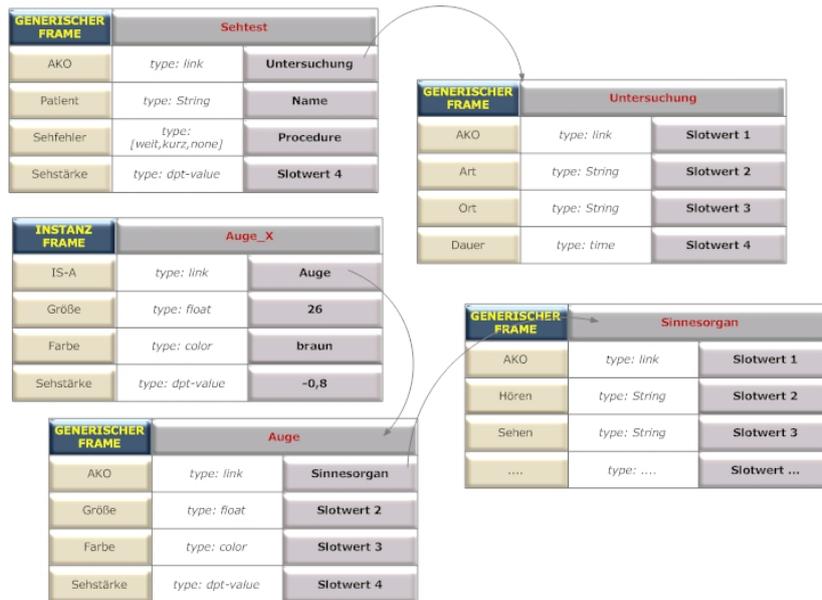


Abbildung 1.5: Graphische Darstellung der Framestruktur zum Pseudocode

Falls eine Information nicht im aktuellen Frame gespeichert ist, können hierarchisch höher liegende Frames auf die gesuchte Information untersucht werden.

Beispiel: Ist das *Auge_X* ein Sinnesorgan? Ja, weil die Superklasse von Auge durch Sinnesorgan gekennzeichnet und über einen ako-Verweis erreichbar ist.

Das Matchingverfahren kann ebenso eine alternative Möglichkeit der Inferenz anbieten. Insbesondere wird Matching in framebasierten Wissensdatenbanken benutzt, um nach dem Frame zu suchen, der die aktuelle Situation am idealsten beschreibt.

Für weitergehende Informationen ist die Literatur [28] zu empfehlen.

1.2.3 Problematik bei inkonsistenten Wissensrepräsentationen

Implementierbare Inferenzverfahren sind zentrales Forschungsgebiet der KR, weil bei großen Datensätzen durch computergestütztes Schlussfolgern viel inferenzierte Information gewonnen werden kann.

Je größer der Datensatz einer Domäne wird, desto komplexer wird die entsprechende Wissensrepräsentation und desto strenger müssen die Anforderungen an ihre Konsistenz sein, damit sinnvolle Ergebnisse beim logischen Schließen erzielt werden können. Man spricht im Forschungsbereich der Logik von **Konsistenz**, wenn die Widerspruchsfreiheit eines axiomatischen Systems ausgedrückt werden soll. Beispielsweise ist eine framebasierte Wissensrepräsentation oder ein semantisches Netz konsistent, wenn die Widerspruchsfreiheit gewährleistet wird.

Semantische Netze und Frames verwenden keinen Kalkül, um Folgerungen aus dem Netz zu ziehen. Abfragemöglichkeiten erfolgen durch das schrittweise Aufsuchen von Relationen. Aufgrund der nicht hinreichend geklärten logischen Semantik ist die Widerspruchs-

freiheit nicht gegeben [29].

Die Semantik der Repräsentationen wird nicht klar definiert, weil die Knoten und Kanten nur durch die Bezeichnungen suggeriert werden. Deshalb sind Inferenzen meist rein operational spezifiziert (i.A. durch die Implementierung) [30].

Die objektorientierte Wissensrepräsentation mit Frames im Vergleich zum semantischen Netz hat zwar den Vorteil, deutlich strukturierter zu sein, jedoch lässt sich auch hier das Problem der logischen Fundierung nicht beseitigen, weil Frames nur eine objektorientierte Repräsentation von semantischen Netzen darstellen. Der Unterschied zu semantischen Netzen besteht darin, dass die Framenamen erweiterte Konzeptknoten darstellen und alle Kanten, die von einem Knoten ausgehen, in ein Paket zusammengelegt werden. Die Frame-Slots, die Verweise auf andere Frames beinhalten, sind analog zu den Kanten eines semantischen Netzes, wobei die Slotfüllungen den Knoten am Ende der Kanten entsprechen.

Da bei semantischen Netzen die passende Inferenzstrategie vom Kontext abhängig ist und aufwändiges Pattern Matching erfordert, sind sie für große Datensätze ungeeignet [84]. Je größer der Datensatz wird, desto unübersichtlicher wird die entsprechende Repräsentation, weil der positive Aspekt der visuellen Intuitivität durch massenhafte Verknüpfungen beeinträchtigt wird.

Es ist ebenso unklar, wie ein Fehlschlag beim Schlussfolgern interpretiert werden soll. Das Problem wird umso größer, wenn Konstrukte Mehrdeutigkeiten aufweisen. Die Korrektheit, Vollständigkeit und Entscheidbarkeit beim Inferenzieren ist aufgrund einer fehlenden formalen Semantik nicht gegeben [85].

Um eine konsistente, vollständige und entscheidbare Wissensrepräsentation reglementieren zu können, werden umfangreiche medizinische Wissensdomänen mittels **Beschreibungslogiken** formatisiert.

Die formale Semantik der Beschreibungslogik wird durch eine leicht verständliche und an natürlichsprachlichen Aussagen angelehnte Syntax gewährleistet. Dieser Repräsentationsformalismus zieht eine adäquate Trennung von intensionalem und assertionalem Wissen heran, damit Schlussfolgerungsprozeduren effizient und fehlerfrei gestaltet werden können.

2 Beschreibungslogik (DL)

2.1 Allgemeines

Anders als der Name suggeriert, ist die Beschreibungslogik (*kurz* DL, *engl.*: Description Logic) keine neue Variante der Logik, die etwa zur Aussagenlogik oder Prädikatenlogik hinzukommen würde. Baader definiert DL wie folgt: „*Description logics (DLs) are a family of knowledge representation languages which can be used to represent the knowledge of an application domain in a structured and formally well-understood way.*“ [31]

Beschreibungslogiken sind eine Familie von Sprachen zur Wissensrepräsentation und sind eine Untermenge der Prädikatenlogik erster Stufe. Die DL ist quasi eine eingeschränkte Form der Prädikatenlogik mit der wichtigen Tatsache, dass sie im Gegensatz entscheidbar ist. Es ist nämlich möglich einen Algorithmus zu entwerfen, der für jedes Element der Menge beantworten kann, ob es eine bestimmte Eigenschaft hat oder nicht. Aufgrund dieser Eigenschaft ist eine konsistente Realisierung von implementierbaren Inferenzverfahren erreichbar.

Aus praktischen Gründen wird eine neue Syntax eingeweiht, dessen Ausdrücke sich vollständig in prädikatenlogische Formeln überführen lassen.

Die Familie der Beschreibungslogiken entfaltet sich systematisch aus den semantischen Netzen und der framebasierten Wissensrepräsentation, weil Inkonsistenzen und semantische Unklarheiten die Forschungen in diesem Bereich ambitionierten. Synonyme Benennungen wie Terminologische Logiken oder Frame-Sprachen bewahrheiten den geschichtlichen Hintergrund der DL-Forschung.

Elementare Bestandteile einer allgemeinen Beschreibungslogik sind Konzepte und Rollen sowie diverse Konstruktoren zur Erstellung komplexerer Definitionen. Konzepte sind nach ihrer prädikatenlogischen Bedeutung unäre Prädikate, wobei die Rollen den binären Prädikaten entsprechen. Konzepte können zu Individuen instanziiert werden, wobei die Relationen zwischen den Individuen durch instanziierte Rollen modelliert werden.

Zur Unterscheidung einzelner Sprachen aus der beschreibungslogischen Sprachfamilie hat sich ein Namensschema etabliert, das die eingesetzten Konstruktoren bestimmten Buchstaben zuordnet und aufführt.

Dieses Namensschema ist mit Vorsicht zu genießen, weil es kein vollständig konsistentes Namensschema darstellt (z.B. ist \mathcal{F} mehrdeutig).

2.2 Abstrakte Syntax und Semantik

In der Theorie der Beschreibungslogiken gelten für die Benennung von Konzepten, Relationen, Konstruktoren und Individuen folgende Konventionen und Notationen [32]:

KONZEPT	Begriff, der für sich selbst steht ohne die Existenz zweier Objekte zu implizieren (vgl. mit unärem Prädikat $C(x)$ in PL1).
INDIVIDUUM	Individuelle Ausprägung (Instanz) eines Konzepts aus der Wissensdomäne.
RELATION	Beziehung, die zwei Konzepte oder Individuen miteinander verbindet. Umgangssprachlich öfters im Zusammenhang mit „is-a“, „has“ oder „a-kind-of“ ausgedrückt (vgl. mit binärem Prädikat $r(x,y)$ mit den freien Variablen x und y).
KONSTRUKTOR	Operator zur Erstellung komplexer Beschreibungen von Konzepten, Relationen und Individuen (vgl. mit den Operatoren in PL1).

Tabelle 2.1: Identifikation von Konzepten, Relationen, Konstruktoren und Individuen

In dieser Diplomarbeit gelten für die Benennung von Konzepten, Rollen und Individuen folgende Konventionen und Notationen:

- A, B bezeichnen atomare Konzepte, d.h. Konzepte ohne komplexe Definition.
- C, D bezeichnen komplexe Konzeptausdrücke.
- r, s, t bezeichnen Relationen.
- a, b bezeichnen Individuen.

In der Entstehung der Beschreibungslogiken spielte die KL-ONE-Eigenschaft eine entscheidende Rolle. Der KL-ONE Ansatz unterscheidet zwischen der terminologischen Ebene (T-Box) und der assertionalen Ebene (A-Box).

Die T-Box enthält das Wissen über die Konzepte einer Domäne, also das terminologische Wissen. Es ist zu vergleichen mit einem Datenbankschema einer Datenbank. Sie besteht aus lauter beschreibungslogischen Definitionen, die Beziehungen zwischen Konzepten und Rollen zueinander realisiert. Diese spezifischen Definitionen bilden in ihrer Gesamtheit die T-Box oder mit anderen Worten eine Terminologie \mathcal{T} .

Die Sammlung von Individuen, die aus einer T-Box \mathcal{T} erstellt wird, heißt A-Box \mathcal{A} . Die A-Box \mathcal{A} enthält das Wissen über konkrete Instanzen der vorhandenen Konzepte und modelliert ihre wechselseitigen Beziehungen. Sie ist zu vergleichen mit den konkreten Daten aus einer Datenbank.

Somit wird eine strukturelle Unterscheidung zwischen der Terminologie einer Anwendungsdomäne und der Menge von individuellen Objekten dieser Terminologie vorgenommen, so dass komplex zusammengesetzte Konzepte und Rollen durch Konstruktoren aufgebaut werden können.

Terminologische Axiome dienen zur Erstellung von Definitionen und haben folgende Form:

- $C \sqsubseteq D$ (Konzeptinklusion)

- $r \sqsubseteq s$ (Rolleninklusion)
- $C \equiv D$ (Konzeptäquivalenz)
- $r \equiv s$ (Rollenäquivalenz)

Eine **vollständige Definition** setzt eine Konzeptbeschreibung C mit einem atomaren Konzept A in eine Äquivalenzbeziehung, sprich $A \equiv C$.

Bsp.: $Patient \equiv Mensch \sqcap \exists leidet_an.Krankheit$

Ein Patient ist ein Mensch, der unter einer bestimmten Krankheit leidet.

Eine **primitive Definition** setzt eine Konzeptbeschreibung C mit einem atomaren Konzept A in eine Inklusionsbeziehung, sprich $A \sqsubseteq C$.

Bsp.: $Infekt \sqsubseteq Krankheit \sqcap \forall hatUrsache.Erreger$

Ein Infekt ist eine Erkrankung dessen Ursache ein gewisser Erreger ist.

Die Syntax der T-Box:

Man bezeichnet eine endliche Menge von primitiven und vollständigen Definitionen als Terminologie bzw. T-Box \mathcal{T} , wenn jedes atomare Konzept genau einmal links von der Definition auftaucht, d.h.

- keine mehrfachen Definitionen wie $\{A \equiv C, A \sqsubseteq D\}$ oder $\{A \equiv D, A \equiv C\}$
- keine zyklischen Definitionen wie $\{A \equiv \forall r.B, B \equiv \exists s.A\}$ (direkt als auch indirekt)

Die Semantik der Konstrukte wird durch eine Interpretationsfunktion $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ definiert.

Die Interpretationsfunktion hat folgende Abbildungseigenschaften:

- Das atomare Konzept A wird auf eine Teilmenge des Objektuniversums $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ abgebildet.
 $A^{\mathcal{I}} = \{a \mid A(a) \text{ „ist wahr in“ } \mathcal{I}\}$
 Bsp.: $Schmerzmittel^{\mathcal{I}} = \{Paracetamol, Aspirin, Ibuprofen...\}$
- Die Relation r wird auf Teilmengen von $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ zugewiesen.
 $r^{\mathcal{I}} = \{(a,b) \mid r(a,b) \text{ „ist wahr in“ } \mathcal{I}\}$
 Bsp.: $hatKrankheit^{\mathcal{I}} = \{(Peter, Diabetis), (Herbert, Hepatitis)...\}$
- Individuen kennzeichnen konkrete Elemente der Domäne $\Delta^{\mathcal{I}}$.

Man nennt die Symbole der linken Seite eines *atomaren* Konzepts der T-Box \mathcal{T} *Namenssymbole* und die, die auf der rechten Seiten stehen werden als *Basissymbole* bezeichnet. Eine Interpretation \mathcal{J} von \mathcal{T} , die nur Basissymbole interpretiert, heißt Basisinterpretation. Existiert eine Interpretation \mathcal{I} , die auch Namenssymbole interpretiert und die gleiche Domäne wie \mathcal{J} hat ($\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$), spricht man von einer **Expansion** bzw. Erweiterung von \mathcal{J} . Wenn jede Basisinterpretation eine Expansion als Modell für \mathcal{T} hat, heißt die Terminologie *definitiv*. Insbesondere Terminologien mit azyklischen Definitionen sind definitiv. Eine Expansion \mathcal{T}' von \mathcal{T} erhält man durch einzelnes Ersetzen der

Namenssymbole durch ihre vorhandenen Definitionen, solange \mathcal{T} azyklisch ist. Die Größe der Expansion kann unter Umständen exponentiell größer werden als die vorhandene Terminologie. Insbesondere gilt für \mathcal{T}' und \mathcal{T} :

- \mathcal{T}' und \mathcal{T} haben die gleichen Namens- und Basissymbole
- \mathcal{T}' und \mathcal{T} sind äquivalent
- \mathcal{T}' und \mathcal{T} sind beide definatorisch

Passende Beispiele können in [32] recherchiert werden.

In einigen Literaturen werden entweder T-Boxen definitiv als zyklensfrei vorausgesetzt oder nach azyklischen, zyklischen und generellen T-Boxen differenziert. *Zyklische* T-Boxen beinhalten keine Ausdrücke der Form:

$A_0 \equiv C_0$ bzw. $A_0 \sqsubseteq C_0, \dots, A_{n-1} \equiv C_0$ bzw. $A_{n-1} \sqsubseteq C_{n-1}$, so dass A_i in C_{i+1} vorkommt, für alle $i < n$ (modulo n).

Eine *generelle* oder *allgemeine* T-Box ist eine endliche Menge von Konzeptinklusionen der Form $C \sqsubseteq D$ mit C, D als Konzeptausdrücken. Konzeptinklusionen zwischen komplexen Konzepten bzw. Konzeptausdrücken führen dazu, dass auf beiden Seiten der Definition komplexe Konzeptbeschreibungen zugelassen werden. Ein solches Axiom bezeichnet man als **General Concept Inclusion** (*kurz*: GCI). Wie man erkennen kann, erlauben generelle T-Boxen im Unterschied zu azyklischen und zyklischen T-Boxen allgemeine Constraints, die zwar die Terminologie ausdrucksstärker machen, jedoch zu einer üblicherweise erhöhten Komplexität beim Schlussfolgern führen. Da $A \equiv C$ äquivalent zu $A \sqsubseteq C, C \sqsubseteq A$ ist, ist jede azyklische T-Box auch eine generelle T-Box, jedoch ist nicht jede generelle T-Box eine (a)zyklische T-Box.

Wird zusätzlich zur *generellen T-Box* noch die Rolleninklusion (*RI*) eingeführt, spricht man von einer *C-Box*, die sogenannte „*Constraint Box*“.

Wenn man primitive Definitionen in einer T-Box ausschließen möchte, ist eine Normalisierung der T-Box angebracht. Eine Terminologie heißt *normalisiert*, wenn sie keine Definitionen mit \sqsubseteq enthält. Die Normalform \mathcal{T}' einer Terminologie \mathcal{T} erhält man durch folgende Umformung:

Jede Definition der Art $A \sqsubseteq C$ wird durch $A = A^* \sqcap C$ ersetzt, wobei A^* ein neuer (nicht benutzter) Begriffsname ist.

Beispiel: $Patient \sqsubseteq Mensch$ wird zu $Patient = Patient^* \sqcap Mensch$

Intuitiv steht $Patient^*$ für die Qualitäten, die einen Patienten von anderen Menschen (semantisch gesehen) unterscheiden. Wenn \mathcal{I} ein Modell einer Terminologie \mathcal{T} ist, dann existiert ein Modell \mathcal{I}' der normalisierten Terminologie \mathcal{T}' (und umgekehrt), so dass für alle Begriffsnamen A , die in \mathcal{T} vorkommen, gilt: $A^{\mathcal{I}} = A^{\mathcal{I}'}$.

Semantik der T-Box:

Eine Interpretation heißt **Modell** für die Terminologie \mathcal{T} genau dann, wenn $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ für alle Axiome in $C \sqsubseteq D$ in \mathcal{T} gilt (analog für $C \equiv D$). Zwei Terminologien $\mathcal{T}_1, \mathcal{T}_2$ sind äquivalent zueinander, wenn sie das gleiche Modell haben.

Die Syntax der A-Box:

- $a : C$ Konzeptinstanziierung
- $(a, b) : r$ Rolleninstanziierung

Beispiel:

$Sam : Patient$

$(Sam, Grippe) : hatKrankheit$

$Sam : (Mann \sqcap (= 2 hatKind Frau))$

Sam hat zwei Töchter und ist ein Patient, weil er an einer Grippe erkrankt ist.

Die Semantik der A-Box:

Unterschiedliche Namen in einer Interpretation referenzieren unterschiedliche Individuen, so dass $a \neq b \Rightarrow a^{\mathcal{I}} \neq b^{\mathcal{I}}$ gilt. Diese Forderung wird als die **unique name assumption** bezeichnet.

Behauptungen (*engl.* assertions) werden benutzt, um auszudrücken, dass entweder ein Element eine Instanz eines Konzepts ist, oder dass zwei Elemente durch eine Relation in Beziehung gebracht wurden.

In PL1-Semantik:

$a : C$ entspricht $C(a)$ und $(a, b) : r$ entspricht $r(a, b)$

Die entsprechende Interpretation:

$a : C$ entspricht $a^{\mathcal{I}} \in C^{\mathcal{I}}$ und $(a, b) : r$ entspricht $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

Eine Interpretation \mathcal{I} bewahrheitet die Konzeptinstanziierung $a : C$ bzw. Rolleninstanziierung $(a, b) : r$, gdw. $a^{\mathcal{I}} \in C^{\mathcal{I}}$ bzw. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ gilt.

Eine Interpretation \mathcal{I} erfüllt die A-Box \mathcal{A} , gdw. alle Konzept- und Rolleninstanziierungen erfüllt sind.

Da das Paar $(\mathcal{I}, \mathcal{A})$ eine konkrete Wissensdatenbank charakterisiert, bezeichnet man es auch als **Ontologie**.

2.2.1 Konzeptbildende Strukturen

Jede Beschreibungslogik ist durch die Menge ihrer zugelassenen Konstruktoren definiert, die eine Definition von zusammengesetzten Konzepten und Rollen erlauben. Die Auswahl an verschiedenen Konstruktoren bestimmt die Komplexität der auf einer Beschreibungslogik arbeitenden Inferenzalgorithmen.

Die Konstruktormenge fungiert als Richtlinie für die Ausdrucksmächtigkeit einer Beschreibungslogik: je größer die Anzahl der Konstruktoren, desto stärker die Ausdrucksmächtigkeit und höher der Sprachumfang verbunden mit einer wachsenden Komplexität beim Inferenzieren. Diese Trade-Off Eigenschaft zwischen Ausdrucksmächtigkeit und Komplexität beim Inferenzieren wird in der Theorie der Beschreibungslogiken über die Jahre hinweg sehr stark erforscht mit dem Ziel, diese zwei Komponenten näher zusammenzubringen, so dass man letztendlich eine Sprache entwickelt, die nicht algorithmisch komplex zu verarbeiten ist und gleichzeitig einen hohen Grad an Ausdrucksstärke aufweist.

Im Folgenden werden die bekanntesten konzeptbildenden Strukturen vorgestellt:

Boolesche Operatoren werden benutzt, um Konzepte untereinander durch Vereinigung und Schnittbildung abzugrenzen.

Syntax:

- $C \sqcap D$ Konzeptkonjunktion
- $C \sqcup D$ Konzeptdisjunktion
- $\neg C$ Konzeptnegation

Beispiele: $Patientin \sqcup Patient$, $Schmerzmittel \sqcap \neg Antibiotikum$

In PL1-Semantik: $C(x) \wedge D(x)$, $C(x) \vee D(x)$, $\neg C(x)$

Die entsprechenden Interpretationen: $C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

Neben der Konjunktion, Disjunktion und Negation von Konzepten sind **Relationsrestriktionen** die wichtigsten Elemente zur Konstruktion von Konzeptdeskriptionen. Es gibt zwei verschiedene Arten von Relationsrestriktionen.

Syntax:

- $\forall r.C$ universelle Quantifikation
- $\exists r.C$ existentielle Quantifikation

Beispiele:

$Mensch \sqcap \exists \text{leidet_an. Krankheit}$

$Krankheit \sqcap \forall \text{hatUrsache. Erreger}$

In PL1-Semantik:

$(\forall r.C)(x) = \forall y(r(x, y) \Rightarrow C(y))$

$(\exists r.C)(x) = \exists y(r(x, y) \wedge C(y))$

Die entsprechenden Interpretationen:

$(\forall r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \forall e((d, e) \in r^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}})\}$

$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists e((d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}})\}$

Damit die Anzahl von möglichen Füllern einer Relation eingeschränkt werden kann, werden **Kardinalitätseinschränkungen** vorgenommen.

Syntax:

- $\geq n r$
- $\leq n r$

Beispiel:

$Patient \sqcap (\geq 2 \text{ hatBeschwerden})$

In PL1-Semantik:

$(\geq n r)(x) = \exists y_1, \dots, y_n(r(x, y_1) \wedge \dots \wedge r(x, y_n) \wedge y_i \neq y_j \text{ für } 1 \leq i < j \leq n)$

$$(\leq n r)(x) = \neg(\geq (n+1)r)(x)$$

Die entsprechenden Interpretationen:

$$(\geq n r)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid |\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\}| \geq n\}$$

$$(\leq n r)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid |\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\}| \leq n\}$$

Es gilt:

$$(\approx n r) \text{ ist äquivalent zu } (\leq n r) \sqcap (\geq n r)$$

In der Rangordnung der Konzeptbeschreibungen gibt es ein allgemeinstes und speziellestes Konzept.

Syntax:

- \top allgemeinste Konzept (Top-Konzept)
- \perp spezielleste Konzept (Bottom-Konzept)

In PL1-Semantik:

$$\top = \text{true} \text{ und } \perp = \text{false}$$

Die entsprechenden Interpretationen:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \text{ und } \perp^{\mathcal{I}} = \emptyset$$

Damit eine Kardinalitätsbeschränkung mit gleichzeitiger Qualifizierung der Rollenfüller erreicht wird, benutzt man **qualifizierende Kardinalitätseinschränkungen**.

Syntax:

- $\geq n r.C$
- $\leq n r.C$
- $= n r.C$

Beispiele:

Patient $\sqcap (\geq 2 \text{ hatBruch.Arm})$ bedeutet, dass ein Patient mehr als 2 Knochenbrüche am Arm hat.

Mensch $\sqcap (= 10 \text{ hat.Finger})$ beschreibt, dass ein Mensch genau 10 Finger hat.

Bemerkung: $(\geq n r)$ lässt sich ebenso mit $(\geq n r.\top)$ simulieren.

In PL1-Semantik:

$$(\leq n r.C)(x) = \forall y_1, \dots, y_{n+1} (r(x, y_1) \wedge \dots \wedge r(x, y_{n+1}) \wedge C(y_1) \wedge \dots \wedge C(y_{n+1}) \Rightarrow \bigvee_{i < j} y_i = y_j)$$

$$(\geq n r.C)(x) = \neg(\leq (n+1) r.C)(x)$$

Die entsprechenden Interpretationen:

$$(\geq n r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid |\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}| \geq n\}$$

$$(\leq n r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid |\{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}| \leq n\}$$

Die Erweiterung stellt die **qualifizierende universelle Quantifikation** dar.

Syntax:

- $\forall r.(CD)$

Alle Füller von r sind vom Typ D und zugleich vom Typ C. Füller von r , die nicht dem Typ D entsprechen, müssen nicht vom Typ C sein.

Beispiel: *Krankenhaus* \sqcap (*hatAngestellte Frau Krankenpfleger*)

Alle angestellten Krankenpfleger im Krankenhaus sind Frauen.

In PL1-Semantik:

$$(\forall r.(CD))(x) = \forall y(r(x, y) \wedge D(y) \Rightarrow C(y))$$

Die entsprechende Interpretation:

$$(\forall r.(CD))^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \forall e((d, e) \in r^{\mathcal{I}} \wedge e \in D^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}})\}$$

Um Beziehungen zwischen den Füllermengen von Relationen auszudrücken, werden **Relations-Füller-Beziehungen** (*engl.: role value maps*) herangezogen.

Syntax:

- $r \subseteq s$ Fülleruntermengenbeziehung
- $r = s$ Füllermengengleichheit

Beispiel: *Patient* \sqcap (*hatKopfschmerzen* \subseteq *hatSchmerzen*)

In PL1-Semantik:

$$(r \subseteq s)(x) = \forall y(r(x, y) \rightarrow s(x, y))$$

$$(r = s)(x) = \forall y(r(x, y) \leftrightarrow s(x, y))$$

Die entsprechenden Interpretationen:

$$(r \subseteq s)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \forall e((d, e) \in r^{\mathcal{I}} \Rightarrow (d, e) \in s^{\mathcal{I}})\}$$

$$(r = s)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \forall e((d, e) \in r^{\mathcal{I}} \Leftrightarrow (d, e) \in s^{\mathcal{I}})\}$$

Eine Relation ist funktional, wenn die Interpretation zu genau einem Wert führt. **Funktionale Relationen** (*engl.: feature/attributes*) müssen die Forderung (=1 f) erfüllen, d.h. $\forall a, b, c, ((a, b) \in f^{\mathcal{I}} \wedge (a, c) \in f^{\mathcal{I}} \Rightarrow b = c)$.

Diese spezielle Form der Relation erlaubt uns Füllerwerte auf Übereinstimmung zu überprüfen. **Füller-Vergleiche** (*engl.: agreement/disagreement*) gelten nur bei funktionalen Relationen.

Syntax:

- $f \doteq g$ Füllerübereinstimmung
- $f \not\doteq g$ keine Füllerübereinstimmung

Beispiel: Seien *hatVerbrennung* und *hatSchmerz* funktionale Relationen in *Patient* \sqcap (*hatVerbrennung* \doteq *hatSchmerz*)

Ein Patient hat Schmerzen an der verbrannten Stelle.

In PL1-Semantik:

$$(f \doteq g)(x) = \exists y, z(f(x, y) \wedge g(x, z) \wedge (y = z))$$

Die entsprechende Interpretation:

$$(f \doteq g) = \{d \in \Delta^{\mathcal{I}} | \exists e((d, e) \in f^{\mathcal{I}} \wedge (d, e) \in g^{\mathcal{I}})\}$$

Um die Werte von numerischen Füllern zweier funktionaler Relationen vergleichen zu können, benutzt man arithmetische Füllervergleiche wie

$f \{< | > | = | \leq | \geq | \neq\} g$.

Beispiel: $Patient \sqcap (hatGewicht_inKg > 200)$ entspricht einem übergewichtigen Patienten mit einem Gewicht von mehr als 200kg.

In PL1-Semantik:

$(f < g)(x) = \exists y, z(f(x, y) \wedge g(x, z) \wedge (y < z))$

Die entsprechende Interpretation:

$(f < g)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \exists a, b((d, a) \in f^{\mathcal{I}} \wedge (d, b) \in g^{\mathcal{I}} \wedge a < b)\}$

Um **explizite Füllerangaben** erreichen zu können, werden Füller bzgl. einer gegebenen Relation ausgeklammert. Somit können aus der Instanzmenge $\{i_1, \dots, i_k\}$ die expliziten Füllerangaben gezielt angesprochen werden.

Syntax:

- $r\{i_1, \dots, i_k\}$ Füllermenge von r
- $r\neg\{i_1, \dots, i_k\}$ keine Füller r

Beispiel:

$Patient \sqcap (benutzt_Medikament\{Allergodil, Cromohexal, Cetirizin\})$ entspricht einem allergiekranken Patienten, der mit den Medikamenten Allergodil, Cromohexal und Cetirizin behandelt wird.

2.2.2 Relationsbildende Strukturen

Neben den konzeptbildenden Strukturen kann jede Beschreibungslogik auch durch relationsbildende Ausdrücke generiert werden. Auch hier bestimmt die Auswahl an verschiedenen relationsbildenden Strukturen die Komplexität von Inferenzalgorithmen und fungiert analog als Maß für die Ausdrucksmächtigkeit der generierten Beschreibungssprache.

Boolesche Relationsoperatoren werden benutzt, um Relationen untereinander durch Vereinigung und Schnittbildung abzugrenzen.

Syntax:

- $r \sqcap s$ Relationenkonzjunktion
- $r \sqcup s$ Relationendisjunktion
- $\neg s$ Relationennegation

Beispiele: $benutztMedikament \sqcap stationiert_in, hatKrankheit \sqcup \neg hatKind$

In PL1-Semantik: $r(x, y) \wedge s(x, y), r(x, y) \vee s(x, y), \neg r(x, y)$

Die entsprechenden Interpretationen: $r^{\mathcal{I}} \cap s^{\mathcal{I}}, r^{\mathcal{I}} \cup s^{\mathcal{I}}, (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus r^{\mathcal{I}}$

Allgemeinste und speziellste Relation existieren implizit in den Definitionen eines festgelegten beschreibungslogischen Systems.

Syntax:

Konstruktor	Syntax	Interpretation
Atomares Konzept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Allgemeinstes Konzept	\top	$\Delta^{\mathcal{I}}$
Speziellstes Konzept	\perp	\emptyset
Konzeptkonjunktion	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Konzeptdisjunktion(\mathcal{U})	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Konzeptnegation(\mathcal{C})	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Universelle Quantifikation	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e((d, e) \in r^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}})\}$
Existentielle Quantifikation(\mathcal{E})	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e((d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}})\}$
Kardinalitätseinschränkung(\mathcal{N})	$\geq n \ r$	$\{d \in \Delta^{\mathcal{I}} \mid \{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\} \geq n\}$
	$\leq n \ r$	$\{d \in \Delta^{\mathcal{I}} \mid \{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\} \leq n\}$
Qualif.Kardinalitätseinschr.(\mathcal{Q})	$\geq \mid \leq n \ r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \{e \in \Delta^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \geq \mid \leq n\}$
Qualif. univ. Quantifikation	$\forall r.(CD)$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e((d, e) \in r^{\mathcal{I}} \wedge e \in D^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}})\}$
Relationsfüller-Beziehung	$r \subseteq s$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e((d, e) \in r^{\mathcal{I}} \Rightarrow (d, e) \in s^{\mathcal{I}})\}$
	$r = s$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e((d, e) \in r^{\mathcal{I}} \Leftrightarrow (d, e) \in s^{\mathcal{I}})\}$
Funktionale Relation	$(= 1 \ f)$	$\forall a, b, c, ((a, b) \in f^{\mathcal{I}} \wedge (a, c) \in f^{\mathcal{I}} \Rightarrow b = c)$
Rollengleichheit (\mathcal{F})	$f \doteq g$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e((d, e) \in f^{\mathcal{I}} \wedge (d, e) \in g^{\mathcal{I}})\}$
Füllwertvergleiche	$f \leq \mid \geq \mid =$	$\{d \in \Delta^{\mathcal{I}} \mid \exists a, b((d, a) \in f^{\mathcal{I}} \wedge (d, b) \in g^{\mathcal{I}} \wedge a \leq \mid \geq \mid = \mid \neq b)\}$
	$\mid \neq g$	
Instanzmenge	$\{i_1, \dots, i_k\}$	$\{i_1^{\mathcal{I}}, \dots, i_k^{\mathcal{I}}\}$
Explizite Füllermenge von r	$r\{i_1, \dots, i_k\}$	$\{d \in \Delta^{\mathcal{I}} \mid \{i_1, \dots, i_k\} \subseteq r^{\mathcal{I}}(d)\}$
Keine Füllermenge von r	$r\neg\{i_1, \dots, i_k\}$	$\{d \in \Delta^{\mathcal{I}} \mid \{i_1, \dots, i_k\} \not\subseteq r^{\mathcal{I}}(d)\}$

Tabelle 2.2: Konzeptbildende Strukturen im Überblick

- U allgemeinste Relation
- ? speziellste Relation

Die entsprechenden Interpretationen: $U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, speziellste Relation = \emptyset

Um eine umfassende Relationsbeschränkung (vgl. mit lokaler Beschränkung in Konzeptdefinitionen) anbieten zu können, werden **Einschränkungen des Ursprungs- und Bildbereichs** vorgenommen.

Syntax:

- $r|_C$ Ursprungsmengeneinschränkung
- $r|_D$ Bildmengeneinschränkung

Beispiele: $benutztMedikament|_{Aspirin}, hatKrankheit|^{Patient}$

In PL1-Semantik:

$$(r|_C)(x, y) = \forall x, y (r(x, y) \Rightarrow C(x))$$

$$(r|_C)(x, y) = \forall x, y (r(x, y) \Rightarrow C(y))$$

Die entsprechenden Interpretationen:

$$(r|_C)^{\mathcal{I}} = \{(d, e) | (d, e) \in r^{\mathcal{I}} \wedge d \in C^{\mathcal{I}}\}$$

$$(r|_C)^{\mathcal{I}} = \{(d, e) | (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$$

Durch die **Relationsumkehrung** können Referenzierungen inverser Beziehungen realisiert werden.

Syntax:

- r^{-1} Relationsumkehrung

Beispiel: $operiert^{-1}$

In PL1-Semantik:

$$(r^{-1})(x, y) = \forall y, x (r(y, x))$$

Die entsprechende Interpretation:

$$(r^{-1})^{\mathcal{I}} = \{(e, d) | (d, e) \in r^{\mathcal{I}}\}$$

Die Verschachtelung bzw. Konkatenierung von Relationen bezeichnet man als **Relationskomposition**.

Syntax:

- $r \circ s$ Relationskomposition

Beispiel: $hatKrankheit \circ benutztMedikament$

In PL1-Semantik:

$$(r \circ s)(x, z) = \exists y (r(x, y) \wedge s(y, z))$$

Die entsprechende Interpretation:

$$(r \circ s)^{\mathcal{I}} = \{(d, f) | \exists e \in \Delta^{\mathcal{I}} (d, e) \in r^{\mathcal{I}} \wedge (e, f) \in s^{\mathcal{I}}\}$$

Durch die iterative Konkatenierung von Relationen können auch transitive Beziehungen von Relationen erzielt werden. Man bezeichnet diesen Vorgang kurz als **transitive Relation**, die folgende Form hat:

Syntax:

- r^+ Transitive Hülle

Die entsprechende Interpretation:

$$(r^+)^{\mathcal{I}} = \bigcup_{n \geq 0} (r^{\mathcal{I}})^n$$

$$\text{Mit } (r^{\mathcal{I}})^0 = \{(d, d) | d \in \Delta^{\mathcal{I}}\} \text{ und } (r^{\mathcal{I}})^{n+1} = (r^{\mathcal{I}})^n \circ r^{\mathcal{I}}$$

2.3 Inferenzdienste

Inferenzdienste, auch Inferenzmaschinen genannt, haben die Aufgabe, neue Aussagen durch das Prinzip der Schlussfolgerung aus einer bestehenden Wissensbasis abzuleiten.

Konstruktor	Syntax	Interpretation
Atomare Relation	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Allgemeinste Relation	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Speziellste Relation	$?$	\emptyset
Relationenkonjunktion(\mathcal{R})	$r \sqcap s$	$r^{\mathcal{I}} \cap s^{\mathcal{I}}$
Relationendisjunktion	$r \sqcup s$	$r^{\mathcal{I}} \cup s^{\mathcal{I}}$
Relationennegation	$\neg r$	$(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus r^{\mathcal{I}}$
Ursprungsmengeneinschränkung	$r _C$	$\{(d, e) (d, e) \in r^{\mathcal{I}} \wedge d \in C^{\mathcal{I}}\}$
Bildmengeneinschränkung	$r _D$	$\{(d, e) (d, e) \in r^{\mathcal{I}} \wedge e \in D^{\mathcal{I}}\}$
Relationsumkehrung	r^{-1}	$\{(e, d) (d, e) \in r^{\mathcal{I}}\}$
Relationskomposition	$r \circ s$	$\{(d, f) \exists e \in \Delta^{\mathcal{I}} (d, e) \in r^{\mathcal{I}} \wedge (e, f) \in s^{\mathcal{I}}\}$
Transitive Relation	r^+	$\bigcup_{n>0} (r^{\mathcal{I}})^n$

Tabelle 2.3: Relationsbildende Strukturen im Überblick

Sie sind Kernbestandteil von Expertensystemen und anderen wissensbasierten Systemen. Die Abb.2.1 zeigt eine mögliche Architektur einer Wissensrepräsentation basierend auf einer Beschreibungslogik. Sie visualisiert den praktischen Umgang einer Beschreibungslogik, in der die Wissensbasis als interagierendes System benutzt wird. Fragestellungen werden im Interface erzeugt und sollen anhand des Inferenzsystems gelöst werden. Man unterscheidet zwischen zwei verschiedenen formulierbaren Fragestellungen:

1. **Bestätigung** (*engl.: validation*)

Das Inferenzsystem greift auf die Wissensbasis zu und beantwortet die Fragestellung entweder mit *True* oder *False*. Diese Art von Bestätigung bzw. Verneinung einer Fragestellung hilft zur Revidierung eines erwarteten Kontexts.

2. **Anfrage** (*engl.: query/retrievals*)

Das Inferenzsystem greift auf die Wissensbasis zu und beantwortet die Fragestellung mit der zugehörigen Menge an Bindungen, die durch die Anfrage impliziert bzw. durch die logische Implikation gefolgert wird. Wenn die Anfrage keine Bindungen impliziert, wird die leere Menge ausgegeben.

Die gelöste Bestätigung oder Anfrage wird im Interface veranschaulicht und steht dem User zur Verfügung.

Durch die Trennung von intensionalem und assertionalem Wissen können somit Anfragen einfacher und korrekt gelöst werden, weil die entsprechende Inferenzprozedur von der jeweiligen Wissensbox abhängig ist.

Der Sinn jedes Schlussfolgerungsdienstes ist es, implizites Wissen explizit zu machen; je detaillierter die Antworten sind, desto ausdrucksstärker wird das explizite Wissen. Die Erkenntnis, dass die Wissensbasis ein Modell hat oder nicht, ist zu plump. Deswegen sind folgende Eigenschaften von Konzeptdeskriptionen aufschlussreich:

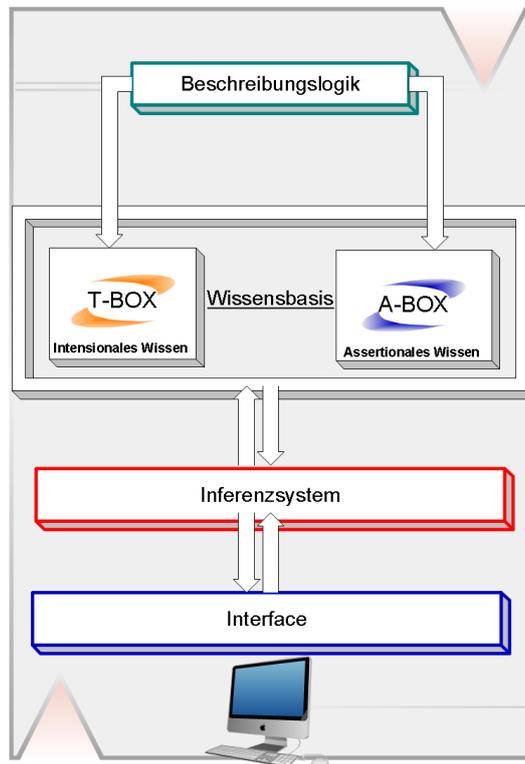


Abbildung 2.1: Architektur einer Wissensrepräsentation mittels Beschreibungslogik

Erfüllbarkeit (*engl.*: **Satisfiability**):

Wenn es ein Modell \mathcal{I} von \mathcal{T} gibt, so dass $C^{\mathcal{I}}$ nicht leer ist ($C^{\mathcal{I}} \neq \emptyset$), dann wird das Konzept C als erfüllbar bezeichnet.

Durch ein Erfüllbarkeitstest kann man aufspüren, ob ein Konzept bereits in sich widersprüchlich ist, oder ob ein Modell dafür existiert. Man übergibt dem Schlussfolgerungsdienst die Konzeptdefinition und bekommt als Antwort ein *True*, falls das Konzept erfüllbar ist, andernfalls ein *False*.

Äquivalenz (*engl.*: **Equivalence**):

Zwei Konzepte C und D sind äquivalent bzgl. einer T-Box \mathcal{T} , gdw. für jedes Modell \mathcal{I} von \mathcal{T} die Aussage $C^{\mathcal{I}} = D^{\mathcal{I}}$ gilt. Man schreibt auch $\mathcal{T} \models C \equiv D$ oder $C \equiv_{\mathcal{T}} D$.

Disjunktheit (*engl.*: **Disjointness**):

Zwei Konzepte C und D sind disjunkt bzgl. einer T-Box \mathcal{T} , gdw. für jedes Modell \mathcal{I} von \mathcal{T} die Aussage $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ gilt.

Eine zentrale Fragestellung in DL-Systemen ist, ob ein Konzept ein anderes Konzept klassifikatorisch impliziert. Diese Eigenschaft wird durch einen Subsumtionstest in Erwä-

gung gezogen. **Subsumtion** (*engl.: Subsumption*) ist ein klassifikatorischer Vorgang, bei dem man ein Konzept unter einen anderen Konzept ordnet.

Ein Konzept C wird von einem Konzept D bzgl. einer T-Box \mathcal{T} **subsumiert**, gdw. für jedes Modell \mathcal{I} von \mathcal{T} die Aussage $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ gilt.
Man schreibt auch $\mathcal{T} \models C \sqsubseteq D$ oder $C \sqsubseteq_{\mathcal{T}} D$.

Somit können implizite Inklusionsbeziehungen komplexer Konzeptausdrücke durch eine Subsumtionsinferenzierung explizit werden. Auch die Individuen, die die jeweiligen Instanzen des konkreten Konzepts darstellen, können durch die Subsumtionsbeziehung explizit werden.

Die Subsumtionsbeziehung ist transitiv: Aus $C \sqsubseteq_{\mathcal{T}} D$ und $D \sqsubseteq_{\mathcal{T}} E$ folgt $C \sqsubseteq_{\mathcal{T}} E$.

Falls $C \sqsubseteq_{\mathcal{T}} D$ und $D \sqsubseteq_{\mathcal{T}} C$ folgt $C \equiv_{\mathcal{T}} D$.

Ergänzend gilt: C und D sind disjunkt $\Leftrightarrow C \sqcap D$ wird subsumiert von \perp .

Die Subsumtionsprozedur wird nicht nur zur Hierarchisierung von Konzepten innerhalb der Terminologie eingesetzt, sondern kann ebenso in ein Erfüllbarkeitstest umgeformt werden, wenn folgende Äquivalenz gezeigt werden kann:

- $C \sqsubseteq D \Leftrightarrow C \sqcap \neg D$ ist unerfüllbar.

Um ein Unerfüllbarkeitstest in einen Subsumtionstest abzuändern, stellt man die Subsumtionsfrage mit $D \sqsubseteq \perp$.

Es gelten weitere Reduktionen auf Unerfüllbarkeit:

- $C \equiv D \Leftrightarrow$ sowohl $C \sqcap \neg D$ als auch $\neg C \sqcap D$ sind unerfüllbar.
- C und D sind disjunkt $\Leftrightarrow C \sqcap D$ ist unerfüllbar.

Folgende Aussagen sind äquivalent zueinander:

- C ist unerfüllbar.
- C wird subsumiert von \perp .
- C und \perp sind äquivalent.
- C und \top sind disjunkt.

Die Inferenzdienste zur Erfüllbarkeit und Subsumtion sind auf die Terminologie einer Wissensbasis beschränkt. In einem DL-System sind auch Fragestellungen bezüglich der Individuen von Interesse, die durch die A-Box \mathcal{A} inferenziert werden. Man spricht von einem **Instanztest**, wenn man feststellen möchte, ob ein Individuum a Instanz eines Konzepts C ist, also ob $a : C$ konsistent ist. Auch hier ist die Analogie der Erfüllbarkeit vertreten: $a : C$ ist konsistent $\Leftrightarrow C$ ist erfüllbar.

Eine A-Box ist konsistent bzgl. einer T-Box, gdw. es eine Interpretation gibt, die sowohl Modell von der A-Box als auch von der T-Box ist. Um die Konsistenz einer A-Box überprüfen zu können, müssen alle geforderten Individuen aus den Konzepten widerspruchsfrei mit einer Interpretationsfunktion instanziiert werden. Ist dies möglich, so ist

die A-Box konsistent und man bezeichnet die Interpretationsfunktion als Modell für die Terminologie. Um bei einer konsistenten A-Box zu erfahren, ob ein Individuum $a \in \mathcal{A}$ Instanz eines Konzepts C ist, kann folgende Äquivalenz ausgenutzt werden: $a : C$ ist konsistent $\Leftrightarrow \{a : \neg C\} \cup \mathcal{A}$ ist inkonsistent.

Im weiteren Verlauf werden hauptsächlich Subsumtionsbeziehungen analysiert, weil das Problem der Subsumtion auf viele andere Inferenzmethoden, wie Konzepterfüllbarkeit, A-Box-Konsistenz, als auch Instanzchecking polynomiell reduziert werden kann. Sie ist für die Hierarchisierung der Terminologie von fundamentaler Bedeutung.

2.3.1 Tableau-Methodik

Die Tableau-Methode ist ein bekanntes syntaktisches Verfahren, um die Gültigkeit eines Ausdrucks zu beweisen. Die Idee ist, dass man die Unerfüllbarkeit eines Ausdrucks C entscheidet indem man systematisch versucht, ein Modell von C zu konstruieren. Ist dies erfolgreich, ist C erfüllbar, ansonsten nicht.

Die Tableau-Methode stellt ein indirektes Verfahren dar, weil die Erfüllbarkeit eines Ausdrucks derart bewiesen wird, dass die Negation des Ausdrucks als widersprüchlich abgeleitet wird (*engl.*: proof by refutation).

Die Tableau Methode konstruiert einen Baum, in dem jeder Knoten mit einer Formel markiert ist. Ein Pfad von der Wurzel zu einem Blatt stellt die Konjunktion aller Formeln der Knoten entlang des Pfades dar; eine Verzweigung stellt eine Disjunktion dar. Durch sukzessive Anwendung der Tableau-Erweiterungsregeln wird der Baum aufgebaut. In einem Tableau ist der Pfad für ein Ausdruck C abgeschlossen, wenn entlang des Pfades sowohl C als auch $\neg C$ auftritt, oder wenn *False* auftritt. Aufgrund der Konjunktion der Ausdrücke entlang des Pfades wird somit ein Widerspruch hergeleitet.

Ein Tableau heißt abgeschlossen, wenn alle seine Pfade abgeschlossen sind. Ein Tableau-Beweis für einen Ausdruck C ist ein abgeschlossenes Tableau für $\neg C$. Die Auswahl der Regeln bei der Erweiterung sind z.T. nichtdeterministisch.

Zum Verständnis können die Tableau-Erweiterungsregel mit konkreten Beispielen in [33] nachgeschlagen werden. Der erste Tableau-Algorithmus zur Resolution in der \mathcal{ALC} -Sprache wurde von Schmidt-Schauss im Jahre 1991 eingeführt. Details zur Realisierung in \mathcal{ALC} und des Blocking-Verfahrens können in [34] repetiert werden. Im gleichen Jahr wurde von Baader die Tableau-Methode um zwei Erweiterungen (mit Zahlenrestriktionen und konkreten Bildbereichen) ergänzt. Im Jahr 1996 wurde das Verfahren durch Sattler auch auf transitive Relationen erweitert.

Inzwischen existieren hoch-optimierte Tableau-Algorithmen für eine Vielzahl von beschreibungslogischen Sprachen, die auch im Vergleich mit klassischen Beweisern mithalten können und effektiv eingesetzt werden.

2.4 \mathcal{AL} -Sprachfamilie

Eine minimale Sprache mit praktischem Nutzen mit dem Namen \mathcal{AL} -Beschreibungslogik (*engl.*: Attributive Language) wurde im Jahre 1991 von Schmidt-Schauß und Smolka ein-

geführt und war ein großer Schritt in der Entwicklung von Beschreibungslogiken, weil sich relativ komplexe Sachverhalte mit einer geringen Anzahl an Konstruktoren beschreiben ließen.

Man verwendet als mögliche Sprachbezeichner $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$.

Nicht alle möglichen Namen, die durch 4 Konstruktoren generiert werden können, bezeichnen auch verschiedene Sprachen. Rein theoretisch sind zwar 16 Sprachbezeichnungen möglich, jedoch erhält man insgesamt 8 Sprachen, die nicht äquivalent zueinander sind. Z.B. ist $\mathcal{ALUE} = \mathcal{ALC}$. Dieser Rückschluss folgt aus der Semantik:

$C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ und $\exists r.C \equiv \neg(\forall r.\neg C)$.

Infolgedessen kann die Vereinigung \mathcal{U} und die existentielle Quantifikation \mathcal{E} mit Komplementen ausgedrückt werden. Weitere bewiesene Äquivalenzen können aus [35] entnommen werden.

Die Konventionen und Notationen für die Generierung von Konzepten, Rollen und Individuen in \mathcal{AL} und \mathcal{AL} -Extensionen sind in Abb.2.2 abgebildet. Hierbei deuten die mit „+“ markierten Felder auf die verfügbaren Konstruktoren innerhalb der jeweiligen Sprache hin.

Die Basissprache \mathcal{AL} und Erweiterungen

	\mathcal{AL}	\mathcal{FL}^-	\mathcal{FL}_0	\mathcal{ALU}	\mathcal{ALF}	\mathcal{ALN}	\mathcal{ALC}
\top	+	+	+	+	+	+	+
$C \sqcap D$	+	+	+	+	+	+	+
$C \sqcup D$	-	-	-	+	-	-	-
$\neg A$	+	-	-	+	+	+	+
$\neg C$	-	-	-	-	-	-	+
\perp	+	-	-	+	+	+	+
$\forall R.C$	+	+	+	+	+	+	+
$\exists R.\top$	+	+	-	+	+	+	+
$\exists R.D$	-	-	-	-	+	-	-
$(\geq n R)$	-	-	-	-	-	+	-
$(\leq n R)$	-	-	-	-	-	+	-

Abbildung 2.2: Tabellarische Darstellung der zugelassenen Konstruktoren in der Basissprache \mathcal{AL} und dessen Extensionen. Graphik aus [36]

Folgende Abb.2.3 soll die einzelnen Zusammenhänge der erweiterten Formen innerhalb der \mathcal{AL} -Sprachfamilie vermitteln.

Da eine vollständige Ausarbeitung der \mathcal{AL} -Sprachfamilie hinsichtlich der Subsumtionskomplexität den Rahmen dieser Arbeit sprengen würde, wird auf [35] und [37] verwiesen. Hierbei werden strukturelle Subsumtionstests und Subsumtions-Algorithmen mit signifi-

\mathcal{AL} -Sprachfamilie (Ausdrucks-mächtigkeit)

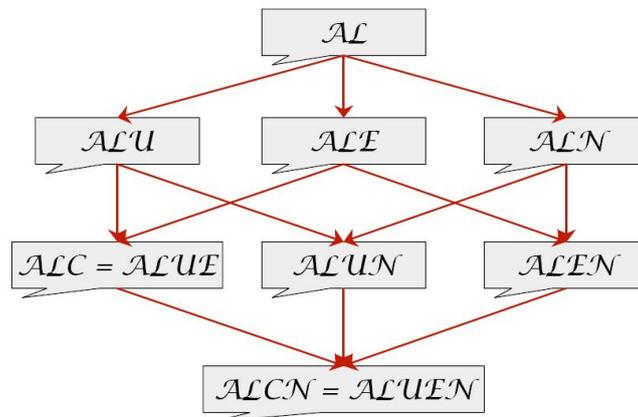


Abbildung 2.3: Graphische Darstellung der zugelassenen Konstruktoren in der Basissprache \mathcal{AL} und dessen Extensionen. Graphik aus [36]

kanten Komplexitäten für diverse Sprachen innerhalb der \mathcal{AL} -Sprachfamilie vorgestellt.

2.4.1 Auszug der Komplexitätsergebnisse für Subsumtion und Erfüllbarkeit in der \mathcal{AL} -Sprachfamilie

Bemerkungen zur Tab.2.4: Auf der linken Seite der Sprachbezeichnung stehen die Konzeptausdrücke und auf der rechten Seite die Rollenausdrücke. Die aufgelisteten Ergebnisse sind ein Auszug aus [37].

Da medizinische Ontologiedatenbanken in der \mathcal{EL} -Sprachfamilie gehandhabt und nicht in \mathcal{AL} realisiert werden, werden im weiteren Verlauf hauptsächlich \mathcal{EL} -Sprachkonstrukte vorgestellt und analysiert. Jedoch muss man bedenken, dass die Sprachen \mathcal{AL} und \mathcal{EL} nicht grundsätzlich verschieden sind oder keinerlei Zusammenhänge aufweisen. Es sollte klar sein, dass alle komplexitätstheoretisch entwickelten und bewiesenen Subsumtions- und Entscheidbarkeitsbetrachtungen in \mathcal{AL} die Entstehung als auch die Entwicklung der \mathcal{EL} -Sprache fundamental beeinflusst haben. Letztendlich können schon einige Unterschiede in der Auswahl der konzept- und relationsbildenden Strukturen die Sprache entscheidend verändern.

Beispielweise bezeichnet man \mathcal{FL}_0 als das Gegenstück von \mathcal{EL} , weil außer dem Allquantor und Existenzquantor die Sprachen keine weiteren Unterschiede aufweisen.

Subsumtion in PTIME: <ul style="list-style-type: none"> • $\mathcal{AL}(\circ)$ • $\mathcal{FL}^-(\sqcap)$ • $(\sqcap, \exists r.C, \{Instanzen\})(\sqcap, ^{-1})$ 	Nicht entscheidbar: <ul style="list-style-type: none"> • $\mathcal{FL}^-(\circ, =)$ • $\mathcal{ALCN}(\circ, \sqcap, ^{-1})$ mit GCIs • $\mathcal{ALCN}(\circ, \sqcup)$ mit GCIs
EXPTIME: <ul style="list-style-type: none"> • $\mathcal{AL}()$ mit GCIs • $(\sqcap, \sqcup, \neg, \exists r.C, \forall r.C, \geq, name.C, \geq, name^{-1}.C)(\sqcup, \circ, *, ^{-1}, id())$ • $\mathcal{SHIQ}^{-1,+}(\sqsubseteq)$ 	PSPACE: <ul style="list-style-type: none"> • $\mathcal{ALC}()$ • $\mathcal{ALEN}()$ • $\mathcal{ALCN}\mathcal{R}()$ • $\mathcal{ALE}()$ mit zyklischer T-Box
NP und co-NP: <ul style="list-style-type: none"> • $\mathcal{AL}(\sqcap)$ • $\mathcal{ALU}()$ • $\mathcal{ALN}^{-1}()$ Erfüllbarkeit in PTIME • $\mathcal{FL}^-(\sqcap, \circ)$ • $\mathcal{AL}()$ mit azyklischer T-Box 	NEXPTIME: <ul style="list-style-type: none"> • $\mathcal{ALC}^{-1}()$ mit konkretem Wertebereich und azyklischer T-Box • $\mathcal{ALC}(\sqcup, \sqcap, \neg)$

Tabelle 2.4: \mathcal{AL} -Sprachfamilie: Komplexitätsergebnisse für Subsumtion und Erfüllbarkeit

2.5 \mathcal{EL} -Sprachfamilie

Ein großer Schritt in der Forschung der leichtgewichtigen Konzeptbeschreibungssprachen (*engl.*: Lightweight Languages) ist die Herleitung der Basissprache \mathcal{EL} , die mit folgenden Konstruktoren definiert ist:

Die \mathcal{EL} -Sprache benutzt ausschließlich den Topkonstruktor, den Konjunktions- und den Existenzquantor zur Realisierung einer Terminologie. Auch hier gilt, wie bei den \mathcal{AL} -Sprachen auch, dass Erweiterungen innerhalb der \mathcal{EL} -Sprachfamilie kenntlich gemacht werden.

Der entscheidende Grund, warum ausgerechnet die \mathcal{EL} -Sprachfamilie sich für große medizinische Ontologiedatenbanken durchgesetzt hat, ist auf ihre **polynomielle Subsumtionskomplexität** zurückzuführen. Der Einfachheit, bedingt durch geringe Konstruktionsvarianten, ist es zu verdanken, dass der Aufbau der Subsumtionshierarchie einer vollständigen \mathcal{EL} -Terminologie „tractable“ ist, d.h. sich polynomiell zur Größe der Ter-

Name	Syntax	Semantik
Konzeptname	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Rollenname	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Top-Konzept	\top	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Konjunktion	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existentielle Restriktion	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{x \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

Tabelle 2.5: Syntax und Semantik von \mathcal{EL}

minologie verhält. Deshalb wird die \mathcal{EL} -Sprachfamilie für diverse große medizinische Ontologien, wie SNOMED CT, Gene Ontology (GO) oder Galen Medical Knowledge Base (Galen) erfolgreich eingesetzt.

Zwar benutzen alle genannten medizinischen Ontologien die \mathcal{EL} -Sprache als Fundament, jedoch sind die folgenden Unterschiede hinsichtlich der verwendeten Extensionen zu beachten [38]:

- Termkonstruktionen in Galen erfolgen in einer erweiterten \mathcal{EL} -Sprache, die zusätzlich die Verwendung von GCIs und transitiven Rollen zulässt [39]. Aufgrund der beiden Extensionen bezeichnet man die verwendete Terminologie in Galen, als C-Box.
- Die Gene Ontology Terminologie kennzeichnet sich durch ihre azyklische T-Box und durch die Verfügbarkeit von transitiven Rollen in \mathcal{EL} aus [40].
- SNOMED CT benutzt \mathcal{EL}^+ mit einer azyklischen Terminologie [41].

Die medizinische Ontologie SNOMED CT wird in Kap.3 aus praktischer und theoretischer Sicht detailliert behandelt.

In den nächsten Abschnitten werden mögliche Erweiterungen innerhalb der \mathcal{EL} -Sprachfamilie vorgestellt und auf ihre Subsumtionskomplexität untersucht. Durch diese theoretische Herangehensweise wird der aktuellste Forschungsstandpunkt der momentan möglichen Extensionsmöglichkeiten mit ihren Auswirkungen auf die Subsumtionskomplexität wiedergeben. Denn das Ziel der Arbeit ist nicht nur die in SNOMED CT implementierte Beschreibungslogik \mathcal{EL}^+ vorzustellen, sondern eine umfassende Analyse der \mathcal{EL} -Sprachfamilie zu führen, damit das Spektrum für weitere Entwicklungsmöglichkeiten in SNOMED CT ausgeweitet und der vollständige Wissenstandpunkt thematisiert wird.

Durch die ständigen Erweiterungen in \mathcal{EL} , ist es den DL-Forschern gelungen, die aktuelle \mathcal{EL}^{++} -Sprache zu entwickeln, die die polynomielle Subsumtionskomplexität trotz vielzähliger Extensionen aufrechterhält.

2.5.1 Subsumtionsalgorithmus und -komplexität in \mathcal{EL}^{++}

Eine erweiterte und nützliche Form der Basissprache \mathcal{EL} stellt die praxistaugliche \mathcal{EL}^{++} dar. Trotz vier weiterer Extensionen (siehe Tab.2.6) bleibt die polynomielle Subsumtionskomplexität erhalten.

Name	Syntax	Semantik
Instanzmenge	$\{a\}$	$\{a^{\mathcal{I}}\}$
Konkrete Domäne	$p(f_1, \dots, f_k)$ mit $p \in \mathcal{P}^{\mathcal{D}_j}$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y_1, \dots, y_k \in \Delta^{\mathcal{D}_j} : f_i^{\mathcal{I}}(x) = y_i \text{ mit } 1 \leq i \leq k \wedge (y_1, \dots, y_k) \in p^{\mathcal{D}_j}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ \dots \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \dots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

Tabelle 2.6: Die Extensionen von \mathcal{EL}^{++}

Durch die Hinzunahme einer bewussten Instanzmenge $\{a\}$ kann ein konkretes Individuum beim Anwenden eines Subsumtions- oder Erfüllbarkeitsalgorithmus gezielt inferenziert werden.

Der konkrete Domänenkonstruktor (*engl.*: concrete domain constructor) stellt eine Schnittstelle zu einer sog. konkreten Domäne dar. Man benutzt diese Art von Konstruktoren, um bspw. konkrete Integer oder Strings ansprechen und einbinden zu können.

Formal ist eine konkrete Domäne \mathcal{D} ein Paar $(\Delta^{\mathcal{D}}, \mathcal{P}^{\mathcal{D}})$ mit $\Delta^{\mathcal{D}}$ als Konzeptmenge und $\mathcal{P}^{\mathcal{D}}$ als die Menge für Prädikatennamen. Jedes $p \in \mathcal{P}$ ist verbunden an eine Stelligkeit $n > 0$ und an eine Extension $p^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$. Um nun eine Schnittstelle zwischen der Domäne und der konkreten Domäne realisieren zu können, braucht man die Menge von Merkmalen, die sogenannte Feature-Menge N_F (*engl.*: set of feature names). Die Interpretationsfunktion $\cdot^{\mathcal{I}}$ einer Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ bildet jeden Feature-Namen $f \in N_F$ auf eine partielle Funktion $f^{\mathcal{I}}$ von $\Delta^{\mathcal{I}}$ zu $\bigcup_{1 \leq i \leq n} \Delta^{\mathcal{D}_i}$ ab.

In Tab.2.6 steht p für ein Prädikat aus einer konkreten Domäne \mathcal{D} mit f_1, \dots, f_k als Feature-Namen. Eine Beschreibungslogik in \mathcal{EL}^{++} kann mit mehreren konkreten Domänen $\mathcal{D}_1, \dots, \mathcal{D}_n$ gleichzeitig realisiert werden, so dass $\Delta^{\mathcal{D}_i} \cap \Delta^{\mathcal{D}_j} = \emptyset$ für $1 \leq i < j \leq n$. Wenn mehrere konkrete Domänen wie $\mathcal{D}_1, \dots, \mathcal{D}_n$ in Anspruch genommen werden, werden diese auch mit $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ gekennzeichnet.

Bsp.: Um Patienten von anderen abgrenzen zu können, die eine bestimmte Körpertemperatur in Grad Celsius aufweisen, wäre folgende Anwendung eines konkreten Domänenkonstruktors sehr hilfreich:

$Patient_{32_35} = Patient \sqcap \geq_{33} (has_temperature) \sqcap \leq_{35} (has_temperature)$, mit $has_temperature$ als Feature-Name und \geq_{33} und \leq_{35} als unäre Prädikate aus einer konkreten Domäne \mathcal{D} . Das Konzept $Patient_{32_35}$ repräsentiert alle Patienten, die eine Körpertemperatur zwischen 32 und 35 aufweisen. Im Zusammenhang mit der Körpertemperatur kann diese Art von Konzepttermkonstruktion für gezielte medizinische Analysen eingesetzt werden, um bspw. absehbare Krankheiten bei vorhandener Hypothermie zu erforschen oder um bakterielles Wachstum bei bestimmter Körpertemperatur zu analysieren.

Wenn alle Bedingungen für \mathcal{EL}^{++} ohne den konkreten Domänenkonstruktor zugelassen sind, spricht man von der beschreibungslogischen Sprache \mathcal{EL}^+ , die in der SNOMED CT

Terminologie Anwendung findet.

Bei angehegender Ausarbeitung des Subsumtionsalgorithmus wird ersichtlich, dass das polynomielle Verhalten trotz der Erweiterung auf eine C-Box durch GCIs und Rollenkomposition bzw. Relationsmapping nicht beeinflusst wird. Wenn man jedoch die beschränkte Form des Relationsmappings (RI) aus Tab.2.6 derart erweitern würde, in dem beliebig viele Rollen konkateniert und mit beliebig vielen konkatenierten Rollen durch eine Inklusion in Verbindung gebracht wären, also $r_1 \circ \dots \circ r_k \sqsubseteq s_1 \circ \dots \circ s_l$ mit $r_1, \dots, r_k, s_1, \dots, s_l$ als Rollen, würde dies zur Folge haben, dass EL unentscheidbar wird [42]. Aus diesem Grund ist nur die beschränkte Form der Inklusion von kompositionierten Rollen (RI) für weitere Betrachtungen relevant.

Um das polynomielle Subsumtionsverhalten in \mathcal{EL}^{++} durch einen Polynomialzeit-Algorithmus aufzeigen zu können, muss jede GCI in der C-Box \mathcal{C} in eine gewisse Normalform transformiert werden. Um eine Normalform erreichen zu können, bezeichnet man $BC_{\mathcal{C}}$ als die kleinste Menge von Konzeptbeschreibungen, die das Topkonzept \top , alle Konzeptnamen in \mathcal{C} und alle Konzeptbeschreibungen der Form $\{a\}$ oder $p(f_1, \dots, f_k)$ in \mathcal{C} beinhaltet.

Eine C-Box \mathcal{C} ist in Normalform, gdw.:

1. alle GCIs eine von den folgenden Formen aufweisen:
 $C_1 \sqsubseteq D, C_1 \sqsubseteq \exists r.C_2, C_1 \sqcap C_2 \sqsubseteq D, \exists r.C_1 \sqsubseteq D$
mit $C_1, C_2 \in BC_{\mathcal{C}}$ und $D \in BC_{\mathcal{C}} \cup \{\perp\}$
2. alle Rolleninklusionen die Form $r \sqsubseteq s$ oder $r_1 \circ r_2 \sqsubseteq s$ haben.

Es wurde in [38] gezeigt, dass die Transformation von einer C-Box \mathcal{C} in eine normalisierte C-Box \mathcal{C}' in linear Zeit erfolgt, so dass die Größe der normalisierten C-Box \mathcal{C}' im Vergleich zu der C-Box \mathcal{C} linear anwächst. Es gilt: $C \sqsubseteq_{\mathcal{C}} D$ gdw., $A \sqsubseteq_{\mathcal{C}'} B$, wobei $\mathcal{C}' = \mathcal{C} \cup \{A \sqsubseteq C, D \sqsubseteq B\}$ ist und A, B neue Konzeptnamen darstellen.

Sei \mathcal{C} eine C-Box in Normalform, die nun klassifiziert werden soll. Der Algorithmus berechnet die Subsumtionsbeziehungen zwischen allen Paaren von Konzeptnamen, die in \mathcal{C} auftreten, d.h. die C-Box \mathcal{C} wird vollständig klassifiziert. Sei $R_{\mathcal{C}}$ die Menge von allen Rollennamen, die in \mathcal{C} auftreten.

Der Algorithmus berechnet:

- die Abbildung \mathbf{S} von $BC_{\mathcal{C}}$ auf eine Untermenge von $BC_{\mathcal{C}} \cup \{\top, \perp\}$ und
- die Abbildung \mathbf{R} von $R_{\mathcal{C}}$ auf eine binäre Relation in $BC_{\mathcal{C}}$.

Mithilfe dieser Abbildungen sollen implizite Subsumtionsbeziehungen sukzessiv explizit werden, indem die Abbildungen folgende Bedeutung suggerieren:

- (I1) $D \in S(C)$ impliziert, dass $C \sqsubseteq_{\mathcal{C}} D$ und
(I2) $(C, D) \in R(r)$ impliziert, dass $C \sqsubseteq_{\mathcal{C}} \exists r.D$.

Im Algorithmus sind diese Abbildungen wie folgt initialisiert:

- $S(C) := \{C, \top\}$ für jedes $C \in BC_{\mathcal{C}}$
- $R(r) := \emptyset$ für jedes $r \in R_{\mathcal{C}}$.

Das Ziel ist nun diese Mengen $S(C)$ und $R(r)$ für jedes $C \in BC_{\mathcal{C}}$ und jedes $r \in R_{\mathcal{C}}$ mit den Ausführungsregeln (*engl.*: completion rules) sukzessiv zu erweitern, bis keine der Ausführungsregeln (CR) mehr anwendbar sind und der Algorithmus terminiert. Am Ende beschreiben die Mengen $S(C)$ und $R(r)$ vollständig alle Verbindungen zu anderen Konzepten und Rollen.

Die Liste der Ausführungsregeln (CR)

- **CR1** Wenn $C' \in S(C), C' \sqsubseteq D \in \mathcal{C}$, und $D \notin S(C)$, dann $S(C) := S(C) \cup \{D\}$.
- **CR2** Wenn $C_1, C_2 \in S(C), C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{C}$, und $D \notin S(C)$, dann $S(C) := S(C) \cup \{D\}$.
- **CR3** Wenn $C' \in S(C), C' \sqsubseteq \exists r.D \in \mathcal{C}$, und $(C, D) \notin R(r)$, dann $R(r) := R(r) \cup \{(C, D)\}$.
- **CR4** Wenn $(C, D) \in R(r), D' \in S(D), \exists r.D' \sqsubseteq E \in \mathcal{C}$, und $E \notin S(C)$, dann $S(C) := S(C) \cup \{E\}$.
- **CR5** Wenn $(C, D) \in R(r), \perp \in S(D)$, und $\perp \notin S(C)$, dann $S(C) := S(C) \cup \{\perp\}$.
- **CR6** Wenn $\{a\} \in S(C) \cap S(D), C \rightsquigarrow_R D$ und $S(D) \not\subseteq S(C)$, dann $S(C) := S(C) \cup S(D)$.
- **CR7** Wenn $con_j(S(C))$ in \mathcal{D}_j unentscheidbar ist und $\perp \notin S(C)$, dann $S(C) := S(C) \cup \{\perp\}$.
- **CR8** Wenn $con_j(S(C))$ die Aussage $p(f_1, \dots, f_k) \in BC_{\mathcal{C}}$ in \mathcal{D}_j impliziert und $p(f_1, \dots, f_k) \notin S(C)$, dann $S(C) := S(C) \cup \{p(f_1, \dots, f_k)\}$.
- **CR9** Wenn $p(f_1, \dots, f_k), p'(f'_1, \dots, f'_k) \in S(C), p \in \mathcal{P}^{\mathcal{D}_j}, p' \in \mathcal{P}^{\mathcal{D}_l}, j \neq l, f_s = f'_t$ für einige s, t und $\perp \notin S(C)$, dann $S(C) := S(C) \cup \{\perp\}$.
- **CR10** Wenn $(C, D) \in R(r), r \sqsubseteq s \in \mathcal{C}$ und $(C, D) \notin R(s)$, dann $R(s) := R(s) \cup \{(C, D)\}$.
- **CR11** Wenn $(C, D) \in R(r_1), (D, E) \in R(r_2), r_1 \circ r_2 \sqsubseteq r_3 \in \mathcal{C}$, und $(C, E) \notin R(r_3)$, dann $R(r_3) := R(r_3) \cup \{(C, E)\}$.

Einige der Regeln benutzen Abkürzungen, die geklärt werden müssen.

Die Regel CR6 benutzt die $\rightsquigarrow_R \subseteq BC_{\mathcal{C}} \times BC_{\mathcal{C}}$ Relation, die wie folgt definiert ist:

$C \rightsquigarrow_R D$ gilt, gdw. es $C_1, \dots, C_k \in BC_{\mathcal{C}}$ gibt, so dass

- $C_1 = C$ oder $C_1 = \{b\}$ für einige Individuen b ,
- $(C_j, C_{j+1}) \in R(r_j)$ für einige $r_j \in R_C (1 \leq j < k)$,
- $C_k = D$ gilt.

Wörtlich formuliert deckt die (\rightsquigarrow_R) -Relation den Fall ab, in der zwei Konzeptbeschreibungen C, D derart klassifiziert sind, dass mehrere Konzeptbeschreibungen C_1, \dots, C_k eine subsumierende Verkettung innerhalb der Relationsrümpfe suggerieren, die durch die (\rightsquigarrow_R) -Relation explizit erkannt werden soll.

Beispiel: Sei $k = 4$

$C \rightsquigarrow_R D$ gilt, gdw.

- $C_1 = \mathbf{C}$
- **j=1** $(C_1, C_2) \in R(r_1) \Rightarrow C_1 \sqsubseteq_C \exists r_1.C_2$
- **j=2** $(C_2, C_3) \in R(r_2) \Rightarrow C_2 \sqsubseteq_C \exists r_2.C_3$
- **j=3** $(C_3, C_4) \in R(r_3) \Rightarrow C_3 \sqsubseteq_C \exists r_3.C_4$
- $C_4 = \mathbf{D}$

Die Regeln CR7 und CR8 führen die Notation mit $con_j(S_i(C))$ ein und benutzen die Erfüllbarkeit und Implikation in einer konkreten Domäne. Wenn p ein Prädikat einer konkreten Domäne \mathcal{D}_j ist, dann betrachtet man die \mathcal{EL}^{++} -Konzeptbeschreibung mit $p(f_1, \dots, f_n)$ als eine prädikatenlogische Formel mit f_1, \dots, f_n als Variablen.

Durch diese boolsche Herangehensweise kann man nun entscheiden, ob solch ein boolscher Ausdruck in \mathcal{D}_j erfüllbar ist, oder ob solch ein Ausdruck einen anderen Ausdruck in \mathcal{D}_j impliziert.

Für eine Menge Γ von $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ -Konzeptbeschreibungen und $1 \leq j \leq n$ definiert man $con_j(\Gamma)$ wie folgt:

$$con_j(\Gamma) := \bigwedge_{p(f_1, \dots, f_k) \in \Gamma \text{ mit } p \in \mathcal{P}^{\mathcal{D}_j}} p(f_1, \dots, f_k)$$

Damit wir annehmen können, dass die Regeln CR7 und CR8 in Polynomialzeit berechnet werden, muss gewährleistet werden, dass die Erfüllbarkeit und Implikation in konkreten Domänen $\mathcal{D}_1, \dots, \mathcal{D}_n$ ebenso in Polynomialzeit entscheidbar sind. Aus diesem Grund werden zusätzliche Bedingungen gestellt, damit der Algorithmus auch für Fälle mit konkreten Domänen die Subsumtionshierarchie in Polynomialzeit aufbauen kann.

Eine konkrete Domäne \mathcal{D} ist *p-zulässig* (engl.: p-admissible), gdw.

1. Erfüllbarkeit und Implikation in \mathcal{D} in Polynomialzeit entscheidbar sind.
2. \mathcal{D} *convex* ist: Wenn eine Konjunktion von Atomen der Form $p(f_1, \dots, f_k)$ eine Disjunktion dieser Atome impliziert, dann impliziert es ebenso eins von seinen Disjunktionen.

Die gesamte Anzahl an ausgeführten Regeln (CR), um eine normalisierte $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ C-Box \mathcal{C} mit p-zulässigen konkreten Domänen $\mathcal{D}_1, \dots, \mathcal{D}_n$ vollständig zu klassifizieren, ist polynomiell beschränkt und jede einzelne Regelanwendung wird in Polynomialzeit berechnet.

Diese wichtige Erkenntnis ist die Konsequenz aus folgenden Punkten:

- Jede Anwendung einer CR führt dazu, dass stückweise ein Element in eine der Mengen $S(C) \subseteq BC_C \cup \{\top, \perp\}$ oder $R(C, D) \subseteq BC_C \times BC_C$ eingefügt wird.
- Die Kardinalität von BC_C ist polynomiell beschränkt in der Größe von \mathcal{C} .
- Die Relation \rightsquigarrow_R kann in Polynomialzeit berechnet werden, indem man die Berechnung anhand eines Graphens realisiert, in der die Erreichbarkeit von Knoten errechnet wird.
- Die konkreten Domänen sind p-zulässig.

Die nächste Aussage soll nun zeigen, warum alle Subsumtionsbeziehungen zwischen den auftretenden Konzeptnamen in \mathcal{C} ermittelt sind, wenn der Algorithmus terminiert.

Sei S die terminierende Abbildung, die durch die Anwendung der Ausführungsregeln beim Klassifizieren einer normalisierten C-Box \mathcal{C} resultiert. Außerdem seien A und B Konzeptnamen, die in \mathcal{C} auftreten.

Dann ist $A \sqsubseteq_C B$, gdw. eins der nachfolgenden zwei Bedingungen erfüllt wird:

- $S(A) \cap \{B, \perp\} \neq \emptyset$,
- es gibt ein $\{a\} \in BC_C$, so dass $\perp \in S(\{a\})$.

Diese Behauptung ist korrekt, weil die Anwendung von (I1) und (I2) mit dem o.g. Initialzustand für S, R die Aussage erfüllt und im weiteren Verlauf des Algorithmus ebenso die Aussage bestätigt, da keine Elemente aus der Menge entnommen werden.

Wenn wir uns CR6 näher anschauen, dann gilt $C, D \sqsubseteq_C \{a\}$, wenn $\{a\} \in S(C) \cap S(D)$. Nun impliziert $C \rightsquigarrow_R D$, dass $C \sqsubseteq_C \exists r_1. \dots \exists r_{k-1}. D$ oder $\{b\} \sqsubseteq_C \exists r_1. \dots \exists r_{k-1}. D$ für einige Individuen b . Diese Herleitung ist auch der Grund, warum D in keinem Modell von \mathcal{C} leer sein kann bzw. D ist in jedem Modell von \mathcal{C} nicht leer, in der auch C nicht leer ist. Zusammengetragen wird durch $C, D \sqsubseteq_C \{a\}$ die Subsumtionsbeziehung $C \sqsubseteq_C D$ impliziert. Dies zeigt ebenso, dass die Regel CR6 fehlerfrei ist, solange (I1) gilt.

Für den ausführlichen Korrektheitsbeweis sei auf [38] verwiesen.

Zusammenfassend wurden folgende Ergebnisse in $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ hergeleitet und ausgearbeitet:

Seien $\mathcal{D}_1, \dots, \mathcal{D}_n$ p-zulässige konkrete Domänen. Dann kann die Subsumtion in $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ mit C-Boxen in Polynomialzeit berechnet werden.

Eine weitere nützliche Extensionsmöglichkeit bzgl. $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ wird in [43] vorgestellt. Hierbei wird die $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ -Sprache mit reflexiven Rollen (*engl.*: reflexive roles) $\epsilon \sqsubseteq r$ und Bereichsrestriktion (*engl.*: range restrictions) $\text{ran}(r) \sqsubseteq C$ derart erweitert, dass die Subsumtion in Polynomialzeit erfolgt, wenn zusätzlich eine syntaktische Restriktion angenommen und eingebunden wird. Es wird gezeigt, dass die Subsumtion *PSPACE* – *hard* (bzw. unentscheidbar) wird, wenn diese Restriktionsbedingung abgeschwächt (bzw. abgeschafft) wird. Zusätzlich wird bewiesen, dass die polynomielle Subsumtionseigenschaft versagt, wenn die Sprache mit symmetrischen Rollen (*engl.*: symmetric roles) erweitert wird. In diesem Fall wird das Subsumtionsproblem *EXPTIME* – *hard*.

Name	Syntax	Semantik
Domänenrestriktion (DR)	$\text{dom}(r) \sqsubseteq C$	$r^{\mathcal{I}} \subseteq C^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Bereichsrestriktion (RR)	$\text{ran}(r) \sqsubseteq C$	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C^{\mathcal{I}}$

Tabelle 2.7: Weitere Extensionen von $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$

Syntaktische Restriktion

Um die vorhandene polynomielle Subsumtionskomplexität mit Bereichsrestriktionen nicht zu verletzen, muss der terminologische Aufbau der Rolleninklusionen restringiert werden. Für eine C-Box \mathcal{C} mit den Rollennamen r, s schreibt man $\mathcal{C} \models r \sqsubseteq s$, gdw. $r = s$ oder \mathcal{C} beinhaltet Rolleninklusionen der Form: $r_1 \sqsubseteq r_2, \dots, r_{n-1} \sqsubseteq r_n$ mit $r = r_1$ und $s = r_n$, siehe Tab.2.6 .

Man notiert $\mathcal{C} \models \text{ran}(r) \sqsubseteq C$, wenn es einen Rollennamen s mit $\mathcal{C} \models r \sqsubseteq s$ und $\text{ran}(s) \sqsubseteq C \in \mathcal{C}$ gibt. Die Restriktion ist nun wie folgt zu verstehen:

Wenn $r_1 \circ \dots \circ r_n \sqsubseteq s \in \mathcal{C}$ mit $n \geq 1$ und $\mathcal{C} \models \text{ran}(s) \sqsubseteq C$, dann $\mathcal{C} \models \text{ran}(r_n) \sqsubseteq C$. Die Restriktion versichert, dass, wenn eine Rolleninklusion der Form $r_1 \circ \dots \circ r_n \sqsubseteq s$, $n \geq 1$ die interpretierte Rollenbeziehung der Form $(d, e) \in s^{\mathcal{I}}$ impliziert, führt die Bereichsrestriktion auf s dazu, dass ein neues Konzept ausschließlich ohne Aufkommen von e eingeführt wird.

Beweis- und implementierungstechnisch betrachtet, kann die Rolleninklusion der Form $\epsilon \sqsubseteq r$ aufgrund seiner Einfachheit in den bekannten Subsumtionsalgorithmus sehr leicht eingebettet werden, ohne dabei die Polynomialzeit zu gefährden. Im Gegensatz dazu stellt die Erweiterung der Sprache mit RR eine Herausforderung dar.

Denn die Frage ist einerseits, wie Bereichsrestriktionen nun in den bekannten Subsumtionsalgorithmus mit passenden Ausführungsregeln integriert werden können und andererseits, ob sie eventuell den Algorithmus derart beeinflussen können, dass die polynomielle Eigenschaft ausklingt.

In [43] wird diese Problematik sehr leicht gelöst, indem man gar nicht beabsichtigt, den bestehenden Algorithmus für die Fälle mit Bereichsrestriktionen anzupassen. Man

eliminiert die vorhandenen RR und führt anschließend den Algorithmus mit der transformierten C-Box aus. Diese Elimination von Bereichsrestriktionen erfolgt in quadratischer Laufzeit, so dass das Ermitteln aller Subsumtionsbeziehungen in der C-Box mit RR in Polynomialzeit gelöst werden kann.

Die transformierte C-Box ohne RR nimmt in der Größe nur quadratisch zu. Bei praktischen Anwendungen sind sogar nur lineare Blow-ups zu erwarten. Die ausführliche Transformation der C-Box kann mit beiliegendem Beweis in [43] repetiert werden.

Implementierungserfolge, wie das CEL-System [44] und andere praktisch-einsetzbare Systeme in [45,46] demonstrieren, dass effiziente Implementierungen von Inferenzsystemen in $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ möglich sind.

Es gibt Inferenzmaschinen, die sich weitestgehend etabliert und Akzente gesetzt haben, wie die Terminology Development Environment (TDE) aus dem Hause der Apelon Corporation, die zum Inferenzieren von \mathcal{EL}^{++} -Fragmenten eingesetzt wird.

Alle vorgestellten Ergebnisse der $\mathcal{EL}^{++}(\mathcal{D}_1, \dots, \mathcal{D}_n)$ -Sprache basieren größtenteils auf den wissenschaftlichen Ausarbeitungen [38] und [43].

2.5.2 Polynomielle Subsumtionskomplexitäten für diverse Extensionen in \mathcal{EL}

Die Subsumtionskomplexität einer beschreibungslogischen Sprache ist von vielen Faktoren abhängig. Dazu gehören die zugelassenen Konstruktoren und Quantoren, als auch die Strukturbeschaffenheit der Terminologie bzgl. der Allgemeinheit, Azyklität oder Zyklität.

Folgende zwei erweiterten \mathcal{EL} -Sprachvarianten stellen einen interessanten Ausgangspunkt für die Analyse der Subsumtionskomplexität dar:

$$\mathcal{EL}^{\cup,(\neg)}(\mathcal{D}) \text{ und } \mathcal{EL}^{\geq, \cup}$$

Die Besonderheit beider Sprachen liegt darin, dass die Subsumtion in polynomieller Zeit berechnet werden kann. Der Untersuchungsschwerpunkt liegt nun darin, wie dies durch einen in Polynomialzeit arbeitenden Algorithmus gezeigt werden kann und, ob diese polynomielle Eigenschaft für alle möglichen Varianten einer T-Box erhalten bleibt.

Im Folgenden wird für die $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ -Sprache gezeigt, dass die Subsumtion bei einer zyklischen als auch azyklischen T-Box in polynomieller Zeit gelöst werden kann.

Die Negation (\neg) steht für die primitive Negation von atomaren Konzepten, wobei die Disjunktion (\cup) auf die Veroderung der Relationen bezogen ist. \mathcal{D} beschreibt eine konkrete Domäne. Um Subsumtionsbeziehungen entscheiden zu können, betrachtet man in [47] keine Konzeptausdrücke, sondern nur atomare Konzepte, weil sie für eine vollständige Subsumtionsanalyse ausreichend sind. Denn $\mathcal{T} \models C \sqsubseteq D$ gilt, gdw. $\mathcal{T}' \models A \sqsubseteq B$ mit $\mathcal{T}' := \mathcal{T} \cup \{A \equiv C, B \equiv D\}$, wobei A und B nicht in \mathcal{T} auftreten dürfen.

Der in [47] vorgestellte Subsumtionsalgorithmus verlangt als Input eine *normalisierte* T-Box \mathcal{T} , die wie folgt aufgebaut ist:

In jeder Äquivalenz $A \equiv C \in \mathcal{T}$ ist C von der Form:

$$A \equiv \underbrace{\prod_{1 \leq i \leq k} L_i \sqcap \prod_{1 \leq i \leq l} \exists r_i . B_i \sqcap \prod_{1 \leq i \leq m} p_i(f_1^i, \dots, f_{n_i}^i)}_C$$

Hierbei werden die in C auftretenden atomaren Konzepte zur besseren Unterscheidung auch primitive Literale genannt und mit L_i gekennzeichnet. Da die $\mathcal{EL}^{\cup,(-)}(\mathcal{D})$ -Sprache die Negation von atomaren Konzepten und die Disjunktion von Relationen erlaubt, kann jedes L_i auch negiert vorliegen und jedes r_i von der Form $r_1 \cup \dots \cup r_n$ sein. Die atomaren Konzepte im Relationsrumpf werden B_i genannt.

Im Folgenden wird die in C auftretende Menge an Literalen mit $L_{\mathcal{T}}(A)$, die Menge an existentiellen Restriktionen mit $E_{\mathcal{T}}(A)$ und die Konjunktion $p_1(f_1^1, \dots, f_{n_1}^1) \wedge \dots \wedge p_m(f_1^m, \dots, f_{n_m}^m)$ mit $Con_{\mathcal{D}}(A)$ bezeichnet. Während der Normalisierung überprüft man auf unerfüllbare Konzeptkonstruktionen. Diese Untersuchung ist relativ einfach, denn ein atomares Konzept A mit der Bedingung $A \equiv C \in \mathcal{T}$ ist unerfüllbar in \mathcal{T} , gdw. eine der folgenden drei Bedingungen gilt:

1. Es gibt ein primitives Konzept P mit $\{P, \neg P\} \in L_{\mathcal{T}}(A)$.
2. $Con_{\mathcal{D}}(A)$ ist unerfüllbar.
3. Es gibt ein $\exists r . B \in E_{\mathcal{T}}(A)$, wobei B nicht erfüllbar ist.

Gesucht sei nun die Entscheidung, ob $A \sqsubseteq B$ in einer normalisierten T-Box gilt.

Wenn das atomare Konzept A in einer normalisierten T-Box \mathcal{T} unerfüllbar ist, gibt der Subsumtionsalgorithmus *YES* aus. Denn aus $A \equiv \perp$ folgt $\perp \sqsubseteq B$. Anderenfalls, wenn das atomare Konzept B unerfüllbar ist, antwortet der Subsumtionsalgorithmus mit *NO*, weil aus $B \equiv \perp$ folgt $A \sqsubseteq \perp$.

Wenn nun sowohl A als auch B erfüllbar sind, wird eine binäre Relation S^* auf die definierten Konzeptnamen in \mathcal{T} angewendet, um alle Subsumtionen der Form $A \sqsubseteq B$ zu finden und paarweise in einer Menge S abzulegen. Damit eine vollständige Hierarchisierung der T-Box \mathcal{T} erreicht werden kann, ist die Relation S^* derart definiert, dass sie bei Erfüllung der Bedingungen in Tab.2.8 iterativ Paare der Form (A, B) in die Menge S einfügt. Die Relation S^* wird durch die Identitätsrelation initialisiert.

Auf den ersten Blick erweckt die Bedingung (C1) den Eindruck, falsch formuliert worden zu sein, weil sie entgegen die Subsumtionsbeziehung aufgestellt ist. Jedoch ist diese Bedingung korrekt, weil jeweils die Kardinalität der Mengen $L_{\mathcal{T}}(A)$ und $L_{\mathcal{T}}(B)$ betrachtet wird. Je größer die Anzahl der Literale in einer Konjunktion ist, umso spezieller wird dementsprechend ein Ausdruck. Wenn das Literal A spezieller ist als das Literal B dann gilt bekannterweise $A \sqsubseteq B$ und somit auch $L_{\mathcal{T}}(B) \subseteq L_{\mathcal{T}}(A)$.

Da die Bedingungen der Relation S^* sich polynomiell zur Größe der T-Box verhalten [47], wird auch deutlich, dass dieser Algorithmus das Subsumtionsproblem in Polynomialzeit löst.

(C1)	$L_{\mathcal{T}}(B) \subseteq L_{\mathcal{T}}(A)$
(C2)	Für jedes $\exists r_B.B' \in E_{\mathcal{T}}(B)$ gibt es ein $\exists r_A.A' \in E_{\mathcal{T}}(A)$, so dass $r_A \subseteq r_B$ und $(A', B') \in S$
(C3)	$Con_{\mathcal{D}}(A)$ impliziert $Con_{\mathcal{D}}(B)$

Tabelle 2.8: $\mathcal{EL}^{\cup,(-)}(\mathcal{D})$: Bedingungen für das Einfügen von (A, B) in S

Sei S_0, \dots, S_n die Sequenz der produzierten Relationen. Um zu zeigen, dass der Algorithmus fehlerfrei ist, ist es erforderlich zu prüfen, ob bei $(A, B) \in S_i$, $i \leq n$ die Aussage $\mathcal{T} \models A \subseteq B$ gilt. Dies wird durch eine vollständige Induktion über i realisiert. Der zugehörige Vollständigkeitsbeweis kann in [47] entnommen werden. Zusammenfassend wurde folgende komplexitätstheoretische Aussage gezeigt:

Das Subsumtionsproblem in $\mathcal{EL}^{\cup,(-)}(\mathcal{D})$ mit zyklischen und azyklischen T-Boxen liegt in der PTIME-Komplexitätsklasse.

Interessanterweise bleibt dieses Ergebnis unverändert, wenn Rollenkonjunktionen der Form $(r \cap s)$ zugelassen sind, mit der Bedingung, dass sie in disjunktiver Normalform (DNF) vorliegen. Wenn die DNF-Annahme verletzt wird, wird das Subsumtionsproblem co-NP-hart.

Der Beweis für die $\mathcal{EL}^{\geq, \cup}$ -Sprache verläuft relativ analog. Wichtig hierbei ist, dass die Rollendisjunktion nur in existentiellen Restriktionen und nicht in Kardinalitätseinschränkungen erlaubt ist. Damit gezeigt werden kann, dass die Subsumtion in dieser Sprache sowohl mit zyklischen als auch azyklischen T-Boxen in polynomieller Zeit gelöst werden kann, wird der Algorithmus für die neuen Bedingungen angepasst. Der Unterschied ist, dass die T-Box nun folgende Normalform besitzt:

Für jedes $A \equiv C \in \mathcal{T}$ ist C von der Form:

$$A \equiv \underbrace{\prod_{1 \leq i \leq k} P_i \cap \prod_{1 \leq i \leq l} \exists r_i.B_i \cap \prod_{1 \leq i \leq m} (\geq n_i s_i)}_C$$

Hierbei sind P_i atomare Konzepte, Relationen r_i von der Form $r_1 \cup \dots \cup r_n$, B_i atomare Konzepte im Relationsrumpf und s_i Rollennamen. Man beschreibt die in C auftretende Menge an atomaren Konzepten als $P_{\mathcal{T}}(A)$, die Menge an existentiellen Restriktionen unverändert mit $E_{\mathcal{T}}(A)$ und die Menge an Kardinalitätseinschränkungen mit $N_{\mathcal{T}}(A)$. Die Bedingungen der $\mathcal{EL}^{\geq, \cup}$ -Sprache für das Einfügen eines Paares (A, B) in S sind in Tab.2.9 aufgelistet.

Der analoge Vollständigkeitsbeweis durch vollständige Induktion kann in [47] entnommen werden.

Zusammenfassend wurde folgende komplexitätstheoretische Aussage für das Subsumtionsproblem in $\mathcal{EL}^{\geq, \cup}$ ausgearbeitet:

(C1)	$P_{\mathcal{T}}(B) \subseteq P_{\mathcal{T}}(A)$
(C2)	Für jedes $\exists r_B.B' \in E_{\mathcal{T}}(B)$ gibt es ein $\exists r_A.A' \in E_{\mathcal{T}}(A)$, so dass $r_A \subseteq r_B$ und $(A', B') \in S$
(C3)	Für jedes $(\geq m r) \in N_{\mathcal{T}}(B)$ gibt es ein $(\geq n r) \in N_{\mathcal{T}}(A)$, so dass $n \geq m$.

Tabelle 2.9: $\mathcal{EL}^{\geq, \cup}$: Bedingungen für das Einfügen von (A, B) in S

Das Subsumtionsproblem in $\mathcal{EL}^{\geq, \cup}$ mit zyklischen und azyklischen T-Boxen liegt in der PTIME-Komplexitätsklasse.

2.5.3 Non-polynomielle Subsumtionskomplexitäten für diverse Extensionen in \mathcal{EL}

In diesem Abschnitt werden Extensionsvarianten mit generellen T-Boxen innerhalb der \mathcal{EL} -Sprachfamilie vorgestellt, deren Subsumtionsproblem nicht mehr in Polynomialzeit verarbeitet werden kann.

Atomare Negation \mathcal{EL}^{\neg}

Sei \mathcal{EL}^{\neg} die Extension von \mathcal{EL} mit Negation auf atomare Konzepte. Da diese Form eine notationale Variante der \mathcal{ALC} -Sprache darstellt, lassen sich wie aus [38] hervorgeht, die Subsumtions- und Erfüllbarkeitseigenschaft in \mathcal{ALC} auf \mathcal{EL}^{\neg} übertragen. Aus diesem Grund ist eine Extension mit atomarer Negation in \mathcal{EL} EXPTIME-vollständig. Diese Komplexität lässt sich ebenso auf die Negation von Konzepttermen C übertragen, indem eine Konzeptnegation C durch das Ersetzen von $\neg A$ zu A erreicht werden kann, wenn zwei GCIs der Form $A \sqsubseteq C$ und $C \sqsubseteq A$ hinzugefügt worden sind.

Somit gilt:

Das Subsumtions- und Erfüllbarkeitsproblem in \mathcal{EL}^{\neg} mit generellen T-Boxen ist EXPTIME-vollständig.

Disjunktion \mathcal{ELU}

Sei \mathcal{ELU} die Extension von \mathcal{EL} mit Disjunktion. Das Subsumtionsproblem in \mathcal{ELU} mit generellen T-Boxen ist EXPTIME, da \mathcal{ELU} ein Fragment von \mathcal{ALC} darstellt. Damit nun eine kleinere obere Schranke als EXPTIME erlangt werden kann, reduziert man das Erfüllbarkeitsproblem in \mathcal{EL}^{\neg} mit generellen T-Boxen auf das Subsumtionsproblem in \mathcal{ELU} mit generellen T-Boxen. Dazu sei nun A_0 ein atomares Konzept und \mathcal{T} eine generelle T-Box in \mathcal{EL}^{\neg} . Für jedes in \mathcal{T} auftretende atomare Konzept A , führt man ein neues atomares Konzept A' ein, mit der Bedingung, dass A' bisher keine Verwendung in \mathcal{T} hat. Zusätzlich wird ein weiteres atomares Konzept L aufgestellt, welches eine besondere Eigenschaft tragen wird.

Nun kann eine Reduktion anhand der Transformierung der T-Box \mathcal{T} erzielt werden. Die

generelle T-Box \mathcal{T}^* wird durch das Ersetzen von jedem Konzept $\neg A$ mit A' in \mathcal{T} und hinzufügen von folgenden zwei GCIs erreicht:

- $\top \sqsubseteq A \sqcup A'$ und $A \sqcap A' \sqsubseteq L$ für jedes $A \in N_C$ in \mathcal{T} ;
- $\exists r.L \sqsubseteq L$.

Es sei darauf hingewiesen, dass $\exists r.L \sqsubseteq L$ äquivalent ist zu $\forall r.\neg L \sqsupseteq \neg L$. Diese Tatsache ermöglicht, dass L die Rolle des Bottomkonzepts bei Gegenmodellen von $A_0 \sqsubseteq_{\mathcal{T}^*} L$ einnimmt.

Hat man eine Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \mathcal{I})$ und eine beschreibungslogische Formel F mit $(F)^{\mathcal{I}} = FALSE$, so heißt \mathcal{I} ein Gegenmodell für F .

Wenn man diese Beobachtung benutzt, ist es nicht schwer nachzuweisen, dass A_0 in \mathcal{T} erfüllbar ist, gdw. $A_0 \not\sqsubseteq_{\mathcal{T}^*} L$.

Das Subsumtionsproblem in \mathcal{ELU} mit generellen T-Boxen ist EXPTIME-vollständig.

Kardinalitätseinschränkungen mit $\mathcal{EL}^{\geq 2}$ und $\mathcal{EL}^{\leq 1}$

Sei $\mathcal{EL}^{\geq 2}$ die Extension von \mathcal{EL} mit einer unteren Kardinalitätseinschränkung der Form $(\geq 2 r)$. Das Subsumtionsproblem in $\mathcal{EL}^{\geq 2}$ ist in der *EXPTIME* Komplexitätsklasse, weil $\mathcal{EL}^{\geq 2}$ ebenso ein Fragment von \mathcal{ALC} mit Kardinalitätseinschränkungen darlegt [48]. Um eine untere Schranke aufzeigen zu können, wird das Subsumtionsproblem mit generellen T-Boxen in \mathcal{ELU} auf das Subsumtionsproblem in $\mathcal{EL}^{\geq 2}$ reduziert. Die Reduktion ist wie folgt aufgebaut:

Seien A_0 und B_0 atomare Konzepte und \mathcal{T} eine generelle T-Box in \mathcal{ELU} .

O.B.d.A wird nun angenommen, dass alle Konzeptinklusionen in \mathcal{T} folgende Form haben:

$C \sqsubseteq D, \exists r.C \sqsubseteq D, C \sqsubseteq \exists r.D, C_1 \sqcap C_2 \sqsubseteq C, C \sqsubseteq C_1 \sqcup C_2$, mit C, D, C_1, C_2 als Konzeptnamen oder \top .

Um \mathcal{T} in eine $\mathcal{EL}^{\geq 2}$ T-Box umwandeln zu können, wird jede GCI der Form $C \sqsubseteq C_1 \sqcup C_2$ mit neuen atomaren Konzepten A, B und einem neuen Rollennamen r umstrukturiert:

$C \sqsubseteq \exists r.A \sqcap \exists r.B, C \sqcap \exists r.(A \sqcap B) \sqsubseteq C_1, C \sqcap (\geq 2 r) \sqsubseteq C_2$.

Es ist zu erkennen, dass $A_0 \sqsubseteq_{\mathcal{T}} B_0$ gilt, gdw. $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$.

Das Subsumtionsproblem in $\mathcal{EL}^{\geq 2}$ mit generellen T-Boxen ist EXPTIME-vollständig.

Sei nun $\mathcal{EL}^{\leq 1}$ die Extension von \mathcal{EL} mit einer oberen Kardinalitätseinschränkung der Form $(\leq 1 r)$. Wie im Fall $\mathcal{EL}^{\geq 2}$ ist das Subsumtionsproblem in $\mathcal{EL}^{\leq 1}$ mit generellen T-Boxen von der Komplexität EXPTIME, weil $\mathcal{EL}^{\leq 1}$ ein Fragment von \mathcal{ALC} mit Kardinalitätseinschränkung ist.

Im Folgenden wird nun eine untere Schranke für das Subsumtionsproblem in $\mathcal{EL}^{\leq 1}$ aufgezeigt, indem man das Subsumtionsproblem in \mathcal{FL}_0^{tf} mit generellen T-Boxen darauf reduziert. \mathcal{FL}_0^{tf} besteht nur aus dem Konjunktionskonstruktor, universellen Quantifikator und verlangt als zusätzliche Bedingung, dass jede Rolle als totale Funktionen interpretiert werden muss. Das Subsumtionsproblem in der \mathcal{FL}_0^{tf} mit generellen T-Boxen wurde bereits von Toman und Wedell in [49] analysiert. Es wurde bewiesen, dass das Subsumtionsproblem für diese Beschreibungslogik EXPTIME-vollständig ist.

Seien A_0 und B_0 atomare Konzepte und \mathcal{T} eine generelle T-Box in \mathcal{FL}_0^{tf} . Man transformiert \mathcal{T} in eine generelle $\mathcal{EL}^{\leq 1}$ -T-Box \mathcal{T}^* , indem man einerseits auf der linken Seite der GCI jedes Subkonzept $\forall r.C$ mit $\exists r.C$ ersetzt und andererseits auf der rechten Seite der GCI jedes Subkonzept $\forall r.C$ mit $(\leq 1 r) \sqcap \exists r.C$ substituiert. Dann ist $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in \mathcal{FL}_0^{tf} , gdw. $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$ in $\mathcal{EL}^{\leq 1}$.

Das Subsumtionsproblem in $\mathcal{EL}^{\leq 1}$ in generellen T-Boxen ist EXPTIME-vollständig.

Interessanterweise kann das Ergebnis von Toman und Wedell auch dafür eingesetzt werden, um zu beweisen, dass das Subsumtionsproblem in \mathcal{FL}_0 mit generellen T-Boxen EXPTIME-vollständig ist.

Nochmal zur Erinnerung: Die beschreibungslogische Sprache \mathcal{FL}_0 entspricht \mathcal{FL}_0^{tf} ohne die Beschränkung, dass die Rollen totale Funktionen sein müssen. In [38] wird gezeigt, dass $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in \mathcal{FL}_0 , gdw. $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in \mathcal{FL}_0^{tf} . Daraus lässt sich schlussfolgern, dass das Subsumtionsproblem in \mathcal{FL}_0 mit dem in \mathcal{FL}_0^{tf} übereinstimmt.

Inverse Rollen \mathcal{ELI}

Sei \mathcal{ELI} die Extension von \mathcal{EL} mit inversen Rollen. Im Folgenden wird gezeigt, dass das Subsumtionsproblem in \mathcal{ELI} mit generellen T-Boxen PSPACE-hart ist, indem das Erfüllbarkeitsproblem in $\mathcal{AL}\mathcal{E}$ mit primitiven T-Boxen darauf reduziert wird. $\mathcal{AL}\mathcal{E}$ ist die Erweiterung von \mathcal{EL} mit universeller Quantifikation und atomarer Negation. Das Erfüllbarkeitsproblem in $\mathcal{AL}\mathcal{E}$ ist PSPACE-vollständig [50].

Sei A_0 ein atomares Konzept und \mathcal{T} eine primitive T-Box in $\mathcal{AL}\mathcal{E}$. Es wird o.B.d.A. angenommen, dass \mathcal{T} nur aus folgenden Konzeptinklusionen besteht:

$A \sqsubseteq B$, $A \sqsubseteq \neg B$, $A \sqsubseteq \exists r.B$, $A \sqsubseteq \forall r.B$, mit A, B als atomare Konzepte.

Nun wird ein neues atomares Konzept L eingeführt und die generelle T-Box \mathcal{T}^* in \mathcal{ELI} definiert, die aus folgenden GCIs besteht:

- $A \sqsubseteq D$ für alle $A \sqsubseteq D \in \mathcal{T}$, wenn D ein Konzeptname oder von der Form $\exists r.B$ ist;
- $\exists r^-.A \sqsubseteq B$ für alle $A \sqsubseteq \forall r.B \in \mathcal{T}$;
- $A \sqcap B \sqsubseteq L$ für alle $A \sqsubseteq \neg B \in \mathcal{T}$;

- $\exists r.L \sqsubseteq L$.

Wie im Fall \mathcal{ELU} ist die Konzeptinklusion $\exists r.L \sqsubseteq L$ äquivalent zu $\neg L \sqsubseteq \forall r.\neg L$. Somit kann erneut garantiert werden, dass L bei Gegenmodellen von $A_0 \sqsubseteq_{\mathcal{T}^*} L$ als Bottomkonzept fungiert. Zusätzlich gilt, dass $\exists r.\neg.A \sqsubseteq B$ äquivalent ist zu $A \sqsubseteq \forall r.B$. Wenn man alle Eigenschaften zusammenfasst, kann man nachweisen, dass A_0 in \mathcal{T} erfüllbar ist, gdw. $A_0 \not\sqsubseteq_{\mathcal{T}^*} L$.

Das Subsumtionsproblem in \mathcal{ELI} in generellen T-Boxen ist PSPACE-hart.

Die exakte Komplexität für das Subsumtionsproblem ist immernoch nicht geklärt. Die beste obere Schranke, die in der DL-Forschung bekannt ist, liegt bei EXPTIME, die aus den Ergebnissen der Beschreibungslogik \mathcal{ALCI} aus dem Jahre 1994 stammen [48].

Subsumtionskomplexität in $\mathcal{EL}^{\geq,(\neg)}$

In diesem Abschnitt wird eine Beweisidee vorgestellt, wie man zeigen kann, dass die Subsumtionskomplexität in $\mathcal{EL}^{\geq,(\neg)}$ co-NP-vollständig ist. Um eine untere Komplexitätsschranke aufstellen zu können, wird das Problem der 3-Färbbarkeit von Graphen auf den Nachweis der non-Subsumtion reduziert. Das Problem der 3-Färbbarkeit von Graphen ist bekannterweise NP-vollständig [51]. Wenn eine Reduktion erfolgreich aufgestellt werden kann, ist der Nachweis von allen non-Subsumtionsbeziehungen der Form $(\not\sqsubseteq)$ NP-vollständig und somit der Nachweis von allen Subsumtionsbeziehungen in einer Terminologie co-NP-vollständig. Ein Graph ist mit drei Farben gefärbt, gdw. alle benachbarten Knoten verschiedene Farben haben.

Gegeben sei nun ein ungerichteter Graph $G = (V, E)$ mit den Eigenschaften, dass für jeden Knoten $v \in V$ ein Konzeptname P_v belegt wird, und dass eine einzige Relation r vorhanden ist.

Dann ist G 3-färbbar, gdw. $C_G \not\sqsubseteq (\geq 4 r)$, mit

$$C_G := \prod_{v \in V} \exists r.(P_v \sqcap \prod_{\{v,w\} \in E} \neg P_w) \quad \text{gilt.}$$

Intuitiv betrachtet; wenn $d \in C_G^{\mathcal{I}} (\geq 4 r)^{\mathcal{I}}$, dann hat d höchstens r -viele Nachfolger. Jeder Nachfolger r würde somit eine von drei Farben beschreiben. Man benutzt die primitive Negation in C_G , damit zwei adjazente Knoten mit gleicher Farbe ausgeschlossen werden können.

Eine obere Schranke für das Subsumtionsproblem in $\mathcal{EL}^{\geq,(\neg)}$ lässt sich auf die obere Schranke des Subsumtionsproblems in \mathcal{ALUN} zurückführen, die in der co-NP Komplexitätsklasse liegt. Für den ausführlichen Beweis sei auf [47] verwiesen.

Das Subsumtionsproblem in $\mathcal{EL}^{\geq,(\neg)}$ ist co-NP-vollständig.

\mathcal{EL} mit	ohne T-Box	azyklisch	zyklisch
$\neg C$	PSPACE	PSPACE	EXPTIME
$\neg A$	PTIME	PTIME	PTIME
$C \sqcup D$	co-NP	PSPACE	EXPTIME
Funktionalität	PTIME	co-NP	PSPACE
$(\geq n r)$	PTIME	PTIME	PTIME
$p(f_1, \dots, f_k)$	PTIME	PTIME	PTIME
$r \cap s$	PTIME	PTIME	PTIME
$r \cup s$	PTIME	PTIME	PTIME
r^-	PTIME	PSPACE	EXPTIME
r^+	co-NP	PSPACE	EXPTIME

Tabelle 2.10: Liste aller Subsumtionskomplexitäten für diverse Extensionsvarianten innerhalb der \mathcal{EL} -Sprachfamilie. Tabelle aus [47]

Die Tab.2.10 ist eine Auflistung aller Subsumtionskomplexitäten für diverse Extensionsmöglichkeiten innerhalb der \mathcal{EL} -Sprachfamilie. Die Zellen mit schwarzer Schriftfarbe bedeuten, dass das jeweilige Problem einer Komplexitätsklasse angehört, wobei die graue Schriftfarbe auf die Vollständigkeit bzgl. der Komplexitätsklasse deutet.

Die vorgestellten Ergebnisse von Extensionsvarianten innerhalb der \mathcal{EL} -Sprachfamilie sind auf die wissenschaftlichen Ausarbeitungen [38] und [47] zurückzuführen.

2.6 Unifikation in \mathcal{EL}

Die **Unifikation** ist eine Methode zur Vereinheitlichung prädikatenlogischer Ausdrücke [52]. Unter Unifikation versteht man die Ersetzung der in den Literalen vorkommenden freien Variablen durch Terme, so dass die Literale zeichenweise identisch sind. Formal ausgedrückt, wird das Gleichungssystem $\Gamma = \{c_1 = d_1, \dots, c_n = d_n\}$ nach den vorkommenden Variablen aufgelöst, so dass die zwei Argumentenlisten (c_1, \dots, c_n) und (d_1, \dots, d_n) unifiziert werden. Allgemein bezeichnet man das Ersetzen von Variablen als *Substitution*. Eine Substitution σ stellt einen *Unifikator* dar, gdw. die Äquivalenz der beiden Argumentenlisten durch die Substitution σ erfolgt. Die Anwendung eines Unifikators auf das Gleichungssystem bezeichnet man als Unifikation.

In der Regel gibt es mehrere Lösungen für ein Unifikationsproblem. Deshalb ist der kleinste gemeinsame Unifikator, auch der allgemeinste Unifikator genannt, von großem Interesse. Denn die Unifikation soll nicht allein für kompositionierte Instanzierungen erfolgen, sondern allgemein wie möglich gehalten werden.

Wenn σ *allgemeinster Unifikator* ist, dann gilt für alle anderen Unifikatoren σ_1 :

- $\exists \sigma_2 : \sigma_1 = \sigma_2 \circ \sigma$

Beispiel:

Beim Unifizieren der beiden Ausdrücke $c_1 = f(X, Y)$ und $d_1 = f(X, Z)$ sind viele Sub-

stitutionen denkbar.

$\sigma_1(X) = \sigma_1(Y) = \sigma_1(Z) = A$ ist allerdings weniger allgemein als

$\sigma_2(X) = A, \sigma_2(Y) = \sigma_2(Z) = B$, da man σ_2 durch $\sigma_3(A) = \sigma_3(B) = A$ noch zu $\sigma_1 = \sigma_3 \circ \sigma_2$ bearbeiten könnte.

Eine bekannte Methode, um Ausdrücke mit dem allgemeinsten Unifikator unifizieren zu können, ist der Robinson-Algorithmus aus [53]. Zwar ist dieser Algorithmus sehr einfach anzuwenden, jedoch ermittelt er das gewünschte Ergebnis in exponentieller Laufzeit.

Unifikationsmethoden werden im Umgang mit Beschreibungslogiken als Inferenzsysteme eingeführt, um hauptsächlich redundante Konstrukte innerhalb der DL-basierten Ontologien zu ermitteln. Bei großen DL-basierten Ontologien können Äquivalenzttests zur Vermeidung von multiplen Konzepttermen sehr aufschlussreich sein, jedoch sind Äquivalenzttests allein nicht ausreichend, um für ein konkreten Konzeptterm entscheiden zu können, ob ein weiterer Konzeptterm mit der gleichen semantischen Bedeutung existiert. Deshalb benutzt man die Unifikation von Konzepttermen, um Äquivalenzen anhand einzelner Substitutionsschritte aufdecken zu können.

Beispiel:

$\exists hatKrankheit.Husten \sqcap \exists hatKrankheit.(Husten \sqcap Schnupfen)$
 $\equiv \exists hatKrankheit.(Husten \sqcap Schnupfen)$

Der Äquivalenzttest würde diese Äquivalenz erkennen und die redundanten Konzeptterme entfernen. Wie schon erwähnt, ist ein Äquivalenzttest allein zur Ermittlung von Redundanzen nicht ausreichend, wie folgendes Beispiel zeigt:

$Frau \sqcap \exists hatKind.Frau \stackrel{?}{\equiv} Weiblich \sqcap Mensch \sqcap \exists hatKind.(Weiblich \sqcap Mensch)$

Der Äquivalenzttest würde in diesem Beispiel keine syntaktische Äquivalenz erkennen, es sei denn man substituiert den Konzeptnamen *Frau* mit dem Konzeptterm *Weiblich* \sqcap *Mensch*. Durch geeignete Substitutionen können semantische Gleichheiten syntaktisch ermittelt werden. Diesen Vorgang bezeichnet man als Unifikation von Konzepttermen.

Die ersten Unifikationsansätze in Beschreibungslogiken tauchten im Jahr 1997 in [54] auf, in der die \mathcal{FL}_0 -Sprache bzgl. des Unifikationsverhalten analysiert wurde.

Die \mathcal{FL}_0 -Sprache lässt folgende Konstruktionen zu:

1. Konjunktionsoperator (\sqcap)
2. Topkonstruktor (\top)
3. Universelle Quantifikation ($\forall r.C$)

Es wurde gezeigt, dass die Unifikation in \mathcal{FL}_0 entscheidbar und ExpTime-complete ist. Für ein gegebenes \mathcal{FL}_0 -Unifikationsproblem kann zwar entschieden werden, ob eine Lösung existiert oder nicht, dennoch ist jede Prozedur an eine exponentielle Laufzeit im worst-case gebunden. Im Zusammenhang wäre es nun interessant zu erkennen, ob sich das Unifikationsverhalten bei ausdrücksschwächeren Beschreibungslogiken ändert. Bisher wurde in diesem Gebiet nur das \mathcal{EL} -**Matching** erforscht, welches ein Spezialfall der Unifikation darstellt, in der eine der beiden Ausdrücke keine Variablen enthält.

Beispiel:

Der Konzeptterm *C* enthält keine Variablen. *X* sei eine Konzeptvariable in *D*. Ein Matching des Konzepts $D = \exists hatKrankheit.X$ mit dem Konzept

$C = \exists \text{hatKind.Frau} \sqcap \exists \text{hatKrankheit.Grippe}$ besitzt einen minimalen matcher $\sigma = \{X \mapsto \text{Grippe}\}$.

Matching ist ebenso eine Möglichkeit uninteressante Teile komplexer Beschreibungen insbesondere in großen Wissensbasen zu filtern.

Aus [55] geht hervor, dass das Entscheidungsproblem **EL-Matching** NP-vollständig ist. Erstaunlicherweise ist das Matching-Entscheidungsproblem in der ausdrucksstärkeren \mathcal{FL}_0 -Sprache „tractable“, also in polynomieller Laufzeit zur Eingabe lösbar. Um das Unifikationsverhalten in \mathcal{EL} erkunden zu können, ist das gründliche Verständnis der Substitution unabdingbar, weil die Substitution als Basisoperation der Unifikation zu Grunde liegt. In erster Linie ist es wichtig, die Menge der Konzeptvariablen N_v von der Menge der Konzeptkonstanten N_c zu unterscheiden.

Konzeptkonstanten können mit anderen Konzepten nicht substituiert werden, wobei die Konzeptvariablen zum Substituieren vorgesehen sind. Damit Variablen von Konstanten unterschieden werden können, sollte ein Standard bei der Notation gesetzt werden.

Eine **Substitution** σ ist eine Abbildung von N_v in die Menge der \mathcal{EL} -Konzeptterme. Die Abbildung auf Konzeptterme ist wie folgt zu verstehen:

- $\sigma(A) := A$ für alle $A \in N_c$
- $\sigma(\top) := \top$
- $\sigma(C \sqcap D) := \sigma(C) \sqcap \sigma(D)$
- $\sigma(\exists r.C) := \exists r.\sigma(C)$

Ein **EL-Unifikationsproblem** ist von der Form $\Gamma = \{C_1 \equiv? D_1, \dots, C_n \equiv? D_n\}$ mit $C_1, D_1, \dots, C_n, D_n$ als \mathcal{EL} -Konzepttermen.

Die Substitution σ ist ein **Unifikator** (oder Lösung) von Γ gdw. $\sigma(C_i) \equiv \sigma(D_i)$ für alle $i = 1, \dots, n$. In diesem Fall bezeichnet man das \mathcal{EL} -Unifikationsproblem Γ als **unifizierbar** (oder lösbar).

Die Lösbarkeit eines \mathcal{EL} -Unifikationsproblems intendiert stets das folgende Entscheidungsproblem: Für ein gegebenes \mathcal{EL} -Unifikationsproblem Γ entscheide, ob Γ unifizierbar ist oder nicht.

Zu einer Menge von Ausdrücken existieren gewöhnlich mehrere Unifikatoren. Damit für ein gegebenes \mathcal{EL} -Unifikationsproblem die auftretenden Unifikatoren sinnvoll verglichen werden können, benutzt man den \leq -Vergleichsoperator (*engl.*: instantiation preorder), der wie folgt definiert ist:

Sei Γ ein \mathcal{EL} -Unifikationsproblem, V die Menge der auftretenden Variablen in Γ und σ, θ zwei Unifikatoren für dieses Problem.

Es gilt $\sigma \leq \theta$ gdw. es eine Substitution λ gibt, so dass $\theta(X) \equiv \lambda(\sigma(X))$ für alle $X \in V$.

Wenn $\sigma \leq \theta$, dann ist θ **eine Instanz von** σ .

Die Substitution λ stellt keinen Unifikator dar, weil es lediglich zur Substitution benutzt wird, um eine Äquivalenz zu einem weiteren Unifikator herzustellen. Man deklariert eine

Menge von Substitutionen M als eine vollständige Menge von Unifikatoren für Γ gdw. es folgende Eigenschaften erfüllt:

1. Jedes Element in M ist ein Unifikator von Γ ;
2. Wenn θ ein Unifikator von Γ ist, dann existiert ein Unifikator $\sigma \in M$, so dass $\sigma \leq \theta$ gilt, d.h. θ ist eine Instanz von σ .

Die Unifikatormenge M von Γ heißt *minimal-vollständig*, wenn es zusätzlich folgende Aussage erfüllt:

3. Wenn $\sigma, \theta \in M$, dann impliziert $\sigma \leq \theta$ die Gleichheit $\sigma = \theta$,
d.h. verschiedene Unifikatoren sind keine Instanzen voneinander.

Die Minimalitätsbedingung dient zur Ermittlung der Menge von allgemeinsten Unifikatoren, d.h. kein Unifikator in der minimal-vollständigen Unifikatormenge kann mit einer Substitution in ein anderen Unifikator überführt werden, es sei denn sie sind gleich.

Die Kardinalität der minimal-vollständigen Unifikatormenge legt den Unifikationstyp für ein gegebenes Unifikationsproblem fest.

Ein \mathcal{EL} -Unifikationsproblem ist vom Typ

- *unitary*, gdw. die minimal-vollständige Unifikatormenge die *Kardinalität 1* hat.
- *finitary*, gdw. die minimal-vollständige Unifikatormenge *endlich* ist.
- *infnitary*, gdw. die minimal-vollständige Unifikatormenge *unendlich* ist.
- *zero*, gdw. keine minimal-vollständige Unifikatormenge existiert.

Die Menge von allen Unifikatoren für ein gegebenes \mathcal{EL} -Unifikationsproblem ist immer *vollständig*. Diese Menge ist üblicherweise unendlich und besteht aus vielen redundanten Unifikatoren, da sie Instanzen von anderen Unifikatoren darstellen.

Für ein unitary oder finitary \mathcal{EL} -Unifikationsproblem, können alle Unifikatoren mit einer endlich-vollständigen Menge repräsentiert werden, wobei dies bei Unifikationsproblemen vom Typ zero oder infnitary nicht möglich ist. Wenn es für ein Problem eine endlich-vollständige Menge von Unifikatoren M gibt, dann kann durch iteratives Entfernen von redundanten Elementen in M die endliche *minimal-vollständige* Menge von Unifikatoren ermittelt werden. Für eine unendlich-vollständige Menge von Unifikatoren ist das iterative Entfernen von redundanten Elementen unendlich, so dass letztendlich die resultierende Menge nicht vollständig wird. Genau diese Tatsache kennzeichnet Unifikationsprobleme vom Typ zero.

Der Unterschied zwischen infitary und zero ist, dass beim Unifikationsproblem vom Typ zero jede vollständige Menge von Unifikatoren verschiedene Unifikatoren σ, θ beinhalten muss, so dass $\sigma \leq \theta$ gilt.

Um zeigen zu können, dass die Sprache \mathcal{EL} einen Unifikationstyp von zero hat, zeigt man, dass es ein \mathcal{EL} -Unifikationsproblem vom Typ zero gibt. Ob dies tatsächlich der Fall ist,

wird durch die nähere Betrachtung der Äquivalenzeigenschaften in \mathcal{EL} ersichtlich.

Ein \mathcal{EL} -Konzeptterm kann durch die Assoziativitäts- und Kommutativitätseigenschaft von \sqcap in einen äquivalenten Ausdruck reduziert werden.

Die Reduktionsregeln sind wie folgt zu verstehen:

- $C \sqcap \top \rightarrow C$ für alle \mathcal{EL} -Konzeptterme C
- $A \sqcap A \rightarrow A$ für alle Konzeptnamen $A \in N_{con} = N_v \cup N_c$
- $\exists r.C \sqcap \exists r.D \rightarrow \exists r.C$ für alle \mathcal{EL} -Konzeptterme C, D mit $C \sqsubseteq D$

Ein \mathcal{EL} -Konzeptterm liegt *reduziert* vor, wenn keine Reduktionsregel anwendbar ist. Insbesondere gilt:

Theorem 1

Seien C, D zwei \mathcal{EL} -Konzeptterme und \widehat{C}, \widehat{D} die reduzierten Formen. Dann ist $C \equiv D$, gdw. \widehat{C} bis auf Assoziativität und Kommutativität der Konjunktion identisch ist mit \widehat{D} .

Anhand der obigen Assoziativitäts- und Kommutativitätseigenschaft der Konjunktion können relevante Rückschlüsse, im Hinblick auf das Konzipieren eines Unifikationsproblems in \mathcal{EL} vom Typ *zero*, gezogen werden:

Lemma 1

Wenn C, D *reduzierte* \mathcal{EL} -Konzeptterme sind, so dass $\exists r.D \sqsubseteq C$, dann ist C entweder \top oder von der Form $C = \exists r.C_1 \sqcap, \dots, \sqcap \exists r.C_n$ mit $n \geq 1$; C_1, \dots, C_n sind reduziert und paarweise verschieden bzgl. Subsumtion; und $D \sqsubseteq C_1, \dots, D \sqsubseteq C_n$.

Umgekehrt, wenn C, D \mathcal{EL} -Konzeptterme sind, so dass $C = \exists r.C_1 \sqcap, \dots, \sqcap \exists r.C_n$ und $D \sqsubseteq C_1, \dots, D \sqsubseteq C_n$, dann $\exists r.D \sqsubseteq C$.

2.6.1 \mathcal{EL} -Unifikationsproblem vom Typ 0

Seien X, Y Variablen.
 Das \mathcal{EL} -Unifikationsproblem $\Gamma := \{X \sqcap \exists r.Y \equiv? \exists r.Y\}$ ist vom Unifikationstyp *zero*.

Beweis:

Eine umfangreiche Beweisführung lässt sich vermindern, indem man zeigt, dass für das Problem jede vollständige Menge von Unifikatoren redundant ist, d.h. es beinhaltet zwei verschiedene Unifikatoren, die bzgl. des \leq -Operators vergleichbar sind.

Sei M eine vollständige Menge von Unifikatoren für Γ . Ein Unifikator von M muss die Variable X auf einen \mathcal{EL} -Konzeptterm nicht äquivalent zu \top oder $\exists r.\top$ abbilden. Wenn man die Substitution τ wie folgt definiert: $\tau(X) = \exists r.A$ und $\tau(Y) = A$; stellt man fest, dass τ ein Unifikator von Γ darstellt.

Für die Menge M bedeutet dies wiederum, dass ein Unifikator σ mit $\sigma \leq \tau$ existieren muss, d.h. es gibt eine Substitution λ , die durch die Instanziierung die beiden Unifikatoren ineinander überführt, formal: $\exists r.A = \tau(X) = \lambda(\sigma(X))$.

Es ist zwar klar, dass die Bedingung durch den \top -Konstruktor erfüllt werden kann, in dem $\sigma(X) \equiv \top$ ($\sigma(X) \equiv \exists r.\top$) die Bedingung $\lambda(\sigma(X)) \equiv \top$ ($\lambda(\sigma(X)) \equiv \exists r.\top$) implizieren würde, so dass $\exists r.A \equiv \top$ ($\exists r.A \equiv \exists r.\top$) gilt. Jedoch ist das Ziel, nicht die Bedingung wahllos zu erfüllen, sondern eine gescheite Substitution zu finden, um die Redundanz eines weiteren Unifikators zu beweisen. Deshalb wird $\sigma \in M$ mit $\sigma(X) \not\equiv \top$ und $\sigma(X) \not\equiv \exists r.\top$ festgelegt. O.B.d.A wird angenommen, dass $C := \sigma(X)$ und $D := \sigma(Y)$ reduziert vorliegen. Da σ ein Unifikator von Γ ist, gilt $\exists r.D \sqsubseteq C$.

Diese Schlussfolgerung erhält man durch die Substitution der Variablen in Γ , also $X \sqcap \exists r.Y \equiv \exists r.Y \Rightarrow \sigma(X) \sqcap \exists r.\sigma(Y) \equiv \exists r.\sigma(Y) \Rightarrow C \sqcap \exists r.D \equiv \exists r.D$ gilt, gdw. $\exists r.D \sqsubseteq C$. Aus Lemma 1 geht hervor, dass C von der Form $C = \exists r.C_1, \dots, \exists r.C_n$ mit $n \geq 1$ ist. C_1, \dots, C_n sind reduziert und paarweise nicht subsumierbar, und $D \sqsubseteq C_1, \dots, D \sqsubseteq C_n$.

Man benutzt nun σ , um einen neuen Unifikator $\hat{\sigma}$ wie folgt zu konstruieren:

$$\hat{\sigma}(X) := \exists r.C_1 \sqcap \dots \sqcap \exists r.C_n \sqcap \exists r.Z$$

$$\hat{\sigma}(Y) := D \sqcap Z$$

mit Z als eine neue Variable, die nicht in C und D auftaucht.

Lemma 1 impliziert, dass $\hat{\sigma}$ ein Unifikator von Γ ist.

Zu zeigen: $\hat{\sigma} \leq \sigma$.

Für diesen Sachverhalt wird eine Substitution λ definiert, die Z auf C_1 abbildet und keine weiteren Variablen verändert.

Somit haben wir $\lambda(\hat{\sigma}(X)) = \exists r.C_1 \sqcap \dots \sqcap \exists r.C_n \sqcap \exists r.C_1 \equiv \exists r.C_1 \sqcap \dots \sqcap \exists r.C_n = \sigma(X)$ und $\lambda(\hat{\sigma}(Y)) = D \sqcap C_1 \equiv D = \sigma(Y)$. Die zweite Äquivalenz gilt aufgrund $D \sqsubseteq C_1$.

Da M vollständig ist, gibt es einen Unifikator $\theta \in M$, so dass $\theta \leq \hat{\sigma}$.

Aus der Transitivität von $\theta \leq \hat{\sigma} \leq \sigma$ gilt $\theta \leq \sigma$.

Da σ und θ Elemente in M sind, ist der Beweis abgeschlossen, wenn $\sigma \neq \theta$ gezeigt ist.

Angenommen sei $\sigma = \theta$ (Widerspruchsbeweis). Dann würde auch $\sigma \leq \hat{\sigma}$ gelten und folglich mit einer Substitution μ die Äquivalenz $\mu(\sigma(X)) \equiv \hat{\sigma}(X)$ erfüllen, d.h.

$$\exists r.\mu(C_1) \sqcap \dots \sqcap \exists r.\mu(C_n) \equiv \exists r.C_1 \sqcap \dots \sqcap \exists r.C_n \sqcap \exists r.Z \quad (\text{Bed.1})$$

Nochmal zur Erinnerung: C_1, \dots, C_n sind reduziert und paarweise nicht subsumierbar. Da $\sigma(X) = \exists r.C_1 \sqcap \dots \sqcap \exists r.C_n$ ebenso reduziert vorliegt und nicht äquivalent zu $\exists r.\top$ ist, kann keines der Konzeptterme C_1, \dots, C_n äquivalent zu \top sein.

Z ist ein Konzeptname, der nicht in C_1, \dots, C_n vorkommt. Aus diesem Grund liegt $\exists r.C_1 \sqcap \dots \sqcap \exists r.C_n \sqcap \exists r.Z$ reduziert vor. Offensichtlich ist jede reduzierte Form für $\exists r.\mu(C_1) \sqcap \dots \sqcap \exists r.\mu(C_n)$ eine Konjunktion von höchstens n -vielen existentiellen Quantifikationen. Demzufolge wird aus Theorem 1 ersichtlich, dass die formulierte Äquivalenz in (Bed.1) nicht erfüllbar sein kann, weil die Kardinalität der Operationen beim reduzierten Ausdruck zum Widerspruch führt.

Zusammenfassend kann man sagen, dass M zwei verschiedene Unifikatoren σ, θ mit der Eigenschaft $\theta \leq \sigma$ beinhaltet. Da M eine vollständige Menge von beliebig vielen Unifikatoren für Γ darstellt, kann es für dieses Unifikationsproblem keine minimal-vollständige Unifikatormenge geben. \square

2.6.2 Entscheidungsprozedur für die \mathcal{EL} -Unifikation

Bevor die Entscheidungsprozedur in seinen Einzelheiten vorgestellt werden kann, müssen noch einige Notationen eingeführt und geklärt werden.

Ein \mathcal{EL} -Konzeptterm wird als *Atom* bezeichnet, gdw. es ein

- Konzeptname (also Konzeptkonstante oder Konzeptvariable) oder
- eine existentielle Restriktion $\exists r.D$

ist.

Ein \mathcal{EL} -Konzeptterm besteht immer aus einer Konjunktion von Atomen, wobei die leere Konjunktion als \top aufgefasst wird. Dies ist die maximale Ausnutzung der vorhandenen \mathcal{EL} -Konstruktoren, die für die Konstruktion eines \mathcal{EL} -Konzeptterm vorgesehen ist.

$At(C)$ entspricht der Menge von Atomen, die in einem Konzeptterm C vertreten sind. Die Menge wird wie folgt induktiv ermittelt:

- $C = \top \Rightarrow At(C) := \emptyset$
- C ist ein Konzeptname $\Rightarrow At(C) := \{C\}$
- $C = \exists r.D \Rightarrow At(C) := \{C\} \cup At(D)$
- $C = C_1 \sqcap C_2 \Rightarrow At(C) := At(C_1) \cup At(C_2)$

Konzeptnamen und existentielle Restriktionen $\exists r.D$, mit D als Konzeptname oder \top , werden *flat Atome* genannt. Ein \mathcal{EL} -Unifikationsproblem Γ ist *flat*, gdw. es Gleichungen folgender Form beinhaltet:

- $X \equiv? C$ mit X als Variable und C als nicht-variables flat Atom.
- $X_1 \sqcap \dots \sqcap X_m \equiv? Y_1 \sqcap \dots \sqcap Y_n$ mit $X_1, \dots, X_m, Y_1, \dots, Y_n$ als Variablen.

Jedes \mathcal{EL} -Unifikationsproblem Γ kann in polynomieller Laufzeit in ein *flat* Unifikationsproblem Γ' in \mathcal{EL} transformiert werden, in dem neue Konzeptvariablen eingeführt und \top eliminiert wird, so dass Γ lösbar ist, wenn Γ' lösbar ist.

Der Unifikator σ heißt *reduziert*, gdw. für alle Konzeptvariablen X in Γ der substituierte \mathcal{EL} -Konzeptterm $\sigma(X)$ keine Konzeptvariablen enthält und reduziert vorliegt. Somit kann der reduzierte Ausdruck nur aus Konzeptkonstanten und existentiellen Restriktionen bestehen. Offensichtlich ist Γ *unifizierbar*, wenn es einen *reduzierten Unifikator* hat. Wenn für ein Unifikationsproblem Γ ein reduzierter Unifikator σ gegeben ist, dann ist $At(\sigma)$ die Menge aller Atome, die durch die Substitution σ über alle auftretenden Variablen resultiert, d.h. $At(\sigma) = At(\sigma(X_1)) \cup \dots \cup At(\sigma(X_n))$ mit X_1, \dots, X_n als alle auftretenden Konzeptvariablen in Γ .

Man bezeichnet die Elemente in $At(\sigma)$ als die Atome von σ .

Lemma 2

Seien C, D, D' \mathcal{EL} -Konzeptterme, so dass $D \sqsubset D'$, wobei C reduziert vorliegt und mindestens ein D enthält. Wenn man in C' alle Vorkommnisse von D mit D' ersetzt und C erhält, dann ist $C \sqsubset C'$. Da D' einen allgemeineren Konzeptterm als D darstellt, wird nach der Ersetzung C' allgemeiner als C . Der ausführliche Beweis erfolgt durch die Induktion über die Länge des Konzeptterms C . Die Einzelheiten des vollständigen Beweises können aus [56] entnommen werden.

Man benutzt die \sqsubset -Beziehung zwischen \mathcal{EL} -Konzepttermen, um eine wohlfundierte Ordnung von reduzierten Unifikatoren zu definieren. Eine *strikte partielle Ordnung* ist eine *transitive* und *irreflexive* Relation. Eine strikte partielle Ordnung \succ auf einer Menge M heißt wohlfundiert, gdw. es keine unendlich absteigende Kette $C_1 \succ C_2 \succ \dots$ in M gibt. Für ein gegebenen reduzierten Unifikator σ auf Γ , sei $S(\sigma)$ die Multimenge über alle \mathcal{EL} -Konzeptterme $\sigma(X)$, wobei X sich über alle vorkommenden Variablen erstreckt.

Sei Γ_{Var} die Menge über alle vorkommenden Variablen im Unifikationsproblem Γ .

Wenn $\Gamma_{Var} = \{X_1, X_2, \dots, X_n\}$, dann gilt $S(\sigma) = \sigma(X_1) \cup \sigma(X_2) \cup \dots \cup \sigma(X_n)$.

Die Multimenge $S(\sigma)$ verfügt über keine Variablen, weil der Unifikator σ reduziert vorliegt. Um zwei reduzierte Unifikatoren σ, θ vergleichen zu können, definiert man $\sigma \succ \theta$, gdw. die Multimengenerweiterung zwischen $S(\sigma) \succ_{mul} S(\theta)$ gilt.

Die Multimengenerweiterung \succ_{mul} auf Multimengen über Unifikatoren U ist allgemein definiert durch::

$S_1 \succ_{mul} S_2 \Leftrightarrow S_1 \neq S_2$ und $\forall u \in U : [S_2(u) > S_1(u) \Rightarrow \exists u' \in U : (u' \succ u \text{ und } S_1(u') > S_2(u'))]$ Wenn \succ wohlfundiert ist, ist auch \succ_{mul} wohlfundiert [57].

Ein reduzierter Unifikator σ für Γ heißt *minimal*, gdw. es keinen weiteren reduzierten Unifikator θ für Γ gibt, so dass $\sigma \succ \theta$ gilt.

Da die \succ Relation wohlfundiert ist, ist jedes Unifikationsproblem lösbar, wenn es einen minimal reduzierten Unifikator hat. Insbesondere gilt für ein *flat* \mathcal{EL} -Unifikationsproblem Γ mit γ als minimal reduzierten Unifikator für Γ ; wenn C ein Atom von γ ist, dann existiert ein nicht-variables Atom D in Γ , so dass $C \equiv \gamma(D)$. Der Beweis aus [56] dient zur Vervollständigung der Aussage.

Aus den ausgearbeiteten theoretischen Ergebnissen kann nun ein nicht-deterministischer Algorithmus zur Entscheidung eines \mathcal{EL} -Unifikationsproblems entwickelt werden, siehe Tabelle 2.11.

Zur Illustration sei folgendes Fallbeispiel mit richtig geratenen Mengen S_{Var} aufschlussreich:

$$\Gamma = \{Vater \sqcap X \sqcap \exists hatKind.Student \equiv Z \sqcap Y \sqcap \exists hatKind.Y\}$$

Wörtlich formuliert: Ein studierender Vater, dessen Kind ebenso ein Student ist.

Zu1) Für jede Variable aus dem Unifikationsproblem wird eine endliche Menge geraten.

$$S_X = \{Vater, Student, \exists hatKind.Student\}$$

$$S_Y = \{Student\}$$

$$S_Z = \{Vater, \exists hatKind.Y\}$$

Zu2) Die Variable Z ist direkt abhängig von der Variable Y , da Y in einem Atom von

Nicht-deterministischer Algorithmus zur Entscheidung eines \mathcal{EL} -Unifikationsproblems Γ
1. Für jede Variable X aus Γ , rate eine endliche, möglicherweise leere Menge S_X bestehend aus nicht-variablen Atomen von Γ . Jedoch sind existentielle Restriktionen mit variablem Rumpf zulässig.
2. Man bezeichnet die Variable X als <i>direkt abhängig</i> von der Variable Y , wenn Y in einem Atom von S_x auftaucht. Wenn es eine Variable gibt, die von sich selbst abhängig ist, dann gibt der Algorithmus FAIL aus. Ansonsten gibt es eine <i>strikte lineare Ordnung</i> $>$ der Variablen in Γ , d.h. $X > Y$ gilt, wenn X von Y abhängig ist.
3. Man definiert die Substitution σ entlang der linearen Ordnung $>$ wie folgt: <ul style="list-style-type: none"> • Wenn X die letzte Variable bzgl. $>$ ist, dann beinhaltet S_X keine anderen Variablen. Man definiert $\sigma(X)$ als die Konjunktion der Elemente in S_X mit \top als die leere Konjunktion. • Angenommen $\sigma(Y)$ wird für alle Variablen mit $Y < X$ definiert. Dann besteht S_X nur aus Variablen Y für die schon $\sigma(Y)$ definiert ist. Wenn S_X leer ist, dann definiert man $\sigma(X) := \top$. Ansonsten wird beim Fall $S_X = \{D_1, \dots, D_n\}$ entsprechend $\sigma(X) := \sigma(D_1) \sqcap \dots \sqcap \sigma(D_n)$ definiert.
4. Prüfe, ob die Substitution σ im vorherigen Schritt ein Unifikator für Γ darstellt. Wenn die Aussage zutrifft, dann geben σ aus; sonst wird FAIL ausgegeben.

Tabelle 2.11: Nicht-deterministischer Algorithmus zur Entscheidung eines Unifikationsproblems Γ in \mathcal{EL}

S_Z auftaucht. Deshalb ist $Z > Y$ und X ist eine unabhängige Variable.

Zu3) Man definiert die Substitution σ für die Variablen wie folgt:

$$\sigma(Y) = Student$$

$$\begin{aligned} \sigma(X) &= \sigma(Student) \sqcap \sigma(Vater) \sqcap \sigma(\exists hatKind.Student) = \\ & Student \sqcap Vater \sqcap \exists hatKind.\sigma(Student) = Student \sqcap Vater \sqcap \exists hatKind.Student \end{aligned}$$

$$\begin{aligned} \sigma(Z) &= \sigma(\exists hatKind.Y) \sqcap \sigma(Vater) = \exists hatKind.\sigma(Y) \sqcap Vater = \\ & \exists hatKind.Student \sqcap Vater \end{aligned}$$

Zu4) Um prüfen zu können, ob im vorherigen Schritt die Substitution σ einen Unifikator für Γ darstellt, wird das Unifikationsproblem nach der Substitution mit σ auf Äquivalenz untersucht.

$$Vater \sqcap \underbrace{Student \sqcap Vater \sqcap \exists hatKind.Student}_X \sqcap \exists hatKind.Student \equiv$$

$$\underbrace{\exists \text{hatKind. Student}}_Z \sqcap \underbrace{Vater}_Y \sqcap \underbrace{Student}_Y \sqcap \underbrace{\exists \text{hatKind. Student}}_Y$$

Durch das Entfernen doppelt vorhandener Atome ergibt sich:

$$Vater \sqcap Student \sqcap \exists \text{hatKind. Student} \equiv \exists \text{hatKind. Student} \sqcap Vater \sqcap Student$$

Da die Äquivalenz durch die Substitution σ erfüllt ist, stellt σ einen Unifikator dar und wird vom nicht-deterministischen Algorithmus als Lösung ausgegeben.

Der Algorithmus ist trivialerweise fehlerfrei, weil nur die Substitutionen ausgegeben werden, die auch Unifikatoren für das angegebene Unifikationsproblem darstellen. Der dezedierte Vollständigkeitsbeweis steht in [56].

Der Algorithmus beruht auf dem Ausprobieren möglicher Fälle und entspricht somit einer *Brute-Force-Methode*. Deshalb steigt der Aufwand an Rechenoperationen proportional zur Anzahl der zu probierenden, möglichen Lösungen an, wobei die Anzahl dieser möglichen Lösungen mit steigendem Umfang der Probleme bei deterministischer Herangehensweise *exponentiell* ansteigt.

Die \mathcal{EL} -Unifikation ist *NP-vollständig!*

Beweis: Ein Entscheidungsproblem L heißt *NP-vollständig*, gdw.:

L in der Klasse NP liegt ($L \in NP$) und gleichzeitig *NP-hart* ist ($\forall L' \in NP : L' \leq_p L$). Die letzte Bedingung bedeutet, dass jedes Entscheidungsproblem in NP durch eine polynomielle Reduktion auf das Problem L reduziert werden kann. Die Reduktion soll deterministisch in Polynomialzeit berechnet werden.

Die *NP-harte* Eigenschaft ist zurückzuführen auf die bewiesene *NP-Vollständigkeit* des \mathcal{EL} -Matchingverfahrens [58].

Um zeigen zu können, dass das Problem durch einen nicht-deterministischen Algorithmus in Polynomialzeit gelöst werden kann, wird jeder Schritt des o.g. Algorithmus komplexitätstheoretisch analysiert.

In Schritt 1 werden die Mengen S_{Var} geraten, welches in NP gelöst werden kann. Die Abhängigkeitsrelation zwischen den Variablen dient zum Ausschließen von zyklischen Strukturen und kann in Schritt 2 in polynomieller Zeit entschieden werden. Jedoch wäre der nächste Schritt mit mehr Aufwand verbunden, weil eine Variable in verschiedenen Atomen von Γ auftreten kann und demzufolge die berechnete Substitution σ eine exponentielle Größe erreichen wird.

Statt die Substitution σ explizit zu berechnen, wird in Schritt 3 σ mit einer azyklischen T-Box gleichgesetzt, die durch Konjunktion der einzelnen Atome entsteht. Denn für jede Konzeptvariable X in Γ , beinhaltet die T-Box \mathcal{T}_σ die Konzeptdefinition $X = \top$, wenn $S_X = \emptyset$ und $X = D_1 \sqcap \dots \sqcap D_n$, wenn $S_X = \{D_1, \dots, D_n\}$ mit $n \geq 1$. Statt σ zu ermitteln, wird \mathcal{T}_σ ausfindig gemacht.

Da in Schritt 2 zyklische Verhältnisse ausgeschlossen werden, folgt, dass \mathcal{T}_σ eine azyklische T-Box ist. Die Größe von \mathcal{T}_σ ist polynomiell zur Größe von Γ , d.h. Schritt 3 ist ebenso polynomiell. Wenn man die Substitution σ auf die einzelnen Variablen anwendet, sieht man, dass dieser Vorgang der Expansion in \mathcal{T}_σ entspricht, d.h. für jede Gleichung $C \equiv^? D$ in Γ , gilt $C \equiv_{\mathcal{T}_\sigma} D$, gdw. $\sigma(C) \equiv \sigma(D)$.

Deshalb kann σ als Unifikator bestätigt werden, wenn für alle Gleichungen der Form $C \equiv^? D$ in Γ die Aussage $C \equiv_{\mathcal{T}_\sigma} D$ zutrifft. Da die Subsumtion und Äquivalenz in \mathcal{EL} bei azyklischen T-Boxen in polynomieller Zeit bestimmt werden kann (siehe Kap.2.5.1), ist der Beweis nun vollständig.

Gleichheitstheorie in $SLmO$

Ursprünglich wurden Unifikationsprobleme nicht in Beschreibungslogiken analysiert, sondern vielmehr in der Gleichheitstheorie [59].

In [60] wird gezeigt, dass das Äquivalenzproblem für \mathcal{EL} -Konzeptterme dem Wortproblem in einer algebraischen Struktur mit monotonen Operatoren (*engl.*: equational theory of semilattices with monotone operators) gleichkommt.

Um die Gleichheitstheorie mit den entwickelten Ansätzen in Einklang zu bringen, zieht man die Signatur Σ_{SLmO} in Betracht. Diese Signatur besteht aus dem binären Funktionssymbol \wedge , einem konstantem Symbol 1 und aus endlich vielen unären Funktionssymbolen f_1, \dots, f_n , die durch folgende algebraische Strukturen und Eigenschaften definiert sind:

$$SLmO := \{x \wedge (y \wedge z) = (x \wedge y) \wedge z, x \wedge y = y \wedge x, x \wedge x = x, x \wedge 1 = x\} \cup \{f_i(x \wedge y) \wedge f_i(y) = f_i(x \wedge y) \mid 1 \leq i \leq n\}$$

Ein \mathcal{EL} -Konzeptterm C , welches nur durch die Rollen r_1, \dots, r_n konzipiert wird, kann anhand der Signatur Σ_{SLmO} in einen Term t_C übersetzt werden, indem jede Konzeptkonstante A mit der übereinstimmenden freien Konstante a , jede Konzeptvariable X mit der übereinstimmenden Variable x , \top mit 1 , \sqcap mit \wedge , und $\exists r_i$ mit f_i ersetzt wird.

Bsp.: Der \mathcal{EL} Konzeptterm $C = B \sqcap \exists r_3.(X \sqcap A) \sqcap \exists r_1.(\top)$ wird übersetzt in $t_C = b \wedge f_3(x \wedge a) \wedge f_1(1)$. Umgekehrt kann auch jeder Term, der durch die Signatur Σ_{SLmO} erstellt wurde, zurück in einen \mathcal{EL} Konzeptterm übersetzt werden. Deshalb gilt die Äquivalenz $C \equiv D$, wenn $t_C =_{SLmO} t_D$.

Infolgedessen entspricht jede Unifikation in \mathcal{EL} einer Unifikation in der Gleichheitstheorie mit $SLmO$ mit unveränderter Komplexität, d.h. $SLmO$ -Unifikation ist ebenso NP-vollständig und vom Typ Null.

Die vorgestellten Ergebnisse und Anmerkungen sind auf die aktuell erschienene wissenschaftliche Veröffentlichung [56] zurückzuführen.

3 SNOMED CT

3.1 Allgemeines

Eine wichtige Voraussetzung für die Funktionsfähigkeit der elektronischen Gesundheitskarte als telematische Infrastruktur ist die inhaltliche Austauschfähigkeit der gespeicherten medizinischen Daten.

Um die semantische Interoperabilität erreichen zu können, muss jedes beteiligte System in der Lage sein, Daten so auszutauschen und zu speichern, dass die genaue medizinische Bedeutung der Daten jederzeit unverfälscht verfügbar ist, und dass die Daten in eine interpretierbare, verständliche Form gebracht werden können.

Ärzte verfassen medizinische Daten, Diagnosen, Befunde, Verfahren, Medikamente überwiegend in Textform. Da textuelle Daten allein wegen der Vielfalt der Schreib- und Ausdrucksweise die semantische Interoperabilität bei der Kommunikation von maschinellen Systemen nicht gewährleisten können, wurden in Kapitel 1.1 diverse medizinische Dokumentationssysteme vorgestellt, die mit ihren unterschiedlichen semantischen Reichhaltigkeiten signifikante Wissensrepräsentationssysteme zum Ausdruck bringen.

Ziel war es zu veranschaulichen, dass die Fähigkeit Daten in unterschiedlichsten Systemen und darüber hinaus sprachunabhängig in gleicher Weise interpretieren zu können, in der Regel eine standardisierte Bezeichnungsweise voraussetzt und zwar eine Dokumentationssprache mit Codes als Notation.

Kodes erfüllen die Anforderungen an die semantische Interoperabilität in höherem Maße als jede sprachliche Repräsentation, insbesondere wenn Inhalte in unterschiedlich feiner Granulierung abgebildet werden sollen. Denn die massenhafte Verknüpfung von einzelnen Begriffen führt zu einer netzartigen Struktur, die anhand von Regeln spezifiziert werden muss, damit im System keine Widersprüche auftreten und die Konsistenz des Systems nicht verletzt wird; spricht man benutzt konsistente Ontologien zur Realisierung von Ordnungssystemen.

Ziel der Forschung ist es, ein ontologiebasiertes System zur Realisierung der Datenspeicherung auf der elektronischen Gesundheitskarte derart einzusetzen, dass letztendlich medizinische Entscheidungen durch die implementierte Ontologie auf der eGK unterstützt werden können.

Das internationale Codesystem ICD-10 aus Kapitel 1.1.2 ist im Gesundheitswesen selbstverständlich und blickt auf eine mehr als hundertjährige Geschichte zurück. Sie wurde durch immer neue Versionen kontinuierlich an die medizinischen Fortschritte angepasst und wird in der BRD standardmäßig für elektronische Dokumentationen eingesetzt. Klassifikationssysteme wie ICD-10 zeichnen sich dadurch aus, dass sie nicht jeden eigenständigen Begriff mit einem Kode versehen, sondern ähnliche Begriffe in Klassen zusammenfassen, die dann gemeinsamen Codes zugeordnet werden.

Bsp.: Angeborene Missbildungen der Trachea sind extrem selten. Aus diesem Grund wurden Stenose und Dilatation der Trachea sowie Anomalien des Trachealknorpels trotz der unterschiedlichen Intention in der Klasse Q32.1 für „Sonstige angeborene Fehlbildungen der Trachea“ zusammengefasst.

Für das eGK-Projekt können Klassifikationssysteme wie ICD-10 aufgrund ihrer genannten Unschärfe nur bedingt eingesetzt werden. Die Begriffe sind zu grob klassifiziert und haben keinen ausgeprägten semantischen Bezug zueinander, so dass eine detaillierte medizinische Entscheidungsunterstützung mit dem ICD-10 System allein aussichtslos ist.

Für sprachübergreifende Anwendungen auf der Gesundheitskarte können derart ausgerichtete Klassifikationssysteme auch nur bedingt eingesetzt werden. Denn die multilinguale semantische Interoperabilität setzt voraus, dass die Inhalte in mehreren Sprachen unmittelbar aus dem Code abgeleitet werden sollen, d.h., dass primär der Code verstanden und der medizinische Inhalt umfassend abgebildet werden soll, damit die repräsentierten Inhalte direkt in die entsprechende Sprache übertragen werden können.

Da Textanalysen im alltäglichen Gebrauch sehr aufwendig sind und die damit verbundenen Leistungen in vielen medizinischen Systemen nicht erbracht werden können, kam man in der eGK-Forschung zum Entschluss die Dokumentationssprache durch Beschreibungslogiken zu realisieren, um die medizinische interoperable Datenspeicherung durch eine konsistente Ontologie zu gewährleisten.

Die für die weitere Entwicklung der eGK und für die Einführung des elektronischen Rezepts notwendige semantische Interoperabilität erfordert nämlich eine umfassende Terminologie, mit der medizinische Befunde, Diagnosen, Medikamente, Verfahren in strukturierter und standardisierter Weise dargestellt werden können.

Dr. Frank Warda, ein Spezialist in diesem Gebiet, brachte in seinem Buch „*Elektronische Gesundheitsakten*“ folgende Meinung hervor:

„Nicht nur für elektronische Akten, sondern auch für viele weitere Prozesse wird im Rahmen der Einführung der Telematik Infrastruktur eine Nomenklatur zur Erschließung der Befunde eingeführt und eingesetzt werden müssen. Je besser die verwendete Terminologie und je strukturierter und standardisierter die Dokumentation sein wird, desto größer wird der Nutzen für alle Beteiligten“ [61].

Zu diesem Zweck rückte die vorhandene medizinische Nomenklatur SNOMED CT in den Mittelpunkt der Forschung. Denn die medizinische Nomenklatur SNOMED CT verfolgt das Ziel, medizinische Aussagen so zu indizieren, dass die inhaltlichen Elemente der Aussage vollständig erfasst werden, wodurch sehr spezielle Suchanfragen mit hohem Recall und hoher Präzision inferenziert werden können.

Zugleich können existierende Querverbindungen zu Wissenssammlungen und Literaturdatenbanken, wie zur ICD-10 oder roten Liste, benutzt und aktualisiert werden.

Ein weiterer Vorteil ist die gegenwärtige Sprachunabhängigkeit, die die angesprochene multilinguale semantische Interoperabilität voraussetzt [62].

3.2 SNOMED CT - Historische Entwicklung

SNOMED CT ist die englische Abkürzung für **S**ystematized **N**omenclature of **M**edicine **C**linical **T**erms.

SNOMED CT blickt auf eine fast vierzigjährige Vergangenheit zurück. Es gab mehrere Entwicklungen in der Forschung mit SNOMED CT, dessen Ursprung in SNOP, der **S**ystematized **N**omenclature of **P**athology, liegt.

SNOMED (ohne den begrifflichen CT Zusatz) entstand als Fortentwicklung der bereits Mitte der sechziger Jahre erschienenen SNOP, die eine mehrdimensionale Nomenklatur darstellt und für die Kodierung pathologischer Befunde eingesetzt wird. SNOP umfasst vier Dimensionen, auch Achsen genannt und zwar die Topographie (T), Morphologie (M), Ätiologie (E) und Funktionsstörung (F), mit denen pathologische Befunde detailliert dargestellt werden können. Jedoch war SNOP für klinische Anwendungen nicht ausdrucksstark genug und wurde mit zusätzlichen Achsen, insbesondere zur Beschreibung der Diagnosen (D), erweitert. Des Weiteren wurde eine Prozeduren (P) und eine Achse zur Beschreibung beruflicher Belastungen eingeführt, die anfangs als Occupation (O) eingeführt, später jedoch in Job (J) umbenannt wurde. Durch diese Erweiterungen wurde SNOP zu SNOMED.

SNOMED wird seit seiner Entstehung hauptsächlich von Mitgliedern des CAP (College of American Pathologists) [63] betreut, die im Jahr 1974 eine erste Testversion herausgaben.

Die Erweiterung zu SNOMED II wurde im Jahr 1979, gefolgt von einem Update im Jahr 1982, veröffentlicht. SNOMED ist eine kompositionelle Klassifikation vom Typ einer Facetten-Klassifikation, d.h. Krankheitsbilder werden durch die Komposition der zutreffenden Kodewerte in den verschiedenen Achsen definiert.

Bsp.: Pneumokokken-Pneumonie ist die Komposition aus den Codewerten der Achsen Diagnose (D-75070 - klinische Pneumonie), Topographie (T-28000 - Lunge). Oder auch aus T-28800 (beide Lungenflügel), Morphologie (M-40000 - Entzündung) und Ätiologie (E-25420 - Streptococcus pneumoniae). Die Kodierung war sehr aufwändig und war in der Praxis nicht durchsetzbar. Hinzu kommt die mehrdeutige Struktur innerhalb der einzelnen Achsen, so dass SNOMED in wenigen Forschungsprojekten Anwendung gefunden hat.

Friedrich Wingert hat SNOMED II für den deutschsprachigen Raum angepasst und mit einigen Erweiterungen im Jahr 1984 publiziert. 1993 wurde SNOMED V.3.0 herausgegeben, die auch als SNOMED International bekannt ist und erstmals die Veterinärmedizin einbezog. Im Jahr 1997 wurden die LOINC-Kodes [64] vollständig in SNOMED integriert. 1998 erschien SNOMED V.3.5 mit anfangs 12 Achsen. Trotz der geringen Akzeptanz und Anwendung wurde kontinuierlich an der Weiterentwicklung von SNOMED gearbeitet, so dass ein Update auf SNOMED RT (Reference Terminology) im Jahr 2000 die Möglichkeiten zur Repräsentation von Begriffsrelationen erheblich erweiterte. SNOMED RT wurde dann mit der Version 3 der in Großbritannien vom National Health Service entwickelten Read Codes (auch Clinical Terms genannt) [65] als SNOMED CT (Clinical Terms) zusammengeführt.

SNOMED CT ist das Ergebnis einer Zusammenarbeit des CAP mit dem NHS und bildet

aus dem jetzigen Standpunkt der Forschung die Basis aller weiteren Entwicklungen und Anwendungen. Ziel war die Vereinigung zweier relativ großer medizinischer Terminologien, SNOMED RT (CAP) und CTV3 (Clinical Terms Version 3), zu einer umfassenderen Terminologie, die weitestgehend alle klinischen Begriffe abdeckt und insbesondere das effiziente Speichern, Wiederfinden und Analysieren von Patientendaten ermöglicht. Die fusionierte Nomenklatur wurde letztendlich SNOMED CT (Clinical Terms) genannt. Eine spanische und eine deutsche Version folgten. Ende 2003 wurde ein Vertrag mit der NLM (National Library of Medicine) über die Abbildungen in UMLS (Unified Medical Language System) abgeschlossen, so dass SNOMED CT in das UMLS integriert wurde. Ziel des UMLS ist es, verschiedene begriffliche Ordnungssysteme derart semantisch anzureichern und miteinander zu verknüpfen, dass konzeptuelle Links zwischen Benutzeranfragen und relevanten Fachinformationen erstellt und Online-Informationssysteme besser erschlossen werden können. Seit April 2007 liegen die Rechte an SNOMED CT bei der International Healthcare Terminology Standards Development Organization (IHTSDO).

System	Jahr	Info
SNOP	1965	Basis für ICD-O
SNOMED	1974	Typ: Facetten-Klassifikation
SNOMED II	1984	Deutsche Übersetzung durch Wingert
SNOMED V. 3.0	1993	Einbeziehung der Veterinärmedizin
SNOMED V. 3.5	1998	Auftakt mit 12 Achsen
SNOMED RT	2000	(Referenz Terms)
SNOMED CT	2002	(SNOMED RT + Read Codes V. 3 (CTV3)) -> Clinical Terms
SNOMED CT	April 2002	Spanische Version
SNOMED CT	April 2003	Deutsche Version
SNOMED CT	2003	Vertrag mit NLM über Integration in UMLS
SNOMED CT	April 2007	Rechte an SNOMED CT bei der IHTSDO

Tabelle 3.1: Die historische Entwicklung von SNOMED CT [66]

Wie man aus der geschichtlichen Entwicklung entnehmen kann, müssen SNOMED und SNOMED CT deutlich voneinander unterschieden werden. Denn SNOMED CT entspricht nicht mehr einer Facetten Klassifikation, sondern ist mittlerweile eine umfassende Terminologie medizinischer Begriffe, dessen Begriffe durch diverse Relationsmethoden zueinander in Beziehung stehen.

SNOMED war ursprünglich auf Postkoordination ausgelegt, d.h. die Beziehungen zwischen den einzelnen Bezeichnungen mussten im Nachhinein vom Nutzer festgelegt werden und waren nicht durch hierarchische Strukturierung vorgegeben.

Anders sieht es jedoch bei den neueren Versionen (ab SNOMED RT) aus. Diese sind eher auf Präkoordination ausgelegt, so dass es nicht mehr erforderlich ist, alle Beziehungen im Nachhinein bestimmen zu müssen. Durch eine festgelegte Definition der hierarchischen

Beziehungen zwischen den Begriffen, wird eine Präkoordination gewährleistet, die mithilfe der Beschreibungslogik formatisiert wird.

Jedoch muss man bedenken, dass die genaue Beschreibung aller Krankheitsbilder, Verfahren, medikamentösen Therapien durch präkoordinierte Konzepte auch in einer umfassenden Terminologie kaum zu erreichen ist. Denn die Bildung aller möglichen Kombinationen von präkoordinierten Konzepten würde die Anzahl an Konzepten in einer Terminologie überdimensional vergrößern. Deshalb wird die Kombination von Konzepten bei Anwendungsfällen unterstützt (Postkoordination), um medizinisches Fachwissen präzise und eindeutig beschreiben zu können.

3.3 Grundaufbau von SNOMED CT

Die umfassende, konzept-basierte klinische Terminologie SNOMED CT zur Verschlüsselung, Speicherung, Wiedergewinnung und Auswertung klinischer Daten setzt sich zusammen aus folgenden Grundkomponenten (*engl.*: core components):

- Konzepte (concepts) und deren
- Attribute (werden in SNOMED CT über Attributsbeziehungen definiert),
- Bezeichnungen (descriptions) und
- Beziehungen (relationships) zwischen den Konzepten.

Alle Komponenten in SNOMED CT verfügen über eine eindeutige SCTID (SNOMED Clinical Terms Identifier). Mithilfe der SCTID ist ein eindeutiges Auffinden und Analysieren von jeder SNOMED CT Komponente möglich, so dass Mehrdeutigkeiten ausgeschlossen sind. Die SCTID steht in keinerlei Zusammenhang mit der Einordnung von Komponenten in der Ontologie oder einer semantischen Bedeutung, d.h. anhand der SCTID können keine hierarchischen Verhältnisse abgelesen werden. Sie ist lediglich eine fortlaufende Nummer, die einmalig und unveränderbar einer Komponente zugeordnet wird. Diese Art der Identifikationskodierung soll eine schnelle Manipulation der Hierarchiestruktur ermöglichen, ohne die Codes jedes Mal verändern zu müssen.

Eine SCTID besteht aus:

- der Item-ID, die die eigentliche ID einer Komponente darstellt.
- der Namespace-ID, die bei inhaltlichen Erweiterungen von SNOMED CT benötigt wird. Falls keine Namespace-ID definiert ist, ist die Komponente Teil des Kerninhalts.
- der Partition-ID, die im Wesentlichen angibt, ob die betreffende Komponente Teil einer Erweiterung ist und zu welchem Typ sie gehört. Folgende Ziffern kennzeichnen den Typ:
00 Concepts (Konzepte)

- 01 Descriptions (Bezeichnungen)
- 02 Relationships (Beziehungen)
- 03 Subsets (Untermengen)
- einer Kontrollnummer (Check-digit), die Fehleingaben ausschließen soll.

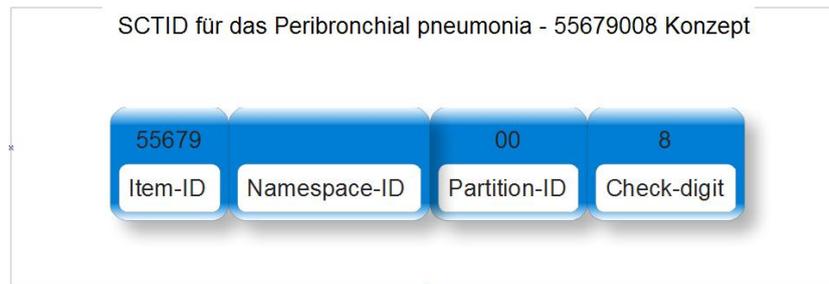


Abbildung 3.1: Graphische Darstellung einer SCTID für das Konzept „Peribronchial pneumonia - 55679008“. Einer Graphik von [66] nachempfunden

3.3.1 Konzepte und Bezeichnungen

Die SCTID für ein Konzept wird als Concept-ID bezeichnet. Ein Konzept beinhaltet mehrere textuelle Bezeichnungen (descriptions). Jedoch sind mindestens zwei Beschreibungen obligatorisch:

Fully Specified Name (FSN) dient der eindeutigen Bezeichnung eines Konzepts und bildet die terminologische Grundlage von SNOMED CT. Die FSN ist eine textuelle Beschreibung mit einem Suffix, der die Zuordnung zu einer Achse ermöglicht (mehr zu SNOMED-CT Achsen im Folgenden).

Preferred Name ist die in der Praxis bevorzugte Bezeichnung für ein Konzept. In der Regel entspricht sie öfters dem FSN ohne den achsenzuordnenden Suffix.

Enthält ein Konzept mehr als diese zwei Bezeichnungen, so handelt es sich um **Synonyme**, die keine Bedeutungsveränderung des Konzepts bewirken. Synonyme erlauben lediglich eine flexiblere Beschreibung, damit unterschiedliche Bezeichnungen für denselben Begriff benutzt und erkannt werden.

Der FSN wird exakt einem Konzept zugeordnet, so dass kein FSN mehr als ein Konzept haben darf. Jedoch ist es prinzipiell möglich, dass ein Preferred Name oder ein Synonym in mehr als nur einem Konzept vorkommen kann. Jede Bezeichnung besteht immer aus einer Description-ID und einer sprachlichen Bezeichnung.

Neben den Bezeichnungen besteht ein Konzept auch aus Beziehungen (relationships), die die Aufgabe haben, Konzepte mit anderen Konzepten in Relation zu bringen. Welche

Relationsarten in SNOMED CT benutzt werden und wie sie zu verstehen sind, werden im nächsten Abschnitt erklärt.

Für ein besseres Verständnis soll die Abb.3.2 zeigen, wie die einzelnen Komponenten beim Aufbau eines Konzepts eingesetzt und genutzt werden.

Concept-ID	
396285007	
1764202015	Bronchopneumonia(disorder)
1776258010	Bronchopneumonia
1785531017	Bronchial pneumonia
IS-A	Pneumonia
ASSOCIATED-MORPHOLOGY	Inflammation
FINDING-SITE	Lung structure
FINDING-SITE	Bronchial structure

Labels in Abbildung 3.2:

- 396285007: Concept-ID
- 1764202015: Description-ID
- Bronchopneumonia(disorder): FSN
- Bronchopneumonia: Hierarchiekennzeichnung durch Suffix
- Bronchopneumonia: Preferred Name
- Bronchial pneumonia: Synonym
- IS-A: Beziehungstyp
- Pneumonia: Konzept (Bezug)
- ASSOCIATED-MORPHOLOGY: Beziehungstyp
- Inflammation: Konzept (Bezug)
- FINDING-SITE: Beziehungstyp
- Lung structure: Konzept (Bezug)
- FINDING-SITE: Beziehungstyp
- Bronchial structure: Konzept (Bezug)

Abbildung 3.2: Schematischer Aufbau eines Konzepts in SNOMED CT am Beispiel Bronchopneumonia. Einer Graphik von [66] nachempfunden

3.3.2 Beziehungen und hierarchische Strukturen

Zusätzlich zur Concept-ID und den Bezeichnungen setzt sich ein Konzept ebenso aus mehreren Beziehungen zusammen. Jede Beziehung besteht jeweils aus der Relationship-ID, den Concept-IDs der zwei in Relation stehenden Konzepten und dem Beziehungstyp (Relationship type). Es gibt etliche Typen von Beziehungen (Relationen) zwischen zwei Konzepten. Jedoch ist ein besonderer Relationstyp von großer Wichtigkeit, da sie über diese Art von Beziehungen die eigentliche semantische Definition der Begriffe zum Ausdruck bringt und die hierarchische Strukturierung einer Terminologie festlegt. Die Rede ist von der **is-a** Beziehung, die schon in Kap.1.2.1 mit einer anderen Intension für semantische Netze eingeführt wurde. Im Vergleich zu einem semantischem Netz stellt die is-a Relation in SNOMED CT keine Instanziierung dar, sondern eher eine ako-Beziehung. Wie schon in Kap.1.2.1 erwähnt, sind sich viele Wissenschaftler in der Namensgebung der einzelnen Relationen nicht ganz einig. Aus diesem Grund sei auf Abb.3.3 verwiesen, die eine Hierarchisierung in SNOMED CT visualisiert.

Aus der oberen Graphik sieht man sehr leicht, dass die is-a Beziehungen eine Hierarchisierung der Konzepte ermöglichen. Sie entsprechen Subtyp/Supertyp-Beziehungen, so dass die Eigenschaften der Supertypen auf die Subtypen vererbt werden.

Die Frage ist nun, wie die komplette Hierarchisierung in SNOMED CT verwaltet wird. Die Antwort auf diese Frage ist sehr einfach, denn an oberster Stelle der Hierarchie befinden sich 19 Konzepte, die sogenannten **Grundkonzepte** (*engl.*: root concepts), die

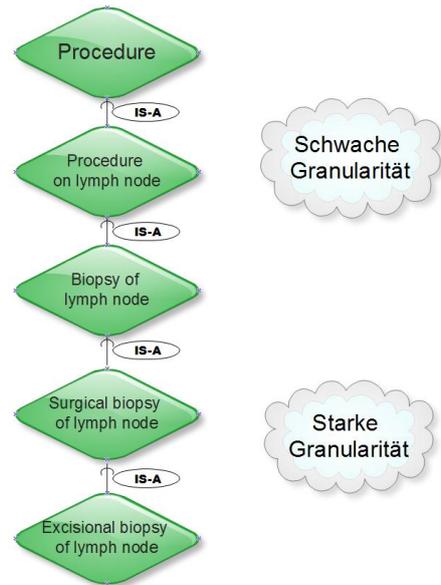


Abbildung 3.3: Schematischer Aufbau der Hierarchie in SNOMED CT am Beispiel Procedure. Einer Graphik von [67] nachempfunden

die Achsen von SNOMED CT bilden (siehe Tab.3.2). Jedes dieser Konzepte spannt sozusagen eine Achse auf.

Es ist bekannt, dass jedes Jahr SNOMED CT bearbeitet und neu veröffentlicht wird. Hierbei kam es oft vor, dass Grundkonzepte umbenannt wurden und neue hinzu gekommen sind. Denn SNOMED CT ist kein endgültiges Produkt, sondern wird von Jahr zu Jahr durch die Aufnahme weiterer Konzepte größer und genauer.

Die wichtigsten Hauptachsen sind Clinical Finding, mit den Subhierarchien Finding und Disease, und die Achse Procedure. Die übrigen Achsen dienen eher der Unterstützung. Die Achse Clinical Finding besteht aus den Konzepten, die für die Dokumentation von Funktionsstörungen und zur Erstellung von Befunden notwendig sind.

Unter Procedure sind alle Aktivitäten bzw. Verfahren im Gesundheitswesen zusammengefasst z.B. Operationen oder diagnostische Verfahren.

Eine der wichtigsten unterstützenden Achsen ist die Achse Attribute, weil sie die möglichen Attribute in Form von Konzepten enthält, über die Attributsbeziehungen definiert werden können.

Darüber hinaus existieren Achsen wie Observable Entity, die für Einheiten stehen, die beobachtet bzw. gemessen werden können, wie Blutdruck, Sehkraft oder Verwundungen etc. Diesen Konzepten können Werte zugewiesen werden, die nicht vordefiniert sind, sondern im Nachhinein von einem Benutzer kontextabhängig festgelegt werden können (Postkoordination). In der Organism Achse befinden sich v.a. Konzepte für krankheits-erregende Organismen wie Bakterien, Viren, etc.

Da eine ausführliche Ausarbeitung der einzelnen Achsen den Rahmen der Arbeit sprengen würde, wird auf [67] verwiesen, in der alle Grundkonzepte mit allen Subkonzepten

detailliert besprochen werden.

Clinical Finding (<i>Klinischer Befund</i>) <ul style="list-style-type: none"> • Finding (<i>Befund</i>) • Disease (<i>Krankheit</i>) 	Physical force (<i>Physikalische Kraft</i>)
Procedure (<i>Verfahren/Eingriff</i>)	Event (<i>Ereignis</i>)
Observable entity (<i>Beobachtbare Einheit</i>)	Environments/geographical locations (<i>Umwelt/geographische Lage</i>)
Body structure (<i>Anatomie</i>)	Social context (<i>sozialer Zusammenhang</i>)
Organism (<i>Organismus</i>)	Situation with explicit context (<i>Kontext-abhängige Situation</i>)
Substance (<i>Substanz</i>)	Staging and scales (<i>Stadieneinteilung und Messverfahren</i>)
Pharmaceutical/biologic product (<i>Pharmazeutisches/biologisches Produkt</i>)	Linkage concept (<i>Verbindende Konzepte</i>) <ul style="list-style-type: none"> • Link Assertion (<i>Relationen vom Typ HL7</i>) • Attribute (<i>Attribute</i>)
Specimen (<i>Untersuchungsmaterial</i>)	Qualifier value (<i>Konzepte für spezielle Werte</i>)
Special concept (<i>Aktives/Inaktives Konzept</i>)	Record artifact (<i>Aufgenommene Informationen</i>)
Physical object (<i>Physikalisches Objekt</i>)	

Tabelle 3.2: Die 19 Grundkonzepte in SNOMED CT - Deutsche Übersetzung bzw. Erklärung in Klammern.

Die Besonderheit der is-a Beziehung ist, dass sie nur zwischen den Konzepten innerhalb einer Achse benutzt wird.

Um eine Relation zwischen zwei Konzepten aus verschiedenen Achsen zu definieren, werden Attributsbeziehungen verwendet. Attribute sind selber Konzepte und befinden sich in der Attribute Achse.

Im Folgenden wird der SNOMED CT Browser VTSL [68] benutzt, um eine Unterscheidung zwischen Attributsbeziehungen und is-a Beziehungen an einem Beispiel repräsentieren zu können.

Das Konzept **Pneumonia** (*Lungenentzündung*) wird durch folgende Bezeichnungen und Relationen vollständig spezifiziert:

- **ConceptId: 233604007**

- Descriptions

Bezeichnungen:

- Fully Specified Name: Pneumonia (disorder)
FSN: Lungenentzündung (Störung)
- Preferred Name: Pneumonia
Bevorzugter Name: Lungenentzündung

- Relationships

Beziehungen:

- is-a Lung consolidation (disorder)
is-a Lungenverfestigung (Störung)
- is-a Pneumonitis (disorder)
is-a Pneumonitis (Störung)
- Associated morphology: Consolidation (morphologic abnormality)
assoziierte Morphologie: Verfestigung (morphologische Anomalie)
- Associated morphology: Inflammation (morphologic abnormality)
assoziierte Morphologie: Entzündung (morphologische Anomalie)
- Finding site: Lung structure (body structure)
Fundort: Lungengewebe (Anatomie)

Die Abb.3.4 ist eine graphische Darstellung des „Finding site“-Attributs aus dem o.g. Beispiel. Die Graphik soll verdeutlichen, dass eine Attributsbeziehung zwischen zwei Konzepten von verschiedenen Achsen stattfindet und alle verketteten is-a Beziehungen immer zu den entsprechenden Grundkonzepten führen.

Im Endeffekt definieren die Attribute und is-a Beziehungen die Bedeutung einzelner Konzepte. Da es unmöglich ist alle möglichen Attribute im Vorhinein festzulegen (Präkoordination), gibt es zusätzlich zu diesen Attributen, auch Attribute, deren Werte im Nachhinein festgelegt werden können (Postkoordination). Beispielsweise von einem Arzt, der einen Befund schreibt.

Neben den bereits genannten Beziehungstypen gibt es auch noch weitere, wie die Beziehung „is-part-of“ zum Festlegen von Ganzes-Teil-Beziehungen oder die Beziehung „same-as“, um sogenannte historische Beziehungen aufzubauen, die auf inaktive Konzepte verweisen.

SNOMED CT verfügt derzeit über 283.000 aktive Konzepte, 732.000 aktive Beschreibungen und 923.000 aktive Beziehungen [94], die entweder in der Konzeptionsphase (Präkoordination) oder bei der Anwendung (Postkoordination) den Konzepten zugeordnet werden. Die Präkoordination von Beziehungen zwischen Konzepten macht nur Sinn, wenn sie immer gültig ist und sich bei Anwendungen niemals ändert.

Die Darstellung postkoordinierter Konzeptkonstrukte erfolgt durch eine leicht verständlich strukturierte „Expression Language“ [69]. Man benutzt das „Konzept-Attribut-Wert“-Konstrukt, um Attributwerte durch Postkoordination einbinden zu können. Konzept und

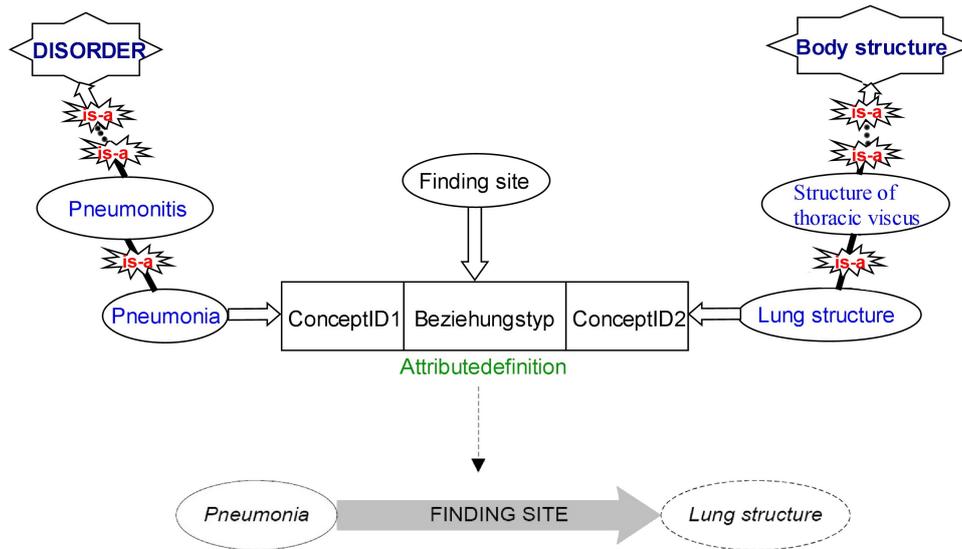


Abbildung 3.4: Beispiel Lungenentzündung: Pneumonia-FINDING SITE-Lung structure. Einer Graphik von [67] nachempfunden

Attribut werden mit einem Doppelpunkt voneinander getrennt. Der Wert des Attributs wird mit dem Attribut gleichgesetzt.

Eine Postkoordination von einer schweren Pyelonephritis auf der rechten Seite, würde man wie folgt ausdrücken:

Pyelonephritis : Lateralität = rechts : Schweregrad = schwer
 45816000 : 272741003 = 24028007 : 246112005 = 2448400

Einzelne Konzepte, die durch eine gemeinsame Eigenschaft auftreten, werden nebeneinander mit einem Additionssymbol aufgestellt. Eine Fraktur von Radius und Ulna würde lauten:

Fraktur der Radius + Fraktur der Ulna
 12676007 + 54556006

Die aufgeführten Beispiele sind aus [69].

Beim Fall einer akuten Pyelonephritis kann sowohl das präkoordinierte Konzept „acute Pyelonephritis“, als auch das postkoordinierte Konzept „Pyelonephritis : Course = Acute“ benutzt werden. Trotz des identischen Informationsgehalts, sollten unterschiedliche Darstellungen bei späteren Routineanwendungen vermieden werden.

Schulz et.al. haben in [70] auf die dabei auftretenden Probleme hingewiesen. Ziel für die zukünftige Entwicklung der SNOMED CT Terminologie sollte es sein, die Verwendung präkoordinierter Konzepte in den Vordergrund zu bringen, so dass Postkoordination nur

zur Verfeinerung und Qualifizierung des Befundes unumgänglich ist.

Für einen weiteren Einblick in den Grundaufbau von SNOMED CT empfiehlt sich der Besuch des SNOMED-CT Browsers [68] oder [71] und ein Einblick in das SNOMED CT User Guide [67].

3.3.3 SNOMED CT - Datenbank Tabellen

Alle vorhandenen Konzepte, Bezeichnungen und Beziehungen bilden den Inhalt von SNOMED CT. Sie werden in Form von Tabellen (*engl.*: tables) gespeichert, die für Implementierungen in der Regel im CSV Format bereitgestellt werden. Das Dateiformat CSV beschreibt den Aufbau einer Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten. In CSV-Dateien können Tabellen oder unterschiedlich lange Listen abgebildet werden [72].

Insgesamt bilden drei Tabellen, die sogenannten Kerntabellen (*engl.*: core tables) das inhaltliche Grundgerüst und zwar die:

1. Konzepttabelle (*engl.*: concepts table),
2. Bezeichnungstabelle (*engl.*: descriptions table) und
3. Beziehungstabelle (*engl.*: relationships table).

Jede Zeile in einer Tabelle steht für ein Konzept, eine Bezeichnung oder eine Beziehung, siehe Abb.3.5 .

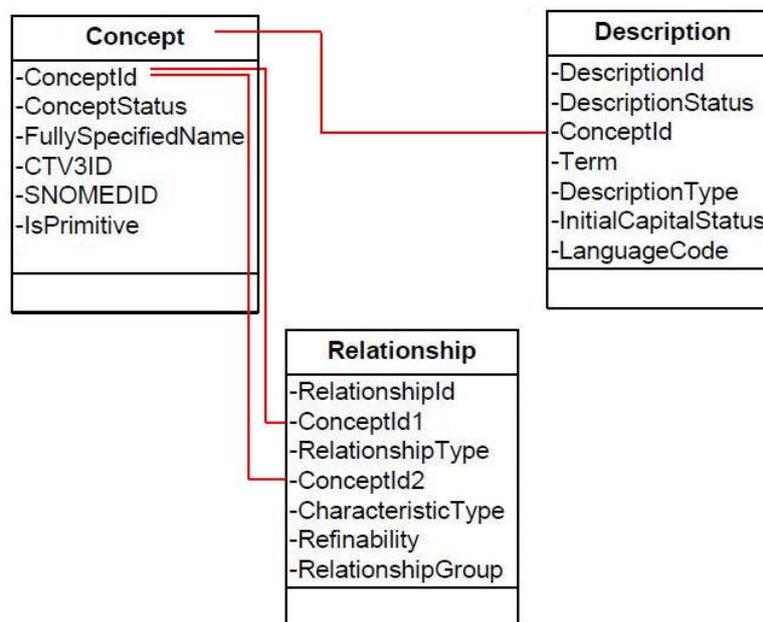


Abbildung 3.5: Core tables in SNOMED CT. Einer Graphik von [67] nachempfunden.

Die **Konzepttabelle** enthält die ConceptID als Primärschlüssel und den FSN als alternativen Schlüssel, sowie historische Verweise auf eine frühere Version SNOMED RT und den Read Code (CTV3), die sich noch aufgrund der Entstehungsgeschichte von SNOMED CT in der Tabelle befinden. Zusätzlich speichert die Konzepttabelle den Konzept-Status (mit aktiv oder nicht aktiv) und gibt an, ob es sich um ein „primitives“ Konzept handelt. Wenn Konzepte durch andere Konzepte ersetzt werden, so dass kein momentaner Bedarf besteht, bietet SNOMED CT die Möglichkeit an, bestimmte Konzepte zu deaktivieren, um das vollständige Entfernen oder Manipulieren überflüssiger Konzepte zu vermeiden. Denn Konzepte sollten innerhalb der Terminologie permanent sein. Sie können als inaktiv deklariert werden, dürfen ihre Bedeutung aber nicht ändern, selbst wenn sie inhaltlich durch die Entwicklung der Medizin irrelevant geworden sind.

Anhand einer „is-primitive“ Markierung kann man Konzepte hervorheben, die vollständig definiert sind (*engl.*: fully defined), oder die von übergeordneten Konzepten nicht deutlich abgrenzbar sind bzw. sich nicht über eigene Beziehungen von Elternkonzepten unterscheiden lassen.

Die Tab.3.3 zeigt die Darstellung eines Konzepts in der Konzepttabelle.

ConceptID	Concept Status	FSN	CTV3ID	SNOMEDID	Is-Primitive
27624003	aktiv	Chronic disease (disorder)	XUAor	DF-00003	0 = Fully defined

Tabelle 3.3: Beispiel für die Darstellung des Konzepts „Chronic disease (disorder)“ in der Konzepttabelle. Daten aus dem SNOMED Browser [71]

Die **Bezeichnungstabelle** hat als Primärschlüssel die DescriptionID und die ConceptID als Fremdschlüssel. Anhand der ConceptID kann die Zuordnung zu einem Konzept ermöglicht werden, welches durch die in DescriptionID suggerierte Bezeichnung beschrieben wird. Hinzu kommen noch die textuelle Beschreibung (*engl.*: term) und der Bezeichnungstyp (FSN, Preferred name, Synonym), sowie der Status der Bezeichnung und der Language Code, der die Sprache der Bezeichnung festlegt. Der Typ einer Bezeichnung wird in DescriptionType abgelegt.

Eine Beziehung in der **Beziehungstabelle** beinhaltet die RelationshipID als Primärschlüssel mit der jeweiligen Partition ID 02, den Relationstyp (is-a, has-causative-agent etc.) und die IDs der Konzepte zwischen denen die Beziehung festgelegt ist. Wenn eine Beziehung durch den is-a Relationstyp aufgebaut wird, gilt immer, dass das erste Konzept ein Subtyp (*engl.*: child concept) des zweiten Konzeptes ist.

Die übrigen Einträge der Beziehungstabelle können aus der Abb.3.5 entnommen werden. Die Anwendung und Bedeutung von gruppierten Relationen (*engl.*: relationship-group) werden in Kap.3.4.2 näher erläutert.

Neben diesen drei genannten Kerntabellen gibt es noch weitere Tabellen, die alle mit den

Kerntabellen in Verbindung stehen. Beispielweise stehen für die Verarbeitung von Untermengen (*engl.*: subsets) weitere Tabellen zur Verfügung. Subsets werden eingesetzt, wenn die verfügbaren Konzepte problem- bzw. aufgabenbezogen zu kleineren Mengen zusammengefasst werden.

Im Hinblick auf eine umfassende Realisierung der elektronischen Gesundheitskarte richtet sich das Augenmerk auf die Cross-Mapping-Tabellen, die für eine großflächige und umfangreiche Etablierung der eGK mit verschiedenen Zugriffsmöglichkeiten auf diverse Terminologien und Klassifikationen von großer Wichtigkeit sind.

Die Aufgabe einer Cross-Mapping-Tabelle ist es, ein SNOMED CT-Konzept auf diverse Schlüssel einer anderen Klassifikation (Bsp.: ICD-10, rote Liste) oder Terminologie abzubilden, so dass eine interdisziplinäre Analyse von medizinischen Sachverhalten erfolgen kann.

3.4 SNOMED CT aus beschreibungslogischer Sicht \mathcal{EL}^+

SNOMED CT ist eine umfangreiche medizinische Ontologie, die die beschreibungslogische Variante \mathcal{EL}^+ aus der \mathcal{EL} -Sprachfamilie benutzt. Man spricht von \mathcal{EL}^+ , gdw. der konkrete Domänenkonstruktor aus \mathcal{EL}^{++} entfällt.

\mathcal{EL}^+ bildet den formalen fundamentalen Aufbau der Terminologie, mit dem Ziel eine konsistente Funktionstüchtigkeit zu gewährleisten. Jedes Konzept in SNOMED CT lässt sich formal in der \mathcal{EL}^+ -Beschreibungslogik darstellen.

Der konkrete Domänenkonstruktor, so wie er formal in Kap.2.5.1 vorgestellt wurde, wird in SNOMED CT bisher derart nicht verwendet, trotz Einhaltung der polynomiellen Eigenschaft beim Inferenzieren und der Effektivität, die sie der Terminologie anbieten würde. Wissenschaftliche Ausarbeitungen sind in dieser Angabe, ob SNOMED CT auf \mathcal{EL}^+ oder \mathcal{EL}^{++} basiert, sehr unschlussig.

In der wissenschaftlichen Ausarbeitung [38] wird SNOMED CT mit der beschreibungslogischen Sprache \mathcal{EL}^{++} suggeriert und zwar heißt es: „*In particular, \mathcal{EL}^{++} is well-suited for the design of life science ontologies, and many existing such ontologies are formulated in this language. Examples include SNOMED CT, GO,...*“.

Auf der anderen Seite heißt es in [41] „...we first introduce the DL \mathcal{EL}^+ , which is an extension of the DL \mathcal{EL} used to define SNOMED CT.“ und „*In particular, the Systematized Nomenclature of Medicine (SNOMED) employs \mathcal{EL} with RIs and acyclic concept definitions.*“ in [44].

Da aus den angebotenen SNOMED CT-Browsern keinerlei Strukturen von konkreten Domänenconstructoren zu erkennen sind, habe ich diesbezüglich eine E-Mail an die International Health Terminology Standards Development Organisation (IHTSDO) geschickt, um eine einsichtige und unbestreitbare fachliche Antwort zu erhalten.

Kent A. Spackman, der Leiter der Terminologie versicherte mir in seiner E-Mail, dass SNOMED CT momentan ohne konkrete Domänen eingesetzt wird und die Beschreibungslogik \mathcal{EL}^+ benutzt.

In seiner E-Mail heißt es: „*In reply to your question, SNOMED CT uses \mathcal{EL}^+ and does not currently employ concrete domains. However, the Australian centre for SNOMED CT*

(NeHTA) has been exploring the possibility of using concrete domains for drug terminology. Michael Lawley of CSIRO, the author of the SnoRocket classifier (based on EL++) has been working to add concrete domains to SnoRocket. That work appears to be in its early phases, but he tells us that it will not take long to implement.“ [73].

Aus der E-Mail ist klar zu erkennen, dass sich das Einbinden von konkreten Domänen in SNOMED CT momentan in der Forschung befindet. Der erwähnte Klassifikationsalgorithmus „SnoRocket“ bietet eine Schnittstelle an, die bei Klassifikationsaufgaben von großen medizinischen Ontologien Anwendung findet [74].

Eine Version des SnoRocket-Algorithmus, die in Java 1.6.0_03 implementiert ist, kann auf einem modernen 2,4 GHz Intel Core 2 Duo Rechner in Windows XP alle Konzepte der SNOMED CT-Datenbank in 54 Sekunden klassifizieren [75].

Im Vergleich dazu steht die alternative Variante mit CEL [44], die auf einer etwas älteren CPU Xeon 3.6 GHz mit RedHat Linux 2.6.9 und in der Java Version 1.6.0_04 implementiert ist, mit 23 Minuten recht mager da.

SnoRocket ist aufgrund seiner Effizienz sehr gefragt. Um die vorhandene Implementierung zu erweitern, hat man sich auf das Klassifizieren von \mathcal{EL}^{++} spezialisiert und erforscht zur Zeit eine konsistente Implementierung von konkreten Domänen.

Da die Integration der konkreten Domänen in SNOMED CT nur eine Frage der Zeit darstellt, wird in Kap.2.5.1 neben der theoretischen Ausarbeitung von \mathcal{EL}^+ ebenso die Beschreibungslogik \mathcal{EL}^{++} mit ihrer ausgiebigen Theorie vorgestellt, damit die Vorteile von konkreten Domänenkonstruktoren im aktuellen Stand der Forschung erkannt werden.

Neben dem konkreten Domänenkonstruktor wird eine weitere beschreibungslogische Konstruktionsmöglichkeit nicht genutzt, die durch ihre polynomielle Eigenschaft den Umgang der SNOMED CT Terminologie effizienter gestalten lassen könnte.

Gemeint sind Instanziierungen bzw. Instanzmengen, die keine Anwendung in SNOMED CT haben. Zwar verfügt SNOMED CT über Konzepte, die keinerlei Unterkonzepte aufweisen, jedoch dürfen diese Konzepte nicht mit Instanzen verwechselt werden.

Eine „instance-of“ Relation wäre in SNOMED CT wünschenswert, damit eine Differenzierung zwischen Konzepten und Instanzen stattfinden kann und beim Postkoordinieren von Diagnosen konkrete Instanzen statt Konzepte angesprochen werden [86].

Um die vollständigen Analogien zwischen den formalen Komponenten in \mathcal{EL}^+ und SNOMED CT zu erkennen, wird auf die Tab.3.4 verwiesen.

Sei C ein Konzept in SNOMED CT. Beschreibungslogisch betrachtet entspricht eine vollständige Konzeptdefinition von C , einer Konjunktion aller Bezeichnungen und Beziehungen von C .

Anlehnung an das Beispiel aus Kap.3.3.2: Das Konzept **Pneumonia** (*Lungenentzündung*) ist durch folgende beschreibungslogische Aussage vollständig spezifiziert:

$$\begin{aligned}
 &Pneumonia \equiv \\
 &LungConsolidation \sqcap Pneumonitis \sqcap \exists associatedMorphology.Consolidation \sqcap \\
 &\quad \exists associatedMorphology.Inflammation \sqcap \exists findingSite.LungStructure
 \end{aligned}$$

\mathcal{EL}^{++} -Komponente	SNOMED CT-Komponente
Konzept C	Konzept <i>Pneumonia, Disease, Inflammation, ...</i>
Rollenname r	Attribute <i>PartOf, FindingSite, ...</i>
Topkonzept (\top)	<i>FSN: SNOMED CT Concept (SNOMED RT+CTV3)</i> mit der <i>ConceptID</i> : 138875005. Dieses Konzept wird auch super root concept genannt und bildet die Wurzel aller SNOMED CT Konzepte.
Existentielle Restriktion $\exists r.C$	Existentielle Restriktionen werden benutzt, um Attributsbeziehungen beschreibungslogisch aufbauen zu können. <i>$\exists associatedMorphology.Inflammation$</i>
Konjunktion $C \sqcap D$	Alle Bezeichnungen und Beziehungen eines SNOMED CT Konzeptes werden in eine Konjunktion gesetzt, um eine äquivalente Aussage zu diesem Konzept zu erhalten bzw. eine vollständige Konzeptdefinition aufzustellen. <i>$Cyst \equiv Disease(disorder) \sqcap \exists AssociatedMorphology.Cyst$</i>
Inklusionsbeziehung (\sqsubseteq)	In SNOMED CT wird die Inklusionsbeziehung verwendet, um die hierarchische Ordnung von allen Konzepten und Attributen (die ebenso als Konzepte gespeichert sind) aufzustellen. Dies geschieht durch die is-a -Beziehung, die ebenso als Attribut in der Attribute-Achse gespeichert ist und beschreibungslogisch betrachtet eine äquivalente Bedeutung zur Inklusion hat. Sie bildet das Fundament für den hierarchischen Aufbau der Terminologie. <i>AllantoicCyst (disorder) is-a Cyst(disorder)</i>
Rolleninklusion $r_1 \circ \dots \circ r_n \sqsubseteq r$	Inklusionsbeziehungen können auf Attribute angewendet werden, um den hierarchischen Aufbau von Attributen festzulegen. <i>After is-a Associated with is-a Concept model attribute is-a Attribute is-a Linkage concept is-a SNOMED CT Concept</i> Kompositionierte Relationen werden mit gruppierten Relationen definiert, siehe Kap.3.4.2 .
GCI $C \sqsubseteq D$	Die Anwendung der Inklusionsbeziehung auf SNOMED CT-Konzepte legt den hierarchischen Aufbau der Konzepte fest und ist für die Terminologie von existentieller Bedeutung. <i>Criminal life style is-a Life style is-a Social context is-a SNOMED CT Concept</i>
Konkreter Domänenkonstruktor $p(f_1, \dots, f_k)$ mit $p \in \mathcal{PD}_i$	Nicht vorhanden!
Instanzmenge $\{a\}$	Nicht vorhanden!

Tabelle 3.4: Die Beschreibungslogik \mathcal{EL}^+ in SNOMED CT. Notationshinweise und Beispiele sind in blauer Schriftfarbe.

3.4.1 SEP-Tripel

Die meisten diagnostischen und therapeutischen Aussagen nehmen Bezug auf die Anatomie. Jedoch können unstrukturierte „part-of“ und „is-a“-Beziehungen zu inkonsistenten Inferenzen führen.

Bsp.: Im Gegensatz zu **Handverletzung is-a Armverletzung** wird **Amputation der Hand (als Ganzes) is-a Amputation des Arms (als Ganzes)** zu Recht nicht abgeleitet. Um solche inkonsistente Aussagen zu umgehen, wird das Zusammenspiel zwischen anatomischen „part-of“-Beziehungen und Subsumtionsbeziehungen mit „is-a“ speziell gelöst. Für sämtliche anatomische Begriffe wird der so genannte **SEP-Tripel**-Ansatz verwendet, der auf Schulz et al. [76,77] zurückgeht.

In diesem Ansatz wird jedes anatomische Konzept aus der Achse „Body-Structure“ als Aggregat aus drei Konzepten modelliert, siehe Abb.3.6 .

Ein künstlich erzeugtes **Strukturkonzept** stellt die Disjunktion eines anatomischen **Entity**-Konzeptes als Ganzes und seiner Teile (**Part**-Konzepte) dar. Die jeweiligen E-Konzepte (bspw. ganzer Arm) und P-Konzepte (bspw. Armteil) werden beide vom künstlichen S-Konzept (bspw. Arm oder Armteil) subsumiert.

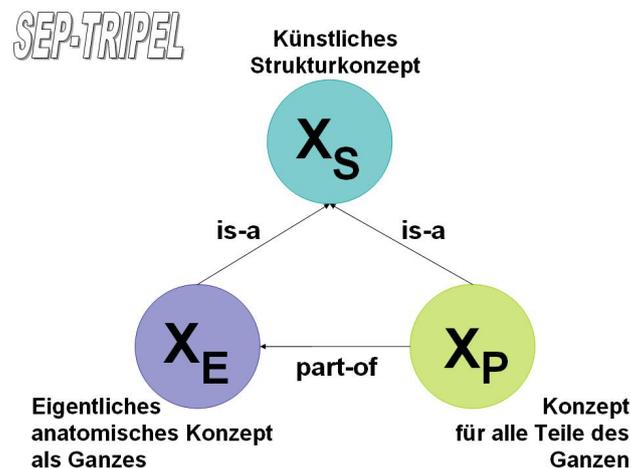


Abbildung 3.6: Schematischer Aufbau des SEP-Tripels. Einer Graphik von [77] nachempfunden.

Der Vorteil dieser dreiecksförmigen Modellierung kommt nun beim Zusammenspiel mit weiteren Relationen (bspw. `hasLocation`) zum Tragen, siehe Abb.3.7 .

Je nachdem, ob man in den Konzeptdefinitionen die Relation auf einen **S**truktur-Knoten (grün gefärbt) oder auf einen **E**ntity-Knoten (rot-gefärbt) anwendet, lassen sich Subsumtionsbeziehungen zwischen den zusammengesetzten Konzepten ableiten oder nicht. Somit kann die Entscheidung über die Knotenzuordnung gezielt für jede Relationsanwendung getroffen werden.

Die SEP-Tripel werden in der Hierarchie derart angeordnet und verknüpft, dass keine Kompositionen an `is-a` Beziehungen hergeleitet werden können, die zu falschen Schluss-

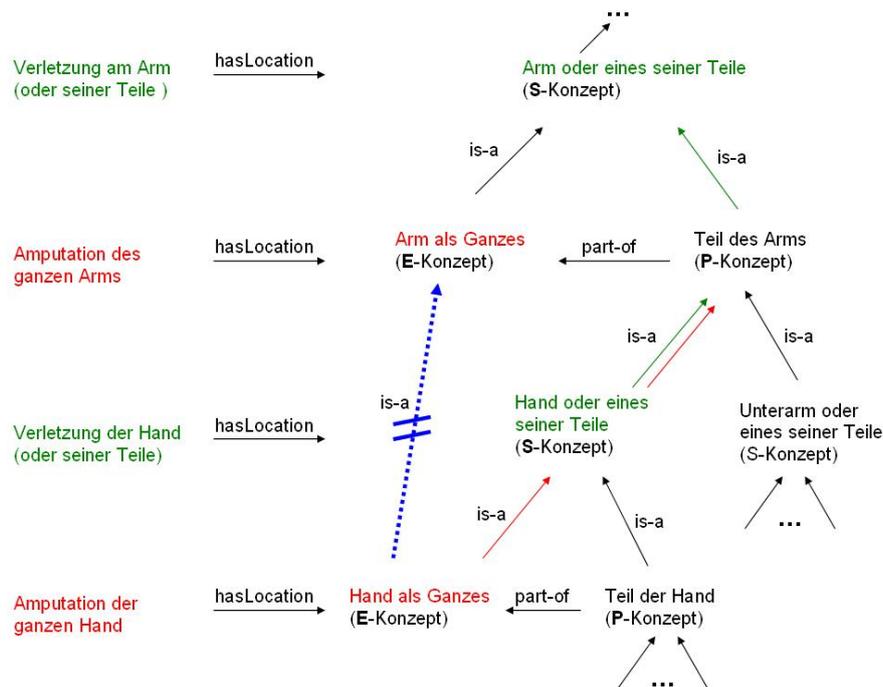


Abbildung 3.7: SEP-Tripel Ansatz zur Repräsentation anatomischer Konzepte am Beispiel „Amputation“. Einer Graphik von [77] nachempfunden.

folgerungen führen. In Abb.3.7 visualisiert der durchgestrichene blaue Pfeil eine falsche Schlussfolgerung, die durch den SEP-Tripel Ansatz umgangen wird. Denn das Konzept „Hand als Ganzes“ soll nicht von dem Konzept „Arm als Ganzes“ subsumiert werden. Der Nachteil einer differenzierten Emulation von transitiven part-of Relationen durch den SEP-Tripel-Ansatz ist die Verdreifachung aller anatomischen Konzepte und die damit verbundene Unübersichtlichkeit für Entwickler und Benutzer. Diese dreifache Ausdehnung wird am Beispiel des Darmtraktes in der nächsten Abb.3.8 wesentlich deutlicher. Appendizitis ist eine Entzündung des Wurmfortsatzes (Appendix) und unterscheidet sich von einer Kolitis, die eine Entzündungen des Grimmdarmes (Kolon) darstellt. Zäkum ist der medizinische Fachbegriff für Blinddarm. Die zugehörigen beschreibungslogischen Notationen sehen wie folgt aus:

$$\begin{aligned}
 Appendix_E &\sqsubseteq Appendix_S \sqsubseteq Zäkum_P \\
 Zäkum_E &\sqsubseteq Zäkum_S \sqsubseteq Kolon_P \\
 Kolon_E &\sqsubseteq Kolon_S \sqsubseteq \top
 \end{aligned}$$

Durch die Relationen **part-of** und **hasLocation** ergeben sich folgende beschreibungslogische Aspekte:

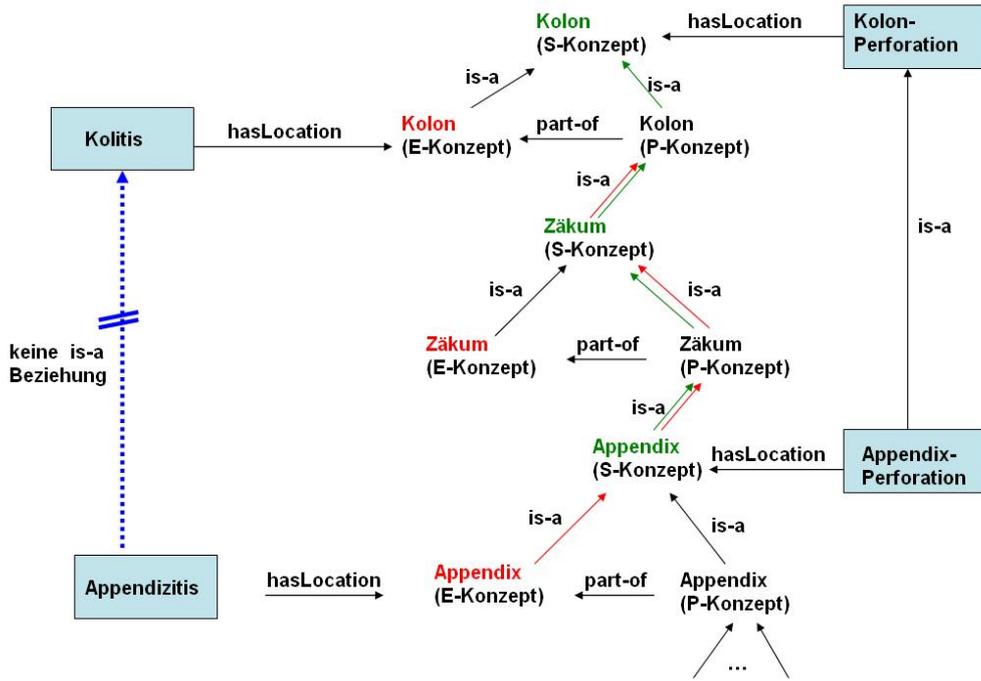


Abbildung 3.8: SEP-Tripel Ansatz zur Repräsentation anatomischer Konzepte am Beispiel „Darmtrakt“, einer Graphik von [77] nachempfunden.

$$Appendix_P \equiv Appendix_S \sqcap \exists part-of. Appendix_E \models Appendix_P \sqsubseteq Appendix_S$$

$$Zäkum_P \equiv Zäkum_S \sqcap \exists part-of. Zäkum_E \models Zäkum_P \sqsubseteq Zäkum_S$$

$$Kolon_P \equiv Kolon_S \sqcap \exists part-of. Kolon_E \models Kolon_P \sqsubseteq Kolon_S$$

$$Appendizitis \equiv Inflammation \sqcap \exists hasLocation. Appendix_E$$

$$Kolitis \equiv Inflammation \sqcap \exists hasLocation. Kolon_E$$

$$Appendizitis \not\sqsubseteq Kolitis$$

$$AppendixPerforation \equiv Perforation \sqcap \exists hasLocation. Appendix_S$$

$$KolonPerforation \equiv Perforation \sqcap \exists hasLocation. Kolon_S$$

$$AppendixPerforation \sqsubseteq KolonPerforation$$

Die transitive Schlussfolgerung von „Appendix part-of Zäkum“ und „Zäkum part-of Kolon“ auf „Appendix part-of Kolon“ kann durch das Schließen in Subsumtionshierarchien emuliert werden. Da $Appendix_E$ von $Appendix_S$ subsumiert wird und diese wiederum von $Zäkum_P$ subsumiert wird, erbt das Konzept $Appendix_E$ die Rollenrestriktion von $Zäkum_P$, nämlich ein Teil des Zäkums zu sein, d.h. $\exists part-of. Zäkum_E$. Diese Abhängigkeit setzt sich entlang der Subsumtionshierarchie nach oben fort.

Letztendlich hat dies zur Folge, dass einerseits die Kolitis in keiner Subsumtionsbeziehung mit Appendizitis steht, und dass andererseits eine Perforation des Grimmdarms als Oberkonzept der Wurmfortsatzperforation erkannt werden kann.

Das Zusammenspiel mit verschiedenen Relationen funktioniert analog zu den Beispielen in Abb.3.7 und Abb.3.8 . Der SEP-Tripel Ansatz ist nicht nur für anatomische Interpretationen zugänglich, sondern kann eine viel weitergehende Modellierung anbieten, etwa die der räumlichen Beziehungen. Ziel ist es Oberflächen, Hohlräume und Überlappungen durch den Einsatz von SEP-Tripeln in die Hierarchie formal zu integrieren.

So können Urin in der Harnblase, das Magenkarzinom im Magen, sowie der Embryo in der Gebärmutter lokalisiert werden. Man sieht, dass es sich hierbei nicht um part-of-Beziehung handeln kann. Für detaillierte Ausführungen dieser Raum- und Zeitphänomene sei auf die Literatur verwiesen [78,79,80].

3.4.2 Gruppierete Relationen

Um eine hohe Granularität anbieten zu können, müssen die dafür vorgesehenen Konzepte und Relationen entweder exakt präkoordiniert vorliegen oder aus Vorhandenem erzeugt werden. Da die Präkoordination von allen Möglichkeiten aussichtslos ist, kann es vorkommen, dass sehr viele Relationen und Konzepte benutzt werden müssen, um den medizinischen Sachverhalt formal festhalten zu können.

Zu beachten ist hierbei; je höher der Anspruch an Granularität ist, desto höher wird die Gefahr einer inkonsistenten Schlussfolgerung. Zu diesem Zweck werden in SNOMED CT **gruppierete Relationen** (*engl.*: relationship-groups) verwendet, um Konzeptdefinitionen konsistent und effizient zu gestalten. Denn ohne genaue Differenzierung von **kompositionierten Relationen (RI)** der Art $r_1 \circ \dots \circ r_n \sqsubseteq s$, können hierarchische Strukturen nicht erkannt werden.

Beispiel aus [81]:

Die Fallot-Tetralogie ist eine Herzkrankheit, die durch vier Syndrome charakteristisch gekennzeichnet ist:

1. Rechtsherzhypertrophie (right ventricle hypertrophy)
2. überreitende Aorta (aorta overriding)
3. Pulmonalstenose (pulmonary valve stenosis)
4. Ventrikelseptumdefekt (incomplete closure of the interventricular septum)

Damit die Fallot-Tetralogie als ein Unterkonzept von Pulmonalstenose erkannt werden kann, muss die Pulmonalstenose wie folgt definiert werden:

PulmonaryValveStenosis \equiv *Disease* \sqcap \exists RG. (\exists *FindingSite*.

PulmonaryValveStructure \sqcap \exists *AssociatedMorphology.Stenosis*)

Die Abkürzung „RG“ steht für eine Gruppe von Relationen. Man interpretiert „RG“ als eine Art „anonyme Eigenschaft“, die die kompositionierten Relationen gruppiert.

SNOMED CT verfügt über 315.550 aktive Konzepte, wovon 87.345 durch gruppierte Relationen definiert sind.

3.5 Anwendungsmöglichkeit der SNOMED CT-Terminologie

SNOMED CT kann im Gegensatz zu anderen Klassifikationssystemen nicht mehr in Buchform benutzt werden. Sie wird in Kombination mit einer Applikationssoftware verwendet, die Zugriff auf die SNOMED CT Ontologiedatenbank hat, um medizinische Sachverhalte in der beschreibungslogischen SNOMED CT-Syntax wiedergeben zu können. Die tatsächliche Integration von SNOMED CT in ein System ist nämlich von dem eigentlichen Inhalt von SNOMED CT getrennt zu sehen, siehe Abb.3.9 .

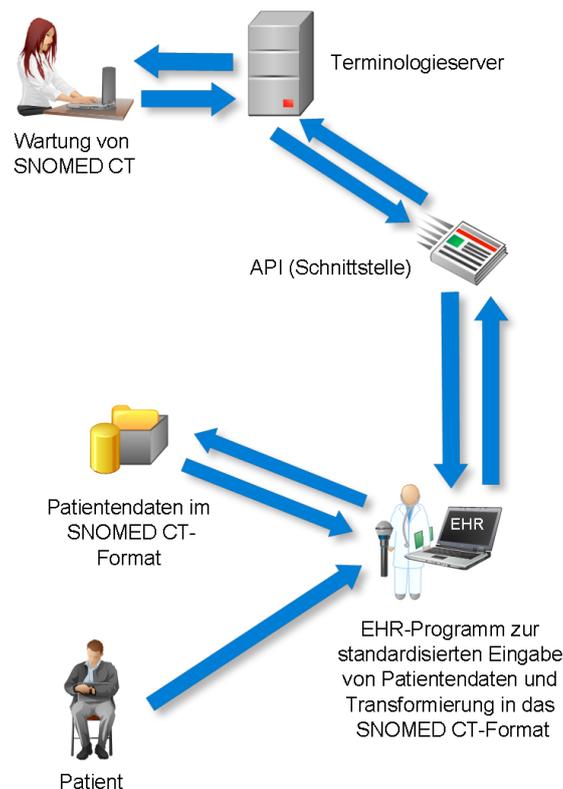


Abbildung 3.9: Software Architektur von SNOMED CT - Eine Implementierungsvariante

Zur Realisierung steht ein Terminologieserver zur Verfügung, der medizinisches Wissen strukturiert ablegt und über die Terminologie als standardisierten Zugriffspunkt verfügbar macht. Durch diese Herangehensweise kann jederzeit auf aktuelles Wissen zugegriffen und Daten ausgetauscht werden, welches einen großen Schritt für die semantische Interoperabilität darstellt. Bei Weiterentwicklungen von SNOMED CT kann somit der

Terminologieserver jederzeit aktualisiert werden, so dass die zukünftigen Konzepte in aktueller Form generiert und gespeichert werden können.

Die Aufgabe der Applikationssoftware auch EHR-Programm (Electronic Health Record Application Program) genannt, ist es nun, medizinisches Wissen bzw. strukturiert eingegebene Patientendaten zu verarbeiten und in SNOMED CT-Syntax zu transformieren. Diese Programme bieten eine Eingabemaske für eine strukturierte Erfassung von medizinischen Sachverhalten an und können aufgrund ihrer Flexibilität ideal zur Realisierung von elektronischen Patientenakten ebenso im Netzwerk eingesetzt werden.

Durch die vorgegebene Eingabemaske wird der Anwender an eine Benutzeroberfläche gebunden, die wiederum an eine feststrukturierte Syntax gekoppelt ist. Textuelle Sachverhalte, die aufgrund ihrer Mehrdeutigkeiten schwer zu kodieren sind, können nun leichter verarbeitet werden.

Da das System eine syntaktisch und zugleich semantisch korrekte Eingabe erzwingt, können falsche Eingaben stark reduziert werden.

Statt einer nachträglichen Kodierung bereits dokumentierter Patientendaten wird durch diese geeignete Form der strukturierten Dokumentation eine hohe Detailtreue in der Kodierung erreicht. Dieses kann bspw. mit einer Software wie „Clinic Win Data“ [82] oder „ID-Logik“ [83] erreicht werden.



Abbildung 3.10: Bild- und Befunddokumentationssystem Clinic Win Data - Screenshot [82]

Das zukünftige Bestreben sollte nun sein, diese Systeme derart zur Verfügung stellen, dass sie über ein Schnittstellensystem (API) eine Verbindung zum Terminologieserver aufbauen und die eingegebenen Daten automatisch in SNOMED CT Ausdrücke umwandeln.

Der klinische Anwender soll in der Lage sein, den zutreffenden SNOMED CT Kode ohne großen Aufwand zu erhalten. Auf Wunsch sollen ihm die gesamte Fülle der hierarchischen und attributiven Beziehungen von SNOMED CT zur Verfügung stehen, so dass er mit geringem Aufwand eine exakte Beschreibung des Krankheitsbildes des Patienten, der angewandten Verfahren, der Medikation usw. erreichen kann. In diesem Fall würde die beschreibungslogische Analyse bereits während des Erstellungsprozesses erfolgen.

Eine geeignete Form der Eingabe durch behandelnde Ärzte und Pflegekräfte bestimmt die Praxistauglichkeit von SNOMED CT. Denn Ärzte sollten in ihrer textuellen Ausdrucksfreiheit nicht zu stark eingeschränkt werden. Ziel ist es einen geeigneten Mittelweg zwischen einer starren syntaktischen und einer ungehinderten textuellen Eingabe zu finden, so dass die Anwendung sich als praxistauglich erweist und damit der Anwender nicht mit beschreibungslogischen Notationen belastet wird. Die mächtige Referenzterminologie soll quasi im Hintergrund unsichtbar umgesetzt werden.

Bspw. bietet ID-Logik neben der Eingabe über die Tastatur auch eine interaktive Spracherkennung an, die den Ablauf einer strukturierten Eingabe benutzerfreundlicher gestalten lässt. Die Abb.3.10 zeigt ein Bildschirmabbild der Clinic Win Data Software der Firma E&L.

Fazit und Ausblick

Die SNOMED CT Terminologie wurde allgemein wie auch im speziellen Bezug zu den verwendeten Wissensrepräsentationsformalissen untersucht, angefangen bei den medizinischen Dokumentationssystemen und Beschreibungslogiken bis hin zu den praktischen Möglichkeiten und Grenzen einer effizienten beschreibungslogischen Realisierung der SNOMED CT Terminologie.

Da die zunehmend elektronische Verarbeitung von Daten und Wissen im Gesundheitswesen auf semantischer Ebene den Einsatz einer maschinell interpretierbaren Terminologie für immer neue Aufgabenstellungen erfordert, besitzt die SNOMED CT Terminologie große Chancen, in den nächsten Jahren die weltweit akzeptierte und zur Anwendung kommende medizinische Referenzterminologie zu werden. Denn die heute eingesetzte ICD-10 Klassifikation wird den Anforderungen der neuen IT Verfahren in der Medizin wie elektronische Gesundheitskarte, elektronisches Rezept und die Möglichkeit einer medizinischen Entscheidungsunterstützung nur unzureichend gerecht.

Aus den Studien [90],[91] und insbesondere aus [89] stellt sich heraus, dass SNOMED CT, trotz einiger Defizite und Unzulänglichkeiten, durchaus für einen praktischen Einsatz geeignet ist und sich relativ gut für eine standardisierte medizinische Dokumentation klinischer Fachbegriffe eignet.

Fehlerhafte Schlussfolgerungen basieren auf ungewollten Wechselwirkungen von Konzepten, die in bestimmten Kontexten falsche Aussagen implizieren. Um Fehler zu vermeiden, haben sich im Verlauf der Jahre mehrere Ansätze wie gruppierte Relationen (Kap.3.4.2) und SEP-Triplets (Kap.3.4.1) etabliert, die viele ungewollte Inferenzen beseitigten. Dennoch ist an vielen Stellen zu erkennen, dass SNOMED CT bisher nur partiell praktisch angewendet worden ist. Dies ist auch einer der Gründe warum Relationen von Konzepten, die mehrere Achsen ansprechen, fehleranfällig sind [86].

Zahlreiche Fehler und Probleme im Detail können vor dem Hintergrund toleriert werden, dass sich die Terminologie in einem langfristigen Entwicklungsprozess befindet.

Eine Stellungnahme der Entwickler [92]: „*In 2003, the US Government licensed SNOMED CT and the NCVHS recommended it as the general terminology for patient medical record information in the US. In the UK, SNOMED CT is a draft national standard and a key element of the NHS National Program for IT. Thus SNOMED is not a theoretical academic exercise, but is being developed with serious expectations and demands for practical usability.*“

In den Ländern wie England, USA und Australien hat man sich für den Einsatz der SNOMED CT Terminologie entschieden, die künftig als standardisierte Terminologie in EHR-Systeme integriert werden soll, um Kosten zu minimieren und die Qualität der Pa-

tientenversorgung zu erhöhen.

Im Hinblick auf ambitionierte Projekte, wie der Einsatz der eGK zur Entscheidungsunterstützung, ist die Einführung der SNOMED CT Terminologie auch in Deutschland nur eine Frage der Zeit. In einem Positionspapier der GMDS-Projektgruppe werden zahlreiche Konsequenzen einer potentiellen Einführung von SNOMED CT dargelegt [87]. Die Zusammenfassung dieses Papiers empfiehlt neben einer Anforderungs-, Kosten- und Nutzenanalyse vor allem eine schnelle und fachübergreifende Auseinandersetzung mit SNOMED CT, die in Deutschland bisher kaum stattgefunden hat.

Aus vielen wissenschaftlichen Quellen wie [69],[87] oder [61] geht hervor, dass SNOMED CT in Zukunft zur Entscheidungsunterstützung und zur Validierung von Gesundheitsdaten herangezogen wird.

Diese Anwendungsmöglichkeiten sehe ich persönlich - aufgrund der derzeit fehlenden Akzeptanz und Erfahrung - als zukünftige Vision, welches noch weiterer Forschung und Weiterentwicklung des Konzeptes bedarf.

Primär sollte eine derartig umfassende Terminologie für das Standardisieren der elektronischen Patienten- und Gesundheitsakten eingesetzt werden, damit schrittweise die in den verschiedenen Krankenhausinformationssystemen eingesetzten, miteinander nicht kompatiblen Data Dictionaries abgelöst werden können.

Erst nach dieser Hürde, kann die Diskussion von Entscheidungsunterstützung ein präsentables Thema werden.

Die Einführung der eGK ist der erste Schritt für die Ermöglichung der semantischen Interoperabilität und einer standardisierten Dokumentation von medizinischen Daten. Trotz der beschlossenen Einführung der eGK sind die Meinungen sehr gespalten. Die Umsetzung stoß auf große Proteste der Ärzte: *„Krankenkassen können die Datenbank beliebig nach Risikopatienten und teuren Kranken durchforsten. Dies führt dazu, dass Ärzte bald nur noch Anweisungen von Kassenmitarbeitern umsetzen und Bürger bald ihre intimen Daten im Internet oder sogar auf dem Schreibtisch des Arbeitsgebers vorfinden.“* [93]. Ärzte befürchten, dass die intimen Daten auf zentralen Servern der Krankenkassen abgelegt und nicht auf der eGK gespeichert werden. Der Datenzugriff würde über das Internet erfolgen und die eGK dabei nur wie ein Schlüssel funktionieren.

Einige Demonstranten sind sogar der Ansicht, dass das Projekt der eGK verfassungswidrig und laut Gesetz (§ 291 a und b SGB V) nichtig sei [88].

Das eGK-Projekt ist erfolgversprechend, wenn persönliche Daten vor Zugriffen durch Dritte geschützt sind und eine effiziente Implementierung der SNOMED CT Terminologie umgesetzt wird.

Eine auf SNOMED CT basierende Datenverwaltung wird für zukünftige interessante und technisch aufwendige Projekte unumgänglich sein, weil die Daten den semantischen Anforderungen gerecht werden müssen, die nur durch Beschreibungslogiken konsistent verarbeitet und formatiert werden können.

Die Zukunft wird zeigen, ob sich SNOMED CT in Deutschland durchsetzen und Akzeptanz finden wird. Fakt ist, dass die zunehmende Verwendung der Informationstechnolo-

gie in der Medizin die Nachfrage nach ausdrucksstärkeren Dokumentationssystemen wie SNOMED CT vergrößert, und dass die vorhandenen Dokumentationssysteme wie ICD-10 für größere bzw. komplexere Projekte auf Dauer nicht geeignet sind.

Literaturverzeichnis

- [1] Personal Health Monitoring-Motivation
<http://www.phmon.de/deutsch/motivation.html>
- [2] Michael Schlander, Oliver Schwarz und Christian Thielscher, *Gesundheitsausgaben in Deutschland: Eine Makroökonomische Analyse ihrer langfristigen Finanzierbarkeit*, 2004
- [3] Tatsachen über Deutschland,
<http://www.tatsachen-ueber-deutschland.de/de/gesellschaft/main-content-08/aeltere-menschen.html>
- [4] Marion Schaefer, *Möglichkeiten und Grenzen der Arzneimitteldokumentation zur Verbesserung der Arzneimitteltherapiesicherheit*, Institut für Klinische Pharmakologie, Charité Universitätsmedizin Berlin
- [5] BMG, *Die elektronische Gesundheitskarte*,
http://www.dak.de/content/files/BMG_eGK_Broschuere0708.pdf
- [6] Florian Leiner, 2006, *Medizinische Dokumentation*, Lehrbuch, 5. Auflage, Seite 2
- [7] Gernot Wersig, *Thesaurus-Leitfaden: Eine Einführung in das Thesaurus-Prinzip in Theorie und Praxis*, Lehrbuch, Verlag Dokumentation Saur KG; 2.Auflage, April 1991
- [8] Wikipedia, *Terminologie*,
<http://de.wikipedia.org/wiki/Terminologie>
- [9] Michael Schuntermann, *Internationale Klassifikation der Funktionsfähigkeit, Behinderung und Gesundheit (ICF) der Weltgesundheitsorganisation (WHO)*,
www.arc-grevenbroich.de/pdf/icf.pdf
- [10] Dimdi, *Universal Medical Device Nomenclature System*,
www.dimdi.de/static/de/mpg/thesauri/umdns/index.htm
- [11] Dimdi, *MeSH - Medical Subject Headings*,
http://www.dimdi.de/static/de/klassi/mesh_u/mls/mesh/meshallg.htm
- [12] Dimdi, *LOINC: Begriffssystem zur Verschlüsselung von Laborwerten und klinischen Untersuchungen*,
www.dimdi.de/static/de/klassi/loinc/index.htm

- [13] Dimdi, *Anatomisch-Therapeutisch-Chemische (ATC) Klassifikation mit definierten Tagesdosen (DDD, Defined Daily Doses)*,
www.dimdi.de/static/de/klassi/atcddd/index.htm
- [14] Bettina Busse, August 2003, *ICD-10 und OPS*, Schriftenreihe zur Förderung der medizinischen Dokumentation, GFMD Verlag
- [15] Thomas Gruber, August 1993, *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Seite 1, Knowledge Systems Laboratory, Stanford University.
<http://tomgruber.org/writing/onto-design.pdf>
- [16] Nardi, Brachman, *An Introduction to Description Logics*, Seite 1,
<http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-01.pdf>
- [17] *Hintergrundwissen: Vom Glossar zur Ontologie*
<http://wissensexploration.de/textmining-hintergrundwissen.php>
- [18] Wikipedia, *Glossar*,
<http://de.wikipedia.org/wiki/Glossar>
- [19] Wikipedia, *Taxonomie*
<http://de.wikipedia.org/wiki/Taxonomie>
- [20] Wikipedia, *Thesaurus*
<http://de.wikipedia.org/wiki/Thesaurus>
- [21] WM 2.0 Wissensmanagement, *Ontologie*,
<http://www.rolotec.ch/wissenswiki/index.php/Ontologie>
- [22] Helbig H., *Künstliche Intelligenz und automatische Wissensverarbeitung*, Hagen, 2.Aufl.1996, S.99
- [23] Software-engineering, *Semantisches Netz*,
<http://www.softwarekompetenz.com/?2409&highlight=repr%C3?sentiert>
- [24] Ipke Wachsmuth, WS 2000/2001, *Alternative Notationen: Semantische Netze; Frames*, Vorlesung: Methoden der Künstlichen Intelligenz, Folie 4
http://www.techfak.uni-bielefeld.de/ags/wbski/lehre/digiSA/Methoden_der_KI/WS0001/methki03.pdf
- [25] Andreas Abecker, *Wissensrepräsentation durch Logik: Diskussion*, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Vorlesungsfolie 11
- [26] Mosbach, SS 2000, *Kapitel 8: Wissensrepräsentation mit Semantischen Netzen und Frames*, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Folie 5
- [27] Institut für Informationssysteme und Computer Medien, *Kap 3.5 Frames*,
http://www.iicm.tu-graz.ac.at/Teaching/theses/2000/_idb9e_/greif/node5.html#SECTION02250000000000000000

- [28] Lämmel, Cleve, 2008, *Künstliche Intelligenz*, 3.Auflage, Hanser Verlag
- [29] Spreckelsen und Spitzer, 2008, *Wissensbasen und Expertensysteme in der Medizin*, 1.Auflage, Vieweg+Teubner Verlag, Seite 133
- [30] FG Wissensverarbeitung, *Kapitel 9 - Frames*, Foliensatz: Künstliche Intelligenz, WS 2008/2009, Universität Kassel, Seite 21
- [31] Wikipedia, *Description Logic*,
http://en.wikipedia.org/wiki/Description_logic
- [32] Franz Baader; Diego Calvanese; Deborah McGuinness; Daniele Nardi; Peter Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, 2003, S.47
- [33] Stefan Müller, *Logik für Linguisten*, Vortragsfolien, 2006, Universität Bremen, S.8
<http://hpsg.fu-berlin.de/~stefan/PS/beweisbaeume-handout-2x2.pdf>
- [34] Prof. Dr. Bernhard Bauer, *Konzepte und Techniken für das Semantic Web*, Seminarband, Universität Augsburg, WS 2007/2008, Seite 73
- [35] Prof. Schmidt-Schauss, *Kapitel 9: Konzeptbeschreibungssprachen*, Skript zur Vorlesung Künstliche Intelligenz, Goethe Universität, Frankfurt am Main, Seite 7,
<http://www.ki.informatik.uni-frankfurt.de/lehre/SS2006/KI/skript/KI-9KL-ONE.pdf>
- [36] Christopher Habel, Vorlesungsskript: *Wissensrepräsentation*, SS 2005, Universität Hamburg, S.18, <http://www.informatik.uni-hamburg.de/WSV/teaching/vorlesungen/WissRep05/WR05-09.pdf>
- [37] Thorsten Liebig, *Wissensmodellierung und wissensbasierte Systeme*, Vorlesungsfolien, SS2007, Universität Ulm, Institut für Künstliche Intelligenz
- [38] Franz Baader; Sebastian Brandt; Carsten Lutz, *Pushing the \mathcal{EL} Envelope*, Institute for Theoretical Computer Science, TU Dresden
<http://lat.inf.tu-dresden.de/research/papers/2005/BaaderBrandtLutz-IJCAI-05.pdf>
- [39] Rector und Horrocks, *Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions*, Medical Informatics Group, Department of Computer Science, University of Manchester, In Proc. Workshop on Ontological Engineering 1997,
<http://www.aaai.org/Papers/Symposia/Spring/1997/SS-97-06/SS97-06-013.pdf>
- [40] The GO Consortium, *Gene Ontology: Tool for the unification of biology*. Nature Genetics, Mai 2000

- [41] Franz Baader und Boontawee Suntisrivaraporn, *Debugging SNOMED CT Using Axiom Pinpointing in the Description Logic \mathcal{EL}^+* , Institute for Theoretical Computer Science, TU Dresden, 2008
<http://lat.inf.tu-dresden.de/research/papers/2008/BaaSun-KRMED-08.pdf>
- [42] Franz Baader, *Restricted role-value-maps in a description logic with existential restrictions and terminological cycles*, Theoretical Computer Science, TU Dresden
<https://eprints.kfupm.edu.sa/62507/1/62507.pdf>
- [43] Franz Baader; Sebastian Brandt; Carsten Lutz, *Pushing the \mathcal{EL} Envelope Further*, Institute for Theoretical Computer Science, TU Dresden
http://www.webont.org/owled/2008dc/papers/owled2008dc_paper_3.pdf
- [44] Franz Baader; Lutz; Suntisrivaraporn, *CEL - a polynomial-time reasoner for life science ontologies*. In Proceedings of IJCAR'06, Volume 4130 of LNAI, Seiten 287-291, Springer Verlag, 2006
- [45] Franz Baader; Lutz; Suntisrivaraporn, *Is tractable reasoning in extensions of the description logic \mathcal{EL} useful in practice?*. In Proceedings of M4M-05, 2005
- [46] Franz Baader; Lutz; Suntisrivaraporn, *Efficient reasoning in \mathcal{EL}^+* . In Proceedings of DL2006, Nummer 19 in CEUR-WS (<http://ceur.ws.org/>), 2006
- [47] Christoph Haase und Carsten Lutz, *Complexity of Subsumption in the \mathcal{EL} Family of Description Logics: Acyclic and Cyclic TBoxes*,
<http://www.informatik.uni-bremen.de/~clu/papers/archive/ecai08a.pdf>
- [48] G.De Giacomo and M.Lenzerini, *Boosting the correspondence between description logics and propositional dynamic logics*. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94), Volume 1, Seiten 205-212. AAAI Press, 1994
<https://www.aaai.org/Papers/AAAI/1994/AAAI94-032.pdf>
- [49] Toman and Wedell, *On reasoning about structural equality in XML: A description logic approach*, Department of Computer Science, University of Waterloo, Canada, 2004
<http://db.uwaterloo.ca/~david/papers-icdt03.pdf>
- [50] D.Calvanese, *Reasoning with Inclusion Axioms in Description Logics: Algorithms and Complexity*, ECAI 1996, 12th European Conference on Artificial Intelligence, 1996
- [51] Hartmut Klauck, Präsentationsfolien: *Black Box Algorithmen*, SS2005, Universität Frankfurt, 2.Seite
http://www.thi.informatik.uni-frankfurt.de/~klauck/VorlBB05/VorlesungBB05_4.pdf,
- [52] Wikipedia, *Unifikation (Logik)*,
[http://de.wikipedia.org/wiki/Unifikation_\(Logik\)](http://de.wikipedia.org/wiki/Unifikation_(Logik))

- [53] J. A. Robinson, *A Machine Oriented Logic Based on the Resolution Principle*, Journal of the ACM, Volume 12, Issue 1, Januar 1965, Seiten 23-41, NY USA
- [54] Franz Baader und Paliath Narendran, *Unification of Concept Terms in Description Logics*. In Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98),
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.9450&rep1&type=pdf>
- [55] Franz Baader und Ralf Küsters, *Matching in Description Logics with Existential Restrictions*. In Proc. of KR2000, Theoretical Computer Science, RWTH Aachen,
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.8372&rep=rep1&type=pdf>
- [56] Franz Baader und Barbara Morawska, *Unification in the Description Logic \mathcal{EL}* , Theoretical Computer Science, TU Dresden, 2009
http://web.comlab.ox.ac.uk/DL2009/proceedings/oral/Baader_Morawska.pdf
- [57] Viorica Sofronie-Stokkermans, *Kap.2.8 Wohlfundierte Ordnungen*, Vorlesung: Automatisches Beweisen, WS2004/2005
www.mpi-inf.mpg.de/~sofronie/teaching/autreas/v10-autreas.pdf
- [58] Ralf Küsters, *Non-standard Inferences in Description Logics*, Springer Verlag, 2001, S.191
- [59] Chapman, *Mathematical Structures*,
http://math.chapman.edu/cgi-bin/structures.pl?Equational_theory
- [60] Viorica Sofronie-Stokkermans, *Locality and subsumption testing in EL and some of its extensions*, Max-Planck-Institut für Informatik,
<http://www.aiml.net/volumes/volume7/Sofronie-Stokkermans.pdf>
- [61] Warda F., *Elektronische Gesundheitsakten*, 2005, Rheinware Verlag, S.41
- [62] Wikipedia, *Systematisierte Nomenklatur des Medizin*,
http://de.wikipedia.org/wiki/Systematisierte_Nomenklatur_der_Medizin
- [63] *College of American Pathologists*, www.cap.org
- [64] *Logical Observation Identifiers Names and Codes (LOINC)*,
<http://loinc.org/>
- [65] *NHS-Read Codes*,
<http://www.connectingforhealth.nhs.uk/systemsandservices/data/readcodes>
- [66] Prof. Dr. Joachim Dudeck, *Einführung in die Referenzterminologie SNOMED CT*, Foliensatz, 2005, Institut für Medizinische Informatik Giessen,
http://www.hl7.de/download/veranstaltungen/jahrestagungen/2005/051028_SNOMED_Tutorial_Goettingen_0.pdf

- [67] CAP, *SNOMED CT User Guide*, January 2007 Release,
http://www.cap.org/apps/docs/snomed/documents/snomed_ct_user_guide.pdf
- [68] *SNOMED CT Browser VTSL*,
<http://terminology.vetmed.vt.edu/SCT/menu.cfm>
- [69] Offizielle Mitteilung der HL7-Benutzergruppe in Deutschland e.V., *HL7-Mitteilungen*, Ausgabe 20, 2006, Seite 20
- [70] Schulz, S. et.al., *Semantic Clarification of the Representation of Procedures and Diseases in SNOMED CT*, In Proc. MIE, 2005, Geneva
- [71] SNOMED CT Browser UVIC-HTG (Health Terminology Group),
<http://web.his.uvic.ca/Research/HTG/vocabulary/Terminologies/SNOMEDCT/Browse.php>, University of Victoria
- [72] Wikipedia, *CSV-Dateiformat*,
[http://de.wikipedia.org/wiki/CSV_\(Dateiformat\)](http://de.wikipedia.org/wiki/CSV_(Dateiformat))
- [73] E-Mail Stellungnahme von Kent A. Spackman, MD PhD, Chief Terminologist, International Health Terminology SDO, 27.10.09
- [74] The Australian E-Health Research Center, *SnoRocket: Ontology Classification Engine*, <http://aehrc.com/hie/snorocket.html>
- [75] Michael J. Lawley, *Exploiting Fast Classification of SNOMED CT for Query and Integration of Health Data*, Proceedings of the 3rd international conference on Knowledge Representation in Medicine (KR-MED 2008), Seite 12
- [76] Schulz S., Kumar A., Bittner T., *Biomedical Ontologies: What part-of is and isn't*, 2006, J Biomed Inform 39 (3): Seiten 350-361,
- [77] Schulz S., Romacker M., Hahn U., *Part-Whole Reasoning in Medical Ontologies Revisited: Introducing SEP Triplets into Classification-Based Description Logics*, 1998, J Am Med Inform Assoc (JAMIA Symposium Supplement): Seiten 830-834
- [78] Mejino JLV, Rosse C, *Conceptualization of anatomical spatial entities in the Digital Anatomist foundational model*. In Proc. of the Annual Symposium of the American Medical Informatics Association (AMIA 99) in Washington DC, November 1999, Seiten 6-10
- [79] Rector AL, *Clinical Terminology: Why it is so Hard?*, 1999, Meth.Inform.Med.: Seiten 239-252
- [80] Schulz S. und Hahn U., *Parts, locations, and holes: Formal reasoning about anatomical structures*. In: Quaglini S., Barahona P., Andreassen S. Proc.of the 8th Conf. on Artificial Intelligence in Medicine (AIME 2001), Portugal. Berlin: Springer (LNAI 2101), Seiten 293-303

- [81] Ronald Cornet und Stefan Schulz, *Relationship Groups in SNOMED CT*, Medical Informatics in a United and Healthy Europe, Seiten 223-227, <http://www.hst.aau.dk/~ska/MIE2009/papers/MIE2009p0223.pdf>
- [82] E&L Systems GmbH, *Clinic Win Data Software*, <http://www.eundl.de>
- [83] conhIT - der Branchentreff der Healthcare IT, *ID-LOGIK Software*, <http://www.virtualmarket.conhit.de/index.php5?id=1170671&Action=showProduct>
- [84] Universität Kassel, *Teil III: Wissensrepräsentation und Inferenz*, Vorlesung Künstliche Intelligenz, WS2007/2008 Seite 6
- [85] Mirjam Minor, *Semantische Netze*, Vorlesung: Data Engineering, WS2000/2001, Universität Berlin, Seite 13
- [86] Schulz; Baader; Suntisrivaraporn, *SNOMED CT's Problem List: Ontologists' and Logicians' Therapy Suggestions*
- [87] gmds, *Positionspapier zur „Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT) in Deutschland*, 2006, Köln, Seite 5
- [88] eGK-Kritik, <http://www.egk-kritik.info>
- [89] Wasserman H.; Wang J., *An applied evaluation of SNOMED CT as a clinical vocabulary for the computerized diagnosis and problem list*, AMIA Annu Symp Proc., 2003, 699-703
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1479961>
- [90] Penz JF; Brown SH; Carter JS; Elkin PL; Nguyen VN; Sims SA; Lincoln MJ, *Evaluation of SNOMED coverage of Veterans Health Administration terms*, Medinfo. 2004;11(Pt 1): 540-4
http://square.umin.ac.jp/DMIESemi/y2004/20041025_2.pdf
- [91] Burgun A; Bodenreider O; Mouglin F; *Classifying diseases with respect to anatomy: a study in SNOMED CT*, AMIA Annu Symp Proc. 2005, 91-95
<http://mor.nlm.nih.gov:8000/pubs/pdf/2005-amia-ab.pdf>
- [92] Spackman K. und Reynoso G., *Examining SNOMED from the Perspective of Formal Ontological Principles: Some Preliminary Analysis and Observations*, In Proc. of KR-MED 2004,
- [93] DailyNet.de, *Deutscher Ärztetag lehnt die Gesundheitskarte ab!*, 18.März 2007,
- [94] IHTSDO, *SNOMED Clinical Terms Overview*, August 2008, Seite 12