Baader-Nutt 1: Basic Description Logics

Ludwig-Maximilians Universität München
Sommersemester 2009
Institut: CIS

Hauptseminar: Beschreibungslogik
Seminarleiter: Hans Leiß, Martin Hofmann
Referentin: Natalia Filatova

Zusammenfassung

- ✓ Beschreibungslogik (BL) als Formale Sprache:
  - Allgemeine Vorstellung von grundliegenden Ideen
- ✓ Struktur der Wissensdarstellung od. *Knowledge Representation* (KR) System:
  - Beschreibungssprachen
  - Syntax und Semantik der AL-Sprachen
  - T Box / Terminologies
  - A Box / Assertions
  - Schlussfolgerungen / Inferences

Beschreibungslogiken (1)
(engl. *Description Logics DL*)

- Familie von Sprachen für Wissensrepräsentation (*Knowledge Representation KR*)
- Wissen über eine Welt, bzw. Anwendungsgebiet:

  Entsprechende Konzepte und Rollen vom Domain (Terminologie) definieren;

  Eigenschaften der Objekten od. Individuen im Domain verwenden.
- Ausstattung mit formale, Logikbasierte Semantik
- Als Basisservice – Logisches Denken: aus implizites Wissen den explizite Wissen zu gewinnen
- In der Anwendung von Bearbeitungssysteme und Menschen liegen:

  Klassifikation von Konzepten und Klassifikation von Individuals:

Beschreibungslogiken (2)
(engl. *Description Logics DL*)

- **Klassifikation von Konzepten** bestimmt „subconcept-superconcept“ Beziehungen *(„Subsumption relationship“)* zwischen Konzepten der gegebene Terminologie. So wird Terminologie in eine Subsumption Hierarchie strukturiert. Diese Hierarchie liefert die Informationen über die Beziehungen zwischen verschiedenen Konzepten; und kann auch den Schlussfolgerungsprozessen beschleunigen.
- **Klassifikation von Individuals** (od. Objekten) bestimmt, ob ein gegebener Individual immer den Fall vom bestimmten Konzept ist und liefert Informationen über Merkmale von Objekten.

Bemerkung:

- Die Konzeptbeschreibungen einer Anwendungsdomäne werden so dargestellt, indem komplexere Ausdrücke durch atomare Konzepte und atomare Rollen durch die jeweiligen Konstruktoren gebildet werden.

Beschreibungslogiken (3)
(engl. *Description Logics DL*)

- Die meisten Beschreibungslogiken sind eine Untermenge der Prädikatenlogik erster Stufe, im Gegensatz zu dieser aber entscheidbar. Dies ermöglicht über eine Beschreibungslogik zu schließen, d.h. aus vorhandenem Wissen ein neues Wissen zu gewinnen. => Schlussfolgerungen

- BL ist KR Formalismus -> KR soll auf die Abfrage entsprechend Logisches Denken beantwortet sein. Also, Logische Verfahren von BL sind in Entscheidungsverfahren beteiligt. Im Unterschied zu 1st PL Theorem-Beweise sollen diese Entscheidungsverfahren für negative und positive Antworte beschlossen sein. Entscheidungsprozess und Schwierigkeiten der Schlussfolgerungen Probleme sind von Ausdrucksmöglichkeiten der BL abhängig. Die meisten Probleme liegen in:
    - ✓ Tausch der Expressivität der BL
    - ✓ Komplexität der Logik

Struktur der KR System, die auf BL basiert ist:

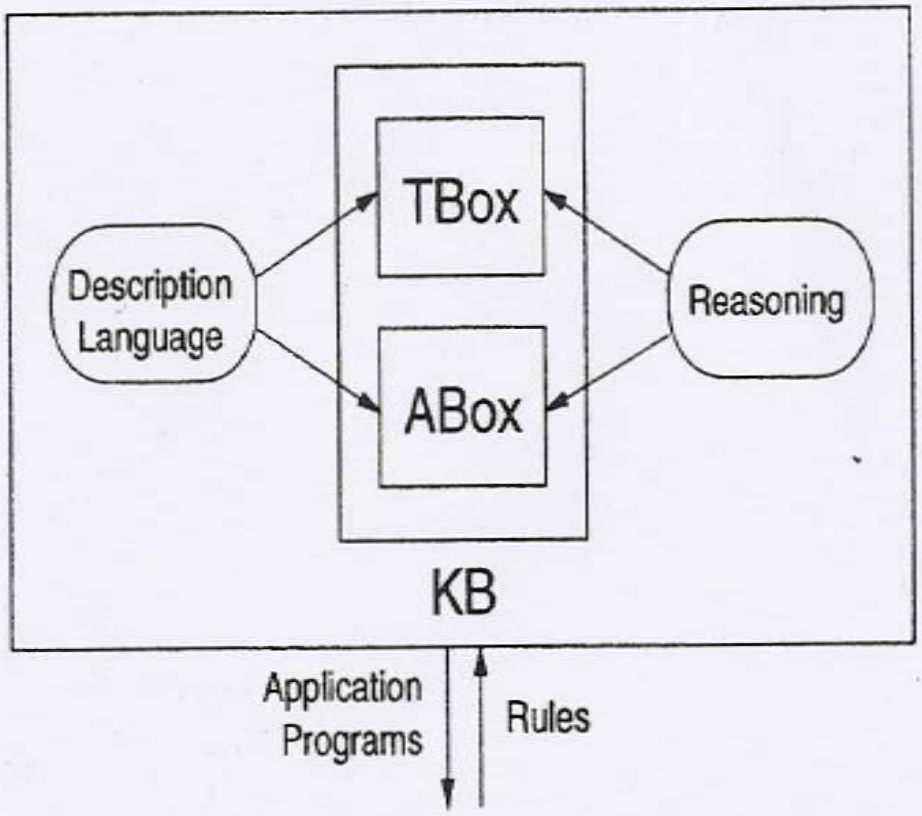

Fig. 2.1. Architecture of a knowledge representation system based on Description Logics.

Definition des grundliegenden Formalismus (1)

- KR gibt die Möglichkeiten die Wissensbasis aufzubauen, den Inhalt zu analysieren und manipulieren.
- Knowledge base (KB) besteht aus 2 Komponenten: TBox und ABox.
- TBox enthält das Wissen über die Konzepte einer Domäne, das terminologische Wissen. => Vokabular von Anwendungsgebiete
- ABox hingegen enthält das Wissen über Entitäten oder Instanzen dieser Konzepte, sowie deren Beziehungen untereinander, und repräsentiert den Zustand der modellierten Welt. => Behauptungen über Individuals im Rahmen der oben genannte Vokabular
- Dieses Vokabular besteht aus Konzepten und Rollen:
  - Konzepten beschreiben die Reihe von Entitäten
  - Rollen beschreiben die binäre Beziehungen zwischen Entitäten

Definition des grundliegenden Formalismus (2)

- Außerdem alle BL Systeme erlauben nicht nur atomare Konzepten und Rollen zu formen, sondern auch eine **komplexe Beschreibung** von diesen Konzepten und Rollen zu bilden. TBox kann benutzt werden um Namen für diese komplexe Beschreibungen zu zuweisen.

- Die Sprache für Beschreibungsbildung ist ein charakteristisches Merkmal jedes BL-System im einzeln. Die BL-Systeme können sich in benutzter Sprache unterscheiden. Die Beschreibungssprache hat Model-Theoretische Semantik. Die Aussagen im TBox und ABox können sich formal mit PL erster Stufe oder ihre Erweiterung identifizieren.

Definition des grundliegenden Formalismus (3)

BL-System speichert nicht nur Terminologien und Behauptungen, sondern bietet auch die logische Folgerung an.

Typische Aufgaben für Terminologie sind:

- zu bestimmen, ob Beschreibung passt (non-contradictory);
- ob eine Beschreibung allgemeiner als andere ist, bzw. ob erste die zweite einordnet;

Probleme der ABox sind:

- rauszufinden, ob die Reihe von Behauptungen folgerichtig ist;
- ob es passende Modell dafür gibt;
- ob Assertions in ABox enthalten diesen bestimmten Entitäten, was genau der Fall des gegebenen Konzeptbeschreibung ist.

Bemerkung:

Ob KB eine Bedeutung hat, überprüfen die „Subsumption“ und „Instance“ Tests (unten)

Beschreibungssprachen (1)

Die **elementaren Beschreibungen** sind Atomare Konzepte und Atomare Rollen .
Die **komplexe Beschreibungen** werden von denen mit Hilfe der „concept constructors“ induktiv konstruiert.

❖ Es wird benutzt:

- **A, B,…** ->für atomare Konzepte
- **R, S,…** ->für atomare Rollen
- **C, D,…** -> für Konzeptbeschreibungen

Die Beschreibungssprachen unterscheiden sich in „constructors“, die sie anbieten. Hier werden die Sprachen aus *AL*-Familie (attributive Sprache) vorgestellt. *AL* ist eine minimale Sprache; die anderen von dieser Familie sind ihre Erweiterungen.

Beschreibungssprachen (2)

Syntax der AL-Sprachen

C, D → A | (atomares Konzept)

⊤ | (universales oder Top-Konzept)

⊥ | (spezielles oder bottom-Konzept)

¬A | (atomare Negation)

C ⊓ D | (Konzeptkonjunktion)

∀ R.C | (universelle Quantifikation / Wertrestriktion)

∃ R.⊤ (beschränkte existentielle Quantifikation)

- Intuitiv bedeutet die Negation von A „alles außer A“ und die Konjunktion von C und D „sowohl C als auch D“.
- Die universelle Quantifikation kann übersetzt werden : „alle x, deren alle Beziehungen vom Typ R mit einem y vom Typ C sind“.
- Die beschränkte existentielle Quantifikation bedeutet „alle x, die eine Beziehung vom Typ R haben“.

Beschreibungssprachen (3)

Semantik der AL-Sprachen

Die Semantik einer AL-Beschreibungslogik basiert auf der Interpretation $I = (\Delta^I, .^I)$, wobei Δ^I als Domäne eine nichtleere Menge ist und die Funktion $.^I$ jeden Konzeptnamen A auf eine Teilmenge A^I von Δ^I abbildet und jeder atomaren Rolle R eine binäre Relation $R^I \subseteq \Delta^I \times \Delta^I$ zuordnet. Die Semantik wird dann wie folgt formuliert:

$\top^I$	=	Δ^I
$\bot^I$	=	{} (leere Menge)
$(\neg A)^I$	=	$\Delta^I \setminus A^I$
$(C \sqcap D)^I$	=	$C^I \cap D^I$
$(\forall R.C)^I$	=	$\{a \in \Delta^I \mid \forall b.\langle a,b\rangle \in R^I \Rightarrow b \in C^I\}$
$(\exists R.\top)^I$	=	$\{a \in \Delta^I \mid \exists b.\langle a,b\rangle \in R^I\}$

Zudem gilt für Äquivalenz zweier Konzepte C und D: $C \equiv D \Leftrightarrow C^I = D^I$.

Beschreibungssprachen (4)

Beispiel von Erweiterungen der AL-Sprache

- ✓ Vereinigung der Konzepten - > ***U*** (Union)
- ✓ Volle Existenz Quantifizierung -> ***E*** (Full Existential Quantification)
- ✓ Zahlbeschränkung -> ***N*** (Number Restrictions)
- ✓ Negation -> ***C*** (C for „Complement“)

AL [U] [E] [N] [C]

ALEN-> ist eine Erweiterung mit den Konstruktoren:
Volle Existenz Quantifizierung und Zahlbeschränkung.

Semantische Äquivalenz durch Negation erlaubt:

Statt *UE* - > *C*

ALUE - > *ALC*

ALUEN - > *ALCN*

Terminologien (1)

- Die Wissensbasis wird mittels BL in zwei Kategorien eingeteilt: die intensionale T-Box und die extensionale A-Box. Das terminologische oder auch definitorische Wissen über die interessierenden Konzepte eines Anwendungsbereichs wird durch die T-Box dargestellt. Das assertionale Wissen in der A-Box hingegen enthält Informationen über die im Anwendungsbereich tatsächlich existierenden Dinge und Beziehungen.
- Eine T-Box *T* stellt die Struktur dar, wie Konzepte und Rollen miteinander in Beziehung stehen und wird durch folgendes Axiom aufgebaut:

Schema:

„Terminological axioms“ -> erklären, wie die Konzepten und Rollen mit einander verbunden sind und werden durch ***2 Arten von Axiomen*** aufgebaut:

Inclusions (Aufnahmen) ***Equalities*** (Gleichheiten)

Semantik der Axiomen ist definiert als „es wird erwartet“. -> Interpretation

Terminologien (2)

- Ist die Menge von Definitionen endlich und kein symbolischer Name ist mehr als einmal definiert, spricht man von einer Terminologie bzw. T-Box. Zudem nennt man die Symbole der linken Seite eines atomaren Konzepts der T-Box T ***Namenssymbol*** und die, die nur auf der rechten Seite stehen können ***Basissymbole***. Eine Interpretation J von T, die nur Basissymbole interpretiert, heißt ***Basisinterpretation***. Existiert eine Interpretation I, die auch Namenssymbole interpretiert und die gleiche Domäne wie J hat, spricht man von einer Extension oder Erweiterung von J. Hat jede Basisinterpretation eine Extension, die Modell für T ist, heißt die Terminologie ***definitorisch***. Insbesondere sind ***azyklische Terminologien***, also **Terminologien, die sich im Gegensatz zu zyklischen Terminologien nicht selbst benutzen**, definitorisch. Eine Expansion T ´ von T erhält man durch schrittweises Ersetzen der Namenssymbole durch ihre Definitionen, solange T azyklisch ist. Für T ´ und T gilt:
  - *T ´ und T haben die gleichen Namens- und Basissymbole*
  - *T ´ ⇔ T*
  - *T ´ und T sind beide definitorisch*

Terminologien (3)

Beispiel für TBox „Familie“

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female}\\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman}\\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild.Person}\\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild.Person}\\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother}\\
\mathsf{Grandmother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild.Parent}\\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap \geqslant 3\ \mathsf{hasChild}\\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild}.\neg\mathsf{Woman}\\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband.Man}
\end{aligned}
$$

Fig. 2.2. A terminology (TBox) with concepts about family relationships.

Terminologien (4)

Beispiel für Erweiterung von TBox „Familie“

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female}) \\
\mathsf{Mother} &\equiv (\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} &\equiv (\mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female})) \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} &\equiv ((\mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female})) \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \\
&\quad \sqcup ((\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \\
\mathsf{Grandmother} &\equiv ((\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \\
&\quad \sqcap \exists \mathsf{hasChild}.(((\mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female})) \\
&\qquad\qquad \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \\
&\qquad\qquad \sqcup ((\mathsf{Person} \sqcap \mathsf{Female}) \\
&\qquad\qquad\quad \sqcap \exists \mathsf{hasChild}.\mathsf{Person})) \\
\mathsf{MotherWithManyChildren} &\equiv ((\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \sqcap \geqslant 3\, \mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} &\equiv ((\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{hasChild}.\mathsf{Person}) \\
&\quad \sqcap \forall \mathsf{hasChild}.(\neg(\mathsf{Person} \sqcap \mathsf{Female})) \\
\mathsf{Wife} &\equiv (\mathsf{Person} \sqcap \mathsf{Female}) \\
&\quad \sqcap \exists \mathsf{hasHusband}.(\mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female}))
\end{aligned}
$$

Fig. 2.3. The expansion of the Family TBox in Figure 2.2.

Terminologien (5)

Terminologien mit „Inclusion Axioms"

- Die einigen Konzepten kann man schwierig komplett definieren. In dem Fall nennen wir Aufnahmen, deren linke Seite atomare ***Spezialisation*** ist.

z.B. wenn ein Wissensingenieur denkt, dass die Definition von „woman" in TBox „Familie" nicht genügend erklärt ist, und kann nicht den Konzept „woman" detailiert definieren, dann kann er annehmen, dass „jede Frau eine Person mit Spezialisation" ist.

- Wenn wir Spezialisation in Terminologie nehmen, dann verliert die Terminologie ihre definitorial Effekt.

Lösung: Generalisierte Terminologie → in Reguläre Terminologie zu transformieren.

Ergebnis: $\text{Woman} = \overline{\text{Woman}} \sqcap \text{Person}$

ABox „World descriptions“

Hier werden die bestimmte Zustände von Anwendungsdomain in Termen von Konzepten und Rollen beschrieben: C(a), R(b,c), wo a,b,c – Individualsnamen

*Zustand, wo **a** gehört zu **C*** *Zustand, wo **c** erfüllt die Rolle **R** für **b***

Beispiel:

PETER, PAUL und MARY – Individualsnamen, dann

Father (PETER) – Peter ist Vater

hasChild (MARY,PAUL) – Paul ist Kind von Mary

→ABox -> *A* = Menge von solche Behauptungen

→TBox *T* verfügt über semantische Relationen zwischen Konzepten und Rollen in Abox.

Semantik und Interpretation von ABox

- Eine Interpretation *I* erfüllt C(a), genau dann wenn $a^I \in C^I$.
- Ein nützlicher Konstruktor für die ABox ist der sog. *set-Konstruktor*. Mit ihm kann man vorgeben, welche Individuen Instanz eines Konzepts sein dürfen. Interpretiert heißt das:

$$\{a_1, ..., a_n\}^I = \{a_1^I, ..., a_n^I\}.$$

Inferences / Schlussfolgerungen (1)

- ✓ KR ist auf BL basiert. BL selbst kann als spezielle Art von Logisches Denken dargestellt werden. Das Ziel von KR geht über Konzepten und Behauptungen hinaus. Wissensbasis, die aus TBox und ABox besteht, hat eigene Semantik mit ihre Menge von Axiomen, die implizites Wissen erhalten. Dieses implizite Wissen kann man explizit durch den Schlussfolgerungen machen.
- ✓ In der Beschreibungslogik werden wie die Wissensbasis selbst die Inferenzprobleme auf T-Box und A-Box aufgeteilt, um durch diese Trennung von Struktur und Individuen eine Verminderung der Komplexität zu erreichen.
- ✓ Das Grundproblem: um eine ausdrucksmächtige Logik zu finden, die gleichzeitig entscheidbares und effizientes Schließen anbietet. Inferenzdienste sollten entscheidbare und vollständige Entscheidungsprozeduren für Anwendungsprobleme darstellen.

Beispiel (Handout):

Aus TBox 2.2 und ABox 2.4 kann man schließen, dass Mary Oma ist, obwohl dieses Wissen steht nicht explizit als Behauptung fest.

Inferences / Schlussfolgerungen (2)

These properties are formally defined as follows. Let $\mathcal{T}$ be a TBox.

Satisfiability A concept C is *satisfiable* with respect to $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is nonempty. In this case we say also that $\mathcal{I}$ is a *model* of C.

Subsumption A concept C is *subsumed* by a concept D with respect to $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$.

Equivalence Two concepts C and D are *equivalent* with respect to $\mathcal{T}$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \equiv_{\mathcal{T}} D$ or $\mathcal{T} \models C \equiv D$.

Disjointness Two concepts C and D are *disjoint* with respect to $\mathcal{T}$ if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ of $\mathcal{T}$.

Inferences / Schlussfolgerungen (3)

Reduktionen:
Reduktion auf Subsumption

Proposition 2.12 (Reduction to Subsumption) *For concepts C, D we have*

(i) *C is unsatisfiable* $\Leftrightarrow$ *C is subsumed by $\bot$;*

(ii) *C and D are equivalent* $\Leftrightarrow$ *C is subsumed by D and D is subsumed by C;*

(iii) *C and D are disjoint* $\Leftrightarrow$ *$C \sqcap D$ is subsumed by $\bot$.*

The statements also hold with respect to a TBox.

Inferences / Schlussfolgerungen (4)

Reduktionen:
Reduktion auf Nicht-Erfüllbarkeit

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Proposition 2.13 (Reduction to Unsatisfiability) *For concepts C, D we have*

(i) *C is subsumed by D $\Leftrightarrow$ $C \sqcap \neg D$ is unsatisfiable;*
(ii) *C and D are equivalent $\Leftrightarrow$ both $(C \sqcap \neg D)$ and $(\neg C \sqcap D)$ are unsatisfiable;*
(iii) *C and D are disjoint $\Leftrightarrow$ $C \sqcap D$ is unsatisfiable.*

The statements also hold with respect to a TBox.

Inferences / Schlussfolgerungen (5)

Reduktionen:
Reduktion auf Äquivalenz

Proposition 2.14 (Reducing Unsatisfiability) *Let C be a concept. Then the following are equivalent:*

(i) *C is unsatisfiable;*
(ii) *C is subsumed by $\bot$;*
(iii) *C and $\bot$ are equivalent;*
(iv) *C and $\top$ are disjoint.*

The statements also hold with respect to a TBox.

Die Generation von BL Systemen

- KRIS
- CRACK
- FACT
- DLP
- RACE

sind auf Subsumption-Test basiert.