Uvod v opisno logiko

I.Savnik, FAMNIT

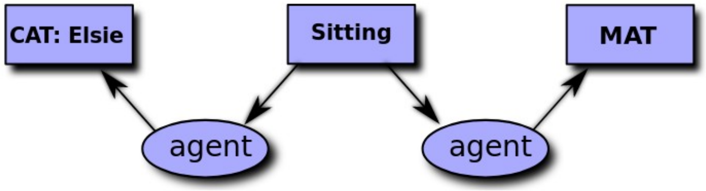
Pregled izvorov

- Okvirji
- Konceptualni grafi
- KL-One
- Logika

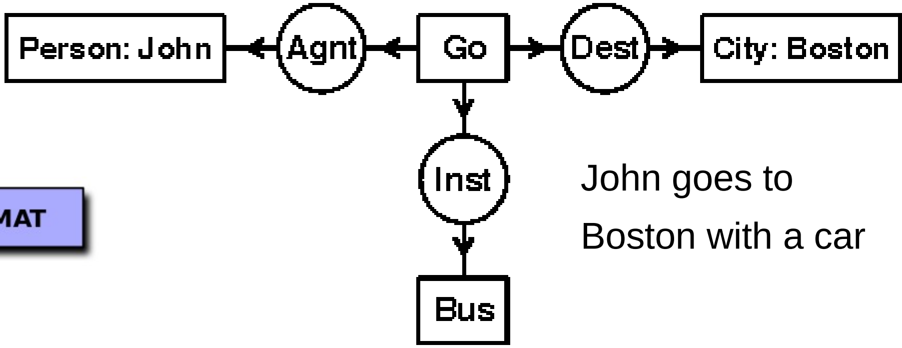
Pregled: okvirji

- Marvin Minsky in his 1974 article "A Framework for Representing Knowledge."
  - A frame is AI structure used to divide knowledge into substructures by representing "stereotyped situations"
  - Izhajajo iz semantičnih mrež
- Okvirji (angl. frames)
  - Vsebuje podatke o stanju, kako se uporablja, kaj narediti v naslednjem koraku in kaj narediti, če pogoji niso izpolnjeni
  - Objekti, koncepti, strukture, razredi, pod-razredi, ...
  - Dovoljujejo izjeme v konkretnih instancah
- Predal (angl. slots)
  - Konkretne vrednosti, procedure, omejitve,privzete vrednosti, druge okvirje

Konceptualni grafi



Elsie the cat is sitting on a mat



John goes to Boston with a car

- John F. Sowa
  - (Sowa 1976) used them to represent the conceptual schem of database systems
  - (Sowa 1984) applied them to a wide range of topics in artificial intelligence, computer science, and cognitive science
- A graphical interface for first-order logic
  - A formula in first-order logic is represented by a labeled graph
  - [Cat Elsie] [Sitting *x] [Mat *y] (agent ?x Elsie) (location ?x ?y)
- Diagrammatic calculus of logics
  - Existential graphs of C.S. Peirce
- Graph-based knowledge representation and reasoning model

Pregled: KL-One

- Knowledge representation system in the tradition of semantic networks and frames
  - Brachman, Schmolze, 1985
  - Frames in KL-ONE are called concepts
    - These form hierarchies using subsume-relations
  - Slot-concept is called roles and the values of the roles are role-fillers
  - First to propose deductive classifier
    - an automated reasoning engine that can validate a frame ontology and
    - deduce new information about the ontology based on the initial information provided by a domain expert

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

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 (logičnih zvez)
- 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

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.
- Druga možnost:
  - Uporaba zelo učinkovitih algoritmov za sklepanje (kompleksno

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 konstruktih, 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 predkiati, razredi)

Primer: Student, Married

{x | Student(x)}
{x | Married(x)}

Vloge: definirajo lastnosti (binarni predikati)

Primer: FRIEND, LOVES

{<x,y> | FRIEND(x,y)}
{<x,y> | LOVES(x,y)}

Primer: opis koncepta

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

Primer:

$\text{Student} \sqcap \exists \text{FRIEND}.\text{Married}$

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

Osnovni opisni jezik $\mathcal{AL}$

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

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

Kvantifikatorji

- Gradnik $\forall$R.C opisuje koncepte z vlogo R, ki ima zalogo vrednosti omejeno na koncept C.
  - $\{x \mid \forall y{:}R(x,y) \longrightarrow C(y)\}$
  - $\forall$CHILD.Doctor ≡ $\{x \mid \forall y{:}CHILD(x,y) \longrightarrow Doctor(y)\}$
  - 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	atomični koncept
Person $\sqcap$ Female	osebe, ki so ženske
Person $\sqcap$ $\neg$Female	osebe, ki niso ženske
hasChild	atomična vloga
Person $\sqcap$ $\exists$hasChild.$\top$	oseba, ki ima enega otroka
Person $\sqcap$ $\forall$hasChild.Female	oseba, ki ima same hkčere
Person $\sqcap$ $\forall$hasChild.$\bot$	oseba, ki nima otrok

Intuitivna semantika

- Intuitivno:
  - Koncepti predstavljajo razrede, možice primerkov.
  - Vloge predstavljajo relacije med pari konceptov.
  - Atomični koncepti so predstavljeni z imeni primitivnih (osnovnih) konceptov.
- Pomen sestavljenih konceptov:
  - Konstrukt ⊓ predstavlja sestavljene koncepte, ki imajo pomen preseka obeh konceptov:

    $$\text{Adult} \sqcap \text{Male} \sqcap \text{Person}$$

  - Konstrukt ⊔ predstavlja sestavljene koncepte, ki imajo pomen unije obeh konceptov.

Formalna semantika $\mathcal{AL}$

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

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

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

Formalna semantika $\mathcal{AL}$

Interpretacijska funkcija *I* je razširjena na opise konceptov z naslednjimi induktivnimi definicijami:

$$\top^I = \Delta$$

$$\bot^I = \varnothing$$

$$(\neg A)^I = \Delta^I - A^I$$

$$(C \sqcap D)^I = C^I \cap D^I$$

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

$$(\exists R.\top)^I = \{x \in \Delta \mid \exists y.(x,y) \in R^I\}$$

C^I je množica individualnih objektov, ki so primerki koncepta C.

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

Ekvivalenca

$C \equiv D \iff C^I = D^I$ za vse interpretacije I

Primer:

$\forall hasChild.Female \sqcap \forall hasChild.Student$

$\equiv$

$\forall hasChild.(Female \sqcap Student)$

Družina jezikov $\mathcal{AL}$

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

- Unija
- Polna eksistenčna kvantifikacija
- Vrednostne omejitve
- Negacija

Unija

Jezik $\mathcal{ALU}$!

C⊔D - unija konceptov

Interpretacija: $(C \sqcup D)^I = C^I \cup D^I$

Polna eksistenčna kvantifikacija

Jezik $\mathcal{ALE}$!

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

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

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

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

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

Negacija poljubnega koncepta

Jezik $\mathcal{ALC}$!

$(\neg C)^I = \Delta^I - C^I$

$\mathcal{AL}$*: pregled

A	$A^I \subseteq \Delta^I$	primitivni koncept
R	$R^I \subseteq \Delta^I x \Delta^I$	primitivna vloga
$\top$	Δ^I	streha
$\bot$	$\emptyset$	tla
$\neg C$	$\Delta^I \setminus C^I$	komplement
$C \sqcap D$	$C^I \sqcap D^I$	konjunkcija
$C \sqcup D$	$C^I \sqcup D^I$	disjunkcija
$\forall R.C$	$\{x \mid \forall y. R^I(x,y) \longrightarrow C^I(y)\}$	univerzalna kvant.
$\exists R.C$	$\{x \mid \exists y. R^I(x,y) \wedge C^I(y)\}$	eksistenčna kvant.

Primer

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

$\text{Oseba} \sqcap$
$(\leq 1\ \text{imaOtroka} \sqcup$
$(>3\ \text{imaOtroka} \sqcap \exists \text{imaOtroka}.\text{Ženska}))$

$\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$;

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 $\mathcal{AL}$ jezike lahko izrazimo z $\mathcal{UEN}$.

Terminologije

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

Terminološki aksiomi

Naj bodo C,D *koncepti* in R, S *vloge*.
Ločimo med naslednjimi dvemi vrstami aksiomov:

Vsebovanost:

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

Enakost:

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

.

Terminološki aksiomi

Semantika aksiomov je definirana na običajen način:

- $C \sqsubseteq D \iff C^I \subseteq D^I$
- $C \equiv D \iff C^I = D^I$

Interpretacija I je rešitev aksioma ali množice aksiomov $\Longrightarrow$ 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:

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

Simbolična imena lahko uporabimo kot okrajšavo v drugih opisih.

Primer: Recimo, da imamo definiran tudi koncept Father.

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

Pomen terminologij

- Naj bo T terminologija.
- Atomične koncepte iz T delimo na:
  - Imenske 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.

Primer: družina

$$\mathrm{Woman} \equiv \mathrm{Person} \sqcap \mathrm{Female}$$
$$\mathrm{Man} \equiv \mathrm{Person} \sqcap \neg\mathrm{Woman}$$
$$\mathrm{Mother} \equiv \mathrm{Woman} \sqcap \exists\mathrm{hasChild.Person}$$
$$\mathrm{Father} \equiv \mathrm{Man} \sqcap \exists\mathrm{hasChild.Person}$$
$$\mathrm{Parent} \equiv \mathrm{Father} \sqcup \mathrm{Mother}$$
$$\mathrm{Grandmother} \equiv \mathrm{Mother} \sqcap \exists\mathrm{hasChild.Parent}$$
$$\mathrm{MotherWithManyChildren} \equiv \mathrm{Mother} \sqcap {>}3\mathrm{hasChild}$$
$$\mathrm{MotherWithoutDaughter} \equiv \mathrm{Mother} \sqcap \forall\mathrm{hasChild.}\neg\mathrm{Woman}$$
$$\mathrm{Wife} \equiv \mathrm{Woman} \sqcap \exists\mathrm{hasHusband.Man}$$

Pomen terminologij

- Naj bo T terminologija.
- Atomične koncepte iz T delimo na:
  - Imenske 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.

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

Interpretacija

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

Human’ ≡ Animal ⊓ ∀hasParent.Human’

Interpretacija

- T je definitabilna, če imajo vse interpretacije osnovnih simbolov natančno eno razširitev, ki je model T.
  - Če poznamo interpretacijo osnovnih simbolov in če je T dobro definirana, potem je pomen imenskih simbolov natančno določen.
  - Angl. definitorial
- Definitabilnost terminologije je povezana s strukturo aksiomov.
- Če terminologija nima ciklov je definitabilna.

Enostaven acikličen TBox

- Koncept A direktno uporablja koncept B v TBox Σ, čče definicija A omenja B
- Koncept A uporablja koncept B, čče obstaja veriga direktnih uporab konceptov vse do B
- TBox je acikličen, če nobeden koncept ne uporablja samega sebe za definicijo

Enostaven acikličen TBox

Imamo aciklično terminologijo T.
Kako dobimo razširjeno terminologijo?

Spremenimo vse definirane koncepte tako, da pojavitve definiranih konceptov na desni strani definicije zamenjamo z definicijo.

Ker je T acikličen je postopek končen.

Postopek imenujemo razširjanje ali razvitje.

Primer: družina (razširjena I)

$$
\begin{aligned}
\text{Woman} &\equiv \text{Person} \sqcap \text{Female}\\
\text{Man} &\equiv \text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})\\
\text{Mother} &\equiv (\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild:Person}\\
\text{Father} &\equiv (\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists\text{hasChild.Person}\\
\text{Parent} &\equiv ((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists\text{hasChild.Person})\\
&\quad \sqcup ((\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild.Person})\\
\text{Grandmother} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild.Person})\\
&\quad \sqcap \exists\text{hasChild.}(((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}))\\
&\quad \sqcap \exists\text{hasChild.Person})\\
&\quad \sqcup ((\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild.Person}))\\
\text{MotherWithManyChildren} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild.Person})\\
&\quad \sqcap >3 \text{ hasChild}\\
\text{MotherWithoutDaughter} &\equiv ((\text{Person} \sqcap \text{Female}) \sqcap \exists\text{hasChild.Person})\\
&\quad \sqcap \exists\text{hasChild.}(\neg(\text{Person} \sqcap \text{Female}))\\
\text{Wife} &\equiv (\text{Person} \sqcap \text{Female})\\
&\quad \sqcap \exists\text{hasHusband.}(\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}))
\end{aligned}
$$

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].

Ciklične terminologije

- Definitabilne teminologije imajo opisno semantiko
- Ciklične terminologije imajo semantiko fiksne točke
  - V nekaterih primerih so smislene
  - V teh primerih ujamemo pomen z najmanjšo ali največjo fiksno točko
- Največja fiksna točka lahko opiše ciklične terminologije
  - Fiksna točka posledica končne količine podatkov

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 koncetualne sheme ≡ podatkovna baza
- ABox ≡ 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.
  - Glej prejšnji primer: MARY je stara mama.
- Ogledali si bomo sklepanje:
  - Koncepti, TBox, ABox ter TBox in ABox

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^I neprazna. Z drugimi besedami je *T* model tudi za C.
- Vsebovanost:
  koncept C je vsebovan konceptu D glede na *T*, če je $C^I \subseteq D^I$ 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 $C^I = D^I$ 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 $C^I \cap D^I = \varnothing$ za vsak model *I* terminologije *T*.

Primer: družina

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

Primer: družina

$$\mathrm{Woman} \sqsubseteq \mathrm{Person}$$
$$\mathrm{Mother} \sqsubseteq \mathrm{Woman}$$
$$\mathrm{Mother} \sqsubseteq \mathrm{Parent}$$
$$\mathrm{Grandmother} \sqsubseteq \mathrm{Mother}$$
$$\mathrm{Woman} \sqcap \mathrm{Man} \sqsubseteq \bot$$
$$\mathrm{Father} \sqcap \mathrm{Mother} \sqsubseteq \bot$$

Redukcija na vsebovanost

Večina DL vsebuje $\sqcap$ in $\bot$.

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

Redukcija na nerešljivost

Če DL vsebuje $\neg$.

- C je vsebovan v D $\Longleftrightarrow$ $C \sqcap \neg D$ nima rešitve
- C in D sta ekvivalentna $\Longleftrightarrow$
  $C \sqcap \neg D \land \neg C \sqcap D$ nimata rešitve
- C in D sta različna $\Longleftrightarrow$ $C \sqcap D$ 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

Sklepanje o ABox

- ABox vsebuje stavke C(a) in R(a, b)
- Konsistentnost ABox je potrebno preveriti
  - Mother(MARY), Father(MARY) ?
- ABox A je konsistenten s TBox T, če obstaja model A in T skupaj
  - Definiramo razširitev ABox A glede na TBox T
    - V A dodamo izjave, ki sledijo iz A glede na T
  - ABox A je konsistenten, čče je razširitev A' glede na T konsistentna

Sklepanje o ABox

- Druge vrste sklepanja
  - Preverjanje instanc: ali lahko dano dejstvo izpeljemo iz A glede na T?
    - $A \models C(a)$ čče $A \cup \{\neg C(a)\}$ ni konsistentna
  - Redukcija sklepanja o konceptih na preverjanje konsistentnosti ABox
    - C je rešljiv čče $\{C(a)\}$ je konsistentna
  - Poizvedovalni problem:
    - Dan je ABox A in koncept C, poišči vsa dejstva $A \models C(a)$
  - Realizacijski problem:
    - Dano je dejstvo a. Poišči najbolj specifične koncepte, ki opisujejo a.

Algoritmi za sklepanje

- Pregled
- Strukturni algoritmi
- Tableaux algoritmi
- Kompleksnost algoritmov

Opisna logika $\mathcal{FL}^-$

- Strukturna opisna logika $\mathcal{FL}^-$:
  - Najenostavnejša opisna logika
    - Sintaksa + Semantika
  - Sklepanje
    - Izračunljivost
    - Kompleksnost
  - Metode sklepanja
    - Algoritem za preverjanje vsebovanosti konceptov.
    - Uglašenost, kompletnost in kompleksnost algoritma.

Slovnica $\mathcal{FL}^-$

$$
\begin{aligned}
C,D \rightarrow\ & A \\
\mid\ & C \sqcap D \\
\mid\ & \forall R.C \\
\mid\ & \exists R
\end{aligned}
$$

A ≡ atomični-koncept
R ≡ atomična-vloga
C,D ≡ koncept

koncept ::= <atomični-koncept> |
<koncept> ⊓ <koncept> |
∃<atomična-vloga> |
∀<atomična-vloga>.<koncept>

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

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

C ⊑ D

C *je-vsebovan-v* D

⇔

Za vsako domeno Δ^I in za vsako interpretacijo $\cdot^I$ nad Δ^I velja: $C^I \subseteq D^I$

⇔

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

Računske lastnosti

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

Strukturni algoritem

- Strukturni algoritem za preverjanje vsebovanosti
  - Algoritem je osnovan na primerjavi strukture izrazov
- Rekurzivni algoritem
  - 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
   $A \sqcap (B \sqcap C) \longmapsto A \sqcap B \sqcap C.$
2. Vse konjunkcije univerzalnih kvantifikatorjev izpostavimo
   $\forall R{:}C \sqcap \forall R.D \longmapsto \forall R.(C \sqcap D).$

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

Osnovni algoritem: $\mathcal{SUBS?}$[C,D]

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

$\mathcal{SUBS?}$[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 $C_j = D_i$
- Če je D_i koncept oblike $\forall R.D'$, potem obstaja C_j oblike $\forall R.C'$ tako da $\mathcal{SUBS?}$[C',D'].

⇑
$\mathcal{O}(|C|*|D|)$

Izrazi urejeni
⇓
$\mathcal{O}(|C|+|D|)$

Enostavni primeri

$\text{Adult} \sqcap \text{Male} \sqsubseteq \text{Adult}$

$\text{Adult} \sqcap \text{Male} \sqcap \text{Rich} \sqsubseteq \text{Adult} \sqcap \text{Male}$

$\forall\text{CHILD}.(\text{Adult}\sqcap\text{Male}) \sqsubseteq \forall\text{CHILD}.\text{Adult}$

$\forall\text{CHILD}.\text{Adult} \sqcap \exists\text{CHILD} \sqsubseteq \forall\text{CHILD}.\text{Adult}$

$\forall\text{CHILD}.\text{Adult} \not\sqsubseteq \exists\text{CHILD}$

$\exists\text{CHILD} \not\sqsubseteq \forall\text{CHILD}.\text{Adult}$

Tableaux algoritmi

- Uporaba negacije za prevod vsebovanosti na (ne)rešljivost
  - C $\sqsubseteq$ D čče C $\sqcap$ $\neg$D ni rešljivo
  - Definiran bo algoritem za $\mathcal{ALCN}$
- Sistemi:
  - Kris [Baader and Hollunder, 1991]
  - Crack [Bresciani et al., 1995]
  - Fact [Horrocks, 1998]
  - Dlp [Patel-Schneider, 1999]
  - Race [Haarslev and Moeller, 2001]

Primer 1

Naj bosta A in B imana konceptov in R ime vloge

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

$$C = \exists R.A \sqcap \exists R.B \sqcap \neg(\exists R.(A \sqcap B)) \text{ ni rešljiv?}$$

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

- Negacije smo potisnili proti simbolom z uporabo De Morganovih pravil (negacijska normalna oblika).
- Hočemo konstruirat interpretacijo I, tako da $C_0^I \neq 0$
- Iščemo proti-primer izjavi C_0 ni rešljiv, ki je hkrati rešitev poizvedbe !

Primer 1

$b \in C_0^I$

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

$(b,c) \in R^I$ $c \in A^I$

$(b,d) \in R^I$ $d \in B^I$

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

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

- Prva alternativa:
  - $c \in A^I$ in $c \in (\neg A)^I \Rightarrow$ protislovje
  - $d \in B^I$ in $d \in (\neg B)^I \Rightarrow$ protislovje

Primer 1

- Druga alternativa:
  - $c \in A^I$ in $c \in (\neg B)^I \Rightarrow$ *OK*
  - $d \in B^I$ in $d \in (\neg A)^I \Rightarrow$ *OK*
- C_0 je rešljiv !
  - Vzamemo: $c \in (\neg B)^I$ in $d \in (\neg A)^I$ in dobimo model, ki je proti-primer
    - $\exists R.A \sqcap \exists R.B \sqsubseteq \exists R.(A \sqcap B)$
    - Model: $\Delta^I = \{b,c,d\}$; $R^I = \{(b,c), (b,d)\}$;
      $A^I = \{c\}$; $B^I = \{d\}$
  - Rezultat:
    - $b \in C_0^I$, $b \in (\exists R.A \sqcap \exists R.B)^I$, $b \notin (\exists R.(A \sqcap B))^I$

Primer 2

- Dodamo dodatno omejitev v prejšnji primer
  - $\exists R.A \sqcap \exists R.B \sqcap \leq 1\ R \sqsubseteq \exists R.(A \sqcap B)$
  - Osnovna ideja:
    - preprečimo alternativo med c in d
    - $c = d \Rightarrow$ ne pride do protislovja
  - Daljši opis:
    - Algoritem se izvaja enako kot prej z dodatno omejitvijo $b \in (\leq 1\ R)^I$
    - $c=d \in (A \sqcap B)$ in hkrati $c=d \in (\neg A \sqcup \neg B)$ kar vedno vodi do protislovja
- C_0 ni rešljiv => $\exists R.A \sqcap \exists R.B \sqcap \leq 1\ R \sqsubseteq \exists R.(A \sqcap B)$

Pravila Tableaux algoritma

- Za vsako eksistenčno omejitev algoritem uvede nove vrednosti za vlogo
  - Vrednosti morajo zadoščati zapisanim omejitvam zaloge vrednosti vloge
- Algoritem obravnava vrednostne omejitve skupaj z obstoječimi omejitvami vloge
  - Doda nove omejitve
- Pri disjunktivnih omejitvah algoritem poskusi obe alternativi
  - V primeru kontradikcije se algoritem rekurzivno vrne v drugo alternativo
- Če je številska omejitev vloge kršena
  - Algoritem poišče druge vrednosti za vlogo

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.