ulm university universität uulm

Semantic Web Grundlagen

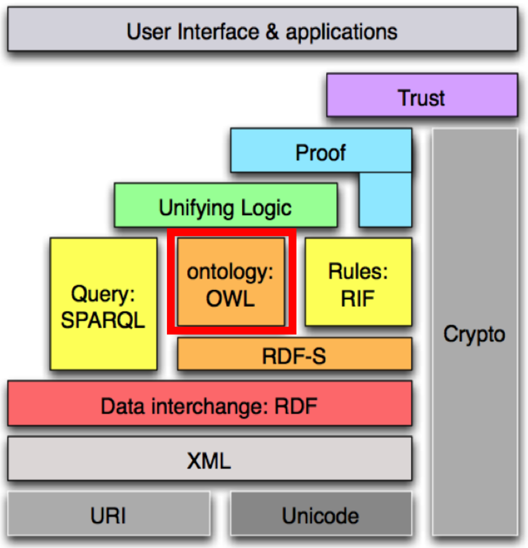
Das Hypertableau Verfahren

Birte Glimm
Institut für Künstliche Intelligenz | 08. Dez 2011

Organisatorisches: Inhalt

Einleitung und XML	17. Okt	SPARQL Syntax & Intuition	12. Dez
Einführung in RDF	20. Okt	Übung 4	15. Dez
RDF Schema	24. Okt	SPARQL Semantik	19. Dez
fällt aus	27. Okt	SPARQL 1.1	22. Dez
Logik – Grundlagen	31. Okt	Übung 5	9. Jan
Übung 1	3. Nov	SPARQL Entailment	12. Jan
Semantik von RDF(S)	7. Nov	SPARQL Implemetierung	16. Jan
RDF(S) & Datalog Regeln	10. Nov	Abfragen & RIF	19. Jan
OWL Syntax & Intuition	14. Nov	Übung 6	23. Jan
Übung 2	17. Nov	Ontology Editing	26. Jan
OWL & BLs	21. Nov	Ontology Engineering	30. Jan
OWL 2	24. Nov	Linked Data	2. Feb
Tableau	28. Nov	Übung 7	6. Feb
Übung 3	1. Dez	SemWeb Anwendungen	9. Feb
Blocking & Unravelling	5. Dez	Wiederholung	13. Feb
Hypertableau	8. Dez	Übung 8	16. Feb

OWL 2



Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Beispiel Standard Tableau

Gegeben sei die folgende TBox $\mathcal{T}$ und ABox $\mathcal{A}$:

$$\mathcal{T} = \{\exists r.A \sqsubseteq A\} \quad C_{\mathcal{T}} = \forall r.(\neg A) \sqcup A$$
$$\mathcal{A} = \{\neg A(a_0), r(a_0, b_1), r(b_1, a_1), \ldots, r(a_{n-1}, b_n), r(b_n, a_n), A(a_n)\}$$

$$a_0 \xrightarrow{r} b_1 \xrightarrow{r} a_1 \xrightarrow{r} b_2 \cdots\cdots \cdots \to a_{n-1} \xrightarrow{r} b_n \xrightarrow{r} a_n$$

Annahme: Wir arbeiten die Knoten im Tableau alphabetisch ab, d.h. a's vor b's

Beispiel Standard Tableau

$C_{\mathcal{T}} = \forall r.(\neg A) \sqcup A$

$$a_0 \xrightarrow{r} b_1 \xrightarrow{r} a_1 \xrightarrow{r} b_2 \cdots\cdots \cdots \to a_{n-1} \xrightarrow{r} b_n \xrightarrow{r} a_n$$

$$L(a_0) = \{\neg A, C_{\mathcal{T}}\} \cup \{\forall r.(\neg A)^1\}$$
$$L(a_1) = \{C_{\mathcal{T}}\} \cup \{\forall r.(\neg A)^2\} \cup \{\cancel{\neg A^{n+2}}\}$$
$$\vdots \quad \vdots$$
$$L(a_{n-1}) = \{C_{\mathcal{T}}\} \cup \{\cancel{\forall r.(\neg A)^n}\} \cup \{\cancel{\neg A^{2n}}\} \cup \{A\}$$
$$L(a_n) = \{A, C_{\mathcal{T}}\} \cup \{\cancel{\forall r.(\neg A)^{n+1}}\} \cup \{\cancel{\neg A^{2n+1}}\}$$
$$L(b_1) = \{C_{\mathcal{T}}\} \cup \{\neg A^1\} \cup \{\cancel{\forall r.(\neg A)^{n+2}}\}$$
$$L(b_2) = \{C_{\mathcal{T}}\} \cup \{\neg A^2\} \cup \{\cancel{\forall r.(\neg A)^{n+3}}\}$$
$$\vdots \quad \vdots$$
$$L(b_n) = \{C_{\mathcal{T}}\} \cup \{\cancel{\neg A^n}\} \cup \{\cancel{\forall r.(\neg A)^{2n+1}}\} \cup \{\cancel{A^{2n+1}}\}$$

Nun erneut Entscheidungen für a_n und $b_1, \ldots, b_n$

Muss das so sein?

- Der Algorithmus konstruiert trotz dependency directed backtracking exponentiell viele Tableauzweige
- Übersetzung der Formel in die Prädikatenlogik erster Stufe:

$$\forall r.(\neg A) \sqcup A$$
$$= \forall x, y. [r(x, y) \wedge A(y) \rightarrow A(x)]$$
$$= \forall x, y. [\neg r(x, y) \vee \neg A(y) \vee A(x)]$$

- Hier hat die Formel keinen echten Nichtdeterminismus (Horn-Klausel)
- Hypertableau macht sich dieses zu nutzen

Idee Hypertableau

- Übersetze die Axiome einer Wissensbasis in die Prädikatenlogik erster Stufe
  - Axiome werden umgeschrieben um nur Formeln bestimmter Struktur zu erhalten
- Axiome werden so übersetzt, dass Nichtdeterminismus möglichst vermieden wird
- Die Formels für die Axiome werden zu Regeln um eine Modellabstraktion zu konstruieren

Einfaches Hypertableau Beispiel

1. Übersetzung in Klauseln

$$
\begin{array}{rl}
A \sqsubseteq A' & A(x) \to A'(x) \\
A \sqsubseteq \exists r.B & A(x) \to \exists r.B(x) \\
D \sqsubseteq E \sqcup F & D(x) \to E(x) \vee F(x) \\
F \sqsubseteq \bot & F(x) \to \bot(x) \\
\exists r.\top \sqsubseteq C & r(x,y) \to C(x) \\
A(a) & \to A(a) \\
(D \sqcap \neg B)(d) & \to D(d) \\
 & \to \neg B(d)
\end{array}
$$

- Existenzquantoren werden weiterhin wie im Tableau behandelt

Einfaches Hypertableau Beispiel

$$
\begin{tikzcd}
a \arrow[d, "r"'] & d \\
v_1 &
\end{tikzcd}
$$

$$
\begin{array}{rl}
A(x) & \to A'(x) \\
A(x) & \to \exists r.B(x) \\
D(x) & \to E(x) \vee F(x) \\
F(x) & \to \bot(x) \\
r(x, y) & \to C(x) \\
 & \to A(a) \\
 & \to D(d) \\
 & \to \neg B(d)
\end{array}
$$

$$L(a) = \{A\} \cup \{A'\} \cup \{\exists r.B\} \cup \{C\}$$
$$L(d) = \{D, \neg B\} \cup \{E^1\}$$
$$L(v_1) = \{B\}$$

- Keine Regel mehr anwendbar $\rightsquigarrow$ Erfüllbarkeit gezeigt
- Aus den Tableau Regeln bleibt nur noch ein Analog zur $\exists$-Regel

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Semantik durch Übersetzung in FOL

Übersetzung von TBox-Aussagen in die Prädikatenlogik mittels der Abbildung π mit C, D komplexe Klassen, r eine Rolle und A eine atomare Klasse:

$$\pi(C \sqsubseteq D) = \forall x.(\pi_x(C) \to \pi_x(D)) \qquad \pi(C \equiv D) = \forall x.(\pi_x(C) \leftrightarrow \pi_x(D))$$

$$
\begin{array}{rlrl}
\pi_x(A) &= A(x) & \pi_y(A) &= A(y) \\
\pi_x(\neg C) &= \neg \pi_x(C) & \pi_y(\neg C) &= \neg \pi_y(C) \\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D) & \pi_y(C \sqcap D) &= \pi_y(C) \wedge \pi_y(D) \\
\pi_x(C \sqcup D) &= \pi_x(C) \vee \pi_x(D) & \pi_y(C \sqcup D) &= \pi_y(C) \vee \pi_y(D) \\
\pi_x(\forall r.C) &= \forall y.(r(x,y) \to \pi_y(C)) & \pi_y(\forall r.C) &= \forall x.(r(y,x) \to \pi_x(C)) \\
\pi_x(\exists r.C) &= \exists y.(r(x,y) \wedge \pi_y(C)) & \pi_y(\exists r.C) &= \exists x.(r(y,x) \wedge \pi_x(C))
\end{array}
$$

Motivation Normalform

- Die gegebene Übersetzung erzeugt für kompelxe Axiome sehr komplexe Formeln

$$\begin{aligned}
&\pi(C \sqsubseteq \exists r.(\forall s.(D \sqcup \exists r.D)))\\
&\forall x.\,[\pi_x(C) \to \pi_x(\exists r.(\forall s.(D \sqcup \exists r.D)))]\\
&\forall x.\,[C(x) \to \exists y.(r(x,y) \land \pi_y(\forall s.(D \sqcup \exists r.D)))]\\
&\forall x.\,[C(x) \to \exists y.(r(x,y) \land \forall x.(s(y,x) \to \pi_x(D \sqcup \exists r.D)))]\\
&\forall x.\,[C(x) \to \exists y.(r(x,y) \land \forall x.(s(y,x) \to \pi_x(D) \lor \pi_x(\exists r.D)))]\\
&\forall x.\,[C(x) \to \exists y.(r(x,y) \land \forall x.(s(y,x) \to D(x) \lor \exists y.(r(x,y) \land \pi_y(D))))]\\
&\forall x.\,[C(x) \to \exists y.(r(x,y) \land \forall x.(s(y,x) \to D(x) \lor \exists y.(r(x,y) \land D(y))))]
\end{aligned}$$

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Strukturelle Transformation

- Strukturelle Transformation führt neue Konzepte für komplexe Unterkonzepte ein

$$\begin{aligned}
& C \sqsubseteq \exists r.(\forall s.(D \sqcup \exists r.D))\\
\rightsquigarrow\ & C \sqsubseteq \exists r.Q_1\\
& Q_1 \equiv \forall s.(D \sqcup \exists r.D)\\
\rightsquigarrow\ & Q_1 \equiv \forall s.Q_2\\
& Q_2 \equiv D \sqcup \exists r.D\\
\rightsquigarrow\ & Q_2 \equiv D \sqcup Q_3\\
\rightsquigarrow\ & Q_3 \equiv \exists r.D
\end{aligned}$$

Optimierung der Strukturellen Transformation

- Müssen wir unbedingt Äquivalenzaxiome einführen?

$$\begin{array}{ll}
A \sqsubseteq \forall r.(\forall r.B) & \exists r.(A \sqcap B) \sqsubseteq C\\
\rightsquigarrow\ A \sqsubseteq \forall r.Q & \rightsquigarrow\ \exists r.Q \sqsubseteq C\\
\rightsquigarrow\ Q \sqsubseteq \forall r.B \quad \checkmark & Q \sqsubseteq A \sqcap B\\
& \rightsquigarrow\ Q \sqsubseteq A\\
& \rightsquigarrow\ Q \sqsubseteq B
\end{array}$$

Hm. Ist das so richtig? Sei $\mathcal{A} = \{r(a,b), A(b), B(b), \neg C(a)\}$

- Bzgl. einer TBox mit dem urspünglichen Axiom war die ABox widersprüchlich
- Bzgl. einer TBox mit den umgeschriebenen Axiomen ist die ABox erfüllbar

Polarität für die Optimierte Transformation

- Wir müssen beachten ob Teilkonzepte positiv oder negativ vorkommen
- $A \sqsubseteq B$ ist $\neg A \sqcup B$
- A kommt daher negativ in dem Axiom vor und B positiv
- Wenn wir ein Konzept ersetzen, das negativ vorkommt, müssen wir $\sqsupseteq$ verwenden

$$\exists r.(A \sqcap B) \sqsubseteq C$$
$$\rightsquigarrow \exists r.Q \sqsubseteq C$$
$$Q \sqsupseteq A \sqcap B$$
$$\rightsquigarrow A \sqcap B \sqsubseteq Q$$

$$C_{\mathcal{T}} = (\forall r.(\neg Q) \sqcup C) \sqcap (\neg A \sqcup \neg B \sqcup Q)$$

Tableau für die Unerfüllbarkeit

$$C_{\mathcal{T}} = (\forall r.(\neg Q) \sqcup C) \sqcap (\neg A \sqcup \neg B \sqcup Q)$$
$$\mathcal{A} = \{r(a,b), A(b), B(b), \neg C(a)\}$$

$$a \xrightarrow{r} b$$

$$L(a) = \{\neg C\} \cup \{C_{\mathcal{T}}, \forall r.(\neg Q) \sqcup C, \neg A \sqcup \neg B \sqcup Q, \forall r.(\neg Q), \neg A\}$$
$$L(b) = \{A, B\} \cup \{\neg Q\} \cup \{C_{\mathcal{T}}, \forall r.(\neg Q) \sqcup C, \neg A \sqcup \neg B \sqcup Q, \forall r.(\neg Q)\}$$
$$\cup \cancel{\{\neg A\}} \cup \cancel{\{\neg B\}} \cup \cancel{\{Q\}}$$

Keine weiteren relevanten Wahlmöglichkeiten
$\rightsquigarrow$ Die Wissensbasis ist unerfüllbar

Optimierte Strukturelle Transformation

- Wir wollen nun die strukturelle Transformation formal definieren
- Pro Teilkonzept soll dabei nur ein neuer Konzeptname eingeführt werden
- Ziel ist eine TBox in eine erfüllbarkeitsäquivalente TBox umzuschreiben, die nur “einfache” Axiome enthält

Polarität von Konzepten

Wir definieren die Polarität eines Konzepts C in einer Formel wie folgt:

- C kommt in C positiv vor,
- C kommt in $\neg D$ positiv (negativ) vor wenn C in D negativ (positive) vorkommt,
- C kommt in $D \sqcap E$ oder $D \sqcup E$ positiv (negativ) vor, wenn C positiv (negativ) in D oder E vorkommt,
- C kommt in $\exists r.D$ oder $\forall r.D$ positiv (negativ) vor, wenn C positiv (negativ) in D vorkommt,
- C kommt in $D \sqsubseteq E$ positiv (negativ) vor, wenn C positiv (negativ) in E vorkommt oder negativ (positive) in D.

Ein Konzept kommt in einer ($\mathcal{ALC}$) TBox $\mathcal{T}$ positiv (negativ) vor, wenn C positiv (negativ) in einem Axiom in $\mathcal{T}$ vorkommt.
$\rightsquigarrow$ Ein Konzept kann gleichzeitig positiv und negativ in einem Axiom vorkommen.

Optimierte Transformation mit Polarität

Sei $\mathcal{T}$ eine $\mathcal{ALC}$ TBox. Für jedes (Teil-)Konzept C in $\mathcal{T}$ führen wir ein frisches atomares Konzept A_C ein und definieren die Funktion st(C) wie folgt:

$$\begin{array}{lll}
\text{st}(A) = A & \text{st}(\neg C) = \neg A_C & \text{st}(\exists r.C) = \exists r.A_C \\
\text{st}(\top) = \top & \text{st}(C \sqcap D) = A_C \sqcap A_D & \text{st}(\forall r.C) = \forall r.A_C \\
\text{st}(\bot) = \bot & \text{st}(C \sqcup D) = A_C \sqcup A_D &
\end{array}$$

Das Ergebnis der strukturellen Transformation einer TBox $\mathcal{T}$ ist eine TBox $\mathcal{T}'$ mit folgenden Axiomen:

- $A_C \sqsubseteq$ st(C) für jedes Konzept C das positiv in $\mathcal{T}$ vorkommt,
- st(C) $\sqsubseteq A_C$ für jedes Konzept C das negativ in $\mathcal{T}$ vorkommt
- $A_C \sqsubseteq A_D$ für jedes GCI $C \sqsubseteq D \in \mathcal{T}$.

Vereinfachung von Axiom

Wir können nun die bekannten Äquivalenzen (Vgl. Vorlesung 5) verwenden, um die Axiome weiter zu vereinfachen:

$$\{C \sqsubseteq D \sqcap E\} \equiv \{C \sqsubseteq D, C \sqsubseteq E\}$$
$$\{C \sqcup D \sqsubseteq E\} \equiv \{C \sqsubseteq E, D \sqsubseteq E\}$$

Ergebnis der Strukturellen Transformation

Mit Hilfe der strukturellen Transformation und den bekannten Äquivalenzen, können wir eine $\mathcal{ALC}$ Wissensbasis in eine erfüllbarkeitsäquivalente umschreiben, die nur Axiome der folgenden Form enthält (A und B atomar):

$$\begin{aligned}
A_1 \sqcap \ldots \sqcap A_n &\sqsubseteq B_1 \sqcup \ldots \sqcup B_m \\
A &\sqsubseteq \exists r.B \\
A &\sqsubseteq \forall r.B \\
\exists r.A &\sqsubseteq B \\
\forall r.A &\sqsubseteq B
\end{aligned}$$

Beispiel zur Optimierten Strukturellen Transformation

$$\overbrace{\underbrace{\forall r.(\overbrace{A \sqcap D}^{C_1})}_{C_3} \sqcap A \sqcap \underbrace{\exists s.(\overbrace{A \sqcap D}^{C_1})}_{C_4}}^{C_7} \sqsubseteq \underbrace{B \sqcup \overbrace{(\underbrace{\exists r.(\overbrace{C \sqcap E}^{C_2})}_{C_5} \sqcap F)}^{C_8} \sqcup \underbrace{\forall r.A}_{C_6}}_{C_9}$$

$$\begin{aligned}
A \sqcap D &\sqsubseteq C_1 & C_6 &\sqsubseteq \forall r.A \\
C_2 &\sqsubseteq C \sqcap E & C_3 \sqcap A \sqcap C_4 &\sqsubseteq C_7 \\
\forall r.C_1 &\sqsubseteq C_3 & C_8 &\sqsubseteq C_5 \sqcap F \\
\exists s.C_1 &\sqsubseteq C_4 & C_9 &\sqsubseteq B \sqcup C_8 \sqcup C_6 \\
C_5 &\sqsubseteq \exists r.C_3 & C_7 &\sqsubseteq C_9
\end{aligned}$$

Beispiel zur Optimierten Strukturellen Transformation

Wir können nun noch die Vereinfachungsregeln anwenden:

$$\overbrace{\underbrace{\forall r.(\overbrace{A \sqcap D}^{C_1})}_{C_3} \sqcap A \sqcap \underbrace{\exists s.(\overbrace{A \sqcap D}^{C_1})}_{C_4}}^{C_7} \sqsubseteq \underbrace{B \sqcup \overbrace{(\underbrace{\exists r.(\overbrace{C \sqcap E}^{C_2})}_{C_5} \sqcap F)}^{C_8} \sqcup \underbrace{\forall r.A}_{C_6}}_{C_9}$$

$$\begin{aligned} A \sqcap D &\sqsubseteq C_1 & C_6 &\sqsubseteq \forall r.A \\ C_2 &\sqsubseteq C & C_3 \sqcap A \sqcap C_4 &\sqsubseteq C_7 \\ \forall r.C_1 &\sqsubseteq C_3 & C_8 &\sqsubseteq C_5 \\ \exists s.C_1 &\sqsubseteq C_4 & C_9 &\sqsubseteq B \sqcup C_8 \sqcup C_6 \\ C_5 &\sqsubseteq \exists r.C_3 & C_7 &\sqsubseteq C_9 \\ C_2 &\sqsubseteq E & C_8 &\sqsubseteq F \end{aligned}$$

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Übersetzung in Klauseln

Eine TBox mit den vereinfachten Axiomen kann nun in Klauseln (geschrieben als Regeln) übersetzt werden:

$$\begin{aligned} A_1 \sqcap \ldots \sqcap A_n &\sqsubseteq B_1 \sqcup \ldots \sqcup B_m \\ A_1(x) \wedge \ldots \wedge A_n(x) &\rightarrow B_1(x) \vee \ldots \vee B_m(x) \end{aligned}$$

Bei $m = 0$ enthält der Regelkopf $\bot(x)$.

$$\begin{aligned} A &\sqsubseteq \exists r.B & A(x) &\rightarrow (\exists r.B)(x) \\ A &\sqsubseteq \forall r.B & A(x) \wedge r(x,y) &\rightarrow B(y) \\ \exists r.A &\sqsubseteq B & r(x,y) \wedge A(y) &\rightarrow B(x) \\ \forall r.A &\sqsubseteq B & &\rightarrow (\exists r.\neg A)(x) \vee B(x) \end{aligned}$$

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Grundbegriffe zum Hypertableau Kalkül

- Für eine TBox $\mathcal{T}$ sei cl$(\mathcal{T})$ die entsprechende Menge der Klauseln
- Wir setzten voraus, dass die ABox nicht leer ist
- Wir setzten voraus, dass die ABox nur Fakten der Form $A(a), \neg A(a), (\exists r.A)(a), (\exists r.\neg A)(a), r(a,b)$ enthält
- Wir schreiben Vars$(\mathcal{T})$ für die Menge der Variablen in cl$(\mathcal{T})$
- Wir schreiben Inds$(\mathcal{A})$ für die Menge der Individuennamen in $\mathcal{A}$

Das Hypertableau Kalkül

HT-Regel: Für $A_1 \wedge \ldots \wedge A_n \rightarrow B_1 \vee \ldots \vee B_m \in$ cl$(\mathcal{T})$ und Mapping σ: Vars$(\mathcal{T}) \rightarrow$ Inds$(\mathcal{A})$ mit $\sigma(A_i) \in \mathcal{A}, \sigma(B_j) \notin \mathcal{A}$ für jedes $1 \leq i \leq n$, $1 \leq j \leq m$
wähle j mit $1 \leq j \leq m$ und setzte $\mathcal{A} = \mathcal{A} \cup \{\sigma(B_j)\}$.

$\exists$-Regel: Für $(\exists r.C)(v) \in \mathcal{A}$ so dass v nicht blockiert ist und es keinen r-Nachfolger v' für v mit $C(v') \in \mathcal{A}$ gibt, setzte $\mathcal{A} = \mathcal{A} \cup \{r(v,v'), C(v')\}$ für v' ein neuer Knoten.

- Ein Tableau wird nun als ABox repräsentiert
- Das Tableau ist widersprüchlich, wenn die ABox $\bot(v)$ oder $A(v)$ und $(\neg A)(v)$ für ein Individuum v und Konzept A enthält

Beispiel Hypertableau

Gegeben sei wieder die TBox $\mathcal{T}$ und ABox $\mathcal{A}$:

$$\mathcal{T} = \{\exists r.A \sqsubseteq A\} \qquad \text{cl}(\mathcal{T}) = \{r(x,y) \wedge A(y) \rightarrow A(x)\}$$

$$a_0 \xrightarrow{r} b_1 \xrightarrow{r} a_1 \xrightarrow{r} b_2 \cdots\cdots \dashrightarrow a_{n-1} \xrightarrow{r} b_n \xrightarrow{r} a_n$$

$$\mathcal{A} = \{\neg A(a_0), r(a_0,b_1), r(b_1,a_1), \ldots, r(a_{n-1},b_n), r(b_n,a_n), A(a_n)\} \cup \{A(b_n)\} \cup \{A(a_{n-1})\} \cup \ldots \cup \{A(a_0)\}$$

Der Hypertableau Algorithmus hat hier keinen Nichtdeterminimus

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Blocking im Hypertableau

Blockierungsmeachanismus war noch undefiniert:

Definition (Blocking)

Ein Individuum $v \in \mathsf{Inds}(\mathcal{A})$ blockiert ein Individuum $v' \in \mathsf{Inds}(\mathcal{A})$ in einer ABox $\mathcal{A}$ direkt, wenn:

1. v' von v erreichbar ist,
2. $\{A \mid A(v) \in \mathcal{A}\} = \{A \mid A(v') \in \mathcal{A}\}$; und
3. es keinen direkt blockierten Knoten v'' gibt so dass v' von v'' erreichbar ist.

Ein Individuum $v' \in V$ ist blockiert wenn entweder

1. v' direkt blockiert ist oder
2. es einen direkt blockierten Knoten v gibt so dass v' von v erreichbar ist.

Blocking im Hypertableau

- Die ABox wird hier wie das Tableau als Graph gesehen in dem jedes Individuum der ursprünglichen ABox die Wurzel eines Baumes bildet
- Blockierung hängt nur noch von atomaren Konzepten ab
- Können wir nicht auch Teilmengenblockierung verwenden?

Beispiel Hypertableau

$$\mathcal{T} = \{ C \sqsubseteq \exists r.C,\ C \sqsubseteq \exists s.D,\ \exists s.D \sqsubseteq E,\ \top \sqsubseteq \forall r.(\neg E)\}$$

$$\mathsf{cl}(\mathcal{T}) = \{ C(x) \rightarrow (\exists r.C)(x),\ C(x) \rightarrow (\exists s.D)(x),\ s(x,y) \wedge D(y) \rightarrow E(x),\ r(x,y) \wedge E(y) \rightarrow \bot(x) \}$$

$$\begin{tikzcd} a \arrow[r, "t"] & v_0 \arrow[r, "r"] \arrow[dr, "s"'] & v_1 \\ & & v_2 \end{tikzcd}$$

$$\begin{aligned}\mathcal{A} = {} & \{(\exists t.C)(a)\} \cup \{t(a,v_0), C(v_0)\} \cup \{(\exists r.C)(v_0)\} \cup \{(\exists s.D)(v_0)\} \\ & \cup \{r(v_0,v_1), C(v_1)\} \cup \{\exists r.C(v_1)\} \cup \{\exists s.D(v_1)\} \\ & \cup \{s(v_0,v_2), D(v_2)\} \cup \{E(v_0)\}\end{aligned}$$

v_0 blockiert v_1 (atomare Konzepte): $\mathcal{L}(v_1) = \{C\} \subseteq \mathcal{L}(v_1) = \{C\} \cup \{E\}$
Keine weiteren Regeln anwendbar

Beispiel Hypertableau

Beim Modellbau kriegen wir Probleme (blockiertes Individuum wird durch den Blockierer inkl. darin gewurzelten Teilbaum ersetzt):

$$\mathsf{cl}(\mathcal{T}) = \{ C(x) \rightarrow (\exists r.C)(x),\ C(x) \rightarrow (\exists s.D)(x),\ s(x,y) \wedge D(y) \rightarrow E(x),\ r(x,y) \wedge E(y) \rightarrow \bot(x) \}$$

$$\begin{tikzcd} a \arrow[r, "t"] & v_0 \arrow[r, "r"] \arrow[d, "s"'] & v_0' \arrow[r, dotted, "r"] \arrow[d, dotted, "s"'] & v_0'' \cdots \\ & v_2 & v_2' & \end{tikzcd}$$

$$\begin{aligned}\mathcal{A} = \{ & (\exists t.C)(a), t(a,v_0), C(v_0), E(v_0), (\exists r.C)(v_0), (\exists s.D)(v_0), \\ & s(v_0,v_2), D(v_2), r(v_0,v_0'), C(v_0'), E(v_0'), \exists r.C(v_0'), \exists s.D(v_0'), \\ & s(v_0',v_2'), D(v_2'), r(v_0',v_0''), C(v_0''), E(v_0''), \exists r.C(v_0''), \exists s.D(v_0''), \ldots\}\end{aligned}$$

Vergleich Blockierung im Tableau und Hypertablau Kalkül

- Teilmengen-Blockierung geht im Hypertableau nicht
- Hinzunehmen der nicht-atomaren Konzepte hilft nicht
- Axiom $\exists s.D \sqsubseteq E$ äquivalent zu $D \sqsubseteq \forall s^{-}.E$

$$\begin{array}{cc}
\exists s.D \sqsubseteq E & D \sqsubseteq \forall s^{-}.E \\
\forall s.(\neg D) \sqcup E & \neg D \sqcup \forall s^{-}.E \\
s(x,y) \wedge D(y) \rightarrow E(x) & D(x) \wedge s^{-}(x,y) \rightarrow E(y) \\
 & D(x) \wedge s(y,x) \rightarrow E(y) \\
 & D(y) \wedge s(x,y) \rightarrow E(x)
\end{array}$$

- Für inverse Rollen braucht auch das Tableau Kalkül auf Gleichheits-Blockierung
- Für das Axiom $\exists s.D \sqsubseteq E$ erzwing das Tableau Kalkül eine Auswahl (GCI $\rightsquigarrow$ Disjunktion)

Agenda

Anmerkungen Hypertablau

- Hypertableau braucht Gleichheits-Blockierung
- Mit Zahlenrestriktionen/Funktionalität dann auch Paar-Blockierung (pairwise blocking)
- Inverse Rollen verschwinden in den Regeln (Variablenpositionen werden getauscht)
- Übersetzung in Regeln in der Realität komplexer (weitere Heuristiken zur Vermeidung von Disjunktionen)
- Zum Auswerten der Regeln kann semi-naive Evaluierung angewendet werden (siehe Vorlesung 7)
- Dependency directed backtracking kann genau wie im Tableau verwendet werden
- (Un)Gleichheit für Funktionalität/Zahlenrestriktionen:
  - Func(f) entspricht der Regel $f(x,y_1) \wedge f(x,y_2) \rightarrow y_1 \approx y_2$
  - spezielle Regel $\approx$-Regel für merging und pruning

Vergleich Tableau und Hypertablau

- Hypertableau hat eine aufwändigere Vorverarbeitung
- Nichtdeterminismus lässt sich oft vermeiden
- Blockierung braucht allerdings immer Gleichheit
- ABox wird durch Regeln erweitert analog zur Erweiterung des Tableau
- Guter Mechanismus zum Auswerten der Regeln wichtig
- Implementierung als ABox statt über Graphstrukturen macht das Überprüfen der Blockierungsbedingungen schwieriger ($\rightsquigarrow$ Optimierung über Hashing)
- Tableau und Hypertableau lassen sich für OWL 2 erweitern (insbesondere Nominale schwierig)
- Hypertableau implementiert in HermiT, Tableau in FaCT++ und Pellet

Agenda

- Motivation
- Wiederholung Übersetzung in die Prädikatenlogik 1. Stufe
- Strukturelle Transformation
- Übersetzung in Klauseln
- Die Hypertableau Regeln
- Blockierung im Hypertableau Kalkül
- Vergleich Tableau und Hypertableau Kalkül
- Zusammenfassung

Zusammenfassung

- Wir haben die grundlegenden Kalküle kennengelernt die auf der Modellkonstruktion basieren
- OWL Profile können auch mit effizienteren Verfahren implementiert werden ($\rightsquigarrow$ consequence-based procedures)
- Tableau und Hypertableau sind korrekt und terminieren
  - Beweise gerade für ausdrucksstärkere Logiken schwierig
- In der Praxis diverse Optimierungen (Blockierung, Caching von Modellteilen, Heuristiken, etc.)
- Anwendbar für mittelgroße Wissensbasen, aber sehr abhängig von der Komplexität der Axiome
- Worst-case für $\mathcal{ALC}$ Wissensbasen: ExpTime, für OWL 1 DL: NExpTime und für OWL 2 DL: 2-NExpTime

Was wir nicht behandelt haben

- Zahlenrestriktionen (ausser Funktionalität im Tableau)
- Datentypen (spezielle Algorithmen)
- Nominale und insb. das Zusammenspiel mit Inversen und Zahlenrestriktionen
- Optimierte Regelübersetzung für das Hypertableau
- Weitere Optimierungen: disjuction learning, told subsumers, etc.
- Genauere Komplexitätsbetrachtungen
- Außer dem Testen der Erfüllbarkeits nur Klassifikation betrachtet, Realisierung (berechnen der Typen von Individuen bzw. der Instanzen von Konzepten)
- APIs (insb. OWL API)