25.10.13

→ Subsymbolische KI

1. Einleitung
2. Logik: Aussagenlogik, Prädikatenlogik erster Stufe
3. ~~Prolog~~
(4. ~~DCG)~~
5. Suchen, Spielen, Planen
6. Constraints
7. Soft Computing
8. ~~Neuronale Netze~~
9. ~~Semantische Netze & Frames~~

~~Lernverfahren~~
- ~~Entscheidungsbäume~~
- ~~Ähnlichkeitsräume~~

— nicht relevant für Klausur

Logik

Wahrheitstabelle

A	B	und $A \wedge B$	einschließl. oder $A \vee B$	entweder oder A XOR B	wenn dann $A \Rightarrow B$	genau dann wenn $A \Leftrightarrow B$
w	w	w	w	f	w	w
w	f	f	w	w	f	f
f	w	f	w	w	w	f
f	f	f	f	f	w	w

Sind folgende Formeln wahr / erfüllbar / falsifizierbar / falsch ?

(1) $A \Rightarrow B$ erfüllbar, falsifizierbar

(2) $A \Rightarrow \neg A$ — " —

(3) $A \Rightarrow A$ tautologisch, erfüllbar

(4) $\neg A \Rightarrow A$ erfüllbar, falsifizierbar

(5) $(A \wedge \neg A) \Rightarrow B$ tautologisch, erfüllbar

(6) $(A \Rightarrow B) \vee (A \Rightarrow \neg B)$ — " —

(7) $(A \Rightarrow B) \wedge (A \Rightarrow \neg B)$ erfüllbar, falsifizierbar

→ Soetwas kommt in der Klausur dran

Beispiel:

Zwei Zeugen beobachten ein Verbrechen. Die Zeugen machen folgende Aussagen:

Zeuge 1	A und B arbeiten nie zusammen.
Zeuge 2	Wenn A der Täter war, dann hat er mit B zusammengearbeitet.

Daraus folgt zwingend, dass A unschuldig ist.

Wir wollen also hier **nicht beweisen**, dass A unschuldig ist, sondern, dass aus der Gültigkeit der Aussagen 1 und 2 **folgt**, dass A unschuldig ist.

Z1: $\neg(A \wedge B)$
Z2: $A \Rightarrow B$
$\Big\} \; \Phi$

Z1 ∧ Z2: $\neg(A \wedge B) \wedge A \Rightarrow B$

$\neg A$: φ

Logisches Folgern: $\models$

Formelmenge Φ — links $\models$ rechts φ

folgern

wenn hier wahr — muss es auch hier wahr sein

Zwei Formel sind äquivalent, wenn:

φ links $\equiv$ rechts ξ

wenn hier wahr — auch hier wahr

muss auch hier wahr sein — wenn hier wahr

Bezeichnung	Äquivalenz	Bezeichnung	Äquivalenz
Implikation	$A \Rightarrow B \equiv \neg A \vee B$	Idempotenz	$A \wedge A \equiv A$
Log. Äquivalenz	$A \Leftrightarrow B \equiv A \Rightarrow B \wedge B \Rightarrow A$	Idempotenz	$A \vee A \equiv A$
De Morgan	$\neg(A \vee B) \equiv \neg A \wedge \neg B$	Absorption	$A \vee (A \wedge B) \equiv A$
De Morgan	$\neg(A \wedge B) \equiv \neg A \vee \neg B$	Absorption	$A \wedge (A \vee B) \equiv A$
Distributivität	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$	Doppelte Negation	$\neg(\neg A) \equiv A$
Distributivität	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$	Kontraposition	$A \Rightarrow B \equiv \neg B \Rightarrow \neg A$
Assoziativität	$A \vee (B \vee C) \equiv (A \vee B) \vee C$		$A \wedge \mathbf{W} \equiv A$
Assoziativität	$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$		$A \wedge \mathbf{F} \equiv \mathbf{F}$
Kommutativität von $\wedge$	$A \wedge B \equiv B \wedge A$		$A \vee \mathbf{W} \equiv \mathbf{W}$
Kommutativität von $\vee$	$A \vee B \equiv B \vee A$		$A \vee \mathbf{F} \equiv A$

Vereinfachen Sie
$\neg((A \vee \neg(B \wedge A)) \wedge (C \vee (D \vee C)))$

$$\equiv \neg((\underbrace{A \vee \neg B \vee \neg A}) \wedge (D \vee (C \vee C)))$$
$$\equiv \neg(\; W \;\wedge\; (D \vee C))$$
$$\equiv \overline{F} \;\vee\; \neg(D \vee C)$$
$$\equiv \neg D \wedge \neg C$$

× von außen nach innen oder
von innen nach außen

Überführen Sie die folgende Formel in KNF!
$\neg A \Leftrightarrow (B \wedge C)$

$$(\neg A \Rightarrow (B \wedge C)) \wedge ((B \wedge C) \Rightarrow \neg A)$$
$$\neg(\neg A) \vee (B \wedge C) \wedge (\neg(B \wedge C) \vee \neg A)$$
$$(A \vee (B \wedge C)) \wedge \underbrace{(\neg B \vee \neg C \vee \neg A)}_{\text{Klausel}}$$

$$\underbrace{(A \vee B)} \wedge \underbrace{(A \vee C)} \wedge \text{— " —}$$

26.10.13

- 90 Min Klausur ⟶ 1 DIN A4 Blatt handschriftlich vorne und hinten beschrieben sowie Taschenrechner

- Wenn eine Variable existiert muss diese mit Quantor versehen werden
- Reihenfolge der Quantoren wichtig

Formel	::= *AtomareFormel* | *KomplexeFormel* | *Quantor Variable [, Variable]* Formel*
AtomareFormel	::= **wahr** | **falsch** | *Prädikat***(***Term [, Term]****)** | *Term* **=** *Term*
Term	::= *Funktion***(***Term [, Term]****)** | *Konstante* | *Variable*
Prädikat	::= **verheiratet** | **hat_geschwister** | ... ⟶ beinhalten Variablen
Funktion	::= **schwester** | ... ⟶ beinhalten Fakten
Quantor	::= ∀ | ∃
Konstante	::= **p** | **q** | **michi** | **norbert** | **ottomar** | **...**
Variable	::= **P** | **Q** | **R** | **S** | ...
KomplexeFormel	::= ¬ *Formel* | (*Formel* ∧ *Formel*) | (*Formel* ∨ *Formel*) | (*Formel* ⇒ *Formel*) | (*Formel* ⇔ *Formel*)

Ersetzen = Substituieren = Unifikation

⟶ Nur Variablen können ersetzt werden, durch:
- andere Variable
- Konstante
- Funktion

⟹ <u>ABER:</u> nur wenn Prädikat gleiche Stelligkeit hat

Liebt(X, Y) Liebt(michi, elsa)

x = michi

y = elsa

liebt(X, Y, von, bis) liebt(michi, elsa)

nicht kombinierbar ohne von, bis

liebt(michi, elsa, 1998, 2003)

→ Prädikate müssen gleich heißen

liebt(X, Y) ≠ liebe(michi, elsa)

In der Umgangssprache wird häufig impliziert quantifiziert:

- Delphine sind intelligent
  `∀X delphin(X) ⇒ intelligent(X)`
- Da sind Karpfen im Teich
  `∃X karpfen(X) ∧ im_teich(X)`
- Jeder Mensch verhält sich zu irgendjemanden loyal
  `∀X mensch(X) ⇒∃Y (mensch(Y) ∧ loyal(X,Y))`

Faustregel:

- All-Quantifizierung i.a. Implikation
- Existenz-Quantifizierung i.a. UND

→ Für Klausur nicht so relevant:

Umformung von natürlich sprachlichen Sätzen in Prädikatenlogik.

4. Paul liebt Paula. → lieben(paul, paula)
5. Alle lieben Paula. → $\forall X$ lieben(X, paula)
6. Jemand liebt Paula. → $\exists X$ lieben(X, paula)
7. Niemand liebt Paula. → $\forall X$ $\neg$ lieben(X, paula)

   oder

   $\neg \exists X$ lieben(X, paula)

8. Manchmal ist Paul einsam.

$\exists z \; zeitraum(z) \wedge einsam(paul, z)$

Zeitabschnitt muss abgebildet werden

"Es gibt einen Zeitraum" → ∃

Zusammenhang z und Paul wichtig

9. Elefanten haben vor Mäusen Angst.

$\forall x, y \; elefant(x) \wedge maus(y) \Rightarrow angst(x, y)$

10. Auf jeden Topf passt ein Deckel.

$\forall x \; topf(x) \Rightarrow \exists y (deckel(y) \wedge passt(x, y))$

Die Semantik der Prädikatenlogik liefert eine formale Grundlage zur Bestimmung des Wahrheitswertes wohlgeformter Ausdrücke.

Ob ein wohlgeformter Ausdruck wahr ist, hängt von der Bedeutung seiner Konstanten, Variablen, Prädikate und Funktionssymbole ab.

- Das heißt von der Abbildung der Konstanten, Variablen, Prädikate und Funktionssymbole auf Objekte und Relationen einer Bezugswelt (Domäne).
- Die Wahrheit von Beziehungen innerhalb der Domäne bestimmt den Wahrheitsgehalt der zugehörigen Ausdrücke.

Die Abbildung von Symbolen auf Objekte und Relationen der Domäne heißt Interpretation.

"closed world assumption"

→ Bei Wissensgewinn (ändern der Datenbasis) können sich Anfrage-ergebnisse ändern

$\equiv$ Äquivalent

$\models$ Folgern (heißt nicht unbedingt ableiten (nachschlagbar))

$\vdash$ Ableiten

$$(\forall X \; (a(X) \wedge b(X)) \Rightarrow (c(X,k) \wedge (\exists Y \; \exists Z \; c(Y,Z) \Rightarrow d(X,Y)))) \vee (\neg \exists X \; e(X))$$

1. Implikation durch Disjunktion

$$(\forall x \; \neg(a(x) \wedge b(x)) \vee ((c(x,k) \wedge (\exists y \; \exists z \; \neg c(y,z) \vee d(x,y)))) \vee (\neg \exists x \; e(x))$$

2. Umwandlung negierter Quantoren

$\neg \exists x \ \varphi(x) \equiv \forall x \ \neg\varphi(x)$ UND $\neg\forall x \ \varphi(x) \equiv \exists x \ \neg\varphi(x)$

[Formel von oben] ...
... $\vee \ (\forall x \ \neg e(x))$

3. Negation nach innen:

$\neg(\varphi \vee \xi) \equiv \neg\varphi \wedge \neg\xi$ und $\neg(\varphi \wedge \xi) \equiv \neg\varphi \vee \neg\xi$

$\dots \ (\neg a(x) \vee \neg b(x)) \vee \dots$

4. Alle Quantoren nach vorne, falls notwendig dabei Umbenennung von Variablen (pränexe Normalform)

$\forall x \ \exists y \ \exists z \ \forall w$ .... [Formel] .... $(\neg e(w))$

⟹ Fuzzy Logik Selbststudium!

Kapitel Suche:

- Wie funktioniert Tiefen- & Breitensuche relevant für Klausur

- Heuristik muss sich auf verwendete Kosten beziehen z.B.: Kosten KM (bei Navi), dann Heuristik auch related to KM

- Heuristik:
  ↳ Schätzen wie viele Kosten ich noch bis zum Zielzustand benötige / was schmeiße ich in die OpenList

- Heuristik steht an jedem Knoten

Knoten n :

Bewertungsfunktion : $f(n)$

Nearest Neighbor : $f(n) = g(n)$

Bestensuche : $f(n) = h(n)$ sortiert alle Kandidaten anhand kleinster Heuristik

Hillclimbing

↳ Hillclimbing in BT:

Nur 1 Kandidat mit Heuristik & Heuristik besser oder gleich Vaterknoten

alle Kandidaten mit Heuristik & Heuristik besser oder gleich Vaterknoten

→ In Klausur nicht auf Teufel komm raus anwenden, Algorithmus muss nicht zu einem Ergebnis führen

Unterschied: Qualität der Heuristik

A Alg(oritmus) $f(n) = g(n) + h(n)$

Heuristik

bisher aufgelaufene Kosten

Bewertungsfunktion für aktuellen Knoten wo ich mich befinde

A* Alg.

als einziges vollständig und optimal

→ Finde ich eine Heuristik die in jedem Knoten nicht überschätzt ist dies ein A* Alg.

= Zulässig

alle Kandidaten entsprechend $f(n) = h(n) + g(n)$ kleinster Wert zuerst

→ Lösen des Suchproblems ist das Finden einer guten Heuristik