

# Uvod v opisno logiko

I.Savnik, FAMNIT

# Opisna logika

- Formalizem za predstavitev podatkov in znanja.
- Združuje naslednje modele:
  - Semantične mreže
  - Objektni model
  - Semantični podatkovni model
  - Ontologije
  - ...
- Strukturiran del predikatnega računa.
- Formalna osnova za predstavitev strukturiranih podatkov in sklepanje na osnovi teh.

# Opisna logika

- Zadnje ime za družino jezikov za predstavitev znanja
- Način predstavitve znanja
  - Predstavitev osnovnih konceptov domene
  - Uporaba konceptov za predstavitev dejstev
- Za razliko od predhodnikov je predstavitev znanja osnovana na logiki
  - Semantika jezika je vezana na logiko
  - Omogočeno je sklepanje: izpeljava znanja, ki je implicitno iz eksplicitno zapisanega znanja

# Opisna logika

- Omogoča uporabo vzorcev sklepanja, ki se uporabljajo v modernih inteligentnih sistemih
  - Ljudje uporabljajo iste vzorce
  - Klasifikacija konceptov in primerkov konceptov
- Klasifikacija konceptov določa pod-koncepte in nad-koncepte
  - Relacija vsebovanja v DL
  - Hierarhija vsebovanja
    - Uporabni podatki o zvezah z ostalimi koncepti
    - Pohitritev sklepanja zaradi hierarhije
- Klasifikacija primerkov pove ali so primerki instance danega koncepta

# Strukturirana omrežja dedovanja

- DL je naslednik strukturiranih omrežij dedovanja [Brachman, 1977b; 1978]
  - Razrešitev dvoumnosti, ki so prisotne v semantičnih mrežah in okvirjih
  - Prva realizacija: KL-One [Brachman and Schmolze, 1985] vsebuje naslednje 3 ideje:
    - Osnovni sitaktični objekti: koncepti, vloge in konstante (unarni, binarni predikati in konstante v PR)
    - Majhna množica konstruktorjev za gradnjo kompleksnih konceptov in vlog iz enostavnih
    - Implicitno znanje lahko izpeljemo z uporabo sklepanja
      - Vsebovanje konceptov (vlog) in relacija instanciranja
      - NE isa relacija kot v semantičnih mrežah!
- PMJ, 2011/12 – Pomembne so relacije def na osnovi lastnosti

# Sklepanje = NP

- Po tem ko so bili predlagani prvi jeziki za predstavitev znanja na osnovi logike so natančno definirani postopki za sklepanje
  - Jeziki začetnih sistemov so bili(?) preveč izrazni
  - Sklepanje je neodločljivo [Schmidt-Schauß, 1989; Patel-Schneider, 1989b]
  - Pokazano je bilo, da je worst-case kompleksnost eksponentna tudi za precej enostavne jezike [Levesque and Brachman, 1987; Nebel, 1988]
- Začetna točka raziskav worst-case kompleksnosti jezikov podobnih KL-One

# Pregled izvorov

- Konceptualni grafi
- KL-One
- Okvirji
- Logika

# Pregled: KL-One

Predstavitev strukture podatkov in znanja:

- Kompleksne zveze med podatki.
- Jezik ima množico gradnikov: koncepti, ISA, vloga,...

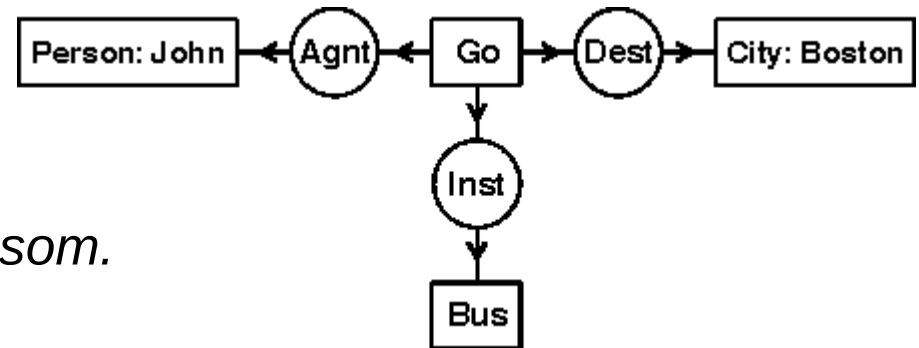
Razlikovanje med konceptualnim nivojem in nivojem instanc.

KL-ONE uporabljen za:

- Avtomatsko klasificiranje
- Določanje vsebovanosti
- Sklepanje ...



# Konceptualni grafi



*Janez gre v Boston z avtobusom.*

- John Sowa, Knowledge representation
- 15 let po objavi knjige se je izkazalo, da se konceptualni grafi lahko izrazijo s predikatnim računom
- Moderna izvedba konceptualnih grafov je opisna logika..

# Pregled: okvirji

- Okvirji (objekti), identifikatorji
  - Razredi in instance
- Predal (atribut)
  - Vrednost
    - Identifikator, privzeta vrednost
  - Vrednostne omejitve
    - Tip, domena. metode

# DL = Strukturirana logika

- Opisna logika je del predikatnega računa.
- Formalizem ne uporablja spremenljivk.
- Opisna logika je razdeljena na dva dela:
  - definicija predikatov (*TBox*)
  - izjave o konstantah (*ABox*)
- Vsaka opisna logika (osnovna) je podmnožica  $L3 = FOL$  brez funkcij, kjer izrazi vsebujejo največ tri spremenljivke.

# Zakaj ne PR?

- Uporaba predikatnega računa direktno brez dodatnih omejitev:
  - Izgubimo strukturo predstavljenega znanja. Ne moremo je uporabiti za vodenje sklepanja.
  - Izrazna moč jezika je prevelika. Izgubimo izračunljivost in s tem učinkovito sklepanje.
  - Izrazna moč sklepanja bi bila prešibka za izražanje zanimivih ter še vedno izračunljivih teorij.

# Aplikacije

- Opisna logika je **uporabljena** v praktičnih sistemih uporabljena za naslednje namene:
  - Osnova semantičnega spleta OWL
  - Konceptualno modeliranje
  - Optimizacija poizvedb in delo z okni
  - Opis pomena naravnih jezikov
  - Integracija podatkov in znanja
  - Dostop do podatkov in inteligentni vmesniki
  - Terminologije in ontologije
  - Upravljanje s programskimi sistemi
  - Planiranje

# Opisni jeziki

- Osnovni opisi so atomični koncepti in atomične vloge.
- Kompleksne opise konceptov gradimo z uporabo konstruktorjev.
  - $A, B$  – atomični koncepti.
  - $R$  – atomične vloge.
  - $C, D$  – opisi konceptov.
- Opisni jeziki se razlikujejo po konstrukcijskih pravilih, ki jih ima jezik na razpolago.
- Jeziki  $\mathcal{AL}$ 
  - $\mathcal{AL}$  (= attributive language) [Schmidt-Schauß & Smolka, 1991]
  - Minimalni jeziki, ki so praktično zanimivi.
  - Ostali jeziki iz te družine so razširitev  $\mathcal{AL}$ .

# Primeri konceptov in vlog

**Koncepti:** definirajo objekte (unarni predikati, razredi)

*Primer:* Student, Married

$$\{x \mid \text{Student}(x)\}$$
$$\{x \mid \text{Married}(x)\}$$

**Vloge:** definirajo lastnosti (binarni predikati)

*Primer:* FRIEND, LOVES

$$\{\langle x, y \rangle \mid \text{FRIEND}(x, y)\}$$
$$\{\langle x, y \rangle \mid \text{LOVES}(x, y)\}$$

# Primer: opis koncepta

- Opisna logika opisuje koncepte.
- Podatki z enako strukturo se združujejo na osnovi skupnih lastnosti množice instanc.

*Primer:*

Student  $\wedge$   $\exists$ FRIEND.Married

$\{x \mid \text{Student}(x) \wedge \exists y. \text{FRIEND}(x,y) \wedge \text{Married}(y)\}$



# Osnovni opisni jezik $\mathcal{AL}$

- ❖ Opis konceptov v  $\mathcal{AL}$  je definiran na osnovi sledečih sintaktičnih pravil:

$C, D \rightarrow A$	(atomični koncept)
$\top$	(univerzalni koncept)
$\perp$	(tla)
$\neg A$	(atomična negacija)
$C \sqcap D$	(presek)
$\forall R.C$	(omejitev vrednosti)
$\exists R.\top$	(omejena eksist. kvantifikacija)

# Kvantifikatorji

- Gradnik  $\forall R.C$  opisuje koncepte z vlogo  $R$ , ki ima zalogo vrednosti omejeno na koncept  $C$ .
  - $\{x \mid \forall y: R(x,y) \rightarrow C(y)\}$
  - $\forall \text{CHILD.Doctor} \equiv \{x \mid \forall y: \text{CHILD}(x,y) \rightarrow \text{Doctor}(y)\}$
  - Način omejitve vrednosti vloge.
- Gradnik  $\exists R$  opisuje koncepte z vlogo  $R$  (za katere je definirana vloga  $R$ ).
  - $\{x \mid \exists y: R(x,y)\}$
  - Koncepti  $x$ , ki imajo vsaj eno vlogo  $R$ .
  - Način definicije vloge koncepta.

# $\mathcal{FL}^-$ , $\mathcal{FL}_0$

- ❖ V  $\mathcal{AL}$  lahko negacijo apliciramo samo na atomične koncepte.
- ❖ Najvišji koncept je dovoljen v okviru eksistenčne kvantifikacije nad vlogami.
- ❖ Pod-jezik, ki ga dobimo s prepovedjo **atomične negacije** imenujemo  $\mathcal{FL}^-$ .
- ❖ Pod-jezik, ki ga dobimo s prepovedjo **omejene eksistenčne kvantifikacije** imenujemo  $\mathcal{FL}_0$

# Primeri

Person, Female

Person  $\sqcap$  Female

Person  $\sqcap$   $\neg$ Female

atomični koncept  
osebe, ki so ženske  
osebe, ki niso ženske

hasChild

Person  $\sqcap$   $\exists$ hasChild. $\top$

Person  $\sqcap$   $\forall$ hasChild.Female

atomična vloga  
oseba, ki ima enega otroka  
oseba, ki ima same hčere

Person  $\sqcap$   $\forall$ hasChild. $\perp$

oseba, ki nima otrok

# Intuitivna semantika

- **Intuitivno:**
  - Koncepti predstavljajo razrede, množice primerkov.
  - Vloge predstavljajo relacije med pari konceptov.
  - Atomični koncepti so predstavljeni z imeni primitivnih (osnovnih) konceptov.
- **Pomen sestavljenih konceptov:**
  - **Konstrukt  $\sqcap$**  predstavlja sestavljene koncepte, ki imajo pomen preseka obeh konceptov:  
$$\text{Adult} \sqcap \text{Male} \sqcap \text{Person}$$
  - **Konstrukt  $\sqcup$**  predstavlja sestavljene koncepte, ki imajo pomen unije obeh konceptov.

# Formalna semantika $\mathcal{AL}$

Interpretacijo  $I = (\Delta', \cdot^I)$  sestavljajo:

- neprazna množica  $\Delta'$  (domena)
- interpretacijska funkcija  $\cdot^I$

$$I : A \longrightarrow A^I \subseteq \Delta'$$

$$I : R \longrightarrow R^I \subseteq \Delta' \times \Delta'$$

# Formalna semantika $\mathcal{AL}$

Interpretacijska funkccija je razširjena na opise konceptov z naslednjimi induktivnimi definicijami:

$$\top' = \Delta$$

$$\perp' = \emptyset$$

$$(\neg A)' = \Delta' - A'$$

$$(C \sqcap D)' = C' \cap D'$$

$$(\forall R.C)' = \{x \in \Delta \mid \forall y. (x, y) \in R' \Rightarrow y \in C'\}$$

$$(\exists R.T)' = \{x \in \Delta \mid \exists y. (x, y) \in R'\}$$

$C'$  je množica individualnih objektov, ki so primerki koncepta  $C$ .

- Zapis  $x \in C'$  pomeni isto kot  $C(x)$ .
- Podobno zapis  $(x, y) \in R'$  pomeni isto kot  $R(x, y)$ .

# Ekvivalenca

$C \equiv D \iff C'=D'$  za vse interpretacije /

Primer:

$\forall \text{hasChild.Female} \sqcap \forall \text{hasChild.Student}$

$\equiv$

$\forall \text{hasChild.}(\text{Female} \sqcap \text{Student})$



# Družina jezikov $\mathcal{AL}$

Jeziki bolj izrazni jeziki od  $\mathcal{AL}$ :

- Unija
- Polna eksistenčna kvantifikacija
- Vrednostne omejitve
- Negacija

# Unija

Jezik  $\mathcal{ALU}$  !

$C \sqcup D$  - unija konceptov

Interpretacija:  $(C \sqcup D)' = C' \cup D'$

# Polna eksistenčna kvantifikacija

Jezik  $\mathcal{ALE}$ !

$\exists R.C$  - obstaja vloga R tipa C

$$(\exists R.C)' = \{a \in \Delta' \mid \exists b.(a,b) \in R' \wedge b \in C'\}$$

$\exists R.C$  je različen od  $\exists R.T$ :

zaloga vrednosti je pri slednjem poljubna medtem, ko je pri prvem omejena na primerke koncepta.

# Vrednostne omejitve

Jezik  $\mathcal{ALN}$  !

$\geq nR$  - najmanj  $n$

$\leq nR$  - največ  $n$

$$(\geq nR)' = \{a \in \Delta' \mid |\{b \mid (a,b) \in R'\}| \geq n\}$$

$$(\leq nR)' = \{a \in \Delta' \mid |\{b \mid (a,b) \in R'\}| \leq n\}$$

# Negacija poljubnega koncepta

Jezik  $\mathcal{ALC}$  !

$$(\neg C)' = \Delta' - C'$$

# $\mathcal{AL}^*$ : pregled

A	$A' \subseteq \Delta'$	primitivni koncept
R	$R' \subseteq \Delta' \times \Delta'$	primitivna vloga
$\top$	$\Delta'$	streha
$\perp$	$\emptyset$	tla
$\neg C$	$\Delta' \setminus C'$	komplement
$C \sqcap D$	$C' \cap D'$	konjunkcija
$C \sqcup D$	$C' \cup D'$	disjunkcija
$\forall R.C$	$\{x \mid \forall y. R'(x, y) \rightarrow C'(y)\}$	univerzalna kvant.
$\exists R.C$	$\{x \mid \exists y. R'(x, y) \wedge C'(y)\}$	eksistenčna kvant.

# Primer

Osebe, ki imajo bodisi največ enega otroka ali več kot tri od katerih je eden ženskega spola.

Oseba  $\Pi$

( $\leq 1$  imaOtroka  $\sqcup$

( $> 3$  imaOtroka  $\Pi \exists$ imaOtroka.Ženska))

*AL*[*U*][*E*][*N*][*C*];

Operacije lahko izrazimo drugimi operacijami.

$$C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$$

$$\exists R.C \equiv \neg \forall R.\neg C$$

Vse *AL* jezike lahko izrazimo z *UEN*.



# Terminologije

- Do sedaj smo predstavljali **razrede** kot kompleksne opise konceptov.
- V nadaljevanju so bomo ogledali **terminološke aksiome**, ki so stavki o razmerjih med koncepti in vlogami.
- Ogledali si bomo posamezne aksiome po tipih.

# Terminološki aksiomi

Naj bodo C,D *koncepti* in R, S *vloge*.

Ločimo med naslednjimi dvema vrstami aksiomov:

Vsebovanost:

$$C \sqsubseteq D \quad (R \sqsubseteq S)$$

Enakost:

$$C \equiv D \quad (R \equiv S)$$

.

# Terminološki aksiomi

Semantika aksiomov je definirana na običajen način:

- $C \sqsubseteq D \iff C' \subseteq D'$
- $C \equiv D \iff C' = D'$

Interpretacija  $I$  je rešitev aksioma ali množice aksiomov  $\implies I$  je model aksioma oz. množice aksiomov.

Dva aksioma sta enaka, če **imata** enaka modela.

# Definicije

Enakost katere leva stran je atomični koncept imenujemo **definicija**.

Definicije uporabljamo za zapis simboličnih imen namesto kompleksnih opisov.

*Primer:*

$$\text{Mother} \equiv \text{Woman} \sqcap \forall \text{hasChild. Person}$$

Simbolična imena lahko uporabimo kot **okrajšavo** v drugih opisih.

*Primer:* Recimo, da imamo definiran tudi koncept Father.

$$\text{Parent} \equiv \text{Mother} \sqcup \text{Father}$$

# Primer: družina

Woman  $\equiv$  Person  $\sqcap$  Female

Man  $\equiv$  Person  $\sqcap$   $\neg$ Woman

Mother  $\equiv$  Woman  $\sqcap$   $\exists$ hasChild.Person

Father  $\equiv$  Man  $\sqcap$   $\exists$ hasChild.Person

Parent  $\equiv$  Father  $\sqcup$  Mother

Grandmother  $\equiv$  Mother  $\sqcap$   $\exists$ hasChild.Parent

MotherWithManyChildren  $\equiv$  Mother  $\sqcap$   $>3$ hasChild

MotherWithoutDaughter  $\equiv$  Mother  $\sqcap$   $\forall$ hasChild. $\neg$ Woman

Wife  $\equiv$  Woman  $\sqcap$   $\exists$ hasHusband.Man

# Stavki TBox

- $A \sqsubseteq C$  definicija primitivnega koncepta
- $A \equiv C$  definicija koncepta
- $C \sqsubseteq D$  vsebovanost koncepta
- $C \equiv D$  enakost koncepta

# Pomen terminologij

- Naj bo  $T$  terminologija.
- Atomične koncepte iz  $T$  delimo na:
  - Simbole  $NT$ , ki predstavljajo **imena** in se pojavijo na levi strani aksiomov.
  - **Osnovne simbole**  $BT$ , ki se pojavijo samo na desni strani aksiomov.
- Imenske simbole pogosto imenujemo **definirane koncepte** in osnovne simbole imenujemo **primitivne koncepte**.
- Pričakujemo, da terminologija definira imena na osnovi primitivnih konceptov.

# Interpretacija

- **Osnovna interpretacija** terminologije T interpretira samo primitivne (osnovne) simbole.
- Naj bo  $\Lambda$  takšna osnovna interpretacija.
- Interpretacija I **razširi**  $\Lambda$  z interpretacijo definiranih simbolov, če je definirana nad isto domeno tako da velja  $\Delta^I = \Delta^\Lambda$  in se ujema v osnovnih simbolih z  $\Lambda$ .



Human'  $\equiv$  Animal  $\wedge$   $\forall$ hasParent.Human'

# Interpretacija

- T je **dobro definirana**, če imajo vse osnovne interpretacije natančno eno razširitev, ki so model T.
- Če poznamo interpretacijo osnovnih simbolov in če je T dobro definirana, potem je pomen imenskih simbolov natančno določen.
- Pravimo tudi, da je T **definitabilna** (angl. definitorial)
- Definiranost terminologije je povezana s strukturo aksiomov.
- **Če terminologija nima ciklov je dobro definirana.**

# Enostaven acikličen TBox

- $A \sqsubseteq C$  definicija primitivnega koncepta
- $A \equiv C$  definicija koncepta
  
- Koncept A **direktno uporablja** koncept B v TBox  $\Sigma$  če definicija A omenja B
- Koncept A **uporablja** koncept B če obstaja veriga direktnih uporab konceptov vse do B
- TBox je **acikličen**, če nobeden koncept ne uporablja samega sebe za definicijo

# Enostaven acikličen TBox

Vsebovanje v acikličnem enostavnem TBox

- $\Sigma \models C \sqsubseteq D$

se lahko prevede na vsebovanje v praznem TBox

- $\models \bar{C} \sqsubseteq \bar{D}$

Kako dobimo  $\bar{C}$  in  $\bar{D}$ ?

- 1) Pretvori  $\Sigma$  v  $\Sigma'$  tako da se vsaka definicija primitivnega koncepta  $A \sqsubseteq C$  spremeni v definicijo koncepta  $A \equiv C \sqcap A^*$ , kjer predstavlja  $A^*$  nov koncept.
  - 2) Spremeni vse definirane koncepte tako, da zamenjaš pojavitve konceptov na desni strani definicije z definicijo v  $\Sigma'$ . Ker je  $\Sigma'$  acikličen je postopek končen.
- Postopek imenujemo **razširjanje** ali **razvitje**.

# Primer: družina (razširjena / )

Woman  $\equiv$  Person  $\sqcap$  Female

Man  $\equiv$  Person  $\sqcap$   $\neg$ (Person  $\sqcap$  Female)

Mother  $\equiv$  (Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild:Person

Father  $\equiv$  (Person  $\sqcap$   $\neg$ (Person  $\sqcap$  Female))  $\sqcap$   $\exists$ hasChild.Person

Parent  $\equiv$  ((Person  $\sqcap$   $\neg$ (Person  $\sqcap$  Female))  $\sqcap$   $\exists$ hasChild.Person)  
 $\sqcup$  ((Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild.Person)

Grandmother  $\equiv$  ((Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild.Person)  
 $\sqcap$   $\exists$ hasChild.(((Person  $\sqcap$   $\neg$ (Person  $\sqcap$  Female))  
 $\sqcap$   $\exists$ hasChild.Person)  
 $\sqcup$  ((Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild.Person))

MotherWithManyChildren  $\equiv$ ((Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild.Person)  $\sqcap$   $>3$  hasChild

MotherWithoutDaughter  $\equiv$  ((Person  $\sqcap$  Female)  $\sqcap$   $\exists$ hasChild.Person)  
 $\sqcap$   $\exists$ hasChild. $\neg$ (Person  $\sqcap$  Female))

Wife  $\equiv$  (Person  $\sqcap$  Female)  
 $\sqcap$   $\exists$ hasHusband.(Person  $\sqcap$   $\neg$ (Person  $\sqcap$  Female))

# Razširitev acikličnih terminologij

**Izrek:** Naj bo  $T$  aciklična terminologija naj bo  $T'$  njena razširitev, potem:

- (i)  $T$  in  $T'$  imata ista imena konceptov in osnovnih simbolov;
- (ii)  $T$  in  $T'$  sta ekvivalentne;
- (iii) obe  $T$  in  $T'$  sta definitabilne (angl. Definitorial).

Razširitev terminologije je lahko eksponentna [Nebel, 1990b].

# Opis sveta

- Opis sveta ali **ABox**
- Opis trenutnega stanja sveta s koncepti in vlogami
- Definicija individualnih objektov
  - Imena in lastnosti objektov
  - Primerki konceptov in vlog
  - Ekstenzija konceptualne sheme  $\equiv$  podatkovna baza
- ABox  $\equiv$  Primerki konceptov in vlog
  - **Konkretni** koncepti
  - **Konkretne** vloge

# Izjave o objektih

MotherWithoutDaughter(MARY)

Father(PETER)

hasChild(MARY,PETER)

hasChild(PETER,HARRY)

hasChild(MARY,PAUL)

# Sklepanje

- Sistem za predstavitev znanja osnovan na DL:
  - Sklepanje z izjavami zapisanimi v bazi.
  - Semantika podatkovne baze izjav DL ima pomen, ki je ekvivalenten **predikatnemu računu**.
- Sistem vsebuje **implicitno znanje**, ki ga lahko dobimo na osnovi sklepanja.
  - Uporaba pravil logičnega sklepanja za izpeljavo novih izjav.



# Sklepanje s koncepti

- Koncepti so definirani na osnovi atomičnih konceptov.
- Med gradnjo podatkovne baze je potrebno preverjati ali imajo na novo dodani koncepti **smisel** v okviru obstoječih.
- Smiselno je, da ima definiran koncept kakšno neprazno **interpretacijo** v podatkovni bazi izjav DL.
- Za takšne koncepte pravimo, da so **rešljivi** oz. **nerešljivi** sicer.

# Sklepanje s koncepti

- **Rešljivost** konceptov je ključnega pomena za sklepanje.
  - Vrsto različnih form sklepanja lahko reduciramo na rešljivost.
  - Preverjanje relacij med koncepti: **splošnost**, **posledica**, itd.
- Problem **vsebovanosti**:
  - Koncept *C* je vsebovan v konceptu *D*, če je vsak model *C* podmnožica interpretacije *D*.
  - Algoritmi za izračun vsebovanosti se uporabljajo tudi za ureditev konceptov TBox v taksonomijo glede na splošnost.
- Relacije med koncepti, ki so zanimive v praksi so tudi **ekvivalenca** in **različnost**.

# Osnovna razmerja

- Rešljivost:  
koncept  $C$  je rešljiv glede na terminologijo  $T$ , če obstaja interpretacija  $T$ , ki je model  $T$  tako da je  $C'$  neprazna. Z drugimi besedami je  $T$  model tudi za  $C$ .
- Vsebovanost:  
koncept  $C$  je vsebovan konceptu  $D$  glede na  $T$ , če je  $C' \subseteq D'$  za vsak model  $I$  terminologije  $T$ , kar zapišemo  $C \sqsubseteq_T D$  ali  $T \models C \sqsubseteq D$ .
- Ekvivalenca:  
koncepta  $C$  in  $D$  sta ekvivalentna glede na  $T$ , če velja  $C' = D'$  za vsak model  $I$  terminologije  $T$ , kar zapišemo  $C \equiv_T D$  ali  $T \models C \equiv D$ .
- Različnost:  
koncepta  $C$  in  $D$  sta različna glede na  $T$ , če velja  $C' \cap D' = \emptyset$  za vsak model  $I$  terminologije  $T$ .

# Primeri

Woman  $\sqsubseteq$  Person

Mother  $\sqsubseteq$  Woman

Mother  $\sqsubseteq$  Parent

Grandmother  $\sqsubseteq$  Mother

Woman  $\sqcap$  Man  $\sqsubseteq$   $\perp$

Father  $\sqcap$  Mother  $\sqsubseteq$   $\perp$

Glej primer družine!

# Redukcija na vsebovanost

Večina DL vsebuje  $\top$  in  $\perp$ .

- $C$  je nerešljiv  $\iff C \sqsubseteq \perp$
- $C$  in  $D$  sta ekvivalentna  $\iff C \sqsubseteq D \wedge D \sqsubseteq C$
- $C$  in  $D$  sta različna  $\iff C \sqcap D \sqsubseteq \emptyset$

# Redukcija na nerešljivost

Če DL vsebuje  $\neg$ .

- C je vsebovan v D  $\iff$  C  $\wedge$   $\neg$ D nima rešitve
- C in D sta ekvivalentna  $\iff$   
C  $\wedge$   $\neg$ D  $\wedge$   $\neg$ C  $\wedge$  D nimata rešitve
- C in D sta različna  $\iff$  C  $\wedge$  D nima rešitve

# Rešljivost

- Vse prej opisane forme sklepanja lahko torej prevedemo na rešljivost.
  - Vsebovanje, ekvivalenca, različnost.
- Algoritem za problem rešljivosti konceptov:
  - Učinkovit algoritem omogoča hitro reševanje vseh oblik sklepanja.
  - Večina rešitev temelji na Tableaux calculi.
- Enostavno tehniko za implementacijo vsebovanosti v TBox na osnovi sintaktične strukture si bomo pogledali kasneje.
- Več o rešljivosti:
  - Še vedno raziskovalni problem
  - Pregled: diploma, magsterij
  - Pregled metode Tableaux za DL je lahko seminar

# Algoritmi za sklepanje

- Pregled
- Tableaux algoritmi
- Strukturni algoritmi
- Kompleksnost algoritmov



# Opisna logika $\mathcal{FL}^-$

Najenostavnejša strukturna opisna logika  $\mathcal{FL}^-$ :

- Sintaksa + Semantika
- Sklepanje
  - Izračunljivost
  - Kompleksnost
- Metode sklepanja
  - Algoritem za preverjanje vsebovanosti konceptov.
  - Uglášenost, kompletnost in kompleksnost algoritma.

# Slovnica $\mathcal{FL}^-$

$C, D \rightarrow A$

|  $C \sqcap D$

|  $\forall R.C$

|  $\exists R$

$A \equiv$  atomični-koncept

$R \equiv$  atomična-vloga

$C, D \equiv$  koncept

koncept ::=  $\langle$ atomični-koncept $\rangle$  |  
 $\langle$ koncept $\rangle \sqcap \langle$ koncept $\rangle$  |  
 $\exists \langle$ atomična-vloga $\rangle$  |  
 $\forall \langle$ atomična-vloga $\rangle . \langle$ koncept $\rangle$

# Strukturna vsebovanost za $\mathcal{FL}^-$ .

Poglejmo si vsebovanost bolj natančno za  $\mathcal{FL}^-$ .

$$C \sqsubseteq D$$

*C je-vsebovan-v D*



Za vsako domeno  $\Delta'$  in za vsako interpretacijo  $\cdot'$  nad  $\Delta'$   
velja:  $C' \subseteq D'$



$$\forall x. C(x) \rightarrow D(x)$$

# Računske lastnosti

- Operacija **vsebovanje** definirana nad koncepti  $\mathcal{FL}$ - ima naslednje računske lastnosti:
  - **odločljiva**
  - **P**
- Oboje lahko preverimo
  - Kako ?

# Strukturalni algoritmi

- **Strukturalni algoritmi** za preverjanje vsebovanosti
  - Algoritem je osnovan na primerjavi strukture izrazov
- **Rekurzivni algoritmi**
  - Osnovna pravila o “ujemanju” izrazov na površju
  - Pravila za izražanje vsebovanosti izraza
  - Preverimo ali se komponente ujemajo

# Normalna oblika

Algoritem ima dve faze:

1. Koncepti so prepisani v normalno obliko
2. Primerjamo strukture konceptov

**Normalna oblika:**

1. Odstanimo vse vgnezdene konjunkcije  
 $A \wedge (B \wedge C) \mapsto A \wedge B \wedge C.$
2. Vse konjunkcije univerzalnih kvantifikatorjev izpostavimo  
 $\forall R:C \wedge \forall R.D \mapsto \forall R.(C \wedge D).$

Normalizirani koncepti so logično ekvivalentni začetnim izrazom. Vsebovanje se ohrani s transformacijo.

# Osnovni algoritem: *SUBS?*[C,D]

Dana sta izraza  $C = C_1 \sqcap \dots \sqcap C_n$  in  $D = D_1 \sqcap \dots \sqcap D_m$ ,  
ki sta v normalni obliki.

*SUBS?*[C,D] vrne TRUE, če in samo če velja za vse  $D_i$ :

- Če je  $D_i$  atomični koncept ali koncept oblike  $\exists R$ ,  
potem obstaja koncept  $C_j$  tako da  $C_i = D_j$
- Če je  $D_i$  koncept oblike  $\forall R.D'$ , potem obstaja  $C_j$   
oblike  $\forall R.C'$  tako da *SUBS?*[ $C',D'$ ].

$$\uparrow \\ O(|C| * |D|)$$

$$\text{Izrazi urejeni} \\ \downarrow \\ O(|C| + |D|)$$

# Enostavni primeri

$Adult \cap Male \subseteq Adult$

$Adult \cap Male \cap Rich \subseteq Adult \cap Male$

$\forall CHILD.(Adult \cap Male) \subseteq \forall CHILD.Adult$

$\forall CHILD.Adult \cap \exists CHILD \subseteq \forall CHILD.Adult$

$\forall CHILD.Adult \not\subseteq \exists CHILD$

$\exists CHILD \not\subseteq \forall CHILD.Adult$



# Tableaux algoritmi

- Uporaba negacije za prevod vsebovanosti na (ne)rešljivost
- $C \sqsubseteq D$  čče  $C \sqcap \neg D$  ni rešljivo
- Definiran bo algoritem za  $\mathcal{ALCN}$

# Primer 1

$$\exists R.A \wedge \exists R.B \sqsubseteq \exists R.(A \wedge B) ?$$

$C = \exists R.A \wedge \exists R.B \wedge \neg(\exists R.(A \wedge B))$  ni rešljiv?



$$C_0 = \exists R.A \wedge \exists R.B \wedge \forall R.(\neg A \sqcup \neg B)$$

Hočemo konstruirat interpretacijo  $I$ , tako da  $C_0' \neq \emptyset$

# Primer 1

$b \in C_0'$

$b \in (\exists R.A)'$     $b \in (\exists R.B)'$     $b \in (\forall R.(\neg A \sqcup \neg B))'$

$(b,c) \in R'$     $c \in A'$

$(b,d) \in R'$     $d \in B'$

$c \in (\neg A \sqcup \neg B)'$    pomeni    $c \in (\neg A)'$  ali  $c \in (\neg B)'$

$d \in (\neg A \sqcup \neg B)'$    pomeni    $d \in (\neg A)'$  ali  $d \in (\neg B)'$

# Literatura

- **Osnovna literatura:**
  - F. Baader, W. Nutt. Basic Description Logics. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002, pages 47-100.
- **Dodatna literatura:**
  - F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. *Studia Logica*, 69:5-40, 2001
  - Donini, F., Lenzerini, M., Nardi, D., Schaerf, A., Reasoning in Description Logics, in: *Principles of Knowledge Representation and Reasoning*, edited by G. Brewka; *Studies in Logic, Language and Information*, CLSI Publications, pp 193-238, 1996.
  - D. Nardi, R. J. Brachman. An Introduction to Description Logics. In the Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press, 2002, pages 5-44.