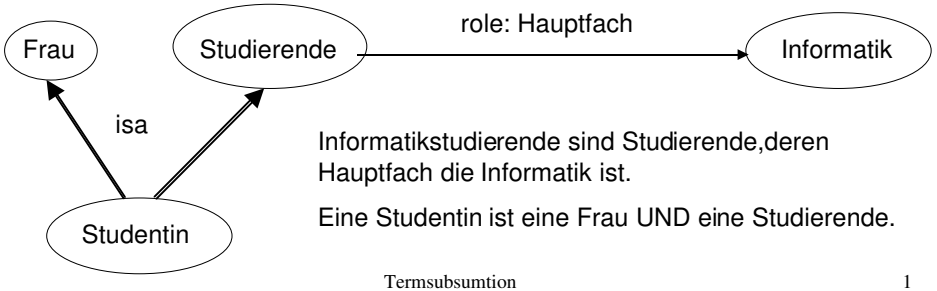
Termsubsumtionsformalismen (KL-ONE)

In epistemischen Primitiven Begriffe definieren. Teile der Signatur und der Interpretation sind vorgegeben. Einige Schlussfolgerungen sind vorbereitet.

≠ logische Ebene!

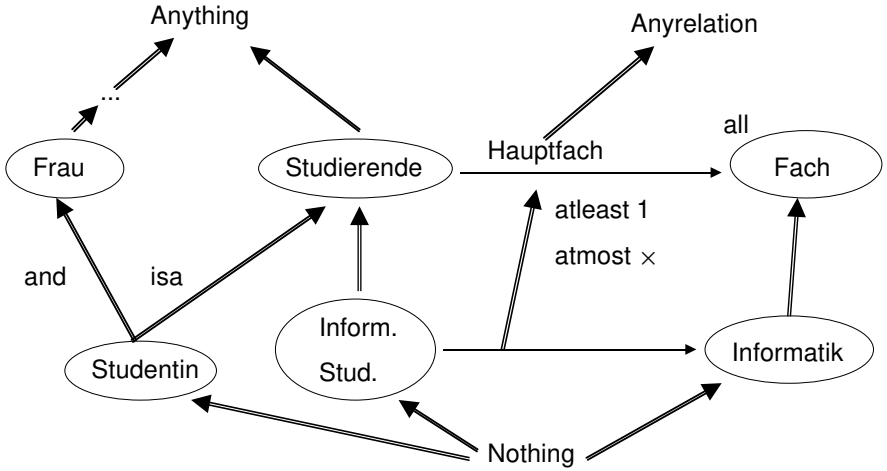


Informatikstudierende sind Studierende,deren Hauptfach die Informatik ist.

Eine Studentin ist eine Frau UND eine Studierende.

T-Box

für terminologisches Schließen



Studentin .=. (and Frau Studierende)

Informatikstudierende .=. (and Studierende (all Hauptfach Informatik))

Syntax von Termsubsumtionsformalismen

<terminology> ::= {<term-introduction> | <restriction> }*

<term-introduction> ::= <concept-introduction> | <role-introduction>

<concept-introduction> ::= <atomic-concept> .=. <concept> |
<atomic-concept> ≤ <concept> |
<atomic-concept> ≤ **ANYTHING**

<role-introduction> ::= <atomic-role> .=. <role> |
<atomic-role> ≤ <role> |
<atomic-role> ≤ **ANYRELATION**

<concept> ::= <atomic-concept> |
(and <concept>+) |
(all <role> <concept>) |
(atleast <number> <role>) |
(atmost <number> <role>)

<role> ::= <atomic-role> |
(androle <role>+)

<restriction> ::= (disjoint <atomic-concept> <atomic-concept>)

<number> ::= <non-negative-integer>

<atomic-role> ::= <identifier>

<atomic-concept> ::= <identifier>

Wohlgeformte, aus epistemischen Primitiven zusammengesetzte Ausdrücke

Semantik epistemischer Primitive und zusammengesetzter Ausdrücke

Knoten: Entscheidung über Begriffszugehörigkeit über notwendige Eigenschaften
d.h. ∀x x∈Begriff | ....

ext

Objekte: a1, a2, a3, ... **D**

Relationen (a1,b1),(a1,b2),(a1,b3),(a2,b1),... **D** x **D**

ext

für alle a ≤ t sei ext(t) ⊇ ext(a)

für alle a.=. t sei ext(t) = ext(a)

ext(**ANYTHING**) = **D**

ext(**ANYRELATION**) = **D** x **D**

ext((and c1 ... cn)) = ∩ ext(ci) wobei i von 1 bis n

ext((all r c)) = {x ∈ **D** | ∀ y : (x,y) ∈ ext(r) --> y ∈ ext(c) }

ext((atleast n r)) = { x ∈ **D** | card ({y ∈ **D** | (x,y) ∈ ext(r)}) ≥ n }

ext((atmost n r)) = { x ∈ **D** | card({y ∈ **D** | (x,y) ∈ ext(r)}) ≤ n }

ext((androle r1 ... rn)) = ∩ ext(ri) wobei i von 1 bis n

für alle (disjoint c1 c2) sei ext(c1) ∩ ext(c2) = {}

Klassifikation

- Gegeben eine TBox und eine neue Begriffsdefinition,
- finde den richtigen Platz für den neuen Begriff.

Der richtige Platz ist:

- der neue Begriff hängt unter seinem speziellsten Oberbegriff,
- der neue Begriff subsumiert alle unter ihm hängenden Begriffe.

Subsumtion

- Logische Struktur (D, ext)
- t subsumiert t´, geschrieben t >> t´, gdw. $\forall$ (D, ext) von T: ext (t) $\supseteq$ ext (t´)
- transitiv: t >> t´, t´>> t´´ dann t >> t´´
- reflexiv: t >> t
- t äquivalent t´gdw. $\forall$ (D, ext) von T: ext (t) = ext (t´)
- t inkohärent in T gdw. $\forall$ (D, ext) von T: ext (t) = { } (unerfüllbar, z.B. (and (atleast 1 r) (atmost 0 r))

T

- Mit and und NOTHING ist T ein Halbverband, d.h.:
- Infimum (größter gemeinsamer Unterbegriff) gibt es immer!
- Die partielle Ordnung ist die Subsumtion >>.
- T behandelt alle Modelle der Begriffsdefinitionen, NICHT bestimmte Mengen von Grundbeispielen!

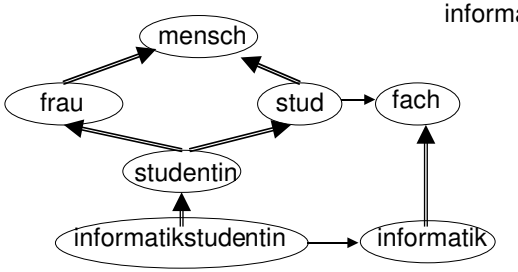
Klassifikationsalgorithmus

- Definitionen glätten, d.h. die Definitionen der Begriffe, die in einer Definition vorkommen, aufschreiben.
- bottom-up: von NOTHING ausgehend, bei jedem Begriff feststellen, ob der neue Begriff von dem Begriff subsumiert wird.
- Der Klassifikationsalgorithmus beruht also auf der Subsumtionsentscheidung.

Beispiel

frau >> studentin?
ja, weil
studentin.=. (and s1 s2)
und
s1= frau

frau ≤ mensch
stud.=.(and mensch
(all studiert fach))
studentin.=. (and frau
(and mensch
(all studiert fach)))
informatikstudentin.=.
(and (and frau
(and mensch
(all studiert fach))
(all studiert informatik))



Subsumtion

subsumes(t,u) ?

t u	androle $t_1...t_n$ and $t_1...t_n$	primitiv	all r_t c_t	atleast n_t r_t	almost n_t r_t
NOTHING	wahr	wahr	wahr	wahr	wahr
primitiv	$\forall_i$ $1\le i\le n$: subsumes (t_i, u)	t = u	falsch	falsch	falsch
all r_u c_u	$\forall_i$ $1\le i\le n$: subsumes (t_i, u)	falsch	subsumes (c_t,c_u) $\wedge$ subsumes (r_u, r_t)	falsch	falsch
atmost 0 r_u	$\forall_i$ $1\le i\le n$: subsumes (t_i, u)	falsch	subsumes (r_u, r_t)	falsch	subsumes (r_u, r_t) $\wedge$ $0 \le n_t$
and $u_1...u_n$ androle $u_1...u_n$	$\forall_i$ $1\le i\le n$: subsumes (t_i, u)	$\exists i, 1\le i\le n$: $t = u_i$	$\exists i$: $1\le i\le n$: subsumes (t, u_i)	$\exists i$: $1\le i\le n$: subsumes (t, u_i)	$\exists i$: $1\le i\le n$: subsumes (t, u_i)
atleast n_u r_u	$\forall_i$ $1\le i\le n$: subsumes (t_i,u)	falsch	falsch	subsumes (r_t, r_u) $\wedge$ $n_t \le n_u$	falsch
atmost n_u r_u	$\forall_i$ $1\le i\le n$: subsumes (t_i, u)	falsch	subsumes (r_u, r_t) bei n_u=0 sonst falsch	falsch	subsumes (r_u, r_t) $\wedge$ $n_u \le n_t$

Subsumtionsalgorithmus

korrekt: wenn wahr für t >> u ausgegeben wird,
gilt ext(t) $\supseteq$ ext(u).

vollständig: wenn ext(t) $\supseteq$ ext(u) gilt,
wird wahr für t >> u ausgegeben.

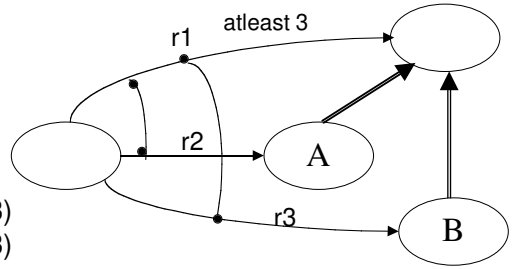
Subsumtion ist entscheidbar, wenn Zyklen ausgeschlossen sind.

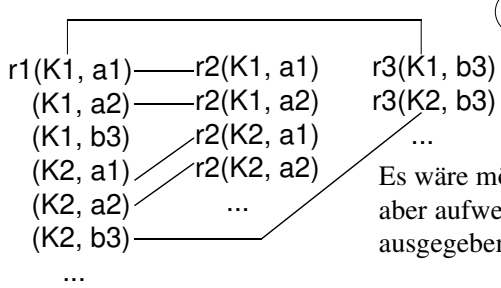
Subsumtionsentscheidung ist polynomiell in der Länge der Definitionen t und u, falls die Rollen primitiv bleiben.

Unvollständigkeit KL-One

t: (atleast 3 r_1)
u: (and (all (androle r1 r2) A)
(all (androle r1 r3) B)
(disjoint A B)
(atleast 2 (androle r1 r2))
(atleast 1 (androle r1 r3))

Schon androle macht die Subsumtion unvollständig! disjoint wird von subsumes nicht berücksichtigt.





Es wäre möglich, disjoint zu berücksichtigen, aber aufwendig! Nicht immer, wenn wahr ausgegeben werden müßte, wird es ausgegeben.

Prinzip

Brachman, Levesque (87):

Handhabbarkeit vs. Ausdrucksfähigkeit eines Wissensrepräsentationsformalismus´.

Man lebt mit polynomieller unvollständiger Subsumtion.

T-Box

Das war die T-Box für

- terminologisches Schließen
- über allen Modellen ohne Grundbeispiele.

Jetzt kommt die A-Box für Assertionen, Grundformeln.

A-Box

Syntax:

<world-description>::=(<obj descr> | <rel descr>)

<obj descr>::=(<atomic concept><obj>)

<rel descr>::=(<atomic role><obj><obj>)

| (<atomic role><obj> (atleast<nr.>))

| (<atomic role><obj> (atmost<nr.>))

Semantik der A-Box

		D	Objekte
Funktion	int	2^D	Begriffe
		2^{DxD}	Rollen

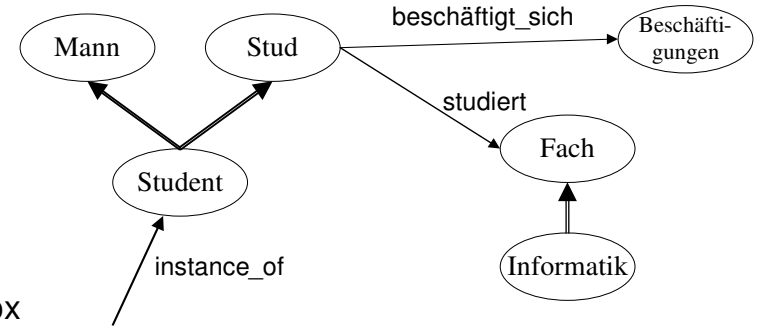
Interpretation (D,int) erfüllt δ

δ = (c o) ist erfüllt gwd.	$int(o) \in int(c)$
(r o p)	$(int(o), int(p)) \in int(r)$
(r o (atleast n))	$\mid \{(int(o),x) \in int(r)\} \mid \geq n$
(r o (atmost n))	$\mid \{(int(o),x) \in int(r)\} \mid \leq n$

Wenn eine Interpretation int alle $\delta_i \in$ A-Box erfüllt, ist sie ein Modell der A-Box. Wenn δ_j in allen Modellen der A-Box erfüllt ist, gilt: A-Box $\mid = \delta_j$

Beispiel

T-Box



A-Box

(Stud Hans)
(beschäftigt_sich Hans (atmost 1))
(beschäftigt_sich Hans Informatik)
(Mann Hans)

Beziehung zwischen T-Box und A-Box

Was studiert Hans?

Was ist der speziellste Begriff für Hans?

(and Stud Mann)

Der allerspeziellste Begriff ist konstruierbar.

Vermutlich gibt es ihn nicht in T-Box.

HansBegriff .=. (and Stud
Mann
(all beschäftigt_sich Informatik)
(atmost 1 beschäftigt_sich))

Klassifikation (HansBegriff)

subsumes(Student, HansBegriff)

Hybride Inferenzen

- Inferenzen in der T-Box
  Vererbung
- Inferenzen zwischen T-Box und A-Box:
  a) Ist ein Objekt ein Grundbeispiel für einen Begriff?
  covers (c, o)?
  b) Welcher Begriff ist der speziellste Begriff, für den ein Objekt ein Grundbeispiel ist?
  realize

covers (c,o)

c ist ein Begriff (T-Box),
o ist ein Objekt beschrieben durch $\delta_1 \ldots \delta_n$ (A-Box)
covers(c, o)
$\delta_i \in$ A-Box $\quad\quad$ δ_i:(<concept> o)
x = (and $\delta_1 \ldots \delta_n$)

subsumes(c, x)

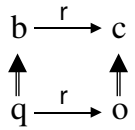
- liefert wahr, dann gilt covers (c,o)
- liefert falsch, dann Fallunterscheidungen

Fall 1

Wenn c eine Beschreibung enthält, die in x, der Beschreibung von o fehlt, kann dennoch covers (c,o) gelten. Nennen wir die fehlenden Beschreibungen in x den Begriff b.

Wenn (r q o) ist in A-Box und
b.=. (all r c) ist in T-Box

dann covers(b, q) ist wahr und damit covers (c,o)
sonst falsch



Beispiel

T-Box:

Human ≤ Anything

Man ≤ Human

Woman ≤ Human

Set ≤ Anything

member ≤ Anyrelation

Team .=. (and Set (all member Human)
(atleast 2 member))

leader ≤ member

Modern-Team .=. (and Team
(atmost 4 member)
(atleast 1 leader)
(all leader Woman))

A-Box:

(Modern-Team team-a)

(Man dick)

(Human mary)

(member team-a dick)

(member team-a harry)

(member team-a mary)

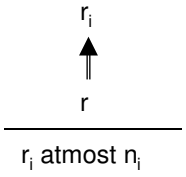
(leader team-a mary)

(member team-a (atmost 3))

covers(Woman mary)?
r: leader
Wenn team-a ein Modern-Team ist, ist mary eine Frau.

Fall 2

Wenn δ_j = (r o (atmost n)) fehlt, aber
δ_i = (r_i o (atmost n_i)) ist in A-Box
und r_i >> r ist in T-Box,



dann sammle alle (r_i o (atmost n_i)) in der A-Box,
füge sie x hinzu,
subsumes(c, x´)
falls es wegen atmost-Verletzung falsch
covers (c,o) falsch
sonst die nächsten Fälle prüfen!

Beispiel

o: hans

x: (Stud hans)
(beschäftigt_sich hans (atmost 1))
(beschäftigt_sich hans informatik)
(Mann hans)

c:

Fall 3

Wenn (r o (atleast n)) fehlt, aber
(r_i o (atleast n_i)) in A-Box und
r_i << r in T-Box
dann zähle Werte von r,
sammle alle (r_i o (atleast n_i)) ,
füge sie x hinzu.
subsumes(c, x´)
falls falsch wegen atleast ist covers(c,o) falsch
sonst weiter!

Fall 4

Wenn (r o oz_i) in A-Box und (all r c_z) T-Box
nicht klappte,
dann teste, ob alle oz_i genannt sind
Wenn ja, für alle oz_i covers(c_z, oz_i) prüfen,
sonst liefere falsch

Realisierungalgorithmus
(realizer)

1.) Propagierung
2.) Abstraktion
3.) Klassifikation

Propagierung

δ = (r o p)
- alle (c_i o) sammeln,
- deren Wertebereichsbeschränkungen für r und r_i >> r sammeln,
- alles als Beschreibung an p reichen

δ = (c o) und
es gibt (r_i o p) in A-Box und c: (all r_i b_i) in T-Box
- sammle alle b_i für r_i und r_j ,
- r_j >> r_i, als Beschreibung an p reichen.

Abstraktion

Die Abstraktion eines Objektes o ist die Konjunktion von:

(and c_1 ... c_n) für alle (c_i o) in A-Box

(atleast n r) Einträge für
(r_i o ...) Einträge und
(r_i o (atleast n_i)) mit r_i << r

(atmost n r) für (r_i o (atmost n_i)) Einträge mit r << r_i

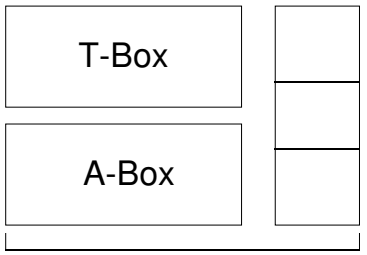
(all r c_z) falls alle Werte p_i bekannt sind für (r_i o p_i)
mit r_i << r, Beschreibungen von
p_i abstrahieren und generalisieren zu c_z.

Klassifikation

... für alle Abstraktionen aufrufen.

meist: Tiefenbeschränkung der Vorwärtsinferenzen

=> Realisierung ist auch unvollständig



„erklärbare Expertensysteme“

Zusammenfassung

Sem. Netze — Frames

Termsubsumptions-formalismen — default Logik

T-Box Syntax
Semantik -> ext
subsumes <- Klassifikation
A-Box Syntax
Semantik -> int
covers <- Realisierung

entscheidbar
unvollständig (falls androle) Also: aufpassen bei falsch!
polynomial (falls kein disjoint)