Projekt: Tableaubeweiser für Beschreibungslogiken
Protokoll zur Sitzung am 15.11.2006

Felix Lindner

17.11.2006

1 Fortführung der Besprechung von "Basic Description Logics", Baader & Nutt

30) Können bestehende Konflikte ('clash') in einem Tableau-Zweig durch nachfolgende Expansionen wieder aufgehoben werden? Zu welchem Zeitpunkt sollte jeweils am besten überprüft werden, ob ein Konflikt besteht?

Eine ABox $\mathcal{A}$ beinhaltet einen *clash* gdw. eines der drei folgenden Kriterien zutrifft:

(i) $\{\bot(x)\} \subseteq \mathcal{A}$ für ein Konstantensymbol x
(ii) $\{A(x), \neg A(x)\} \subseteq \mathcal{A}$ für ein Konstantensymbol x und ein Konzept A.
(iii) $\{(\leq nR)(x)\} \cup \{R(x, y_i) | 1 \leq i \leq n+1\} \cup \{y_i \neq y_j | 1 \leq i < j \leq n+1\} \subseteq \mathcal{A}$ für Konstantensymbole $x, y_1, \ldots, y_{n+1}$, ein $n > 0$ und ein Rollensymbol R.

Von Interesse war insbesondere die dritte Regel. Steht x mit n+1 vielen paarweise verschiedenen y_i über die Rolle R in Beziehung, dann liegt ein Konflikt vor. Als Beispiel wurde hier genannt:

> Studenten nehmen an höchstens 1000 Lehrveranstaltungen teil.
> Es findet sich ein Student, der an 1001 Lehrveranstaltungen teilnimmt.
> *Clash.*

Bestehende Konflikte werden durch nachfolgende Expansionen innerhalb eines Tableau-Zweiges nicht aufgehoben. Dies gilt auch für (iii), denn das Wissen, dass die y_i ungleich sind, bleibt erhalten.

31) In 2.3.2.2. wird erläutert, wie der Platzbedarf für die Expansion systematisch verringert werden kann. Gib ein Beispiel an, bei dem der vorgestellte Weg hilfreich ist. Welche Eigenschaft der Beschreibungslogiken erlaubt diese Vereinfachung?

Baader & Nutt unterbreiten Vorschläge, die Entscheidung, ob ein $\mathcal{ALCN}$-Konzept erfüllbar ist, in PSpace-Komplexität zu erreichen. Ein Vorschlag besteht darin, die Zahlen in den Mindestrestriktionen zur Basis 1 zu repräsentieren. Für größere Basen könnte die zu repräsentierende Anzahl exponentiell zur Größe der Repräsentation sein. Durch die lange Kodierung des Problems aber steigt der Platzbedarf, der durch die Expansion hinzukommt, relativ zur Größe der Repräsentation nicht mehr in exponentiellem Maße an. Letztlich wird dadurch aber kein Platz gespart.

Des Weiteren besteht nicht die Notwendigkeit, alle Nachfolger der Mindestregel zu generieren. Wurde bereits durch eine $\rightarrow_{\exists}$-Regel ein Nachfolger generiert, dann muss die $\rightarrow_{\geq}$-Regel gar nicht angewandt werden. Ansonsten reicht es, einen einzigen Nachfolger einzuführen. Auf diese Weise wird festgestellt, ob es überhaupt sinnvoll ist, Individuen anzunehmen, die das Konzept erfüllen könnten. Um die Inkonsistenz bezüglich wiedersprüchlicher Anzahlrestirktionen aufzudecken, werden die Zahlen direkt verglichen.
Das Clash-Kriterium lautet $\{(\leq n\, R)(x), (\geq m\, R)(x)\} \subseteq \mathcal{A}, n, m \in \mathcal{N}, n < m$. Beispiel: $(\geq 5)R(x), (\leq 3)R(x)$

In diesem Zusammenhang wurde auf den Fall hingewiesen, dass $R \sqsubseteq T$ (Rolleninklusion). Dieser muss berücksichtigt werden.
Beispiel: $hatTochter \sqsubseteq hatKind$, $(\geq 5)hatKind(x)$ $(\leq 3)hatTochter(x)$

So wie auf Seite 85 ("Basic Description Logics") beschrieben, werden wir die Mindestregel nicht anwenden.

33) Welche TBox-Axiome lassen sich nicht durch Expansion der Beschreibungen verarbeiten und müssen daher in der verallgemeinerten Fassung des Tableau-Verfahrens behandelt werden? Gib weitere Fälle an, die nicht explizit von Baader & Nutt in 2.3.2.4 genannt werden.

Es können solche TBox-Axiome nicht durch Expansionen der Beschreibungen behandelt werden, die in der Form $C \sqsubseteq D$ vorliegen, wobei C und D komplexe Konzepte sind (generalisierte Inklusionsaxiome).

Anstatt zu expandieren fassen wir die TBox in einem einzigen Axiom $\top \sqsubseteq C_{TBox}$ zusammen. Dabei enthält das Konzept C_{TBox} die gesamte Information der TBox. Das Axiom besagt, dass jede Konstante zu diesem Konzept gehören muss. Bei Konstanteneinführungen von a_i durch $\rightarrow_\exists$-Regel bzw. $\rightarrow_\geq$-Regeln wird geprüft, ob $C_{TBox}(a_i)$ erfüllt ist. Dadurch verringert sich aber auch die Komplexität der Ausdrücke entlang eines Tableau-Zweiges nicht und die Termination des Verfahrens ist nicht mehr ohne Weiteres sichergestellt. Als Notbremse fungieren Blockierungsmechanismen.

35) Wie funktioniert das Blockieren der Regeln $\rightarrow_\exists$ und $\rightarrow_\geq n$? Wann ist zu prüfen, ob Blockierung vorliegt? Wie wird eine Blockierungskonstellation semantisch interpretiert? Wie lässt sie sich in Bezug auf die Tableau-Expansion und das Aufdecken von Konflikten motivieren?

Die Anwendung der $\rightarrow_\exists$-Regel bzw. $\rightarrow_\geq$-Regel auf eine Konstante x wird in einer ABox $\mathcal{A}$ von einer Konstanten y blockiert gdw. $\{D(x) \in \mathcal{A}\} \subseteq \{D' | D'(y) \in \mathcal{A}\}$. Man darf mehr über y wissen, als über x, aber alles was man über x weiß, weiß man bereits über y. Solange keine Information vorliegt, dass x von y verschieden ist, müssen keine neuen Nachfolger von x generiert werden.
Beispiel: $\exists R.\top(y)$ wird expandiert und in der ABox steht R(y,a). Dann kennt man einen Nachfolger a von y. Die Expansion $\exists R.\top(x)$ wird daraufhin blockiert, wenn keine Information vorliegt, dass x und y unterscheidbar sind.

Erst dann, wenn man über x etwas weiß, das man über y nicht weiß, muss die Blockade aufgehoben werden. Ist die Information über x gleich der Information über y, dann reicht es, $\rightarrow_\exists$- bzw. $\rightarrow_\geq$-Regel für einen von beiden anzuwenden.

Dieses Vorgehen führt letztlich wieder zu garantierter Termination.

32) Was muss beachtet werden, wenn man mit dem Tableau-Verfahren nicht nur Konzeptkonsistenz, sondern allgemeiner ABox-Konsistenz prüfen will?

Es wurden Überlegungen angestellt, welche Modifikationen auf die ABox des Benutzers $ABox_{Benutzer}$ angewendet werden müssen, damit sie anhand des

Tableau-Verfahrens auf Konsistenz geprüft werden kann. Für $ABox_{Benutzer}$ ist es sinnvoll, die Unique Name Assumption anzunehmen, nicht aber im Beweiser selbst. Denn hier spricht gegen die Unique Name Assumption, dass Annahmen darüber getroffen werden, dass zwei Konstanten x und y gleich sind, solange über x und y nicht explizit die Verschiedenheit bekannt ist. (vgl. 35) über Blockierungsmechanismen). In einem Normalisierungsschritt könnte die Unique Name Assumption explizit gemacht werden ($ABox_{Benutzer} \Rightarrow_{UNA \rightarrow \neq} ABox_{Tableau}$).

In diesem Zusammenhang wurde ferner überlegt, ob auch die Negationsnormalform hergestellt werden sollte. Dies wurde aber nicht mit hoher Priorität versehen, da dann einige Optimierungen, die darauf abzielen, möglichst früh Konflikte zu erkennen, nicht mehr anwendbar wären.

2 Erste Planungen für die Implementationsphase

Wir haben erste Teilaufgaben spezifiziert und Designfragen aufgestellt, über die im weiteren Verlauf entschieden werden muss.

<table>
<tr><th>Aufgabe</th><th>wird bearbeitet von</th></tr>
<tr><td>- Beschreibungslogik auswählen</td><td></td></tr>
<tr><td>- Beispiel zum Testen</td><td></td></tr>
<tr><td>- Parser</td><td>RI</td></tr>
<tr><td>- Tableaubeweiser
- Expansion
- Clash-Detektion
- Blockierung</td><td>NB, AS, JW</td></tr>
<tr><td>- Zusammenspiel der Komponenten
- Reasoning-Service → Tableau-Übersetzung
- Initiales Tableau</td><td>FL, OG</td></tr>
<tr><td>- Schnittstellen: 'Benutzer', API</td><td></td></tr>
<tr><td>- Fensterschnittstelle
- Dateien Laden</td><td>JW, RI</td></tr>
<tr><td>- Ausgabe (mit Begründung)</td><td></td></tr>
<tr><td>- Syntax
- LISP-ähnlich [1]
- OWL-DL [2]
- Eigene</td><td></td></tr>
<tr><td>- Auswahl: Welche Reasoning-Services? [3]
- mit/ohne TBox
- Konsistenz
- Retrieval</td><td></td></tr>
<tr><td>- TBox: Welche Komplexität?
- evtl. Vorverarbeitung/Umformung
- interne/externe Einschränkungen</td><td></td></tr>
<tr><td>- natürliche Sprache: Englisch?</td><td></td></tr>
</table>

[1] Wir werden uns am *Concrete Syntax* (Appendix 1, Description Logic Terminology, Baader) orientieren
[2] Wird eventuell berücksichtigt. OG beschäftigt sich mit der DIG-API.
[3] In erster Version Konsistenzcheck (mit/ohne TBox), Retrieval in späteren Versionen