

Semantische Netze Strukturierte Repräsentationen

Inheritance Networks

- Konzepte mit Attributen
- is-a und instance-of Relationen

Probleme:

- Konflikte bei Mehrfachvererbung
- Überschreiben von Eigenschaften in Spezialisierungen
(wenn Pinguin ein Vogel ist, der nicht fliegt, wieso nicht Stein als Vogel modellieren, der nicht fliegt, nicht lebt, und keine Federn hat?)

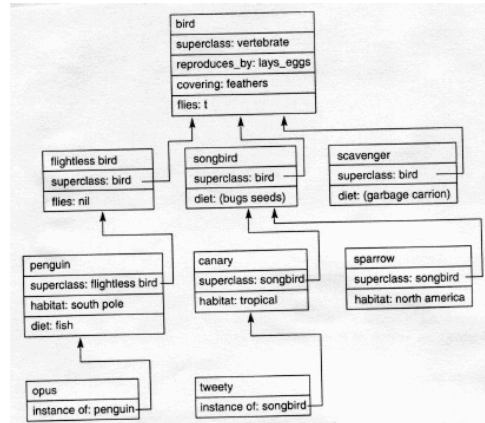
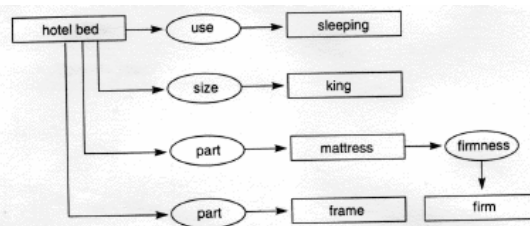


Figure 9.27 Inheritance system description of birds.

Semantische Netze Strukturierte Repräsentationen

Semantische Netze



Frames



Figure 9.23 Conceptual graph and frame descriptions of a hotel bed.

Strukturierte Repräsentationen - Frames

- Erweiterung der Inheritance Networks um *Procedural Attachment*
- *Generische Frames beschreiben Klassen, Instanzen einzelne Objekte*
- *Vererbung, i.a. auch Mehrfachvererbung und Überschreiben von Eigenschaften*
- *Frames haben Slots, jeder Slot hat einen oder mehrere Werte*
- *„Dämonen“-Prozeduren an Slots gebunden um Seiteneffekte bei Slot-Zugriff zu modellieren*
 - **if-added**: aufgerufen, wenn neuer Wert in Slot eingetragen wird
 - **if-removed**: aufgerufen, wenn Wert aus Slot entfernt wird
 - **if-replaced**: aufgerufen, wenn Wert des Slots geändert wird
 - **if-needed**: aufgerufen, wenn fehlender Wert eines Instanz-Frames aus generischen Frame berechnet werden muß. Facette cache bedeutet, daß Wert in Instanz gespeichert wird
 - **if-new**: aufgerufen, wenn neuer Instanz-Frame erzeugt wird
 - **range**: überprüft Wertebereich eines neuen Frames
 - **help**: wird aufgerufen, wenn range-Test Fehler liefert
 - o.ä.

14

Frames Beispiel

```
cylinder ako object with
  height:
    range
      number(new value) and new value > 0
    help
      print("Height must be a positive number")
    if_needed
      ask
    if_removed
      remove volume from this cylinder
    cache
      yes;
  radius:
    range
      number(new value) and new value > 0
    help
      print("Radius must be a positive number")
    if_needed
      ask
    if_removed
      remove cross_section from this cylinder
    cache
      yes;
  cross_section:...
  volume:
    ...
```

15

Frames

Minsky, 1975

- (Generische) Frames repräsentieren stereotype Situationen, z.B. Wohnzimmer, die auf spezifische Situation angepaßt werden

Schank & Abelson, 1977

- Scripts sind spezielle Frames, die typische Verläufe von Ereignissen beschreiben, z.B. Restaurant-Script

1970er / 1980er Jahre

- Frame-Systeme FRL, KRL, ...; auf Allgemeinheit ausgelegt
- Behandlung von Mehrfachvererbung, siehe z.B. Winston, 1993
- KL-ONE und Nachfolger: formale, deklarative Semantik
- Inheritance Networks mit Ausnahmen: formale Semantik z.B. Abb. auf nichtmonotone Logik

CYC-Projekt

- Ziel: Modellierung von Commonsense-Wissen; über 5000 Frames (Stand 1993)

heute

- eher spezialisierte Systeme; in Bielefeld z.B. ERNEST (Bild- und Sprachverstehen), COAR
- prominent: OWL Web Ontology Language (<http://www.w3.org/TR/owl-ref/>)

16

Regel-basierte Systeme (Produktionssysteme)

- Anwendungen von Frames / Semantischen Netze eher im Sprach, Bildverstehen
- Anwendungen von Regel-basierten Systemen eher im Problemlösen

if-then-Regeln

- if C1, C2, ... then A1, A1, ...
- „Deduktive“ Regel-basierte Systeme: if- und then-Komponenten sind Assertionen; then-Komponenten werden zum Arbeitsspeicher hinzugefügt
- „Regel-basierte Reaktionssysteme“: then-Komponenten können beliebige Aktionen sein, z.B. Assertion hinzufügen oder löschen, Aufruf von Prozeduren (Winston, 1993)

Psychologische Plausibilität

- z.B. Newell & Simon, 1972
- „Kognitive Architekturen“ z.B. SOAR, ACT-R

17

Regel-basierte Systeme

Deduktive Regel-basierte Systeme

- Modellierung von Taxonomien

if X is-a sparrow then X is-a bird

- Modellierung von Merkmalen von Konzepten

if X is-a bird then X has-feathers, X can-fly

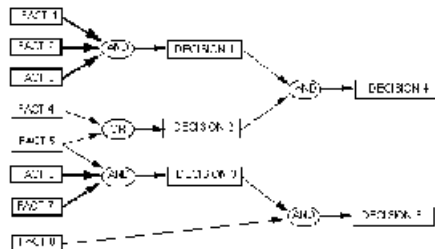
- Regeln zur Klassifikation von Objekten, über die bestimmte Eigenschaften bekannt sind

**if X is-a mammal, X has pointed teeth, X has claws, X has forward-pointing eyes
then X is-a carnivore**

Regel-basierte Systeme

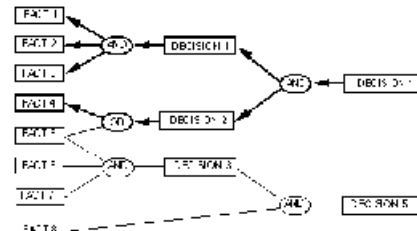
Forward-Chaining

- aus gegebenen Fakten werden durch Vorwärtsanwendung der Regeln Schlußfolgerungen / Aktionen abgeleitet
- z.B. Produktionssysteme



Backward-Chaining

- Aus Hypothese über Entscheidung wird durch Rückwärtsanwendung nach Fakten zur Rechtfertigung der Entscheidung gesucht
- z.B. Prolog



Regel-basierte Systeme (Produktionssysteme)

Komponenten

- Menge von if-then-Regeln;
then-Komponenten Hinzufügen/Löschen von Assertionen, Aufruf von Prozeduren
- Arbeitsspeicher mit temporären Daten
- Inferenzmaschine mit Forward Chaining

Match-Resolve-Act Cycle

```
loop
  match conditions of rules with contents of working memory
  if no rule matches then stop
  resolve conflicts
  act (i.e. perform conclusion part of rule)
end loop
```

20

Regel-basierte Systeme (Bsp. Bagger)

Bagger hilft beim Packen von Einkaufstüten

- check-order-step: überprüft Einkäufe des Kunden; macht Vorschläge zum Kauf weiterer Waren
- bag-large-items-step: große Waren einpacken, große Flaschen zuerst
- bag-medium-item-step: dann mittelgroße Waren packen; Gefrorenes in Plastiktüten
- bag-small-items-step: kleine Waren Einpacken, wo Platz ist

```
B1 if    step is check-order
      potato chips are to bagged
      there is no Pepsi to be bagged
then
  ask customer whether he would
  like a bottle of Pepsi

B2 if    step is check-order
      then
      delete step is check-
      order
      add step is bag-large-
      items
```

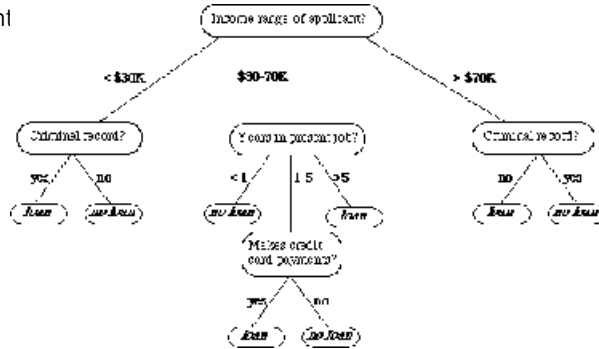
Konfliktlösungsstrategien (falls mehrere Regeln anwendbar sind)

- Rule-Ordering: Regeln in Liste angeordnet, erste anwendbare Regel feuert
- Weitere, z.B. Specificity-Ordering, Recency-Ordering, Size-Ordering, .

21

Decision Trees

auch bekannt als „Discrimination Nets“
 „hartverdrahtete“ Ent



Baum

- innere Knoten repräsentieren Auswahlmöglichkeiten
- Blätter repräsentieren Klassifikation oder Entscheidungen
 z.B. Klassifikation von Objekten mit bekannten Merkmalen

Decision Trees

- Induktives Lernen von Decision Trees aus Beispielen
 (z.B. ID3, erfolgreich in Expertensystemen eingesetzt)

Ziel: Klassifikation von Objekten anhand von Attributen

Verfahren: Bewertung des Informationsgewinns eines Attributes.

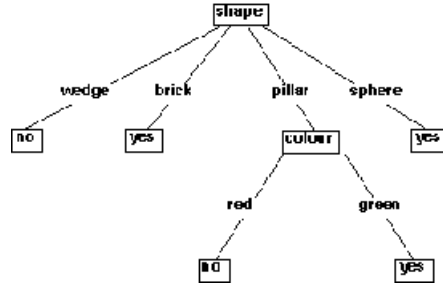
- Wie gut kann Objekt ohne/mit Attribut klassifiziert werden?
- Welches Attribut hat größten Informationsgewinn?

Decision Trees

- ID3
 - Gegeben: Menge von Beispielen in Attribut-Wert Repräsentation
 - Ziel: Entscheidungsbaum zur Klassifikation neuer Objekte

Positive Beispiele
 medium blue brick
 small red sphere
 large green pillar
 large green sphere

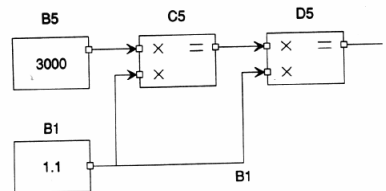
Negative Beispiele
 small red wedge
 large red wedge
 large red pillar



Constraint-Netze

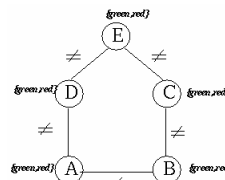
Arithmetische Constraint-Netze

- Variablen-Knoten enthalten numerische Werte
- Operationsknoten erzwingen arithmetische Constraints zw. Terminalwerten
- Anwendung z.B. in Spreadsheets



Symbolische Constraint-Netze

- Knoten repräsentieren Variablen (mit endlicher Anzahl von Werten)
- Kanten repräsentieren (binäre) Constraints
- Anwendungen u.a. Sprachverstehen, temporales Schlußfolgern uvm.



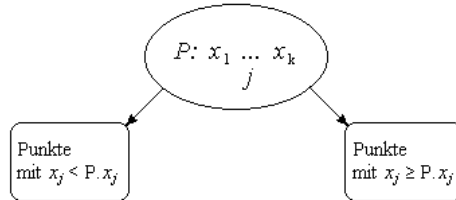
➤ „Constraintpropagierung“

K-d tree

- Es sollen Punkte P_i des k -dimensionalen Raums gespeichert werden.
- Punkte sind Koordinaten (x_1, x_2, \dots, x_k) gegeben.
- Durchführung effizienter Suchverfahren.

Jeder Knoten des Baums enthält

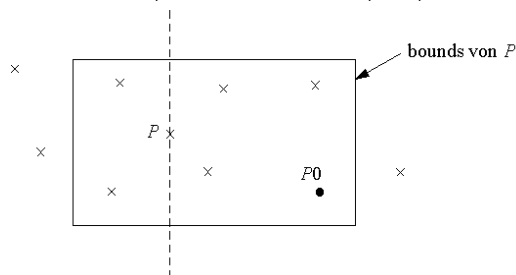
- einen Punkt P
- eine Dimensionsangabe j
- 2 Zeiger auf Nachfolger



Der linke Unterbaum enthält alle Punkte deren x -Komponente $< P.x_j$ ist, der rechte Unterbaum alle anderen.

K-D tree

- Bounds: Jeder Knoten des k -d-trees ist die Wurzel eines Teilbaums (aller Punkte eines bestimmten Quaders).
- Für den Gesamtbaum sind die bounds:
 $[Min\ x_1 .. Max\ x_1, Min\ x_2 .. Max\ x_2, \dots, Min\ x_k .. Max\ x_k]$



- Bei jedem Knoten wird die j -te Komponente der bounds gemäß der j -ten Komponente des in ihm enthaltenen Punktes unterteilt (der Quader wird in j -Richtung zerschnitten).

K-d tree

- **Suchen nach einem Punkt P_0 :**

Das ist das normale Suchen; die Dimensionsangabe j des gerade erreichten Knotens gibt an, welche Komponenten von P_0 und P man vergleichen muß, um zu erfahren, in welchem Unterbaum weiter zu suchen ist.

- **Suche alle Punkte, die in einem Quader Q liegen (region search):**

Das Gebiet (bounds), für das der Teilbaum mit dem aktuellen Knoten als Wurzel zuständig ist, muß man bei der Suche mitschleppen und immer auf den neuesten Stand bringen. Falls das P des aktuellen Knotens in Q liegt, dann P ausgeben. Für jeden Unterbaum prüfen, ob sein Gebiet mit Q nicht disjunkt ist; wenn das der Fall ist, wird er durchsucht.

- ...

K-d tree

- **Suche die m nächsten Punkte zu einem gegebenen P_0 :**

Man benutzt Hilfsliste *mbest* mit m nächsten der bisher angetroffenen Punkte des Baums (Am Anfang sind das weniger als m) und *mdist* als Distanz von P_0 zu dem schlechtesten dieser Punkte.

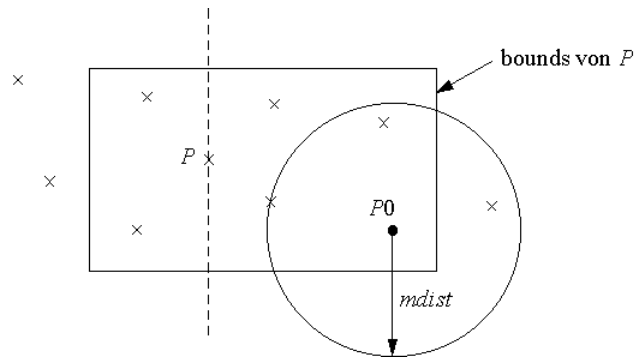
Suchen als rekursive Prozedur, die bei Erreichen des Knotens mit Punkt P folgendes durchführt:

1. Falls Distanz zwischen P und P_0 kleiner als *mdist* ist, dann wird P in *mbest* eingeordnet und *mdist* korrigiert.
2. Durch einen rekursiven Aufruf sucht man zuerst im *besseren* Unterbaum, das ist der auf der Seite von P_0 . Man findet unter Umständen bessere Punkte für *mbest*, die auch *mdist* verkleinern.
3. Nach Rückkehr erst prüfen, ob die Untersuchung des *schlechteren* Unterbaums noch etwas bringen kann, **bounds overlap ball test**: Nur wenn die *bounds* dieses Unterbaums sich mit dem *ball*, eine k -dimensionale Kugel um P_0 mit dem Radius *mdist*, überschneidet, kann er noch Punkte enthalten, die um weniger als *mdist* von P_0 entfernt sind. Nur dann wird auch dieser Unterbaum durchsucht; die Skizze zeigt einen Fall, in dem der linke Unterbaum keine brauchbaren Punkte mehr enthalten kann.

K-d tree

•Suche die m nächsten Punkte zu einem gegebenen $P0$:

- die Skizze zeigt einen Fall, in dem der linke Unterbaum keine brauchbaren Punkte mehr enthalten kann.



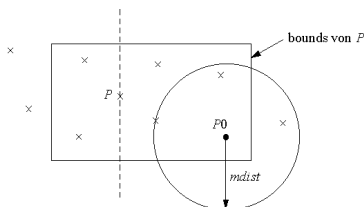
30

K-d tree

...

- Prüfung, ob die Suche jetzt vollständig beendet werden kann, d.h. ob auf dem Rückweg von P zur Wurzel des Gesamtbaums das Untersuchen von eventuell noch nicht bearbeiteten schlechteren Unterbäumen unterbleiben kann,

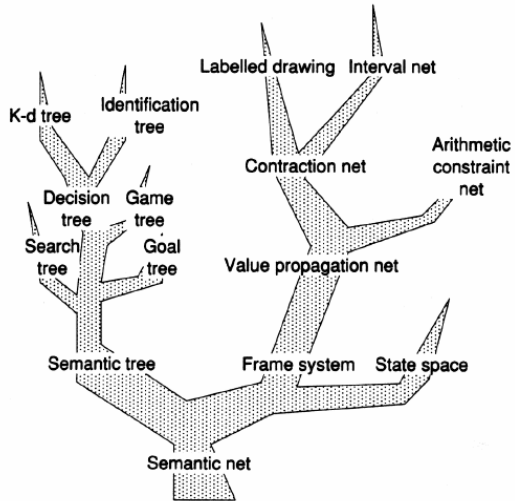
ball within bounds test: Wenn der *ball* mit Radius $mdist$ vollständig in den *bounds* von P enthalten ist, dann können die außerhalb dieses Quaders liegenden Punkte (alle Punkte der vorher erwähnten schlechteren Teilbäume sind solche) nichts mehr bringen; -> Suche endgültig abbrechen!



Die Skizze zeigt das Gegenteil, der ganz rechts liegende Punkt gehört unter Umständen zu einem schlechteren Unterbaum einer höheren Ebene und wird später noch in m best eingeordnet.

31

Übersicht: Netzwerkartige Repräsentationen



(Winston 1993)