

Reasoners for the Semantic Web

Kaon & Kaon2

Seminar A.I. Tools
Universität des Saarlandes
Wintersemester 2006/07

Erik Endres
email@endrese.de

Dozenten: Michael Kipp
Alassane Ndiaye
Dominik Heckmann
Michael Feld

Abstract

Nach einem Vorschlag des World Wide Web Begründers Tim Berners-Lee soll dieses um maschinenverstehbare Daten erweitert werden. Das so erweiterte Netz wird auch Semantisches Web genannt und die maschinenverstehbare Daten werden in den meisten Fällen in Form einer Ontologie realisiert. Ist das Wissen maschinenlesbar vorhanden, benötigt man Software, die diese Daten verarbeitet. Diese Funktion wird durch Reasoner übernommen. Sie können über gegebene Ontologien schlussfolgern und damit Wissen ableiten, das nicht explizit beschrieben ist. Dadurch können Reasoner Anfragen beantworten, die Daten auf Konsistenz prüfen und vollständig klassifizieren. Das Semantische Web stellt aufgrund seiner Eigenschaften besondere Anforderungen an einen Reasoner.

Inhaltsverzeichnis

| | |
|---|----------|
| 1 Einleitung..... | Seite 1 |
| 2 Reasoner..... | Seite 1 |
| 2.1 Grundlagen..... | Seite 1 |
| 2.2 Prädikatenlogik & Beschreibungslogik..... | Seite 3 |
| 2.3 Inferenz..... | Seite 3 |
| 2.4 Aufgaben & Anforderungen..... | Seite 4 |
| 2.5 Beispiele..... | Seite 6 |
| 3 Kaon..... | Seite 7 |
| 4 Kaon 2..... | Seite 8 |
| 5 Zusammenfassung..... | Seite 10 |
| 6 Literaturangaben..... | Seite 11 |

1 Einleitung

An das Semantische Web werden große Erwartungen gestellt. Suchmaschinen sollen auf die schwierigsten Anfragen schnell und sicher die richtige Antwort finden, Informationen sollen automatisch auf ihre Korrektheit überprüft werden können. Damit das funktionieren kann, müssen die Informationen in einer maschinenlesbaren Form vorhanden sein. Dies geschieht in Form einer Ontologie, aber dies ist nur die Voraussetzung um das Semantische Web zu realisieren. Sobald genügend Informationen vorhanden sind, muss Software diese Information auswerten können. Diese Arbeit erledigen die Reasoner, die die Information nicht nur wiedergeben und durchsuchen können, sondern auf der Basis einer formalen Logik auch implizit vorhandenes Wissen ableiten können. Allerdings werden sie für die besonderen Anforderungen im Semantischen Web vorbereitet werden müssen. Im folgenden werden Reasoner und die Grundlage, auf der sie arbeiten, besprochen. Danach wird als ein Vertreter dieser Software KAON2 und deren Vorgänger KAON vorgestellt.

2 Reasoner

2.1 Grundlagen

Ontologien stellen eine formale Wissensrepräsentation dar, die es ermöglicht Wissen in einer maschinenlesbaren Form festzuhalten. Im Unterschied zu Taxonomien, die nur eine hierarchische Gliederung darstellen, stellen Ontologien ein Netzwerk von Informationen mit logischen Relationen dar. Zusätzlich enthalten Ontologien Inferenz- und Integritätsregeln, die das logische Schließen und eine Überprüfung der Gültigkeit ermöglichen. Auf dieser Wissensrepräsentation setzen Reasoner auf und leiten neues Wissen durch Inferenz ab. Daher werden Reasoner auch Inferenzer genannt. Als Inferenz oder Schlussfolgern bezeichnet man den Vorgang, wenn aus einer Anzahl von Fakten, den Prämissen, ein neues Faktum abgeleitet wird, die Konklusion.

Reasoner

| |
|--|
| <p>Prämisse</p> <p>„Alle Griechen sind sterblich“,</p> <p>„Sokrates ist ein Grieche“</p> <p>Konklusion</p> <p>„Sokrates ist sterblich“</p> |
|--|

Beispiel 2.1 Schlussfolgerung

Als formale Beschreibungssprache für das Semantische Web wurde vom World Wide Web Consortium die Web Ontologie Language (OWL) spezifiziert. Die Syntax von OWL basiert auf RDF und erlaubt es, Ausdrücke ähnlich der Prädikatenlogik zu formulieren. Zur Veranschaulichung stellt Beispiel 2.2 die Prämisse des Beispiels 2.1 in OWL dar. [2]

```
<owl:Class rdf:ID="Grieche">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue>
        <Sterblichkeit rdf:ID="Sterblich"/>
      </owl:hasValue>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hat_Lebensspanne"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<Grieche rdf:ID="Sokrates"/>
```

Beispiel 2.2 Prämisse aus Bsp. 2.1 in OWL Syntax

OWL gibt es in drei unterschiedlich mächtigen Ausprägungen OWL Full, OWL DL und OWL Lite. Wobei für die Reasoner OWL DL von besonderem Interesse ist, da es sich hierbei um eine entscheidbare Untermenge der Prädikatenlogik handelt. OWL DL ist den Beschreibungslogiken zuzurechnen, was auch durch den Namen verdeutlicht

wird; DL steht für description logic. Damit man die Arbeitsweise von Reasonern besser verstehen kann, lohnt sich zunächst ein kurzer Blick auf die zugrunde liegenden Logiken. [3]

2.2 Prädikatenlogik & Beschreibungslogik

Die Prädikatenlogik ist eine Familie logischer Systeme, die die Aussagenlogik erweitert. Sie wird auch Quantorenlogik genannt. In der Prädikatenlogik lassen sich Aussagen wie: „Sokrates ist ein Grieche“ in das Prädikat „_ ist ein Grieche“ und den Bezeichner „Sokrates“ unterteilen. Zusätzlich können die Bezeichner mit Quantoren versehen werden: „Alle Griechen sind sterblich“. Hier ist „_ sind sterblich“ das Prädikat, „Griechen“ der Bezeichner und „Alle“ der Allquantor. Zusätzlich gibt es den Existenzquantor und die Identität. Eine direkte Umsetzung der Prädikatenlogik in ein Programmiersystem stellt die Programmiersprache Prolog dar.

Die Beschreibungslogiken sind eine Untermenge der Prädikatenlogik, aber sind im Unterschied zu dieser entscheidbar. Beispiele für unentscheidbare Konstrukte sind das Russell oder Barber Paradox¹. So trägt in Prolog der Programmierer die Verantwortung dafür, dass das geschriebene System entscheidbar ist. Im Semantischen Web wäre dies allerdings keine geeignete Lösung. [2]

Beschreibungslogiken werden in eine T-Box und eine A-Box unterteilt. Die T-Box enthält sämtliche Konzepte und die A-Box alle Instanzen dieser Konzepte.

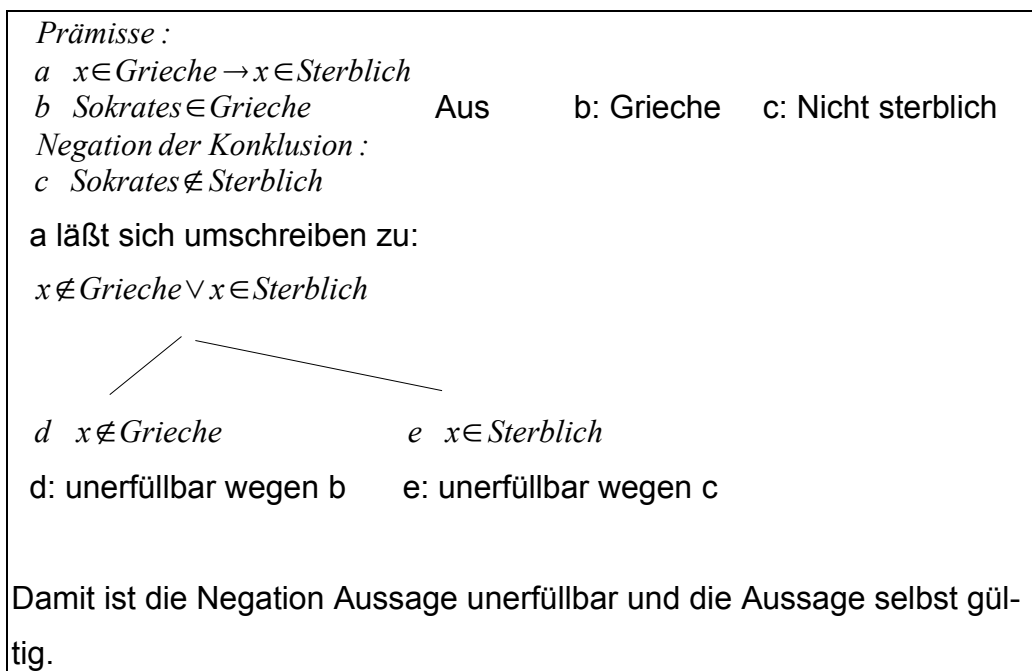
2.3 Inferenz

Die Regeln zum Schließen über eine bestimmte Logik werden auch Kalkül genannt. Konkrete Kalküle für Prädikatenlogiken sind zum Beispiel: Hilbertkalkül, Baumkalkül und Resolution. [1]

¹ Das Russell oder Barber Paradox stammt vom Mathematiker Bertrand Russell und lautet wie folgt: „Der Barbier von Sevilla rasiert alle Männer von Sevilla, nur nicht die, die sich selbst rasieren.“ [2]

Reasoner

Bei den Reasonern wird meistens das Baumkalkül, auch Tableaux Methode oder Bethkalkül genannt, verwendet. Bei dieser Methode wird nicht die Gültigkeit einer Aussage bewiesen, sondern die Ungültigkeit widerlegt. Daher werden solche Kalküle auch als Widerlegungskalküle bezeichnet. Die Ungültigkeit wird dabei durch die Unerfüllbarkeit der Negation der Aussage gezeigt. Die Negation der Aussage besteht dabei aus der Disjunktion von Prämisse und der negierten Konklusion. [2] Im folgenden Beispiel wird durch das Baumkalkül die Gültigkeit des Beispiels 2.1 gezeigt:



Beispiel 2.3 Anwendung des Baumkalküls auf Bsp.2.1

Mit steigender Ausdrucksmächtigkeit der Sprache wird auch das Schließen immer schwieriger, was sich direkt auf die Anforderungen an die Reasoner auswirkt. So stellt Prolog vollständig die Prädikatenlogik dar, verlangt aber zur Verwendung das Erlernen einer vollständigen Programmiersprache. [4]

2.4 Aufgaben & Anforderungen

Die Aufgaben der Reasoner sind:

- Klassifikation
- Konsistenzprüfung

Reasoner

- Bearbeiten von Anfragen

Zur Erfüllung dieser Aufgaben sind an den Reasoner besondere Anforderungen gestellt:

- Korrektheit
Alle vom Reasoner abgeleiteten Aussagen müssen gültig sind.
- Vollständigkeit
So sind bei der Klassifikation sämtliche Konzepte zu finden denen eine Instanz zugeordnet ist.
- Entscheidbarkeit
Der Reasoner sollte in endlicher Zeit feststellen können, ob eine Aussage gültig oder ungültig ist.
- Effizienz
Der Reasoner muss eine Anfrage möglichst effizient beantworten können.

Die Erfüllbarkeit dieser Anforderungen ist im wesentlichen durch die Ausdrucksmächtigkeit der Repräsentationsprache bestimmt. Über sehr einfache Repräsentationssprachen kann ein Reasoner auch sehr effizient schließen. Darüber hinaus werden im Semantischen Web weitere Anforderungen an einen Reasoner gestellt:

- Skalierbarkeit
Das Semantische Web ist ein sich rasant vergrößerndes Netz und wird wohl auch in der Zukunft weiter wachsen. Es ist wichtig, dass ein Reasoner auch mit einer sich stark vergrößernden Wissensbasis effizient umgehen kann.
- Adaptive Performanz
Das Semantische Web ändert sich permanent. Neues Wissen kommt hinzu, anderes Wissen verschwindet wieder. Ein Reasoner muss daher auch mit einer sich ständig ändernden Wissensbasis effizient umgehen können.
- Robustheit
Das WWW steckt voller fehlerhafter Informationen. Auch das Semantische Web wird mit fehlerhaften Daten umgehen können müssen. *„Inkonsistenzen, wie widersprüchliche Formeln, Ausnahmen und semantische „broken links“ dürfen die Funktionsfähigkeit des Inferenzsystems nicht beeinträchtigen.“* [5]

Reasoner

- **Integration verteilten Wissens**
Ontologien können eine Vielzahl weiterer Ontologien, die im gesamten Semantischen Web verteilt sind, einbinden. Ein Reasoner muss diese sinnvoll miteinander verbinden können.
- **Verständlichkeit**
Das Semantische Web soll als Erweiterung des WWW auch weiterhin von Menschen benutzt werden können, ohne dass diese eine lange Einarbeitungszeit benötigen. Daher müssen auch die Reasoner einfach zu benutzen sein.

Diese Anforderungen im Semantischen Web stehen zum Teil im Gegensatz zu den normalen Anforderungen an einen Reasoner. So ist die Vollständigkeit meist weniger wichtig als die Skalierbarkeit. Für eine semantische Suche ist es wichtiger, dass die Ergebnisse nach wenigen Sekunden statt nach Stunden vorliegen, als dass bei einer Suche wirklich jeder Treffer gefunden wird. Dabei sollte die Korrektheit aber möglichst nicht verloren gehen. [5]

2.5 Beispiele

FaCT++

ist ein Open Source Reasoner auf C++ Basis für OWL

Pellet

ist ein Open Source Reasoner auf Java Basis für OWL

RacerPro

ist ein kommerzieller Reasoner auf Java Basis für OWL [6]

Alle oben vorgestellten Reasoner arbeiten mit dem Baumkalkül. Im Gegensatz dazu arbeitet **KAON2** mit einer anderen Methode und wird im Folgenden vorgestellt.

3 Kaon

Der Name KAON Toolsuite steht für **K**arlsruhe **O**ntology and Semantic Web tool suite. Sie wurde von der Universität Karlsruhe und dem Forschungszentrum Informatik in Karlsruhe entwickelt. Die erste Version erschien im Jahr 2002 als Open Source Software. Die KAON ToolSuite besteht aus:

- KAON
enthält Frontend und Core
- KAON Extensions
enthält DLP, KAON Server und KAONtoEdit
- TextToOnto

Den Aufbau der kompletten Toolsuite stellt Diagramm 3.1 dar.

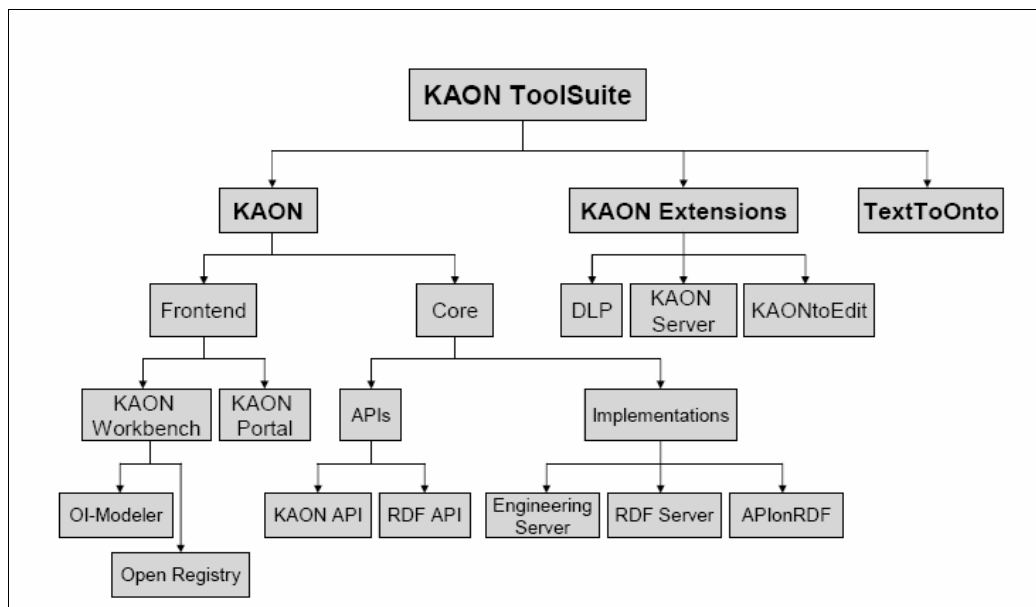


Diagramm 3.1 Übersicht KAON ToolSuite [7]

KAON kann Ontologien auf RDF Basis erzeugen und effizient in Datenbanken ablegen. Die Erweiterung DLP (Description Logic Programs) ermöglicht einen effizienten Reasoner, indem es die Ontologie in ein Logikprogramm übersetzt. [7]

Auf dieser Basis wurde KAON2 entwickelt.

4 Kaon 2

Im Gegensatz zur KAON Toolsuite ist KAON2 keine Open Source Software mehr, sondern wird durch die Karlsruher Firma Ontoprise GmbH vertrieben. Allerdings ist die Nutzung für Forschung und Lehre weiterhin kostenlos. Die erste Version erschien im Jahr 2005. KAON2 besitzt keine Benutzeroberfläche, sondern wird nur noch über die API oder über das DIG Interface gesteuert.

Gegenüber KAON wurden folgende Verbesserungen vorgenommen:

API für OWL-DL

KAON2 wurde um eine API für OWL-DL, SWRL und F-Logik erweitert. Dabei sind die Sprachen hinsichtlich des Reasoning eingeschränkt. OWL-DL ist auf die SHIQ(D) Teilmenge beschränkt, die keine Enumeration enthält.

```
<owl:Class>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Sterblich"/>
    <owl:Thing rdf:about="#Unsterblich"/>
  </owl:oneOf>
</owl:Class>
```

Beispiel 4.1 Enumeration in OWL Syntax

DIG Interface

Anwendungsprogrammierer arbeiten meistens mit vollkommen anderen Programmiersprachen als die Reasoner. Durch die gestiegene Bedeutung von Web Services, die wieder auf anderen Sprachen basieren, wird dies noch verstärkt. Deshalb wurden mehrere Entwürfe für ein unabhängiges Interface erstellt. Dies hat verschiedene Vorteile: Man ist von der Programmiersprache unabhängig, die API kann mit einer zum Verwendungszweck passenden Formalismus definiert werden, es wird ein Mechanismus zur Kommunikation mit dem Beschreibunglogiksystem bereitgestellt und das Beschreibunglogiksystem kann verändert werden ohne die Anwendung zu beeinflussen.

Diese Idee wurde zuerst in einem CORBA Interface für den FaCT Reasoner implementiert. Da dieses sich aber noch sehr an der zugrunde liegenden Lisp-Syntax orientierte, konnten Sprachen wie DAML+OIL nicht verarbeitet werden. Auch das RACER System wurde durch ein sprachenunabhängiges API erweitert, das allerdings mit einem geringen Abstraktionsgrad dem Programmierer viel Verantwortung übertrug. Schließlich wurde durch die Description Logic Implementation Group (DIG) ein neues Interface definiert.

Das DIG besteht aus XML Schemas, daraus abgeleiteten Java XMLBeans und einer Java Reasoners API, die die XMLBeans benutzt. Fast alle neueren Reasoner, wie auch KAON2, sind zu der Reasoner API kompatibel. [8]

Inferenz Engine

KAON2 enthält nun einen Reasoner für Anfragen in SPARQL Syntax oder F-Logik. SPARQL ist eine Abfragensprache für das Semantische Web, die durch das W3C als Nachfolger für RDF Query Language, RDQL, entwickelt wurde. Diese Inferenz Engine arbeitet allerdings nicht mit dem Baumkalkül, sondern übersetzt die Wissensbasis in ein disjunktes Datalog Programm. Datalog ist eine auf Prolog basierende Datenbankprogrammiersprache.

Durch die Übersetzung in ein Datenbankprogramm kann KAON2 bei sehr großen Ontologien mit sehr großen Instanzenmengen sehr effizient arbeiten. Für die T-Box ist allerdings ein höherer Aufwand notwendig. Da im Semantischen Web normalerweise die A-Box einer Ontologie wesentlich größer ist, als die T-Box, ist KAON2 hier ein sehr effizienter Reasoner. [9]

5 Zusammenfassung

Das Semantische Web wird eine wesentliche Verbesserung des WWW sein, und dabei spielen Reasoner eine zentrale Rolle. Ist erst einmal das Wissen als Ontologie vorhanden, können Reasoner damit sehr effizient und gezielt Anfragen beantworten. Allerdings müssen sie dabei auf die besonderen Anforderungen im Semantischen Web vorbereitet werden. In dieser Hinsicht ist KAON2 besonders interessant, da es Ontologien mit sehr großen Instanzenmengen überaus effizient verarbeiten kann.

6 Literaturangaben

- [1] Brockhaus Enzyklopädie
in 24 Bänden, 19. Auflage, Brockhaus GmbH, Mannheim
- [2] www.wikipedia.de
Stand 18.03.2007
- [3] <http://www.w3.org/2004/OWL>
Stand 18.03.2007
- [4] Description Logics Basics, Applications, and More
www.cs.man.ac.uk/~horrocks/Slides/ecai-handout.pdf
- [5] Inferenzsysteme für das Semantic Web
Steffen Balzer, Universität Ulm Abteilung Künstliche Intelligenz, 28. Februar 2003
- [6] <http://www.cs.man.ac.uk/~sattler/reasoners.html>
Stand 18.03.2007
- [7] KAON – An Overview
Thomas Gabel, York Sure and Johanna Voelker, Universität Karlsruhe Institut AIFB, 7. April 2004
- [8] <http://dig.sourceforge.net>
Stand 16.03.2007
- [9] <http://kaon2.semanticweb.org>
Stand 18.03.2007