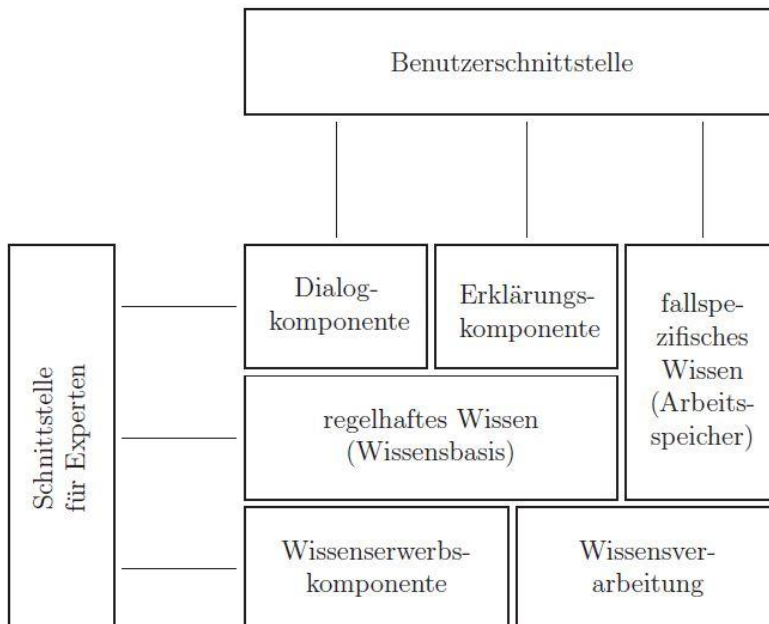


Fragenkatalog

1. Was ist der Unterschied zwischen Experten Systemen und Wissensbasierten Systemen?

Die Unterscheidung der Herkunft der Art des in der Wissensbasis vorhandenen Wissens wird als einfaches Kriterium gewertet, um zwischen einem Expertensystem und allgemeiner einem wissensbasierten System zu unterscheiden. Expertensysteme sind demnach gerade spezielle wissensbasierte Systeme, bei denen das Wissen letztlich von Experten stammt. Demnach sind viele existierende wissensbasierte Systeme Expertensysteme.

2. Expertensysteme sind ja spezielle wissensbasierte Systeme. Welche Komponenten enthalten denn üblicherweise solche Expertensysteme?



3. Wie ist die Architektur eines wissensbasierten Systems aufgebaut?

Wichtig ist die Trennung zwischen der Darstellung des Wissens über den betreffenden Problembereich (Wissensbasis) und der Verarbeitung dieses Wissens (Wissensverarbeitung).

4. Die Regelbasis ist ja auch Teil eines Expertensystems. Was sind denn Regeln?

Regeln sind formalisierte Konditionalsätze. Form: Wenn (IF) A Dann (Then) B – wenn A erfüllt dann B. Wenn Teil ist Prämisse oder Antezedenz, Dann Teil Konklusion oder Konsequenz.

5. Was sind regelbasierte Systeme?

Regelbasierte Systeme sind sozusagen die Grundform eines wissensbasierten Systems. Zu den ältesten und bewährtesten Formen der Wissensrepräsentation in Computersystemen gehören die sog. Wenn-dann-Regeln. Einerseits stellen sie Wissen in einer gut verständlichen Weise dar, andererseits lassen sich (deterministische) Regeln mit Hilfe der klassischen Logik adäquat verarbeiten. Die regelbasierten Systeme fundieren also auf gut verstandenen und erprobten Techniken mit Tradition. Auch heute noch werden sie in klar strukturierten Bereichen, bei denen es lediglich auf 0-1-Entscheidungen ankommt, eingesetzt.

6. Auf welche Weise werden Schlussfolgerungen in regelbasierten Systemen gezogen? Wie arbeitet eine Regelbasis? Wie kann man in regelbasierten Systemen beweisen?

Mit der Vorwärtsverkettung (forward chaining) und der Rückwärtsverkettung (backward chaining)

7. Was sind die 8 Schritte bei der Entwicklung eines wissensbasierten Systems?

Problembeschreibung: Funktionalität des zu entwickelnden Systems festlegen

Wissensquellen: Welche Quellen kommen für das Wissen in Frage (z.B. DB, Buch, menschliche Experten)

Design: Inferenzmodelle, Struktur der Wissensdarstellung, Schnittstellen Design

Entwicklungswerkzeuge: Vorhandene Werkzeuge oder Programmiersprachen

Entwicklung Prototyp: wichtig frühzeitige funktionierende Prototypen (Rapid Prototyping)

Testen des Prototyps: Verzahnt mit Entwicklung

Verfeinerung und Generalisierung: Weiterentwicklung des Prototypen

Wartung und Pflege: Fehlerbeseitigung, Erweiterungen

8. Was bedeutet Deduktives Schließen?

Aus dem Wissen, dass zum Starten eines Autos eine aufgeladene Batterie notwendig ist und dass bei einem gegebenen Auto die Batterie leer ist, kann man schließen, dass der Wagen nicht gestartet werden kann. Diese Art von Schließen nennt man deduktives Schließen.

9. Was bedeutet Induktives Schließen?

Andererseits könnte man aus der wiederholten Beobachtung, dass ein Auto nicht startet und die Batterie leer ist, die Regel ableiten, dass ein Auto, das eine leere Batterie hat, nicht gestartet werden kann. Diese Art der Inferenz von regelhaftem Wissen nennt man induktives Schließen.

10. Was bedeutet Abduktives Schließen?

Eine dritte Art der Schlussfolgerung wird angewendet, wenn man aus dem Wissen, dass ein Auto mit leerer Batterie nicht gestartet werden kann und dass sich ein gegebener Wagen nicht starten lässt, schließt, dass die Batterie leer ist. Diese Art von Schlussfolgerung, die nach einer Erklärung für Beobachtungen sucht, nennt man auch abduktives Schließen.

11. Welches ist das einzige sichere Schließen?

Das deduktive Schließen

12. Was ist ein Beweis?

Eine Folge von Schritten einer korrekten Inferenzprozedur, die B aus W ableitet, ist ein Beweis.

Z.B. kann man mittels Widerspruch (Resolution) beweisen.

13. Was ist probabilistisches Schließen?

Probabilistisches Schließen hat zum Ziel, eine Variante des Alltagsschließens (mit Wahrscheinlichkeiten) zu formalisieren. Es beruht auf der mathematischen Wahrscheinlichkeitstheorie, die den konsistenten Umgang mit Wahrscheinlichkeiten regelt.

14. Was ist Schließen mittels Fuzzy-Logik?

Ähnlich wie die probabilistischen Logiken liefert sie graduelle Aussagen unter Verwendung reeller

Zahlen aus dem Einheitsintervall, doch die Bedeutung einer solchen Gradzahl ist nicht die einer Wahrscheinlichkeit. Vielmehr geht es in der Fuzzy-Logik um die Beschreibung sog. vager Prädikate, das sind Prädikate, bei denen die Abgrenzung zwischen wahr und falsch nicht nur schwer fällt, sondern auch zu unangemessenen Schlussfolgerungen führen kann. Ein typisches Beispiel ist hier das Prädikat "groß". Ein 1,85 m großer Mann ist sicherlich groß, auch einen Mann von 1,80 m würde man noch als groß bezeichnen – aber ist ein Mann mit der Körpergröße 1,79 m nun schon "nicht groß", also klein? Die Fuzzy-Logik modelliert in solchen Fällen graduelle Abstufungen und arbeitet mit Regeln der Form "je mehr . . . desto höher", die dem menschlichen Verständnis nachempfunden sind. Insbesondere im Bereich Steuern und Regeln lässt sich so eine gute Simulation menschlicher Verhaltensweisen realisieren.

15. Was sind Signaturen in Aussagenlogik?

Im logischen System der Aussagenlogik ist eine Signatur eine Menge von (nullstelligen) Namen, die Aussagenvariable genannt werden. So ist etwa $\Sigma_{AL} = \{\text{Fieber, Krank, Arbeitsunfähig}\}$ eine aussagenlogische Signatur, die drei verschiedene Aussagenvariablen zur Verfügung stellt. Ein logisches System stellt eine Menge von Signaturen zur Verfügung. Wenn man eine Wissensbasis W damit aufbauen will, legt man zuerst eine Signatur fest und damit die Namen, die in W auftreten können.

Eine aussagenlogische Signatur Σ ist eine Menge von Bezeichnern genannt Aussagenvariablen.

16. Was sind Formeln?

Für jede Signatur Σ eines logischen Systems stellt dieses System eine Menge von Formeln zur Verfügung. Die Formeln ermöglichen es, Dinge über die zu repräsentierende Welt auszudrücken. Von den Formeln nehmen wir an, dass sie nach bestimmten Regeln wohlgeformt sind (well-formed formula). Oft sind wohlgeformte Formeln rekursiv aufgebaut, so dass man aus sog. atomaren Formeln mit logischen Verknüpfungsoperatoren, den sog. Junktoren, schrittweise komplexere Formeln aufbauen kann. Die meisten Logiken stellen binäre Verknüpfungsoperatoren zur Verfügung, die zwei Formeln als Konjunktion ("und") bzw. Disjunktion ("oder") miteinander verbinden.

17. Was ist „erfüllbar“?

erfüllbar (konsistent, Konsistenz) gdw. $Mod_{\Sigma}(F) \neq \emptyset$, d.h., wenn sie von wenigstens einer Interpretation erfüllt wird.

18. Was ist „falsifizierbar“?

falsifizierbar gdw. $Mod_{\Sigma}(F) \neq Int(\Sigma)$, d.h., wenn sie von wenigstens einer Interpretation nicht erfüllt (d.h. falsifiziert) wird.

19. Was ist „allgemeingültig“?

allgemeingültig (Tautologie) gdw. $Mod_{\Sigma}(F) = Int(\Sigma)$, d.h., wenn sie von jeder Interpretation erfüllt wird.

20. Was ist „unerfüllbar“?

unerfüllbar (widersprüchlich, inkonsistent, Kontradiktion) gdw. $Mod_{\Sigma}(F) = \emptyset$, d.h., wenn sie von keiner Interpretation erfüllt wird.

21. Erklären sie den Algorithmus Vorwärtsverkettung

Datengetriebene Inferenz:

Eingabe: Eine Regelbasis RB (Objekte und Regeln),
eine Menge \mathcal{F} von Fakten.

Ausgabe: Die Menge der gefolgerten Fakten.

1. Sei \mathcal{F} die Menge der gegebenen (evidentiellen) Fakten.
2. Für jede Regel **if** A **then** B der Regelbasis RB überprüfe:
Ist A erfüllt, so schließe auf B ;
 $\mathcal{F} := \mathcal{F} \cup \{B\}$
3. Wiederhole Schritt 2, bis \mathcal{F} nicht mehr vergrößert werden kann.

Abbildung 4.3: Algorithmus zur datengetriebenen Inferenz

22. Beschreiben sie die datengetriebene Inferenz (Vorwärtsverkettung)

Bei der Vorwärtsverkettung werden Regeln in gewohnter Weise transitiv miteinander verknüpft. Das bekannte fallspezifische Wissen wird als Ausgangspunkt für den Schlussfolgerungsprozess genommen. Aus erfüllten Prämissen wird auf die Wahrheit der Konklusion geschlossen, und diese abgeleiteten Fakten gehen erneut als faktisches Wissen in den Inferenzprozess ein. Das Verfahren endet, wenn keine neuen Fakten mehr abgeleitet werden können. Nützlich wenn man einen allgemeinen Überblick über den Zustand des Systems will.

23. Wenden Sie die Vorwärtsverkettung an.

Regeln:

$R1: \text{ if } A \wedge B \text{ then } H$	$\left\{ \begin{array}{l} R2a: \text{ if } C \text{ then } I \\ R2b: \text{ if } D \text{ then } I \end{array} \right.$
$R2: \text{ if } C \vee D \text{ then } I$	
$R3: \text{ if } E \wedge F \wedge G \text{ then } J$	$\left\{ \begin{array}{l} R4a: \text{ if } H \text{ then } K \\ R4b: \text{ if } I \text{ then } K \end{array} \right.$
$R4: \text{ if } H \vee I \text{ then } K$	
$R5: \text{ if } I \wedge J \text{ then } L$	
$R6: \text{ if } K \wedge L \text{ then } M$	

Faktenmenge $F = \{H, C, E, F, G\}$

Durch R2 kommt I dazu, durch R3 kommt J dazu, durch R4 kommt K dazu

$F = \{H, C, E, F, G\} \cup \{I, J, K\}$

Durch R5 kommt L dazu, durch R6 kommt M dazu.

$F = \{H, C, E, F, G, I, J, K\} \cup \{L, M\}$

24. Erläutern sie die Rückwärtsverkettung

Zielorientierte Inferenz:

Eingabe: Eine Regelbasis RB (Objekte und Regeln),
eine (evidentielle) Faktenmenge \mathcal{F} ,
eine Liste von Zielen (atomaren Anfragen) $[q_1, \dots, q_n]$

Ausgabe: *yes*, falls alle q_i ableitbar sind, sonst *no*

BACKCHAIN($[q_1, \dots, q_n]$)

wobei BACKCHAIN wie folgt rekursiv definiert ist:

```
BACKCHAIN( $[q_1, \dots, q_n]$ ) (1)
  if  $n = 0$  then return(yes); (2)
  if  $q_1 \in \mathcal{F}$  (3)
    then return(BACKCHAIN( $[q_2, \dots, q_n]$ )) (4)
    else for each Regel  $p_1 \wedge \dots \wedge p_m \rightarrow q$  aus  $RB$  mit  $q_1 = q$  (5)
      do if BACKCHAIN( $[p_1, \dots, p_m, q_2, \dots, q_n]$ ) = yes (6)
        then return(yes) (7)
      endfor (8)
  endif (9)
return(no) (10)
```

Abbildung 4.5: Algorithmus zur zielorientierten Inferenz

25. Erklären sie die zielorientierte Inferenz (Rückwärtsverkettung)

Braucht man den Zustand eines bestimmten Knotens bietet sich im Gegensatz zur Vorwärtsverkettung die Rückwärtsverkettung an. Auch hier liegt das Prinzip der transitiven Verknüpfung von Regeln zu Grunde. Man geht nicht von gegebenen Daten, sondern einem Zielobjekt aus. Das System durchsucht dann die Regelbasis nach geeigneten Regeln die das Zielobjekt in der Konklusion enthalten.

Anfangs besteht die Zielliste aus den Zielobjekten z.B. $[q_1]$ Liste leer \rightarrow Backchain liefert *yes*.

Ist q_1 in \mathcal{F} enthalten wird Backchain mit der Resteliste ohne q_1 aufgerufen. Anderenfalls wird RB nach Regeln durchsucht die das Zielobjekt enthalten. Findet es solche Regeln \rightarrow Verfahren wendet sie an indem die Regeln in Backchain rekursiv aufgerufen wird aber statt q_1 mit den Prämissen der Regeln. Ist der Aufruf für keine der Regeln erfolgreich kann das Zielobjekt nicht abgeleitet werden und Backchain liefert *no*. Es kann passieren, dass Backchain nicht terminiert (bei z.B. $p \rightarrow q, q \rightarrow p$)

26. Nennen sie die Äquivalenzen für die Aussagenlogik

Theorem 3.30 (Äquivalenzen für die Aussagenlogik) *Es gelten:*

1.	$F \wedge F \equiv F$ $F \vee F \equiv F$	(Idempotenz)
2.	$F \wedge G \equiv G \wedge F$ $F \vee G \equiv G \vee F$	(Kommutativität)
3.	$(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$ $(F \vee G) \vee H \equiv F \vee (G \vee H)$	(Assoziativität)
4.	$F \wedge (F \vee G) \equiv F$ $F \vee (F \wedge G) \equiv F$	(Absorption)
5.	$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$	(Distributivität)
6.	$\neg\neg F \equiv F$	(Doppelnegation)
7.	$\neg(F \wedge G) \equiv \neg F \vee \neg G$ $\neg(F \vee G) \equiv \neg F \wedge \neg G$	(de Morgansche Regeln)
8.	$\left. \begin{array}{l} F \vee G \equiv F \\ F \wedge G \equiv G \end{array} \right\}$ falls F allgemeingültig	(Tautologieregeln)
9.	$\left. \begin{array}{l} F \vee G \equiv G \\ F \wedge G \equiv F \end{array} \right\}$ falls F unerfüllbar	(Unerfüllbarkeitsregeln)
10.	$F \Rightarrow G \equiv \neg G \Rightarrow \neg F$	(Kontraposition)
11.	$F \Rightarrow G \equiv \neg F \vee G$	(Implikation)
12.	$F \Leftrightarrow G \equiv (F \Rightarrow G) \wedge (G \Rightarrow F)$	(Koimplikation)

27. Woraus besteht eine Logik?

Aus einer Signatur, Formelmenge, Menge von Interpretationen, Erfüllungsrelation

28. Was ist ein Atom?

Ein Aussagensymbol welches nicht weiter zerlegt werden kann – in einer Formel $A \vee B$ sind z.B. A und

Definition 3.42 (atomare Formel, Atom) Eine *atomare Formel* (oder *Atom*) über einer Signatur $\Sigma = (Func, Pred)$ und einer Menge V von Variablen wird wie folgt gebildet:

- (1) p falls $p \in Pred$ und p hat die Stelligkeit 0
- (2) $p(t_1, \dots, t_n)$ falls $p \in Pred$ mit der Stelligkeit $n > 0$ und $t_1, \dots, t_n \in Term_{\Sigma}(V)$

□

B Atome.

29. Wie ist die Inferenzrelation definiert?

Ganz allgemein gesprochen handelt es sich darum, aus gegebenem Wissen W neues Wissen B abzuleiten. Ist B nun eine Folgerung von W , so sind W und B durch eine Inferenzrelation R miteinander verbunden: $(W, B) \in R$. Die klassische Aufteilung der Inferenz in Deduktion, Induktion und Abduktion geht auf C. S. Peirce zurück.

30. Was ist ein Entscheidungsbaum?

Ein Entscheidungsbaum ist eine Datenstruktur, ein geordneter, gerichteter Graph, der der Darstellung von Entscheidungsregeln dient. Die grafische Darstellung als Baumdiagramm veranschaulicht hierarchisch aufeinanderfolgende Entscheidungen.

31. Um welche Art des Lernens handelt es sich beim Entscheidungsbaum?

Um induktives Lernen

32. Wie ist ein Entscheidungsbaum aufgebaut?

Die Inneren Knoten entsprechen den Attributen der Objekte, die Kanten sind die Werte der Objekte und die Blattknoten entsprechen den Klassifikationen.

33. Worauf kommt es bei der Konstruktion eines Entscheidungsbaums an?

Es soll sich um einen kompakten Baum handeln der auch neue Fälle klassifizieren kann.

34. Wie wird ein Entscheidungsbaum aufgebaut?

Mit dem DT – Algorithmus

```
function DT(E, A, default)

Eingabe: E Menge von Beispielen
         A Menge von Attributen
         default Default-Klassifikation
Ausgabe: Entscheidungsbaum

if E = ∅
then return default
else if alle Elemente in E haben die gleiche
      Klassifikation c ∈ {ja, nein}
then return c
else if A = ∅
then Fehler "gleiche Beispiele mit unterschiedlicher
      Klassifikation"
else a := ChooseAttribute(A, E)
      T := neuer Entscheidungsbaum
           mit Wurzelmarkierung a
      for each Attributwert wi von a do
        Ei := {e ∈ E | a(e) = wi}
        Ti := DT(Ei, A \ {a}, MajorityVal(E))
        hänge an den Wurzelknoten von T
           eine neue Kante mit Markierung wi
           und Unterbaum Ti an
      end
return T
```

Abbildung 5.6: Verfahren *DT* zur Generierung von Entscheidungsbäumen

35. Wie funktioniert das Entscheidungsbaumlernverfahren?

Auswahl des ersten Attributs → für jede durch die verschiedenen Attribute bestimmte Teilmenge der Trainingsmenge neue Instanz des Lernproblems.

Vier Fälle für die rekursiven Lernprobleminstanzen:

1. Beispielmenge leer: kein Beispiel mit Attribut-Wert Kombination vorhanden → Default Klassifikation möglich

2. Falls alle Beispiele gleiche Klassifikation haben wird diese ausgegeben.

3. Attributmenge leer aber noch pos. oder neg. Beispiele vorhanden: es gibt in der ursprünglichen Trainingsmenge Bsp. Mit selben Attributwerten aber unterschiedlicher Klassifikation. Entweder ist Trainingsmenge falsch oder zusätzliche Attribute sind notwendig → Algo bricht mit Fehlermeldung ab

4. Falls noch pos. und neg. Bsp. Vorhanden sind und Attributmenge nicht leer ist, wird das beste Attribut gemäß seiner „Wichtigkeit“ ausgewählt.

36. Was versteht man unter einer Warenkorbanalyse?

Das Erkennen von Zusammenhängen zwischen gekauften Produkten welche gleichzeitig – also in der gleichen Transaktion – gekauft wurde. Damit kann feststellen z.B. 20% aller Personen die Chips und Cola kauften haben auch Bier gekauft.

37. Wie kann man Warenkorbanalysen durchführen?

Mit Hilfe von Assoziationsregeln

38. Was sind Assoziationsregeln?

Assoziationsregeln sind probabilistische Wenn-dann-Regeln welche Beziehungen zwischen verschiedenen Dingen bzw. Merkmalen beschreiben. Sie beschreiben gewisse Zusammenhänge und Regelmäßigkeiten zwischen verschiedenen Dingen. Sie haben die Form $A \rightarrow B$ wobei A und B disjunkte Itemmengen sein müssen – d.h. sie dürfen kein gemeinsames Element haben.

39. Welche Elemente haben Assoziationsregeln?

Items, Transaktion, Datenbasis, Support und Confidence

40. Was sind Items, Transaktion und Datenbasis?

Items sind die Dinge deren Beziehungen zueinander beschrieben werden sollen – z.B. verschiedene Waren aus einem Supermarkt. $I = \{i_1, i_2, \dots\}$ eine (endliche) Menge solcher Items, Eine beliebige Teilmenge $X \subseteq I$ wird Itemmenge genannt. Eine Transaktion $t \subseteq I$ ist eine Itemmenge. Die Datenbasis $D = \{t_1, t_2, \dots\}$ bestehe aus einer Menge solcher Transaktionen. Eine Transaktion ist in unserem Beispiel ein Einkauf bestehend aus x Items.

41. Was sind Support und Confidence?

Der Support einer Itemmenge X ist der (relative) Anteil aller Transaktionen aus D, die X enthalten:

$$\text{support}(X) = \frac{|\{t \in \mathcal{D} \mid X \subseteq t\}|}{|\mathcal{D}|}$$

Der Support von $X \rightarrow Y$ ist der Support der Itemmenge $X \cup Y$

$$\text{support}(X \rightarrow Y) = \text{support}(X \cup Y)$$

Die Konfidenz von $X \rightarrow Y$ ist der (relative) Anteil derjenigen X enthaltenden Transaktionen, die auch Y enthalten:

$$\begin{aligned} \text{confidence}(X \rightarrow Y) &= \frac{|\{t \in \mathcal{D} \mid (X \cup Y) \subseteq t\}|}{|\{t \in \mathcal{D} \mid X \subseteq t\}|} \\ &= \frac{\text{support}(X \rightarrow Y)}{\text{support}(X)} \end{aligned}$$

Support und Konfidenz sind also nichts anderes als (bedingte) relative Häufigkeiten bzw. (bedingte) Wahrscheinlichkeiten.

Beispiel 3 elementige Itemmenge hat 6 Kombinationen für die conf Berechnung.

42. Welche Teilprobleme gibt es bei der Suche nach Assoziationsregeln?

-) Finde alle Itemmengen, deren Support über der minsupp-Schwelle liegt; diese Itemmengen werden häufige Itemmengen genannt.

-) Finde in jeder häufigen Itemmenge (siehe erstes Teilproblem) I alle Assoziationsregeln $I' \rightarrow (I-I')$ mit $I' \subset I$, deren Konfidenz mindestens minconf beträgt.

43. Wie ist ein logisches System aufgebaut?

Es gibt z.B. Aussagenlogik und Prädikatenlogik 1 Stufe.

Diese haben folgende Komponenten: Signatur, Formel, Interpretation und Erfüllungsrelation

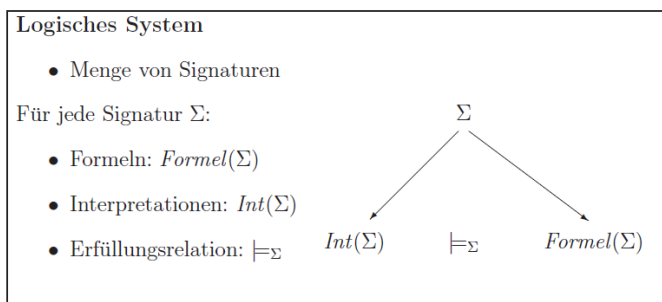


Abbildung 3.2: Komponenten eines logischen Systems

44. Was ist eine aussagenlogische Signatur?

Eine aussagenlogische Signatur Σ ist eine Menge von Bezeichnern genannt Aussagenvariablen.

45. Was ist eine aussagenlogische Formel?

Für eine aussagenlogische Signatur Σ wird die Menge $Formel(\Sigma)$ der aussagenlogischen Formeln wie folgt gebildet.

-) eine atomare Formel ist eine aussagenlogische Formel, die nur aus einer Aussagen-Variable besteht

-) falls A und B aussagenlogische Formeln sind, dann sind auch die folgenden Konstrukte aussagenlogische Formeln wobei die darin auftretenden Operationssymbole \neg , \wedge Junktoren heißen.

$\neg A$ Negation „nicht A“

$(A \wedge B)$ Konjunktion „A und B“

$(A \vee B)$ Disjunktion „A oder B“

$(A \rightarrow B)$ Implikation „wenn A dann B“

$(A \leftrightarrow B)$ Koimplikation oder Äquivalenz „A genau dann wenn B“

46. Was ist eine aussagenlogische Interpretation?

Sei Σ eine aussagenlogische Signatur, dann ist eine Abbildung $I: \Sigma \rightarrow \text{BOOL}$ eine aussagenlogische Interpretation für Σ . $Int(\Sigma)$ ist eine Menge aller Σ Interpretationen. \rightarrow eine Aussagenvariable kann durch eine Belegung beliebig interpretiert werden.

47. Was ist die Erfüllungsrelation in der Aussagenlogik?

Beispiel 3.6 (Erfüllungsrelation in der Aussagenlogik) In der Belegung I_1 aus Beispiel 3.5 gilt $I_1(\text{Fieber}) = \text{true}$. Daher sagen wir, dass die Belegung (oder Interpretation) I_1 die Formel *Fieber* erfüllt. Wir notieren dies mit

$$I_1 \models_{\Sigma_{AL}} \text{Fieber}.$$

48. Was ist die semantische Äquivalenz?

Zwei Formeln F und G sind semantisch äquivalent ($F \equiv G$) falls für alle Interpretationen I gilt:

$$\llbracket F \rrbracket_I = \llbracket G \rrbracket_I$$

49. Was ist der Aufbau der Prädikatenlogik 1. Stufe?

In der Prädikatenlogik 1. Stufe können wir nicht nur einfache Aussagen repräsentieren. Die Welten, die wir hier beschreiben können, können enthalten:

- Objekte, z.B.: Menschen, Zahlen, Farben
- Funktionen auf den Objekten, z.B.: Nachfolger, Leukozytenanzahl der Blutprobe, ...
- Aussagen (wie in der Aussagenlogik)
- Eigenschaften von Objekten, z.B.: groß, gelb, negativ
- Relationen zwischen den Objekten, z.B.: Großvater von, kleiner als

Eine Signatur $_$ der Prädikatenlogik 1. Stufe stellt ein entsprechendes Vokabular zur Verfügung.

50. Was ist die Signatur der Prädikatenlogik 1. Stufe?

Eine PL1 Signatur $\Sigma = (\text{Func}, \text{Pred})$ besteht aus einer Menge Func von Funktionssymbolen und einer Menge Pred von Prädikatensymbolen. Dabei hat jedes Symbol $s \in \text{Func}$ u Pred eine feste Stelligkeit ≥ 0 . Ein Funktionssymbol mit der Stelligkeit 0 heißt Konstante. Generell setzen wir für jede PL1 Signatur voraus, daß es mindestens eine Konstante gibt.

51. Was ist das Universum PL1?

Das Universum oder auch Trägermenge genannt ist eine beliebige nicht-leere Menge. Sie enthält alle Objekte der Interpretation.

52. Was ist die Interpretation in der PL1?

Definition 3.35 (Interpretation) Sei $\Sigma = (Func, Pred)$ eine Signatur. Eine Σ -Interpretation $I = (U_I, Func_I, Pred_I)$ besteht aus

- einer nichtleeren Menge U_I , genannt *Trägermenge* (engl. *carrier set*) (oder: Universum, Individuenbereich, Diskursbereich, Domäne)
- einer Menge $Func_I$ von Funktionen

$$Func_I = \{f_I : \underbrace{U_I \times \dots \times U_I}_{n\text{-mal}} \rightarrow U_I \mid f \in Func \text{ mit der Stelligkeit } n\}$$

- einer Menge $Pred_I$ von Relationen

$$Pred_I = \{p_I \subseteq \underbrace{U_I \times \dots \times U_I}_{n\text{-mal}} \mid p \in Pred \text{ mit der Stelligkeit } n\}$$

Die Menge der Σ -Interpretationen wird mit $Int(\Sigma)$ bezeichnet. □

53. Was ist ein Individuum?

Nullstellige Funktionssymbole werden durch Objekte oder Individuen der betrachteten Welt, d.h. durch Elemente des Universums interpretiert.

54. Wie sieht die Entwicklung eines wissensbasierten System aus?

- 1) Problembeschreibung: hier wird das zu lösende Problem definiert
- 2) Wissensquellen: Festlegung aus welchen Quellen das notwendige Wissen gewonnen werden kann z.B. Datenbanken, menschliche Experten etc.
- 3) Design: Definition welche Strukturen für die Wissensdarstellung benötigt werden, welche Arten von Inferenzen geleistet werden sollen etc.
- 4) Entwicklungswerkzeug: Festlegung welche Entwicklungswerkzeuge in Frage kommen oder ob man direkt auf Programmiersprachen zurückgreifen soll.
- 5) Entwicklung eines Prototyps: bei wissensbasierten Systemen sind frühe funktionsfähige Prototypen wichtig um die ursprünglichen Anforderungen zu überprüfen.
- 6) Testen des Prototyps: mit Phase 5 eng verzahnt → Überprüfung des Prototyps.
- 7) Verfeinerung und Generalisierung: hier können bisher untergeordnete Funktionen welche bisher nicht berücksichtigt wurden implementiert werden.
- 8) Wartung und Pflege: Fehler müssen beseitigt werden bzw. neue Funktionen hinzugefügt werden.

55. Wie wird die Inferenzrelation nach Pierce charakterisiert?

Deduktion: die getroffenen Schlussfolgerungen sind stets korrekte Schlussfolgerungen. Man schließt von einem allgemeinen auf einen speziellen Fall. Bsp. Aus dem Wissen, dass zum Starten eines Autos eine aufgeladene Batterie notwendig ist, und dass bei einem gegebenen Auto die Batterie leer ist, kann man schließen, dass der Wagen nicht gestartet werden kann.

Induktion: hier erschließt sich regelhaftes Wissen aus einzelnen Sachverhalten. Das abgebildete Wissen muss nicht stets wahr sein. Bsp. Aus der wiederholten Beobachtung, dass ein Auto nicht startet und die Batterie leer ist wird die Regel „Auto mit leerer Batterie startet nicht“ abgeleitet.

Abduktion: ist das Erschließen eines Sachverhalts aus einer vorliegenden Beobachtung aufgrund regelhaften Wissens. Bsp. Wenn man aus dem Wissen, dass ein Auto mit leerer Batterie nicht gestartet werden kann und dass sich ein gegebener Wagen nicht starten lässt schließt, dass die Batterie leer ist. Auch nicht notwendigerweise wahr.

56. Was ist das Ziel von nicht-monotonen Schließen?

Die Zielsetzung nichtmonotoner Logiken besteht darin, revidierbares Schließen zu ermöglichen. Es können Schlussfolgerungen die sich als falsch herausgestellt haben zurückgenommen werden und alternative Schlussfolgerungen abgeleitet werden.

57. Was ist ein Default?

In der Default-Logik geht es um unsichere Regeln. Default Regeln sind:

-) Regeln mit Ausnahmen
-) Regeln die im Allgemeinen meistens oder typischerweise gelten
-) Regeln die gelten solange nicht das Gegenteil explizit bewiesen worden ist.

58. Wie lautet die mathematische Notation des Default?

Allgemein können wir also sagen, dass ein *Default* δ ein Ausdruck der Form

$$\delta = \frac{\varphi : \psi_1, \dots, \psi_n}{\chi} \quad (8.5)$$

ist, wobei die einzelnen Komponenten von δ wie folgt bezeichnet werden:

φ	(Default)Voraussetzung (prerequisite)
$\{\psi_1, \dots, \psi_n\}$	(Default)Begründungen (justifications) ³
χ	(Default)Konsequenz (consequent)

59. Zeigen sie ein Beispiel für einen Default

Weg_zur_Uni : nehme_Bahn
nehme_Bahn

60. Was ist MYCIN?

Mycin ist ein medizinisches Konsultationssystem welches zur Diagnose und Therapie von Infektionskrankheiten durch Antibiotika eingesetzt wird. Es wurde in den 70er Jahren entwickelt.

61. Wie ist MYCIN aufgebaut?

-) das Herzstück sollte ein interaktives entscheidungsunterstütztes Programm sein. Es gibt einen Frage-Antwort Dialog zwischen Benutzer und System.
-) neben den subjektiven Bewertungen des Dialogs sollten Labordaten analysiert werden
-) es sollte in der Lage sein gespeichertes Wissen darzulegen und Schlussfolgerungen zu erklären (Erklärungskomponente)
-) neues Wissen soll eingegeben werden können um die Leistungsfähigkeit zu steigern (Wissensakquisition)
-) neues Wissen sollte nicht zu Umstrukturierung führen (modulare Wissenspräsentationsform)

62. Was ist ein wissensbasiertes System – Was ist das Besondere daran?

Ein wissensbasiertes System ist ein intelligentes Informationssystem in dem Wissen mit Methoden der Wissensrepräsentation und Wissens Modellierung abgebildet und nutzbar gemacht. Wichtig ist die Trennung zwischen der Darstellung des Wissens über den betreffenden Problembereich (Wissensbasis) und der Verarbeitung dieses Wissens (Wissensverarbeitung).

63. Was für Logiken gibt es?

Aussagenlogik und Prädikatenlogik

64. Welche Lernstrategien gibt es?

Direkte Eingabe von neuen Wissen uns Auswendiglernen – keine Interferenz oder andere Art von Wissenstransformation erforderlich. Direktes Speichern von gegebenen Daten ohne Schlussfolgerung.

Lernen durch Anweisungen – hier wird aufbereitetes Wissen vorgegeben. Lernender muss dieses Wissen aufnehmen und intern verarbeiten. Wissen muss effektiv verwendet werden → nur gewisse Schlussfolgerungen notwendig.

Lernen durch Deduktion – hier wird aus vorhandenen Wissen mittels deduktiver Schlussweisen neues Wissen abgeleitet.

Lernen durch Analogie – neue Fakten und Fähigkeiten dadurch erlernt, dass man vorhandenes Wissen an neue Situationen anpasst → ausgeprägte Fähigkeiten für Schlussfolgerungen notwendig. Problemlösungen müssen aus dem vorhanden Wissen herausgefunden werden, das ausgewählte Wissen muss transferiert und angepasst werden und als neues Wissen abgelegt werden.

Beim Lernen aus Beispielen ist es Aufgabe des Lernenden eine allgemeine Konzeptbeschreibung zu erstellen die alle vorher gegebenen Beispiele umfasst und evtl. vorhandene Gegenbeispiele ausschließt. Hier sind größere Schlussfolgerungsfähigkeiten notwendig.

Lernen durch Beobachtungen und durch Entdeckungen – bei dieser Form des Lernens handelt es sich um die anspruchsvollste Form des Lernens – es ist eine induktive Form des Lernens. Keine Steuerung durch den Lehrenden mehr möglich. Die gemachten Beobachtungen können verschiedene Konzepte umfassen die alle mehr oder weniger gleichzeitig gelernt werden müssen.

65. In welchem Bereich liegen bei Mycin die Sicherheitsfaktoren?

Eine reelle Zahl zwischen -1 bis +1

66. Was sind die Sicherheitsfaktoren bei Mycin?

Sicherheitsfaktoren drücken allgemein den Grad des Glaubens an eine Hypothese aus. Das Ziel einer Anfrage mit Sicherheitsfaktoren ist nicht einfach die Aussage ob ein Zielobjekt wahr oder falsch ist, sondern eine Angabe eines Faktors für die Sicherheit des Zielobjektes.

67. Wie lauten bei Mycin die Propagationsregeln?

1. *Konjunktion:*

$$CF[A \wedge B] = \min\{CF[A], CF[B]\}$$

2. *Disjunktion:*

$$CF[A \vee B] = \max\{CF[A], CF[B]\}$$

3. *serielle Kombination:*

$$CF[B, \{A\}] = CF(A \rightarrow B) \cdot \max\{0, CF[A]\}$$

4. *parallele Kombination:* Für $n > 1$ ist

$$CF[B, \{A_1, \dots, A_n\}] = f(CF[B, \{A_1, \dots, A_{n-1}\}], CF[B, \{A_n\}])$$

wobei die Funktion $f : [-1, 1] \times [-1, 1] \rightarrow [-1, 1]$ folgendermaßen definiert ist:

$$f(x, y) := \begin{cases} x + y - xy & \text{wenn } x, y > 0 \\ x + y + xy & \text{wenn } x, y < 0 \\ \frac{x+y}{1-\min\{|x|, |y|\}} & \text{sonst} \end{cases}$$

68. Welche Inferenzregeln gibt es?

Modus ponens besagt, daß man bei Vorliegen von Formeln F und $F \rightarrow G$ die Formel G ableiten kann.

Modus tollens ist die Umkehrung von MP. Liegen die Formeln $F \rightarrow G$ und $\neg G$ vor kann $\neg F$ abgeleitet werden.

Die \wedge -Einführung besagt, daß aus der Gültigkeit von zwei Formeln deren Konjunktion geschlossen werden kann.

Die \wedge -Elimination besagt, daß man aus einer Konjunktion auf ein Konjunktionsglied schließen kann.

69. Was ist die Folgerung?

Aus F folgt logisch G (geschrieben $F \models_{\Sigma} G$) genau dann, wenn jede Interpretation, die F erfüllt, auch G erfüllt.

70. Was ist ein Kalkül?

Ein Kalkül besteht aus einer Menge von logischen Axiomen und Inferenzregeln. Die Axiome sind eine Menge von elementaren Tautologien (pos. Kalkül) oder einer Menge von elementaren Widersprüche (negativer Kalkül). Die Inferenzregeln sind eine Menge von Vorschriften nach denen weitere Formeln abgeleitet werden können.

71. Wie kann man das beste Attribut auswählen beim Entscheidungsbaum?

Das Entscheidungsbaumlernsystem ID3 formalisiert die Idee von guten und schlechten Attributen durch die Berücksichtigung des Informationsgehalts der Attribute.

72. Was ist der Informationsgehalt?

Der Informationsgehalt wird wie in der Informatik üblich in Einheiten von bits gemessen wobei 1 bit dem Informationsgehalt von einer Ja/Nein Antwort entspricht zu deren Antwort man vorher keine Information vorliegen hatte.

Berechnung mittlerer Informationsgehalt (auch Entropie genannt):

$$H(P(v_1); \dots; P(v_k)) = - \sum_{i=1}^k P(v_i) \log_2 P(v_i)$$

Wert liegt zwischen 0 und 1 – 0 wenn nur pos oder nur neg. Bsp vorhanden sind (Informationsgewinn wird maximal) und 1 bei 50:50.

73. Was ist der Informationsgewinn (information gain?)

Der Informationsgewinn durch das Attribut a ist die Differenz zwischen der ursprünglichen Information und der Restinformation

$\text{Gain}(a) = I(E) - I(E|a \text{ bekannt})$ berechnet. Wird vom Verfahren ID3 verwendet. $\text{Gain}(a)$ ist der absolute Informationsgewinn

$$I(E_i) = H\left(\frac{p_i}{p_i + n_i}; \frac{n_i}{p_i + n_i}\right) \text{ bit}$$

$$\begin{aligned}
I(E \mid a \text{ bekannt}) &= \sum_{i=1}^k P(a = w_i) \cdot I(E_i) \\
&= \sum_{i=1}^k \frac{p_i + n_i}{p + n} \cdot H\left(\frac{p_i}{p_i + n_i}; \frac{n_i}{p_i + n_i}\right) \text{ bit}
\end{aligned}$$

Kann bei vielen Datensätzen wie z.B. ID Nummern falsche Daten liefern – daher verwendet Verfahren C4.5 gain ratio

$$\text{gain ratio}(a) = \frac{\text{gain}(a)}{\text{split info}(a)}$$

wobei $\text{split info}(a)$ die sog. Entropie des Attributes a ist

$$\text{split info}(a) = H(a) = - \sum_{i=1}^k P(a = w_i) \log_2 P(a = w_i)$$

74. Was ist das Occam's Razor?

„Bevorzuge die einfachste Hypothese, die konsistent mit allen Beobachtungen ist“

Ziel des Lernens ist es einen Entscheidungsbaum zu generieren der nicht nur die Beispiele der gegebenen Trainingsmenge korrekt klassifiziert, sondern auch möglichst kompakt ist. Im Allgemeinen ist ein kleiner Entscheidungsbaum, der konsistent mit allen Beispielen ist der korrekt als ein großer komplexer Entscheidungsbaum.

75. Schreiben sie in PL1 „Alle Vögel können fliegen“

$$\forall x \text{ Vögel}(x) \rightarrow \text{fliegen}(x)$$

76. Wie ist der Operator $Cn(F)$ definiert?

Definition 3.14 (klassisch-logische Inferenzoperation) Die Funktion

$$\begin{aligned}
Cn &: 2^{\text{Formel}(\Sigma)} \rightarrow 2^{\text{Formel}(\Sigma)} \\
Cn(\mathcal{F}) &:= \{G \in \text{Formel}(\Sigma) \mid \mathcal{F} \models G\}
\end{aligned}$$

heißt *klassisch-logische Inferenzoperation*. □

Definition 3.15 (deduktiv abgeschlossen, Theorie) Eine Formelmenge $\mathcal{F} \subseteq \text{Formel}(\Sigma)$ mit

$$Cn(\mathcal{F}) = \mathcal{F}$$

heißt (*deduktiv abgeschlossen*). Formelmengen \mathcal{F} bzw. ihr deduktiv abgeschlossenes Pendant $Cn(\mathcal{F})$ werden auch als *Theorien* bezeichnet. □

Eine (deduktiv abgeschlossene) Theorie ist daher ein *Fixpunkt* des Operators Cn .

die Inferenzoperation ist eine Abbildung die einer Menge von Formeln die Menge aller Formeln zuordnet die sich aus ihr logisch schlussfolgern lässt.

77. Erläutern sie den Apriori Algorithmus?

```

Apriori( $\mathcal{D}$ )

Eingabe: Datenbasis  $\mathcal{D}$ 
Ausgabe: Menge häufiger Itemmengen

 $L_1 := \{\text{häufige 1-Itemmengen}\}$ 
 $k := 2$ 
while  $L_{k-1} \neq \emptyset$  do
   $C_k := \text{AprioriGen}(L_{k-1})$            % neue Kandidatenmengen
  for all Transaktionen  $t \in \mathcal{D}$  do
     $C_t := \{c \in C_k \mid c \subseteq t\}$    % in  $t$  enthaltene
                                          % Kandidatenmengen

    for all Kandidaten  $c \in C_t$  do
       $c.\text{count} := c.\text{count} + 1$ 
    end for
  end for
   $L_k := \{c \in C_k \mid c.\text{count} \geq |\mathcal{D}| \cdot \text{minsupp}\}$ 
   $k := k + 1$ 
end while
return  $\bigcup_k L_k$ 

```

Abbildung 6.1: Der Apriori-Algorithmus

78. Erläuterung AprioriGen

```

AprioriGen( $L_{k-1}$ )

Eingabe: Menge häufiger ( $k-1$ )-Itemmengen  $L_{k-1}$ 
Ausgabe: Obermenge der Menge häufiger  $k$ -Itemmengen

 $C_k := \emptyset$ 
for all  $p, q \in L_{k-1}$  mit  $p \neq q$  do
  if die ersten  $k-2$  Elemente von  $p$  und  $q$  sind gleich, % Items sind geordnet:
     $p = \{e_1, \dots, e_{k-2}, e_p\}$ ,           %  $e_1 < e_2 < \dots < e_{k-2} < e_p$ 
     $q = \{e_1, \dots, e_{k-2}, e_q\}$          %  $e_1 < e_2 < \dots < e_{k-2} < e_q$ 
    und  $e_p < e_q$  then
       $C_k := C_k \cup \{\{e_1, \dots, e_{k-2}, e_p, e_q\}\}$ 
    end if
end for
for all  $c \in C_k$  do
  for all ( $k-1$ )-Teilmengen  $s$  von  $c$  do % Teilmengencheck
    if  $s \notin L_{k-1}$  then
       $C_k := C_k \setminus \{c\}$ 
    end if
  end for
end for
return  $C_k$ 

```

Abbildung 6.2: Der AprioriGen-Algorithmus

79. Erklären sie den Apriori Algorithmus und AprioriGen

Im ersten Durchlauf bestimmt der Algorithmus die Menge L_1 der häufigen 1-Itemmengen, indem er einfach zählt, wie oft jedes Item in den Transaktionen aus \mathcal{D} vorkommt. Jede weitere, k -te Iteration p.P.

($k \geq 2$) besteht aus drei Schritten: Im ersten Schritt werden aus den im ($k-1$)-Durchlauf bestimmten häufigen Itemmengen, L_{k-1} , mittels AprioriGen die Kandidaten-Itemmengen, C_k , berechnet. Im zweiten Schritt (for-Schleife in Apriori) wird für jede Kandidatenmenge $c \in C_k$ in dem Zähler $c.\text{count}$

die Anzahl der Transaktionen $t \in D$ gezählt, die c enthalten. Für jede Transaktion $t \in D$ werden dazu zunächst die in t enthaltenen Kandidatenmengen C_t bestimmt. Im dritten Schritt werden in L_k alle diejenigen Kandidatenmengen $c \in C_k$ aufgenommen, deren Support in D größer als minsupp ist (d.h. $c.\text{count} \geq |D| \cdot \text{minsupp}$). Als Ergebnis liefert Apriori schließlich die Vereinigung aller L_k . Der Algorithmus benötigt m Iterationen, wobei m die Kardinalität der größten häufigen Itemmengen ist. Seine Ausgabe ist die Menge aller häufigen Itemmengen, d. h. die Menge aller Itemmengen I mit $\text{support}(I) \geq \text{minsupp}$. Der Kandidaten-Algorithmus AprioriGen nutzt aus, dass jede Teilmenge einer häufigen Itemmenge wieder häufig sein muss, insbesondere, dass also jede $(k-1)$ -Teilmenge einer Menge aus L_k in L_{k-1} enthalten sein muss. Zwei häufige $(k-1)$ -Itemmengen $p, q \in L_{k-1}$, die in genau $(k-2)$ Elementen übereinstimmen, werden vereinigt und ergeben so eine neue k -Itemmenge. Aus der so gewonnenen Menge möglicher Kandidaten werden dann durch den anschließenden Teilmengencheck noch diejenigen Mengen entfernt, deren $(k-1)$ -

Teilmengen nicht alle in L_{k-1} liegen. Da alle $(k-1)$ -Teilmengen häufig sein müssen, können wir bereits bei der Erzeugung der k -Itemmengen eine weitere Einschränkung vornehmen: Der

Algorithmus AprioriGen nimmt an, dass alle Items lexikographisch geordnet sind, und zwei häufige $(k-1)$ -Itemmengen werden nur dann für die Erzeugung einer k -Itemmenge verwendet, wenn die gemäß dieser Ordnung ersten $k-2$ Elemente e_1, e_2, \dots, e_{k-2} in beiden Mengen gleich sind und die Mengen sich daher nur im jeweils größten Element unterscheiden.

80. Was ist die Monotonie?

Da die klassisch-logische Folgerungsrelation immer *alle* Modelle, d.h. alle erdenklichen Möglichkeiten, mit einbezieht, ist C_n *monoton*, d.h.

$$\text{aus } \mathcal{F} \subseteq \mathcal{H} \text{ folgt } C_n(\mathcal{F}) \subseteq C_n(\mathcal{H}) \quad (8.2)$$

81. Welche nicht monotonen Logiken kennen sie?

TMS – Truth Maintenance System teilt sich in JTMS (Justification TMS) und ATMS (Assumption TMS) auf.

Sie verwalten Abhängigkeiten zwischen Aussagen oder Daten und führen bei Vorliegen neuer Information ein maschinelles System von einem Zustand in einen anderen über, wobei die Konsistenz ("truth") erhalten bleibt (sofern überhaupt möglich). Insbesondere veranlassen sie die Rücknahme von Aussagen, die in Konflikt mit der neuen Information stehen.

Ein TMS ist immer mit einer Inferenzkomponente gekoppelt, die klassisch-deduktive Schlussfolgerungen vollzieht.

Bei JTMS wird eine Begründung für alle Regeln benötigt.

Bei ATMS verwaltet und berechnet eine Menge von Annahmen.

Default Logik nach Reiter

Erläuterung siehe 53-55

82. Wie funktioniert ChooseAttribute beim Entscheidungsbaumlernverfahren?

Über den Informationsgehalt (siehe Frage 71) den Informationsgewinn (information gain) und über gain ratio (verbessertes Informationsgewinn bei C4.5) inkludiert split info.

83. Was ist der Beweis durch Widerspruch?

Dieses Theorem besagt, dass sich die semantische Folgerbarkeit auf die Unerfüllbarkeit einer Formel(menge) zurückführen lässt.

Theorem 3.18 (Logisches Folgern durch Widerspruch) *Seien F, G aussagenlogische Formeln oder geschlossene PL1-Formeln. Dann gilt:*

- F ist allgemeingültig gdw. $\neg F$ ist unerfüllbar.
- $F \models G$ gdw. $\neg(F \Rightarrow G)$ ist unerfüllbar.
- $F \models G$ gdw. $F \wedge \neg G$ ist unerfüllbar.

84. Was ist ein CBR?

CBR steht für Case Base Reasoning – fallbasiertes Schließen. Hier besteht die primäre Wissenbasis nicht aus generischen Regeln, sondern aus einer Sammlung von Fällen in denen spezifische frühere Erfahrungen gespeichert sind. Neue Probleme, mit denen das System konfrontiert wird, werden dadurch gelöst, dass der relevanteste Fall aus der Falldatensammlung herausgesucht wird und dessen Lösung in geeigneter Form auf das neue Problem übertragen wird. Hier ist das Schließen also kein regelbasierter, sondern ein erinnerungsbasierter Prozess.

85. Auf welchen Annahmen beruht ein CBR System?

-) Ähnliche Probleme haben ähnliche Lösungen. Lösungen bereits bewältigter Probleme liefern einen guten Ausgangspunkt für die Behandlung eines neuen Problems.

-) Jedes Problem ist anders aber der Typ der Aufgabenstellung wiederholt sich. Wenn man wiederholt mit einem Problemtyp konfrontiert wird, findet man im Allgemeinen leichter und schneller eine Lösung als beim ersten Mal.

„Erinnern ist nützlich, und Erfahrungen dürfen genutzt werden“

86. Was sind die Anwendungsgebiete des fallbasierten Schließen?

CBR ist praktisch in allen Fach- und Lebensbereichen anzufinden. Eines der wichtigsten Beispiele ist die juristische Rechtsprechung, hier wird oft mit vergleichbaren Fällen argumentiert. In der Medizin wird mit jedem neuen Patienten ein neuer Fall assoziiert. Verkaufsagenten im elektronischen Handel verwenden CBR um Kunden entsprechende Produkte anzubieten.

87. Welche Grundtypen des fallbasierten Schließen gibt es?

Problemlösendes CBR: Ziel hier ist die Lösung eines ähnlichen, früheren Problems als Lösung eines neuen Problems vorzuschlagen und ev. anzupassen.

Interpretatives CBR: hier steht nicht die Lösung eines Problems im Mittelpunkt sondern die adäquate und differenzierte Beurteilung einer Situation. Dafür werden ähnliche Fälle aus der Fallbasis herausgesucht und mit der neuen Situation verglichen. Die Unterschiede werden dahingehend untersucht ob diese für die aktuelle Situation relevant sind.

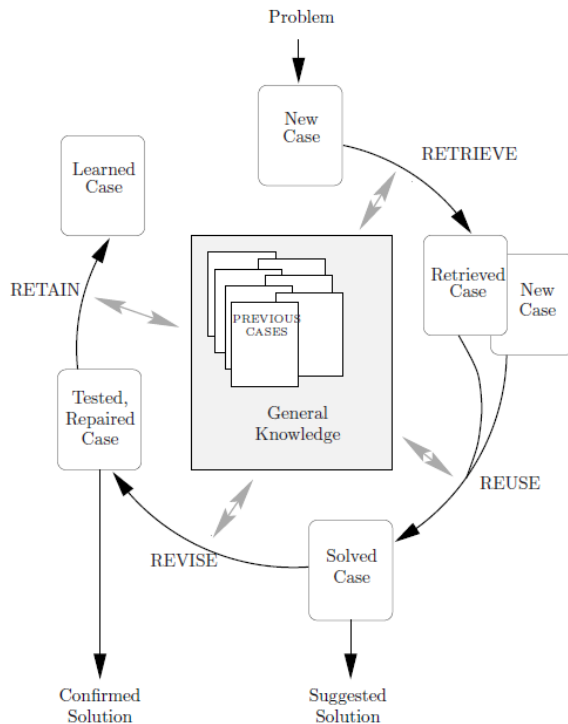
88. Wie funktioniert der CBR Zyklus?

1 Selektierung (Retrieve) des ähnlichen Falls oder Fälle

2 Wiederverwendung (Reuse) des in den gefundenen Fällen gespeicherten Wissens um die Aufgabenstellung zu lösen.

3 Überprüfung (Revise) der vorgeschlagenen Lösung

4 Aufnahme (Retain) der des neuen Falls in die Fallbasis durch Integration.



89. Was ist der Versionsraum?

Sei B eine Menge von Trainingsbeispielen, dann ist die Menge

$$V_B = \{h \in L_C \mid h \text{ ist korrekt und vollständig bzgl. } B\}$$

der Versionsraum bzgl. der Beispielmenge.

Die Grundidee des Versionsraumverfahrens ist es, nicht einzelne Hypothesen auszuwählen und bei späteren Widersprüchen die getroffene Auswahl zurückzunehmen, sondern zu jedem Zeitpunkt die Menge aller bis dahin noch möglichen Hypothesen zu repräsentieren.

90. Wie funktioniert das Versionsraum-Lernverfahren?

Es handelt sich um ein inkrementelles Lernverfahren. B ist die Menge der bisher verarbeiteten Beispiele, S und G repräsentieren den Versionsraum V_B . Für jedes neue Beispiel e müssen S und G überprüft und gegebenenfalls angepasst werden. Für eine Hypothese $h \in S \cup G$ die mit e übereinstimmt d.h. $h(e)=1$ falls pos. Bsp. oder $h(e)=0$ falls neg. Bsp. Braucht man nichts zu tun. Wenn h nicht mit e übereinstimmt kann folgendes passieren.

1 e ist für h fälschlicherweise negativ d.h. $h(e)=0$ obwohl e ein pos. Bsp. ist

2 e ist für h fälschlicherweise pos. d.h. $h(e)=1$ obwohl e ein neg. Bsp. ist

Tritt einer dieser Fälle auf gehen wir wie folgt vor:

1 e ist für s falsch pos. \rightarrow s ist zu allgemein – S enthält nur speziellste Hypothesen daher nicht weiter spezialisierbar \rightarrow s muss aus S entfernt werden.

2 e ist für s falsch neg. \rightarrow s ist zu speziell und muss verallgemeinert werden bis e abgedeckt ist

3 e ist für g falsch pos. → g ist zu allgemein und muss spezialisiert werden bis e nicht mehr abgedeckt wird

4 e ist für g falsch neg. → g ist zu speziell und kann daher nicht mehr verallgemeinert werden (weil G nur allgemeine Hypothesen hat) g muss aus G entfernt werden.

```

function VS
  Eingabe: Konzeptlernaufgabe mit den Sprachen  $L_E$  und  $L_C$ 
          und Folge von Trainingsbeispielen
  Ausgabe: Versionenraumrepräsentation aller Konzepte, die vollständig
          und korrekt bzgl. der eingegebenen Beispiele sind

  Initialisiere S zu der Menge der speziellsten Hypothesen aus  $L_C$ 
  Initialisiere G zu der Menge der allgemeinsten Hypothesen aus  $L_C$ 

  for each neues Trainingsbeispiel e do
    • if e ist ein positives Beispiel then
      • entferne aus G alle Hypothesen g mit  $g(e) = 0$ 
      • for each h ∈ S mit  $h(e) = 0$  do
        • entferne h aus S
        • füge zu S alle Hypothesen h' hinzu mit:
          - h' ist minimale Verallgemeinerung von h bzgl. e und
          - es gibt eine Hypothese  $g \in G$  mit  $h' \leq g$ 
      • entferne aus S jede Hypothese, die (echt) allgemeiner als
        eine andere Hypothese in S ist

    • if e ist ein negatives Beispiel then
      • entferne aus S alle Hypothesen s mit  $s(e) = 1$ 
      • for each h ∈ G mit  $h(e) = 1$  do
        • entferne h aus G
        • füge zu G alle Hypothesen h' hinzu mit:
          - h' ist minimale Spezialisierung von h bzgl. e und
          - es gibt eine Hypothese  $s \in S$  mit  $s \leq h'$ 
      • entferne aus G jede Hypothese, die (echt) spezieller als
        eine andere Hypothese in G ist
  
```

91. Wie kann das Versionenraumlernverfahren terminieren?

1 S und oder G sind leer → Versionenraum kollabiert zur leeren Menge → keine konsistente Hypothese für die Trainingsbeispiele aus dem Hypothesenraum L_C .

2 S und G sind identische einelementige Mengen $S = G = \{h\}$. Hypothes h ist das einzige Konzept aus L_C das konsistent bzgl. der Trainingsmenge ist.

3 Alle Bsp. sind bearbeitet, S und G sind nicht leer und enthalten unterschiedliche Hypothesen. Alle Hypothesen im Versionenraum zwischen S und G sind konsistent bzgl. der Trainingsmenge.

92. Was ist die Hamming Ähnlichkeit?

Es handelt sich hier um ein einfaches und grobes Ähnlichkeitsmaß. Die Hamming Distanz zweier Fälle ist definiert durch:

$$\text{Dist}_H(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Stimmen x und y überein ist der Hamming Abstand zwischen ihnen 0, sind die komplett verschieden beträgt der Abstand n – ist also maximal. Das Ähnlichkeitsmaß soll zwischen 0 und 1 sein. Dafür berechnet man:

$$\text{Sim}_H(x, y) = 1 - \frac{\text{dist}_H(x, y)}{n}$$

93. Was ist die gewichtete Hamming Ähnlichkeit?

Um eine Priorisierung von verschiedenen Attributen vornehmen zu können kann man den Attributen verschiedene Gewichte zuordnen. $w_i \geq 0$

$$\text{Sim}_H^w(x, y) = \frac{\sum_{i=1}^n w_i (1 - |x_i - y_i|)}{\sum_{i=1}^n w_i}$$

Der Gewichtungsfaktor w_i gibt an wie stark das i -te Merkmal die Gesamtähnlichkeit beeinflussen soll.

94. Wann ist ein Kalkül vollständig und korrekt?

Zweck eines Kalküls K ist es eine synthetische Ableitungsrelation \vdash zwischen Formeln zu definieren, dies soll die semantische Folgerungsrelation \models möglichst gut nachbilden. Ein Kalkül ist korrekt, wenn alle dadurch definierten Ableitungen auch semantische Folgerungen sind, d.h. wenn für beliebige Formeln F und G gilt:

$F \vdash G$ impliziert $F \models G$

Ein Kalkül ist vollständig, wenn dadurch alle semantischen Folgerungen abgeleitet werden können, d.h. wenn für beliebige Formeln F und G gilt:

$F \models G$ impliziert $F \vdash G$

95. In PL1 ausdrücken:

1. Jeder liebt jemand
2. Es gibt jemand der jeden liebt

1 $\forall x \exists y \text{ lieben}(x, y)$

2 $\exists y \forall x \text{ lieben}(x, y)$

96. Was ist die disjunkte und konjunktive Normalform?

Eine Formel F ist in disjunkter Normalform (DNF) wenn sie von der Form $K_1 \vee K_2 \vee \dots \vee K_n$ ist, wobei K_i die Konjunktionen von Literalen sind.

F ist in konjunktiver Normalform (KNF) wenn sie von der Form $D_1 \wedge D_2 \wedge \dots \wedge D_n$ ist, wobei D_i Disjunktionen von Literalen sind.

97. Was ist das Resolutionskalkül?

Im Resolutionskalkül geht es nicht direkt um das Ableiten einer Formel G aus einer Formelmenge F . Die zu beweisende Formel G muss negiert werden und zu F hinzugefügt werden. Anschließend muss die Unerfüllbarkeit von $F \wedge \neg G$ gezeigt werden. Damit ist $F \models G$ (die logische Folgerung) bewiesen. Das Resolutionskalkül arbeitet auf Formeln in Klauselform und hat nur eine Inferenzregel die Resolutionsregel.

98. Wofür kann KDD verwendet werden?

KDD steht für Knowledge Discovery in Database.

KDD kann angewendet werden für:

Produktion: Information zur Prozessoptimierung

Ökonomie: Warenkorbanalyse, Supermarkt-Design

Bankenwesen: Bonitätsanalyse, Aufdeckung Kreditkartenmissbrauch

E-Commerce: Kundenprofile

Geologie: Erdbebenanalyse

Wissenschaft: Informationsgewinnung bei wissenschaftlichen Phänomenen

Internet: Suchen nach relevanten Informationen

99. Welche Schritte umfasst der KDD Prozess?

1 Hintergrundwissen und Zielsetzung: Ziele des KDD definieren, Bereichsspezifisches Wissen zur Verfügung stellen

2 Datenauswahl: eine Menge von Daten auswählen

3 Datenbereinigung: Ausreißer müssen beseitigt werden, Datenbasis muss bereinigt werden

4 Datenreduktion und -Projektion: Datenmenge wird komprimiert

5 Modellfunktionalität: Feststellung des Zwecks des Data-Mining

6 Verfahrensauswahl: Bestimmung des Verfahren

7 Data-Mining: eigentlicher Data-Mining-Prozess

8 Interpretation: gewonnene Information wird aufbereitet

100. Was bedeutet Syntax und Semantik einer Wissensrepräsentationssprache?

Die Syntax einer Wissensrepräsentationssprache legt fest wie die Sätze dieser Sprache festgelegt sind.

Die Semantik bestimmt auf welche Begriffe der zu repräsentierenden Welt sich die Sätze beziehen. Erst mit einer solchen semantischen Beziehung erlangen die Sätze der Repräsentationssprache eine Bedeutung, die z.B. festlegt ob ein Satz der Sprache in einer gegebenen Welt wahr oder nicht wahr ist.

101. Welche Beziehungen gelten für logische Formeln?

Eine Formel ist allgemeingültig wenn sie nicht falsifizierbar ist.

Eine Formel ist dann unerfüllbar wenn sie nicht erfüllbar ist.

Eine Formel ist genau dann allgemeingültig wenn ihre Negation unerfüllbar ist.

Eine Formel ist genau dann erfüllbar wenn ihre Negation falsifizierbar ist.

102. Was ist das Deduktionstheorem?

$F \models G$ gdw. $\models F \Rightarrow G$

$\models F \Rightarrow G$ drückt die Allgemeingültigkeit der Implikation aus im Gegensatz $F \Rightarrow G$ ist nur eine syntaktische Formel

103. Was sind die Entscheidbarkeitsresultate?

M ist entscheidbar gdw. es gibt einen Algorithmus der für jedes x angibt ob $x \in M$ oder nicht gilt

M ist unentscheidbar gdw. M ist nicht entscheidbar

M ist semi-entscheidbar gdw. es gibt einen Algorithmus der für x angibt das $x \in M$ gilt.