Universität Bielefeld

4 Spezielle Schlussverfahren

11. Vorlesung: Probabilistisches Schließen; Nichtmonotones Schließen

Methoden der Künstlichen Intelligenz

Ipke Wachsmuth WS 2012/2013

Unsicherheit, Unvollständigkeit

4. Wie lassen sich auch bei unsicherem und unvollständigem Wissen Schlüsse ziehen?

- In der klassischen Logik kann nur ausgedrückt werden, dass eine Aussage wahr oder falsch ist, jedoch nicht, dass man eine Aussage für wahrscheinlich hält oder über ihr Zutreffen nichts weiß.
  - In Anwendungen kommen solche Fälle häufig vor (Unsicherheit bzw. Unvollständigkeit des Wissens).
  - Wissensrepräsentation und Schlussverfahren müssen entsprechend erweitert werden.
- Hauptansätze:
  - probabilistisches Schließen
  - nichtmonotones Schließen

Knappe Quelle hierzu:

- Puppe, Einf. in Expertensysteme, Kap. 7 und 8

Probabilistisches Schließen

- In angewandten Aufgabenstellungen besteht oft Unsicherheit über das Gelten von Bedingungen und Folgerungen (vor allem im Bereich der Diagnostik).
- Basis des probabilistischen Schließens ist die Bewertung von Aussagen mit einer „Wahrscheinlichkeit“, die den Grad ihrer Unsicherheit repräsentiert.
  - Unsicherheiten können aus Statistiken abgeleitet sein (Wahrscheinlichkeiten im engeren Sinn) oder von Experten geschätzt (Evidenzen oder Sicherheitsfaktoren).

Beispiel: Regel mit Evidenzen

```
Wenn   1. der Typ der Infektion Meningitis ist
 und   2. keine Labordaten verfügbar sind
 und   3. der Typ der Meningitis bakteroid ist
 und   4. der Patient älter als 17 Jahre ist
 und   5. der Patient Alkoholiker ist
dann   gibt es Evidenz für E. Coli (0.2) und
       Diplococcus (0.3)
```

„Certainty Factors“

(Evidenzen / Sicherheitsfaktoren)

aus dem MYCIN-System zur Diagnose bakterieller Infektionen des Blutes und des Hirns

Diagnosebewertung: Basisalgorithmus

1. Starte mit der A-priori-Wahrscheinlichkeit aller Diagnosen D_i
2. Für jedes Symptom: Modifiziere die Wahrscheinlichkeit aller Diagnosen gemäß den Symptom-Diagnose-Wahrscheinlichkeiten
3. Selektiere die wahrscheinlichste Diagnose

setzt voraus: (am besten auf Grundlage von Statistiken)

- Abschätzungen der Symptom-Diagnose-Wahrscheinlichkeiten aller relevanten Symptom-Diagnose-Paare
- Abschätzungen der symptomunabhängigen A-priori-Wahrscheinlichkeiten der Diagnosen

Das Theorem von Bayes

erlaubt Berechnung der wahrscheinlichsten Diagnose D_i unter Annahme der Symptome $S_1 \ldots S_m$ aus:

- den A-priori-Wahrscheinlichkeiten $P(D_i)$ einer Menge von n Diagnosen und
- den bedingten Wahrscheinlichkeiten $P(S_j/D_i)$ (der statistischen Häufigkeit des Auftretens der Symptome bei geg. Diagnose D_i)

(zu berechnen für jedes i)

$$P(D_i / S_1 \,\&\ldots\&\, S_m) = \frac{P(D_i) * P(S_1 / D_i) *\ldots* P(S_m / D_i)}{\sum_{j=1}^{n} P(D_j) * P(S_1 / D_j) *\ldots* P(S_m / D_j)}$$

Einfaches Beispiel zu Bayes

Angenommen Statistiken zeigen folgende Wahrscheinlichkeiten:

- P(Bronchitis) = 0.05
- P(Husten) = 0.2
- P(Husten/Bronchitis) = 0.8 (d.h. wenn jemand Bronchitis hat, beobachtet man in 80% aller Fälle auch Husten)

Dann gilt hier (vereinfachte Formel):

$$P(\text{Bronchitis/Husten}) = P(\text{Bronchitis}) * \frac{P(\text{Husten/Bronchitis})}{P(\text{Husten})}$$

$$= 0.05 * \frac{0.8}{0.2} = 0.2$$

d.h. die Wahrscheinlichkeit, dass ein Patient mit Husten Bronchitis hat, ist 20% – viermal so hoch wie die A-priori-Wahrscheinlichkeit von Bronchitis.

Voraussetzungen für Bayes

1. Symptome dürfen nur von der Diagnose abhängen und müssen untereinander unabhängig sein
2. Vollständigkeit der Diagnosemenge
3. wechselseitiger Ausschluss der Diagnosen
4. fehlerfreie und vollständige Statistiken zur Gewinnung der A-priori-Wahrscheinlichkeiten der Diagnosen und der bedingten Symptom-Diagnose-Wahrscheinlichkeiten
5. Konstanz der Wahrscheinlichkeiten

Diese Voraussetzungen sind in Diagnose-Anwendungsbereichen in der Regel verletzt!

Verringerung von Fehlerquellen

- Berücksichtigung von Symptomkombinationen durch spezielle Regeln (z.B. beim MYCIN-System)
- Abschwächung der Bedingung sich wechselseitig ausschließender Diagnosen (z.B. INTERNIST-System)
- Angabe von Wahrscheinlichkeitsintervallen anstelle von -werten (Dempster-Shafer-Theorie)
- Etablierung allgemeiner Grobdiagnosen, die mit zusätzlichem Wissen verfeinert werden
- Getrennte Bewertung von positiver und negativer Evidenz; diagnose-bezogene Verrechnungsschemata

Aber: Verrechnungsschemata nicht immer theoretisch fundiert oder Voraussetzungen für fundierte Verfahren verletzt!

probabilistisch *vs.* nichtmonoton

- Probabilistisches Schließen ist in vielen Fällen problematisch.
- Besser deshalb: So viel „hartes“ Schließen wie möglich und so wenig probabilistisches Schließen wie nötig.
- Wo es möglich ist: Repräsentation der Unsicherheiten als Ausnahmen von Regeln (→ nichtmonotones Schließen)

Probabilistisches Schließen (z.B.)

$A \rightarrow B$ (70%)

Die Regel gilt in 70% der Fälle.

Nichtmonotones Schließen (z.B.)

$A \rightarrow B$ UNLESS C

Die Regel gilt, außer wenn eine der Ausnahmen aus C zutrifft.

Problem der Unvollständigkeit

Bei Vorliegen unvollständiger Daten kann es vorkommen, dass bereits vorgenommene Ableitungen aufgrund neuer Information zurückgenommen werden müssen, z.B.:

- Erwartungswerte (Defaults) werden überschrieben
- Bekanntwerden von Ausnahmen von Regeln
- Bekanntwerden gegenteiliger Evidenz für etablierte Schlüsse
- Korrektur oder zeitliche Änderung von Eingabedaten
- Einführung neuer Annahmen in der Wissensbasis

Bekanntwerden einer Ausnahme

Beispiel (PROLOG mit Closed-World Assumption):

„Closed-World Assumption“ (CWA) : Wenn F nicht beweisbar ist, wird $\neg F$ angenommen.

```
fliegt(X) :- vogel(X), not abnorm(X).
abnorm(X) :- pinguin(X).
vogel(tweety).
>fliegt(tweety)? ==> yes
```

Durch das neue Faktum wird klar, dass eine Ausnahme vorliegt.

```
fliegt(X) :- vogel(X), not abnorm(X).
abnorm(X) :- pinguin(X).
vogel(tweety).
pinguin(tweety).
>fliegt(tweety)? ==> no
```

Einführung neuer Annahmen

hier am Beispiel eines erweiterten Frame-Systems:

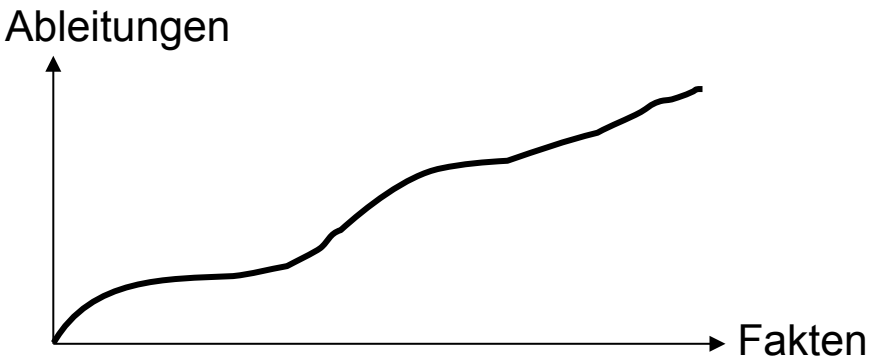
```
(defframe auto
  (slots (sitze 5)
         (zylinder 4)
         (räder 4)))

(defframe sportwagen
  (supers auto)
  (slots (sitze 2)
         (preis hoch) ))

(definstance flitzer of
             sportwagen)
```

```
>Hat flitzer 4 zylinder?
>JA
```

Erweiterung des Frames `sportwagen`:

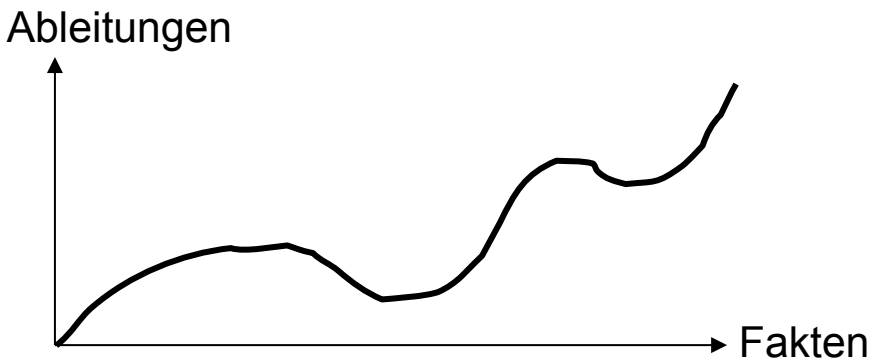
```
(defframe sportwagen
  (supers auto)
  (slots (sitze 2)
         (preis hoch)
         (zylinder 6) ))

>Hat flitzer 4 zylinder?
>NEIN
```

Nichtmonotones Schließen



Monotones Schließen:
Kenntnis weiter Fakten erhält alle bisherigen Ableitungen

PL-1 ist monoton!



Nichtmonotones Schließen:
Kenntnis weiter Fakten kann vorher abgeleitete Fakten unableitbar machen

in PL-1 nicht modellierbar!

(Konklusionen plausibel, nicht korrekt)

Ansätze zur Formalisierung

(alle haben wesentliche Effizienzprobleme)

- Modale nichtmonotone Logiken
  - logische Sprache wird um sog. Modaloperatoren erweitert
- Default-Logiken
  - Einführung nichtmonotoner Inferenzregeln
- Circumscription
  - aus einer Prämissenmenge nichtmonoton ableitbare Formeln werden definiert als aus einer Obermenge der Prämissen *monoton* ableitbare Formeln

hier nicht weiter betrachtet
(außer ggf. in den Übungen)

bei Interesse siehe:

- Görz et al., Kap. „Nicht-monotones Schließen“

Pragmatischer Ansatz: „Belief Revision“

Man nimmt an, dass eine Formel gilt („konsistent ist“), solange ihre Negation nicht bewiesen wurde:

- „unbeweisbar“ wird ersetzt durch „bisher nicht bewiesen“
- Ableitbarkeit relativ zum Problemlösungszustand definiert
- Abhängigkeiten der einzelnen Schlüsse von Vorbedingungen werden protokolliert
- falls sich später herausstellt, dass eine Annahme falsch war: Schluss zurücknehmen

–> Truth-Maintenance-Systeme
(erheblich effizienter als nichtmonotone Logiken)

Problem für TMS (auch: RMS)

- Von einer revidierten Schlussfolgerung können weitere Schlussfolgerungen abhängen, die vor der Revision schon abgeleitet waren.
  - Die Rücknahme eines Elements in diesem Netzwerk von Schlussfolgerungen kann eine Kettenreaktion auslösen.
- Aufgabe der „Belief Revision“ (durch Truth-Maintenance-Systeme oder Reasoning-Maintenance-Systeme):
  - Herstellung eines Zustandes, der entstanden wäre, wenn man das neue oder geänderte Faktum gleich von Anfang an berücksichtigt hätte.

Revision von Schlussfolgerungen

Einfachstes Verfahren:

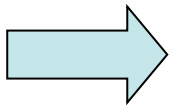
- Neuberechnung aller Schlussfolgerungen aus den veränderten Eingabedaten

Verbesserung:

- Abspeichern eines Protokolls aller Schlussfolgerungen und Neuberechnung ab dem Zeitpunkt der ersten Verwendung der kritischen Schlussfolgerungen (Backtracking).

Wesentlich ökonomischer:

- Nur die Schlussfolgerungen korrigieren, die tatsächlich von einer Änderung betroffen sind.



Datenabhängigkeit

Problem der Datenabhängigkeit

Eintrag in DATABASE:	$p \rightarrow q$
späterer Eintrag:	p
Vorwärtsverkettung, Eintrag:	q
Löschen der Voraussetzung:	~~p~~
q ist nicht mehr begründet !	~~q~~

Löschen von Einträgen ==> Löschen von Folgerungen

D.h. das Fortbestehen des Datums q in DATABASE ist abhängig vom Fortbestehen der Voraussetzung p .

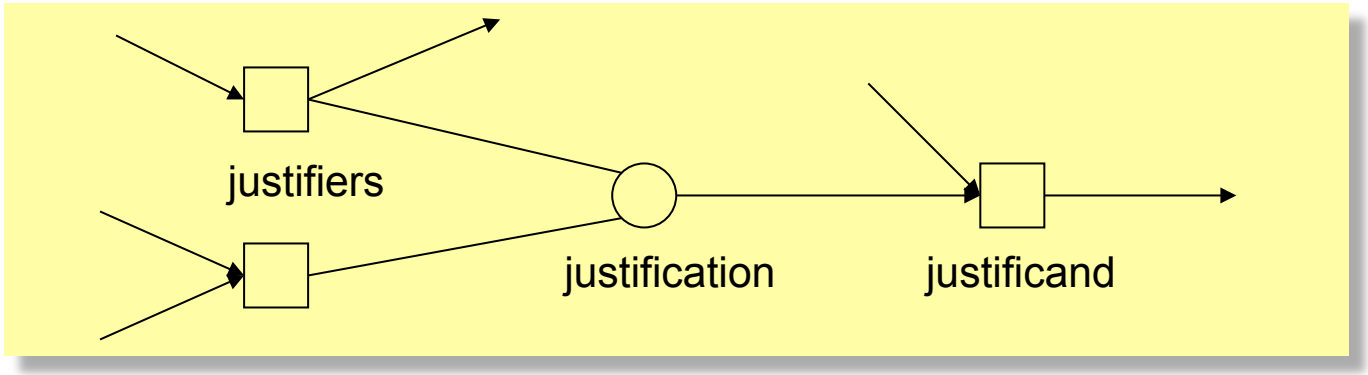
zu Datenabhängigkeiten:

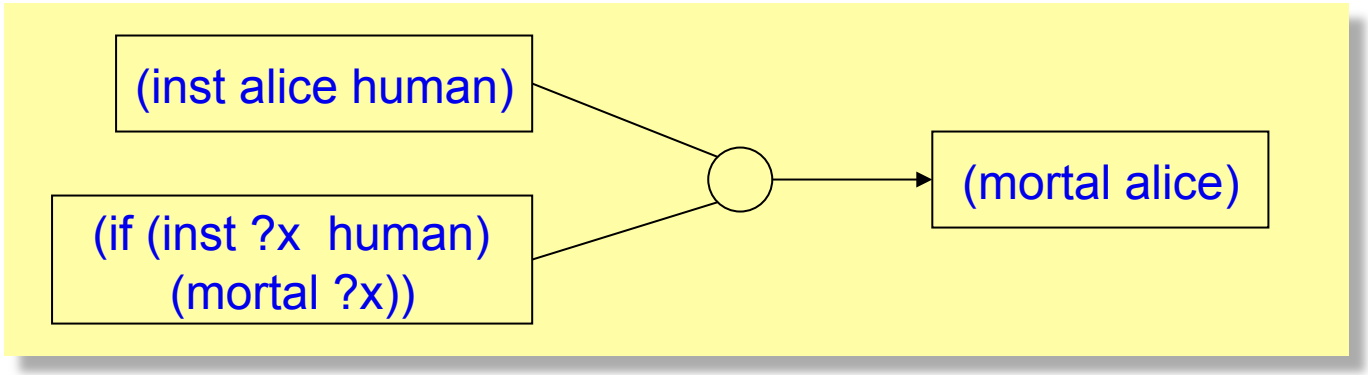
- Charniak & McDermott, Kapitel 7, Seite 411-416

Datenabhängigkeit: Lösungsidee

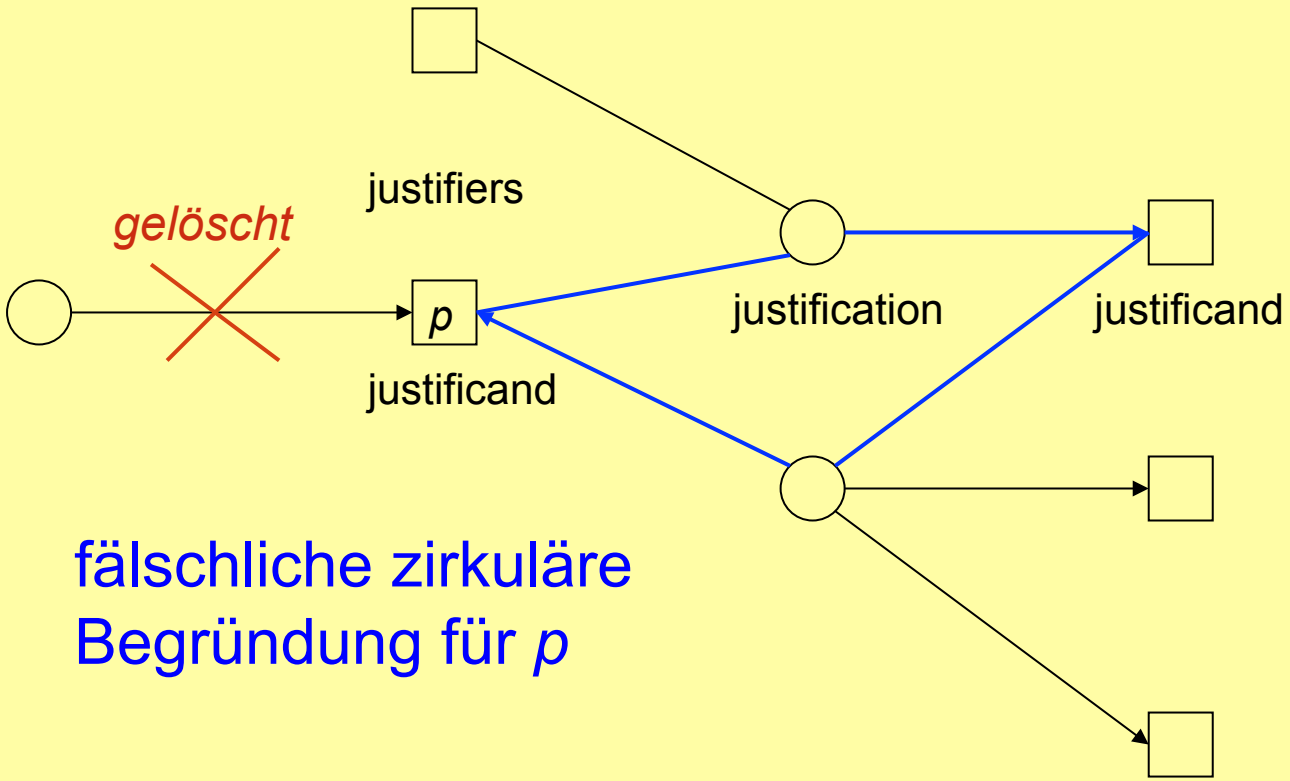
Begründungsnotiz für Einträge (data dependency note)

• Grund (justifier) • Begründung (justification) • Begründetes (justificand)

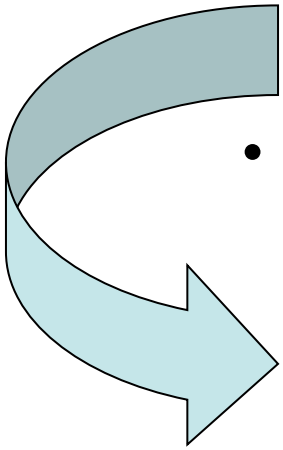




Technische Schwierigkeit



Grundidee für „Belief Revision“



- für jede gemachte Schlussfolgerung ihre Begründung mit abspeichern
- bei Rücknahme eines Faktums (rekursiv) überprüfen, welche Begründungen dadurch ungültig werden

Wichtigste Ansätze:

JTMS (Justification-based TMS)

ATMS (Assumption-based TMS)

– Beim JTMS werden direkte Begründungen, beim ATMS Basisannahmen, die einer Begründung zugrunde liegen (in Form sog. gelabelter Kontexte) abgespeichert.

Basisalgorithmus des JTMS

Eingabe: Änderung eines Faktums

Ausgabe: Propagierung der Änderung mit Herstellung eines konsistenten Zustandes

1. Wenn ein Eingabedatum oder eine Schlussfolgerung sich ändert, überprüfe alle damit verbundenen Begründungen;
2. Wenn eine Begründung ungültig wird, überprüfe, ob die Schlussfolgerung noch weitere Begründungen hat;
3. Wenn eine Schlussfolgerung keine gültigen Begründungen mehr hat, ziehe sie zurück und rufen den Algorithmus rekursiv mit der zurückgezogenen Schlussfolgerung auf, sonst: keine Änderung nötig.

Zur Klarstellung

Ein Truth-Maintenance-System

- ist ein System zur „Wahrheitswartung“: Revision der bisher geglaubten Annahmen („beliefs“)
- speichert und verwaltet Abhängigkeiten zwischen Fakten in einem Netzwerk aus Rechtfertigungen („justifications“)
- kennt nicht den mit den Knoten assoziierten Inhalt
- wird zwischen je zwei Ableitungsschritten eines Inferenz-systems angestoßen (durch „GATEKEEPER“)

Frohe Weihnachten:

- am 8. Januar 2013 geht’s weiter...