Description Logics
(Beschreibungslogiken)

Seminar Semantic Web

Paulius Friedt
Friedrich – Schiller Universität Jena (Erasmus)
Universität Vilnius, Litauen

Überblick

- Einführung
  - Wissensrepresentation
  - Ontologien
- Grundlagen von Beschreibungslogiken
  - Die $\mathcal{AL}$ Beschreibungslogik
  - $\mathcal{AL}$ Syntax
  - $\mathcal{AL}$ Semantik
  - Die Erweiterungen von $\mathcal{AL}$
  - TBox, ABox
- Inferenzen und Tableau – Based Algorithmus

Einführung: Wissensrepresentation

Warum Wissensrepresentation:

- Das Wissen formal zu definieren
- Automatisches Reasoning

Arten:

- Frame-Based
- Network-Based
- Ontologien

Einführung: Ontologien

Was ist eine Ontologie:

- formale Repräsentation von Wissen einer Welt
- Eigenschaften dieser Welt
- Bedingungen die in dieser Welt gelten

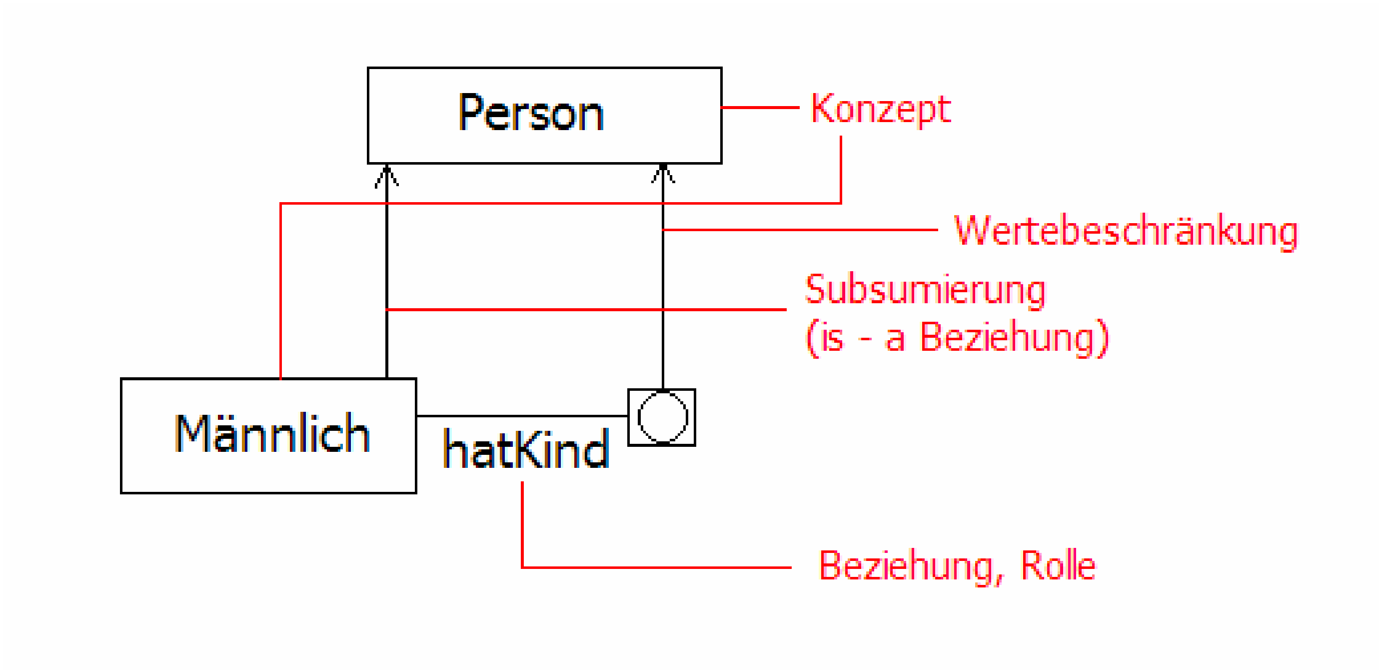
Ontologiesprachen:

- Konzepte
- Eigenschaften dieser Konzepte
- Beziehungen zwischen den Konzepten

- Unterscheiden sich in ihrer Ausdruckskraft.
- Meistens durch Diagrame Ausgedrückt (ER – Diagramme, UML – Diagramme).

Einführung: Wissensrepresentation

Wissensrepresentation durch netzwerkbasierende Struktur



Einführung: Beschreibungslogiken

- man braucht streng beschriebene Semantik

⇨ Beschreibungslogiken

- Familie von Beschreibungslogiken gehört zu den auf Logik basierenden Wissensrepresentationssprachen

- Basierend auf Konzepte und Rollen
- Stammen von Semantic Networks und KL-ONE
- Früher genannt als terminological logics, concept languages

Beschreibungslogiken

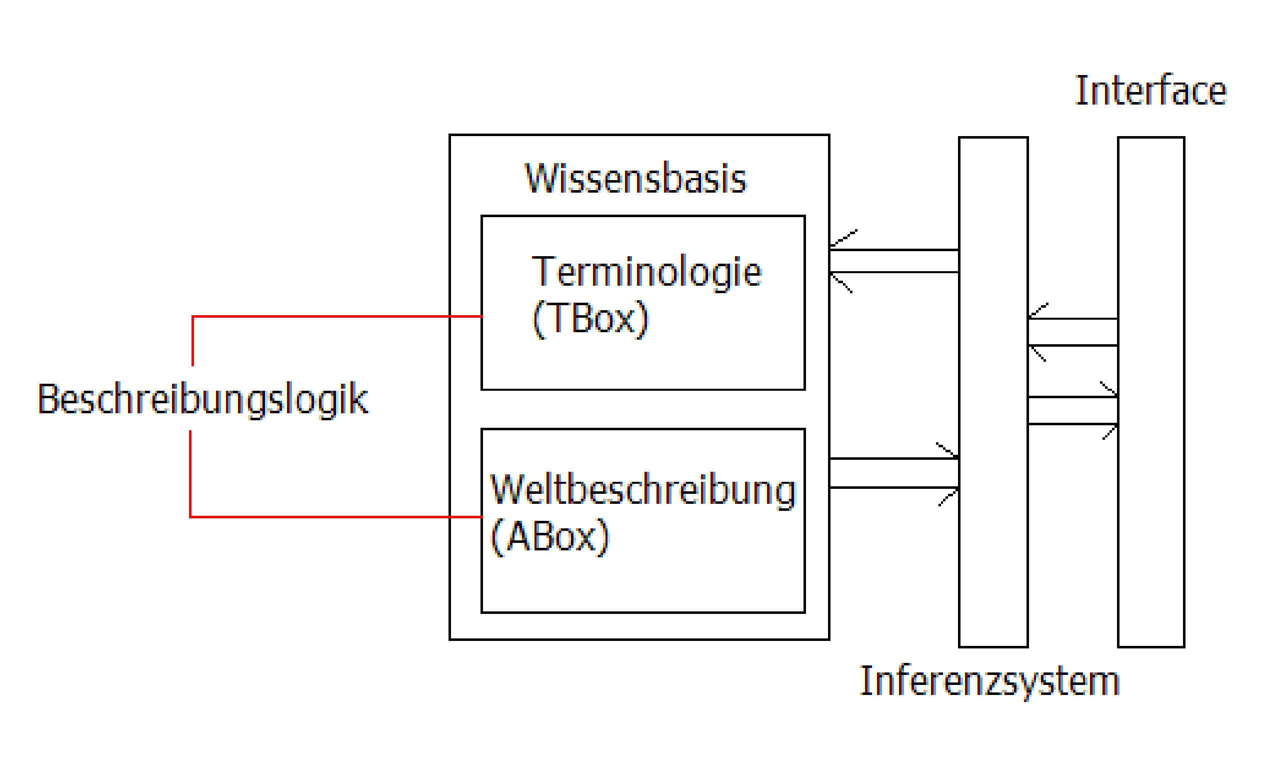
- Basiert auf die FOL (First Order Logics).
  BL sind ein entscheidbares Fragment der FOL

Eigenschaften von BLs

- Wohl definierte Semantik
- Entscheidbarkeit
- Berechenbare und vollständige Inferenzdienste

Beschreibungslogiken

Architektur eines WR Systems basiertes auf BL



$\mathcal{AL}$ Syntax

Grundlegende Bestandteile der BL sind

- Konzepte (C, D)
- Rollen (R)

- Neue Konzepte können mit Hilfe von Konstruktoren erzeugt werden

C, D $\rightarrow$	A	atomares Konzept
|	$\top$	universelles Konzept
|	$\perp$	Bottom – Konzept
|	$\neg$A	atomare Negation (nur für atomare Konzepte)
|	C$\sqcap$D	Schnitt
|	$\forall$R.C	Wertebeschränkung
|	$\exists$R.$\top$	Eingeschränkte existenzielle Quantifizierung

Beispiel:

- Person $\sqcap$ $\exists$hatKind.$\top$ Person, die Kinder hat

$\mathcal{AL}$ Semantik

Interpretation $\mathcal{I}$ $(\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$

- $\triangle^{\mathcal{I}}$ - Domäne, $\triangle^{\mathcal{I}}$ nicht leere Menge
- $\cdot^{\mathcal{I}}$ - Interpretationsfunktion
  - für alle atomaren Konzepte A: $A^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$
  - für alle Rollen R: $R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$

Erweiterungen

- $\top^{\mathcal{I}} = \triangle^{\mathcal{I}}$
- $\bot^{\mathcal{I}} = \emptyset$
- $(\neg A)^{\mathcal{I}} = \triangle^{\mathcal{I}} \setminus A^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{ a \in \triangle^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}} \}$
- $(\exists R.\top)^{\mathcal{I}} = \{ a \in \triangle^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \}$

- Äquivalenz $X \equiv Y$ gdw. $X^{\mathcal{I}} = Y^{\mathcal{I}}$
- Inklusion $X \sqsubseteq Y$ gdw. $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$, wobei X, Y Konzepte oder Rollen

$\mathcal{AL}$ Semantik: Beispiel

$\triangle^{\mathcal{I}}$ = {John, Peter, Jim, Sarah, Mike}

(Männlich)$^{\mathcal{I}}$ = {John, Peter, Jim, Mike}

(Hatkind)$^{\mathcal{I}}$ = {(John, Sarah), (Jim, Peter), (Sarah, Mike)}

Definition von einem komplexen Konzept

Vatter $\equiv$ Männlich $\sqcap$ $\exists$hatKind.$\top$

(Vatter)$^{\mathcal{I}}$ = (Männlich $\sqcap$ $\exists$hatKind.$\top)^{\mathcal{I}}$ = (Männlich)$^{\mathcal{I}}$ $\cap$ ($\exists$hatKind.Person)$^{\mathcal{I}}$ =

= {John, Peter, Jim, Mike} $\cap$ { a $\in$ $\triangle^{\mathcal{I}}$ | $\exists$b.(a,b) $\in$ (hatHind)$^{\mathcal{I}}$ } =

= {John, Peter, Jim, Mike} $\cap$ {John, Jim, Sarah} = {John, Jim}

$\mathcal{AL}$ Beschreibungslogik: Erweiterungen $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$

- Volle Negation $\mathcal{C}$ $\quad (\neg C)^{\mathcal{I}} = \triangle^{\mathcal{I}} \setminus C^{\mathcal{I}}$
- Vereinigung $\mathcal{U}$ $\quad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- volle existenzielle Quantifizierung $\mathcal{E}$
  $$(\exists R.C)^{\mathcal{I}} = \{\, a \in \triangle^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \,\}$$
- Anzahlbeschränkung (number restriction) $\mathcal{N}$
  $$(\geqslant nR)^{\mathcal{I}} = \{\, a \in \triangle^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \geqslant n \,\}$$
  $$(\leqslant nR)^{\mathcal{I}} = \{\, a \in \triangle^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \leqslant n \,\}$$

$\mathcal{AL}$ Beschreibungslogik: Erweiterungen $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$

Vereinigung und existenzielle Quantifizierung können durch Negation ausgedrückt werden

- $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$
- $\exists R.C \equiv \neg \forall R.\neg C$

- Jede $\mathcal{C}$ – Sprache ist eine $\mathcal{UE}$ – Sprache.

TBox: Terminologien

eine Menge von terminologischen Axiomen die intensionales Wissen von der Welt beschreiben. Die Grundlage für die Ontologien

Axiome

- Gleichung $A \equiv C$, $R \equiv S$, hinreichende Bedingungen
- Inklusion $A \sqsubseteq C$, $R \sqsubseteq S$, notwendige Bedingungen

  A, C – Konzepte und R, S – Rollen

- Definition – eine Gleichung deren linke Seite ein atomares Konzept ist
- die Konzepte, die auf der linke Seite der Definition stehen – Namenssymbole $\mathcal{N}_{\mathcal{T}}$
- die Konzepte, die nur auf der rechten Seite auftreten – Basissymbole $\mathcal{B}_{\mathcal{T}}$
- Endliche Menge von Definitionen $\mathcal{T}$ heisst Terminologie (TBox), wenn kein Namenssymbol zwei Mal definiert ist

TBox: Terminologien

- Eine Interpretation $\mathcal{I}$ erfüllt $\mathcal{T}$ gdw. $\mathcal{I}$ erfüllt jedes Axiom aus $\mathcal{T}$.
  $\mathcal{I}$ ist ein Modell von $\mathcal{T}$
- Zwei Mengen von Axiomen sind äquivalent gdw. sie den gleichen Modell haben
- Eine Interpretation $\mathcal{J}$, die nur die Basissymbole $\mathcal{B}_{\mathcal{T}}$ definiert, heißt Basisinterpretation
- Interpretation $\mathcal{I}$ heißt eine Erweiterung von $\mathcal{J}$ gdw. $\triangle^{\mathcal{I}} = \triangle^{\mathcal{J}}$ und $A^{\mathcal{I}} = A^{\mathcal{J}}$ $\forall A \in \mathcal{B}_{\mathcal{T}}$
- Terminologie $\mathcal{T}$ heißt definitorisch wenn es genau eine Erweiterung $\mathcal{I}$ gibt, die ein Modell von $\mathcal{T}$ ist
- Terminologie $\mathcal{T}$ definitorisch gdw. $\mathcal{T}$ azyklisch

- Wenn man jedes Namenssymbol in der rechten Seite durch seine Definition ersetzt erhält man Definitionen an deren rechte Seite nur Basissymbole stehen. Das Prozess nennt man Entfaltung von $\mathcal{T}$ ($\mathcal{T}'$)
  - $\mathcal{T}$ und $\mathcal{T}'$ haben dieselben Namens- und Basissymbole
  - $\mathcal{T}$ und $\mathcal{T}'$ sind äquivalent
  - Wenn $\mathcal{T}$ definitorisch, dann $\mathcal{T}'$ auch definitorisch

TBox Beispiel

Frau $\equiv$ Person $\sqcap$ Weiblich

Mann $\equiv$ Person $\sqcap$ Männlich

Mutter $\equiv$ Frau $\sqcap$ $\exists$hatKind.Person

Vater $\equiv$ Mann $\sqcap$ $\exists$hatKind.Person

Elternteil $\equiv$ Vater $\sqcup$ Mutter

Großmutter $\equiv$ Frau $\sqcap$ $\exists$hatKind.Eternteil

Namenssymbole $\mathcal{N}_{\mathcal{T}}$ = {Frau, Mann, Mutter, Vater, Elternteil, Großmutter}

Basissymbole $\mathcal{B}_{\mathcal{T}}$ = {Person, Weiblich, Männlich}

Entfaltung

Mutter $\equiv$ <u>Frau</u> $\sqcap$ $\exists$hatKind.Person $\equiv$ Person $\sqcap$ Weiblich $\sqcap$ $\exists$hatKind.Person

Großmutter $\equiv$ <u>Frau</u> $\sqcap$ $\exists$hatKind.<u>Elternteil</u> $\equiv$

$\equiv$ Person $\sqcap$ Weiblich $\sqcap$ $\exists$hatKind.<u>Elternteil</u>

$\equiv$ Person $\sqcap$ Weiblich $\sqcap$ $\exists$hatKind.(Person $\sqcap$ Männlich

$\sqcap$ $\exists$hatKind.Person $\sqcup$ Person $\sqcap$ Weiblich $\sqcap$ $\exists$hatKind.Person)

ABox: Weltbeschreibung

Eine endliche Menge von Aussagen über die Individuen, die extensionales Wissen über die Welt bezüglich einer TBox beschreibt

- die Aussagen der Form C(a) oder R(a, b)
  a, b – Individuen, C – Konzept, R – Rolle

- Interpretation $\mathcal{I}$ ($\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}}$) wird um Individuen $a^{\mathcal{I}} \in \triangle^{\mathcal{I}}$ erweitert
  $\mathcal{I} \models C(a)$ gdw. $a^{\mathcal{I}} \in C^{\mathcal{I}}$ und $\mathcal{I} \models R(a, b)$ gdw. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

ABox Beispiel

Weiblich(SARAH)

Vater(JOHN)

hatKind(JOHN, SARAH)

Inferenz der TBox

Inferenzdienste der TBox

- Erfülbarkeit
  Konzept C heißt erfülbar gdw. es existiert ein Modell $\mathcal{I}$ von $\mathcal{T}$, so dass $C^{\mathcal{I}} \neq \emptyset$

- Subsumierung
  D subsumiert C bzgl. $\mathcal{T}$ gdw. $\forall$ Modell $\mathcal{I}$ von $\mathcal{T}$ gilt die Inklusion $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
  ($\mathcal{T} \vDash C \sqsubseteq D$)

- Äquivalenz
  C ist äquivalent D bzgl. $\mathcal{T}$ gdw. $\forall$ Modell $\mathcal{I}$ von $\mathcal{T}$ gilt: $C^{\mathcal{I}} = D^{\mathcal{I}}$ ($\mathcal{T} \vDash C \equiv D$)

- Disjunktheit
  C und D sind disjunkt bzgl. $\mathcal{T}$ gdw. $\forall$ Modell $\mathcal{I}$ von $\mathcal{T}$ gilt: $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$

Inferenz der ABox

Inferenzdienste der ABox

- Konsistenz
  $\mathcal{A}$ ist konsistent wenn ein Modell von $\mathcal{A}$ existiert

- Konsistenz bzgl. einer TBox $\mathcal{T}$
  $\mathcal{A}$ ist konsisten bzgl. einer TBox $\mathcal{T}$ wenn ein gemeinsames Modell von $\mathcal{A}$ und $\mathcal{T}$ existiert

- Instanzcheck $\mathcal{A} \models C(a)$
  Kann durch Konsistenzcheck ausgedrückt werden
  $\mathcal{A} \models C(a) \Leftrightarrow \mathcal{A} \cup \{\neg C(a)\}$ inkonsistent

- Die Erfüllbarkeit des Konzepts kann auf Instanzcheck reuduziert werden
  C ist erfüllbar gdw. $\{C(a)\}$ ist konsistent

Inferenz

- Inferenzdienste der TBox können auf Subsumierung bzw. auf Unerfüllbarkeit reduziert werden.

- Reduktion auf Subsumierung
  - C ist unerfüllbar $\Leftrightarrow$ C $\sqsubseteq \bot$
  - C $\equiv$ D $\Leftrightarrow$ C $\sqsubseteq$ D und D $\sqsubseteq$ C
  - $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset \Leftrightarrow C \sqcap D \sqsubseteq \bot$

- Reduktion auf Unerfüllbarkeit
  - C $\sqsubseteq$ D $\Leftrightarrow$ C $\sqcap \neg$D unerfüllbar
  - C $\equiv$ D $\Leftrightarrow$ sowohl C $\sqcap \neg$D als auch $\neg$C $\sqcap$ D unerfüllbar
  - $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset \Leftrightarrow$ C $\sqcap$ D unerfüllbar

- Eliminierung der TBox $\mathcal{T}$ durch die Entfaltung auf $\mathcal{T}'$
  - C ist erfüllbar bzgl. $\mathcal{T}$ gdw. C' erfüllbar
  - $\mathcal{T} \models C \sqsubseteq D$ gdw. $\models C' \sqsubseteq D'$
  - $\mathcal{T} \models C \equiv D$ gdw. $\models C' \equiv D'$
  - C und D disjunkt bzgl $\mathcal{T}$ gdw. C' und D' disjunkt

Structural Subsumption Algortihmus

- macht einen Subsumierungstest von zwei Konzepten in einer Normalen Form
- anwendbar nur für nicht Ausdruckskräftige BL Sprachen
- BL Sprachen mit Schnitt, Negation, existenziele Quantifizierung ist das Algorithmus nich komplett.

Tableau-Based Algorithmus für $\mathcal{ALCN}$

Anstatt ein Subsumierungstest auf die Konzeptdefinition zu machen, macht das Tableau Algorithmus einen Erfüllbarkeitstest

$C \sqsubseteq D \Leftrightarrow C \sqcap \neg D$ unerfüllbar

- Zwingt keine einzigartige Individuennamen für die ABox
  Führt Ungleichheit - Assertions (inequality assertions) $x \neq y$ wenn $x^{\mathcal{I}} \neq y^{\mathcal{I}}$ (symetrisch)

- C_0 – ein $\mathcal{ALCN}$ Konzept in Negation Normaler Form (negation normal form). Die Negation steht nur vor den atomaren Konzepten.

- Um die Erfüllbarkeit zu testen, startet das Algorithmus mit einer ABox $\mathcal{A}_0 = \{C_0\}$
- Die Transformationsregeln werden angewandt auf die ABox bis keine mehr angewandt werden können.
- Wenn die in diesem Weg rausgekommene "komplette" ABox keine Widersprüche (auch clash genannt) enthält, dann ist $\mathcal{A}_0$ konsistent und C_0 erfüllbar

Tableau-Based Algorithmus für $\mathcal{ALCN}$

- Weil die Transformationsregel mehr als eine ABox erzeugen kann, betrachten wir Mengen von ABox'en $\mathcal{S} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$

- $\mathcal{S}$' ist rausgekommen durch die Anwendung von Transformationsregeln auf $\mathcal{S}$
  $\mathcal{S}$ ist konsistent gdw. $\mathcal{S}$' ist konsistent
- $\mathcal{S}$' ist konsisten wenn ein von $\mathcal{A}'_i$ konsistent ist

- Transformation terminiert immer

- Clash
  - $\{\perp(x)\} \subseteq \mathcal{A}$
  - $\{A(x), \neg A(x)\} \subseteq \mathcal{A}$
  - $\{(\leq nR)(x)\} \cup \{R(x, y_i) \mid 1 \leq i \leq n + 1\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n + 1\} \subseteq \mathcal{A}$

- Es ist entscheidbar ob $\mathcal{ALCN}$ – Konzept erfüllbar ist

Tableau-Based Algorithmus: Transformationsregeln

The $\rightarrow_{\sqcap}$-rule
Condition: $\mathcal{A}$ contains $(C_1 \sqcap C_2)(x)$, but it does not contain both $C_1(x)$ and $C_2(x)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C_1(x), C_2(x)\}$.

The $\rightarrow_{\sqcup}$-rule
Condition: $\mathcal{A}$ contains $(C_1 \sqcup C_2)(x)$, but neither $C_1(x)$ nor $C_2(x)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C_1(x)\}$, $\mathcal{A}'' = \mathcal{A} \cup \{C_2(x)\}$.

The $\rightarrow_{\exists}$-rule
Condition: $\mathcal{A}$ contains $(\exists R.C)(x)$, but there is no individual name z such that $C(z)$ and $R(x, z)$ are in $\mathcal{A}$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C(y), R(x, y)\}$ where y is an individual name not occurring in $\mathcal{A}$.

The $\rightarrow_{\forall}$-rule
Condition: $\mathcal{A}$ contains $(\forall R.C)(x)$ and $R(x, y)$, but it does not contain $C(y)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C(y)\}$.

The $\rightarrow_{\geq}$-rule
Condition: $\mathcal{A}$ contains $(\geqslant n\, R)(x)$, and there are no individual names $z_1, \ldots, z_n$ such that $R(x, z_i)$ $(1 \leq i \leq n)$ and $z_i \neq z_j$ $(1 \leq i < j \leq n)$ are contained in $\mathcal{A}$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{R(x, y_i) \mid 1 \leq i \leq n\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$, where $y_1, \ldots, y_n$ are distinct individual names not occurring in $\mathcal{A}$.

The $\rightarrow_{\leq}$-rule
Condition: $\mathcal{A}$ contains distinct individual names $y_1, \ldots, y_{n+1}$ such that $(\leqslant n\, R)(x)$ and $R(x, y_1), \ldots, R(x, y_{n+1})$ are in $\mathcal{A}$, and $y_i \neq y_j$ is not in $\mathcal{A}$ for some $i \neq j$.
Action: For each pair y_i, y_j such that $i > j$ and $y_i \neq y_j$ is not in $\mathcal{A}$, the ABox $\mathcal{A}_{i,j} = [y_i / y_j]\mathcal{A}$ is obtained from $\mathcal{A}$ by replacing each occurrence of y_i by y_j.

Die Komplexität

- Mit der ansteigenden Ausdruckskräftigkeit der BL Sprachen, steigt auch die Komplexität der Reasoning – Algorithmen

Sprache	$\models C \sqsubseteq D$	$\models C(a)$
$\mathcal{AL}$	P	P
$\mathcal{ALE}$	NP	PSPACE
$\mathcal{ALC}$	PSPACE	PSPACE
$\mathcal{SHIQ}$	EXPTIME	EXPTIME

- $\mathcal{SHIQ}$ - BL Sprache mit folgenden Erweiterungen:

  inverse, transitive Rollen, Sub – Rollen (subroles) und
  Qualified – Anzahbeschränkung ($\geqslant$ nR.C – z.B. $\leq$1 hatKind.Männlich : hat meistens einen Sohn)

Ende

Danke für die Aufmerksamkeit

Fragen ?