

Grundlagen für Anwendungen:

Ausdrucksstarke Beschreibungslogiken

Motivation

Um immer komplexer werdende Probleme fehlerfrei lösen zu können, sind häufig neuartige Ansätze und Techniken erforderlich. Dieses gilt auch für das Gebiet der anwendungsorientierten Informationstechnologie.

Wie lässt sich die Erstellung verlässlicher und anpassbarer Softwaresysteme effektiv unterstützen, wie sehen automatische Überprüfungstechniken für definierte Modelle aus - Fragen, die der Beitrag von Ralf Möller zu beantworten versucht.

Die Probleme anwendungsorientierter Informationstechnologie werden zunehmend komplexer, da vielfach tiefere ontologische Modellierungen zur Lösung eingesetzt werden sowie Aspekte der Verteilung und Sicherheit zu beachten sind. Verschiedene Ansätze zur Bewältigung der Komplexität wurden vorgeschlagen: Wissensbasierte Systeme, objektorientierte Rahmensysteme, vereinheitlichte Modellierungssprachen (z.B. UML: Unified Modeling Language), Spezifikationsprachen für das Verhalten von technischen Systemen, Modellierungssprachen für Arbeitsvorgänge (Workflow Modeling Languages) oder Organisationsmodelle für Geschäftsprozesse - um nur einige zu nennen. Allen Ansätzen gemeinsam ist das Ziel, nämlich die Erstellung benutzbarer, verlässlicher und anpassbarer Softwaresysteme zu

unterstützen, sowie ihr Bedarf an automatischen Überprüfungstechniken für definierte Modelle. Ferner verlangen viele Anwendungskontexte seit jeher Systeme, mit denen komplexe Probleme bzw. deren Teilprobleme auf der Basis von deklarativen Modellen der Anwendungsdomäne automatisch gelöst werden können. Hier bedeutet automatisches Lösen, dass die Problemlösung durch allgemeine Dienste bzw. vorgefertigte Bausteine und nicht durch Programmierung eines speziellen, anwendungsabhängigen Spezialverfahrens erbracht wird. In diesem Beitrag werden praktische und theoretische Forschungsergebnisse zur Entwicklung von Modellierungswerkzeugen zur Aufstellung deklarativer Modelle präsentiert. Diese können einerseits auf Konsistenz geprüft werden und andererseits als Basis für automatische Problemlösungsprozesse in verschiedenen Anwendungskontexten dienen.

Automatische Verarbeitung deklarativer Modelle

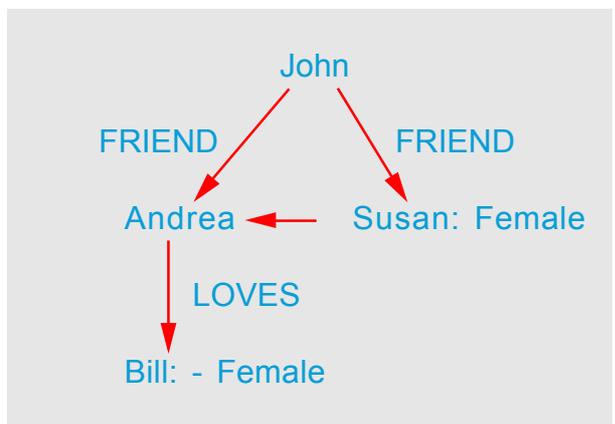
Um eine automatische Verarbeitung von deklarativen Modellen zu ermöglichen, muss die Modellierungssprache auf einer formalen Semantik beruhen. Damit lassen sich dann Inferenzprobleme formal definieren. Das Lösen eines Inferenzproblems eines bestimmten Typs durch ein Softwaresystem wird häufig auch als Inferenzdienst bezeichnet. Algorithmen zur Realisierung von Inferenzdiensten können unter Bezugnahme auf die Semantik der Repräsentationssprache daraufhin bewertet werden, ob sie vollständig und korrekt sind und auch terminieren. Je nach Ausdrucksmächtigkeit der Repräsentationssprache kann die Lösung von Inferenzproblemen durch unterschiedliche Verfahren erfolgen.

So kann zum Beispiel ein Inferenzproblem durch Suche über einem extensional gegebenen Datenbestand gelöst werden (vgl. z.B. klassische Datenbanksysteme). Das Inferenzproblem wird hier auch als Anfragebeantwortung bezeichnet. Bei komplexeren Anwendungen werden zunehmend ausdrucksstärkere Repräsentations- und Anfragesprachen untersucht, so dass die Be-

antwortung von Anfragen nicht mehr nur in einer reinen Wiedergabe der explizit gespeicherten Daten besteht. Während in Datenbanken klassischer Art nur eine definite Informationsrepräsentation erfolgen kann, also ein Tupel entweder Element einer Relation ist oder nicht bzw. ein Attributwert gegeben ist oder nicht, sind in vielen Anwendungskontexten auch indefinite Informationen zu verarbeiten. Über ein Objekt einer Anwendung kann z.B. nicht bekannt sein, ob es männlich oder weiblich ist. Trotzdem lassen sich bestimmte Anfragen eindeutig beantworten.

Ein bekanntes Beispiel ist in der Grafik wiedergegeben. Hier steht John in der Beziehung FRIEND zu Andrea und Susan. Weiterhin LOVES Susan Andrea und Andrea LOVES Bill. Über Bill sei nun bekannt, dass "nicht Female" gilt, während Susan ein Mitglied von "Female" ist. Nun könnte eine Anfrage lauten, ob für John ein Freund (Beziehung "FRIEND") existiert, der weiblich ist (für den also "Female" gilt) und der wiederum jemanden liebt (Beziehung "LOVES"), für den "nicht Female" gilt (der also männlich ist).

Die Frage scheint auf den ersten Blick nicht beantwortbar zu sein, da nichts über Andrea bekannt ist, d.h. es liegt keine definite Information vor. Nun ist allerdings zu beachten, dass eine Fallunterscheidung möglich ist. Nehmen wir an, dass Andrea weiblich ist. Dann lautet die Antwort auf die obige Anfrage "ja", denn John hat Andrea als weiblichen Freund (laut Annahme gilt "Female" für Andrea) und Andrea wiederum liebt (Beziehung "LOVES") Bill, für den "nicht Female" gilt. Im anderen Fall, wenn also für Andrea "nicht Female" gilt, lautet die Antwort ebenfalls "ja". Die Begründung kann der Leser leicht selbst



anhand der Grafik geben. Das Beispiel zeigt, dass auch bei indefiniter Information (hier über Andrea) definite Antworten möglich sind. Wird nun "Female" bzw. "nicht Female" in einer Datenbank über zwei mögliche Werte eines entsprechenden Attributes repräsentiert, dann wäre für Andrea an entsprechender Stelle ein Null-Wert eingetragen. Eine Fallunterscheidung über die möglichen Werte, wie oben skizziert, erfolgt bei Datenbanken bei der Anfragebeantwortung nicht.

Die Beantwortung von Anfragen über indefiniten Repräsentationsformen bedingt ein Schlußfolgern über implizite Informationen. Obwohl neuere Datenbanksysteme sich mehr und mehr in diese Richtung bewegen, liegt der Fokus von klassischen Datenbanksystemen doch eher auf Themen wie Persistenz und Transaktionen oder auch Effizienz in Bezug auf Speicherbedarf, Speichergeschwindigkeit und Suchgeschwindigkeit. Die Beantwortung von Anfragen, bei denen über implizite Informationen (bzw. Sachverhalte) geschlossen werden muss, fällt in den Bereich der sog. Inferenzsysteme. Da ausdrucksstärkere Repräsentationssprachen in vielen Anwendungen benötigt werden, spielen Inferenzen aber auch im Datenbankkontext eine immer größere Rolle. Es überrascht daher nicht, dass die Ergebnisse von Forschungsarbeiten zu formalen Inferenzsystemen und zur Wissensrepräsentation im Rahmen der konventionellen Informatik immer bedeutsamer werden (und umgekehrt).

Inferenzsysteme und deren Anwendung

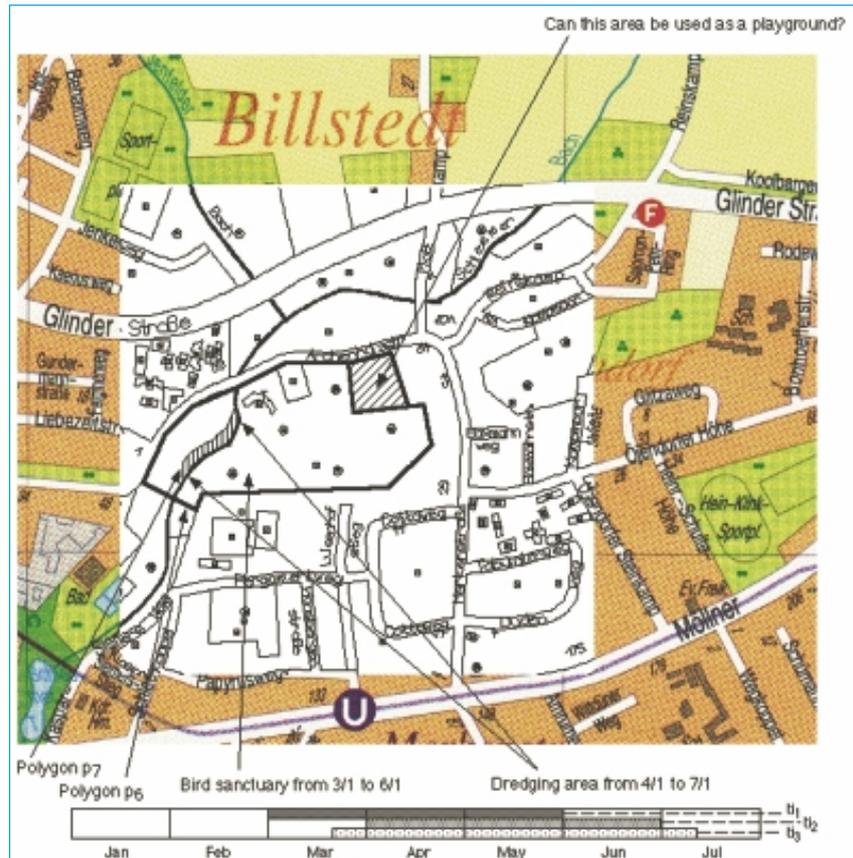
Unter dem Begriff (formale) Inferenzsysteme seien Systeme zusammengefasst, die Theoreme in einer entsprechend ausdrucksstarken Logik beweisen können. Wichtige Anwendungen von Inferenzsystemen sind sog. Informationssysteme: Diese liefern Auskünfte über einen bestimmten Informationsbestand. Werden die Hauptleistungen durch formale Inferenzsysteme für ausdrucksstarke Beschreibungssprachen erbracht, spricht man von einem deduktiven Informationssystem. Ein derartiges System muss allerdings nicht auf einer monolithischen Architektur basieren, es kann auch durch mehrere kooperierende Teilkomponenten realisiert sein. In jüngster Zeit haben In-

ferenzsysteme eine große Bedeutung für die Informationsrecherche in verteilten Systemen gewonnen (vgl. z.B. die Initiative des Semantic Web).

Die Entwicklung von formalen Inferenzsystemen bzw. Theorembeweisern hat eine lange Tradition, wobei unterschiedliche Sichtweisen mit verschiedenem mathematischem Hintergrund Eingang fanden. Für viele Anwendungsprobleme eignen sich *logikbasierte* Modellierungs- und Schlussysteme. Hier wird in diesem Kontext ein anwendungsorientierter Ansatz verfolgt, in dem theoretische Resultate über die Entscheidbarkeit von ver-

schiedenen Repräsentationssprachen durch praktische Arbeiten zu effizienten Beweisprozeduren ergänzt werden. Effiziente Beweisprozeduren wiederum sind notwendig für die praktische Realisierung von Anfragebeantwortungssystemen mit ausdrucksstarken Repräsentationssprachen. Die Entwicklung von Algorithmen, die vollständig und korrekt sind sowie terminieren, aber im Durchschnittsfall nicht in eine kombinatorische Explosion hineinfluten, ist ein vielversprechendes Forschungsfeld.

Wie schon erwähnt, können formale Beweissysteme in einem Informationsrecherche-Szenario zur Beantwortung von Anfragen eingesetzt werden. In diesem Kontext müssen Repräsentationssysteme für die Praxis eine adäquate Ausdruckskraft für eine breit gefächerte Art von Phänomenen und Effekten aufweisen. So bildet zum Beispiel für geographische Informationssysteme (GIS) räumliches Hintergrundwissen über Artefakte und natürli-



Geografisches Informationssystem



Informationssystem RACER

che Objekte die Basis für Problemlösungsprozesse. Wenn Anwendungsprobleme durch Inferenzdienste gelöst werden sollen, ist die Festlegung einer formalen Semantik für die verwendeten Repräsentationskonstrukte unumgänglich, da nur so die Aufgabe von Inferenzdiensten definiert werden kann. Dabei ist das Zusammenspiel von räumlichem bzw. zeitlichem und terminologischem Wissen auf einer semantischen Ebene von besonderer Bedeutung. Die systematische Integration des konzeptuellen, begrifflichen Schließens mit Aspekten des räumlichen Schließens ist ein noch junges Forschungsgebiet, zu dem in der hier beschriebenen Arbeit durch die Entwicklung von vollständigen und korrekten Inferenzalgorithmen wichtige

Grundlagen gelegt wurden. Um eine automatische Verarbeitung von deklarativen Modellen zu ermöglichen, sollten auch im Kontext von Informationsrechersystemen vorzugsweise Repräsentationssprachen untersucht werden, für die vollständige und korrekte, terminierende Inferenzalgorithmen existieren.

Informationssysteme sind nicht die einzigen Anwendungsfelder, in denen formale Inferenzsysteme erfolgreich eingesetzt werden können. Durch das stetig wachsende Interesse am elektronischen Handel in verteilten Systemen wird das Problem der Verifikation vordringlich. Es geht dabei darum, durch Auswertung eines Modells zu prüfen, dass bestimmte, meist interagierende Dienste auch unter allen Umständen erbracht werden können. Als weiteren bedeutsamen Anwendungskontext für die formale Verifikation sind Telekommunikationssysteme zu nennen. In diesen Systemen wird eine Vielzahl von ebenfalls stark interagierenden Diensten angeboten. Die Kombinierbarkeit von Diensten stellt dabei ein großes Problem dar. Daher besteht vielfach der Wunsch, ausgehend von einem formalen Modell eines Systems und einer Menge von Einschränkungen (Invarianten), zu prüfen, ob ein fehlerfreier Betrieb des Systems gewährleistet werden kann. In diesem Kontext werden vielfach sogenannte Algorithmen zur Modellüberprüfung (*Model*

Checking Algorithms) eingesetzt. Diese prüfen, ob eine spezielle Realisierung eines Systems bestimmte Anforderungen erfüllt. Grundannahme bei der Modellierung ist dabei, dass das Verhalten des Systems als endlicher Zustandsraum repräsentierbar ist. Zustandsübergänge werden durch Aktionen bzw. Ereignisse ausgelöst. Anforderungen legen fest, dass in bestimmten Zuständen bestimmte Prädikate gelten müssen. Die Modellüberprüfung erlaubt dann eine Aussage darüber, ob eine gegebene Systemrealisierung die Anforderungen erfüllt.

Problematisch ist dabei, dass nur endliche Zustandsräume behandelt werden können und auch nur zustandsorientierte Aspekte von Anwendungen betrachtet werden können. Daher werden in zunehmendem Maße auch Theorembeweiser zur Überprüfung von Spezifikationen eingesetzt. Theorembeweiser können mit unendlichen Zustandsräumen umgehen und unterstützen mit einem Konsistenztest auch eine Überprüfung der Datenmodelle.

Weitere Anwendungsgebiete, wie z.B. das Verstehen natürlicher Sprache oder auch die Konstruktion bzw. Konfiguration von technischen Geräten, sollen hier nur erwähnt werden. Damit Deduktionstechniken erfolgreich in Anwendungen eingesetzt werden können, müssen ausdrucksstarke Modellierungs- und Repräsentationssprachen verwendet werden und effiziente Beweiserimplementierungen verfügbar sein.

Inferenzsysteme für ausdrucksstarke Beschreibungslogiken

Eine Familie von logikbasierten Repräsentationssprachen, die für die obengenannten Aufgaben sehr gut geeignet ist, wird als Beschreibungslogiken bezeichnet (*Description Logics, DLs*). Mit Beschreibungslogiken wird eine rechnerinterne Modellierung von Objekten und Phänomenen der Realwelt vorrangig aus einer objektzentrierten Sicht vorgenommen. Es werden hierzu Konstrukte zur Formulierung von sog. Konzepten und Rollen (binäre Relationen) sowie entsprechende, auf der formalen Semantik der Repräsentationssprache basierende Inferenzdienste bereitgestellt (z.B. Konsistenzprüfung, Klassifi-

kation). In letzter Zeit sind auch Beschreibungslogiken mit n-stelligen Relationen untersucht worden. Lassen sich Teilprobleme einer Anwendung unter Rückgriff auf Beschreibungslogiken als entsprechende Inferenzprobleme formulieren, so stehen theoretisch abgesicherte und systematisch getestete Beweissysteme zur Lösung dieser Teilprobleme zur Verfügung.

Erfahrungen mit Anwendungen belegen, dass ausdrucksstarke Formalismen benötigt werden, damit nicht auf Ad-hoc-Lösungen für Teilprobleme zurückgegriffen werden muss. Die Entwicklung von Inferenzsystemen, die auf vollständigen und korrekten Algorithmen basieren und im mittleren Fall (*average case*) ein gutes Laufzeitverhalten zeigen, ist für ausdrucksstarke (aber entscheidbare) Beschreibungslogiken ein relativ neues Forschungsgebiet. Innerhalb der durchgeführten Forschungsarbeiten wurden zu den Problemstellungen der aufgezeigten Bereiche Lösungen erarbeitet. Folgende Hauptpunkte sind hervorzuheben:

- Theoretische Arbeiten zur Entwicklung von Kalkülen für ausdrucksstarke Beschreibungslogiken.
- Entwicklung von optimierten, vollständigen und korrekten Algorithmen für praxiserhaltende Schlussverfahren (insbesondere für das Schließen mit Individuen).
- Implementierung eines Beschreibungslogik-Inferenzsystems namens *RACER* zur Durchführung von Untersuchungen zum durchschnittlichen (*average case*) Verhalten von Inferenzalgorithmen für Anwendungswissensbasen. Hierzu gehören empirische Analysen der Performanz des Inferenzsystems *RACER* z.B. für Formalisierungen von großen Ontologien (*Ontology Engineering*).
- Erweiterung der Theorie zu Beschreibungslogiken zur Einbeziehung von Schlüssen bzgl. Einschränkungen über kontinuierlichen und diskreten Domänen (sog. *constraints*) mit Anwendungen im Bereich der Konfigurierung.
- Theoretische Arbeiten zur semantikbasierten Integration von Techniken zum räumlichen (und zeitlichen) Schließen in das konzeptuelle Schließen mit Beschreibungslogiken.
- Aufzeigen der praktischen Verwendung der beschreibungslogischen Model-

lierungskonstrukte für Anwendungen aus verschiedenen Kontexten: Inferenz-basierte bzw. deduktive Informationssysteme, geographische Informationssysteme und Agenten-orientierte Informationssysteme.

In der Arbeit wird aufgezeigt, dass sich Beschreibungslogiken in vielfältiger Weise zur Modellierung und Problemlösung in Anwendungssystemen einsetzen lassen. Obwohl nicht bestritten wird, dass in bestimmten Kontexten auch andere Logikformalismen erfolgreich als Basis von Inferenzsystemen zum Einsatz kommen können (z.B. Datalog in sog. deduktiven Datenbanken), bildet gerade das breite Anwendungsspektrum für Beschreibungslogiken die Motivation für die aufgezeigten Forschungsarbeiten.

Zusammenfassung

Mit der Arbeit konnten Grundlagen für ein Repräsentationsmedium entwickelt werden, so dass mit automatischen Inferenzdiensten Lösungen für (Teil-)Probleme einer Anwendung berechnet werden können. Dadurch ist eine kosteneffizientere Entwicklung von robusten und anpassbaren Anwendungen mit erweiterter Funktionalität möglich. Die Entwicklung von optimierten Algorithmen verdeutlicht, dass die Dienste, die durch automatische Inferenzsysteme erbracht werden, nicht ohne weiteres durch Ad-hoc-Techniken der Softwareentwicklung realisiert werden können. Wissenschaftliche Arbeit ist notwendig, um Beweistechniken anzuwenden, so dass Korrektheit, Vollständigkeit und Terminierung auf der Ebene der Algorithmen gezeigt werden können. Dedizierte Programmentwicklungs- und Testtechniken garantieren, dass die Implementation dieser Algorithmen (möglichst) fehlerfrei ist. Einmal entwickelt, können Inferenzsysteme in verschiedenen Anwendungskontexten wirkungsvoll zum Einsatz kommen.

Formale Inferenzsysteme und speziell auch Beschreibungslogiken können als sich wandelnde Disziplinen angesehen werden. Vor einigen Jahren noch bestand eine große Kluft zwischen Theorie und Praxis. Mit der Entwicklung von Optimierungstechniken jedoch zeigt sich, dass theoretische Ergebnisse bezüglich Entscheidbarkeit und Komplexität direkt bedeutsam sind für die Implementierung von

praktischen Inferenzsystemen. Die Notwendigkeit von Optimierungstechniken in einer praktischen Implementierung stellt neue Anforderungen an theoretische Arbeiten bzgl. Beweistechniken, Analyse von Durchschnittsfällen usw. Neue Optimierungstechniken, die in *RACER* implementiert sind und auf der Basis von theoretischen Ergebnissen entwickelt wurden, ermöglichen die Realisierung von neuartigen Anwendungen.

Nicht zuletzt die diskutierten Anwendungsbeispiele belegen die Signifikanz der erzielten Ergebnisse für die Wissensrepräsentation und die Informatik generell. Vieles wurde erreicht, doch vieles ist auch noch offen. Es bleibt zu hoffen, dass durch diese Arbeit neue Forschungen angeregt werden.



Ralf Möller; FH Wedel

Der Beitrag beruht auf der Habilitationsschrift des Autors (R. Möller: *Expressive Description Logics: Foundations for Practical Applications*; Universität Hamburg, Fachbereich Informatik (2001))