

3 Wissensrepräsentation und Modellierung

5. Vorlesung: Repräsentationssprachen

Methoden der Künstlichen Intelligenz
Bernhard Jung WS 2001/2002

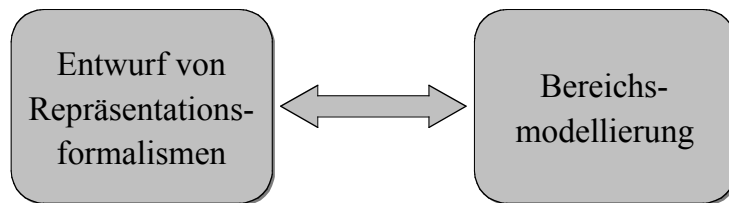


Übersicht

◆ Aspekte der Wissensrepräsentation

- ◆ Conceptual Primitives Ansatz
- ◆ Frame-Sprachen
- ◆ KL-ONE-artige Sprachen

Wissensrepräsentation: Aufgaben



- ◆ Repräsentationsformalismen
 - Beschreibungsmittel für symbolisch darzustellendes Wissen
 - automatische aufgabengesteuerte Berechnung von Inferenzen über der Menge kodierten Wissens
- ◆ Modellierung bestimmter Fachgebiete und Weltbereiche („Domänen“)

Repräsentationsformalismen

Klassifikation nach Mylopoulos & Levesque (1984)

- ◆ Logische Repräsentationsformalismen
deklarativ; z.B. PL1, modale/temporale u.a. Erweiterungen von PL1, PROLOG
- ◆ Netzwerk-artige Repräsentationsformalismen
Graphen: Knoten = Konzepte oder Objekte, Kanten = Relationen / Assoziationen
z.B. semantische Netze wie Conceptual Dependency, Conceptual Graphs
- ◆ Strukturierte Repräsentationsformalismen
Erweiterung der Netzwerkrepräsentationen:
Knoten als komplexe Datenstrukturen mit benannten Slots,
Slotwerte: numerisch, symbolisch, Zeiger auf andere Knoten; procedural attachment
z.B. Frames, Scripts
- ◆ Prozedurale Repräsentationsformalismen
Wissen in Form von Instruktionen
z.B. if..then..Regeln (->Produktionssysteme)

Adäquatheit von WR-Sprachen

- ◆ Epistemologische Adäquatheit (McCarthy & Hayes, 1969)
 - Ausdrucksstärke: können alle relevanten Fakten und Regeln repräsentiert werden?
 - z.B. PL1 sehr ausdrucksstark
- ◆ Heuristische Adäquatheit (McCarthy & Hayes, 1969)
 - Sind die durchzuführenden Inferenzen überhaupt bzw. mit den zu Verfügung stehenden Ressourcen möglich? Sind Zwischenschritte des Lösungswegs im Formalismus repräsentierbar?
 - z.B. Theorembeweisen in PL1 unentscheidbar, in Praxis aufwendig
- ◆ Ergonomische Adäquatheit
 - Wie verständlich sind die Repräsentationen?
 - Klarheit und Präzision, leichte Veränderbarkeit (Woods, 1987)
 - z.B. Modellierung in PL1 i.a. mühsam
- ◆ Kognitive Adäquatheit
 - z.B. Ursprünge semantischer Netze und Frames auch kognitiv motiviert

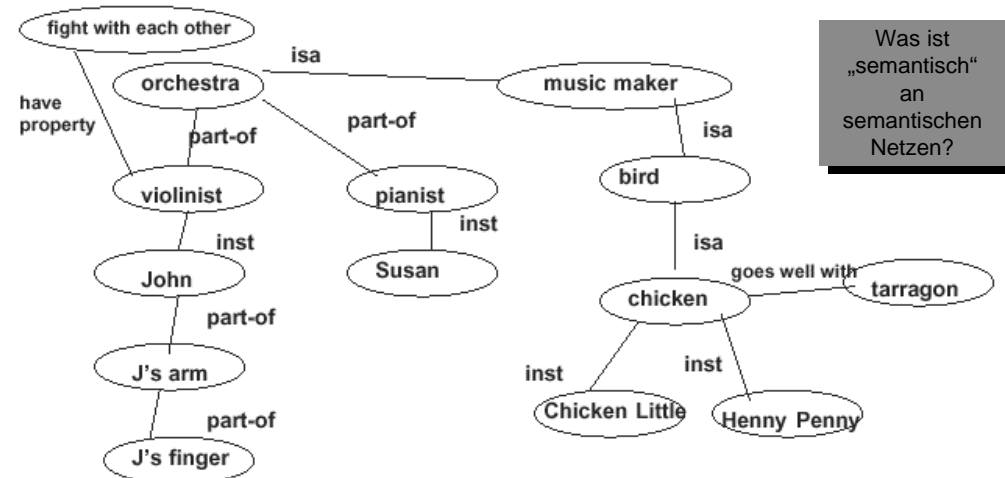
Semantik (I)

Bedeutung der sprachlichen Konstrukte von WR-Formalisten kann über verschiedene Semantikkonzeptionen festgelegt werden.

- ◆ Operationelle Semantik
 - Bedeutung durch Algorithmen definiert, die über Sprachkonstrukten operieren
 - Typisch für frühe Semantische Netze und Frames, z.B. Babylon
Die Semantik des Objekt-orientierten Teils einer Babylon-Wissensbasis ist wesentlich durch die Vererbungsstrategie bestimmt. Sie ist operationell definiert und beruht auf der Semantik des Lisp-Maschinen-Flavor-Systems (Chrastaller et al., 1989)
 - Nachteil: Semantik ist systemabhängig, damit nicht unmittelbar übertragbar
- ◆ Äquivalenzsemantik
 - Übersetzung in WR-Formalismus mit bekannter Semantik
 - z.B. Frames -> PL1 (Hayes, 1980)

Semantik (II)

- ◆ Deklarative Semantik
 - Syntaktische Strukturen der WR werden über eine Interpretationsfunktion zu Elementen von abstrakten Strukturen in Bezug gesetzt
 - z.B. Semantik der PL1
 - z.B. Semantik von KL-ONE-artigen Sprachen (Konzept -> Menge von Individuen)
 - Vorteile
 - „Konsistenz“ einer Wissensbasis kann formal gefaßt werden
 - Berechnung von Subsumtionsbeziehungen zwischen Konzepten (Extension des einen Konzepts ist Teilmenge der Extension des anderen Konzepts)
 - „Korrektheit“ und „Vollständigkeit“ von Inferenzverfahren, z.B. zur Berechnung von Subsumtionsbeziehungen, sind jetzt definiert
 - Nachteil
 - Semantik ausschließlich extensional, intensionale Aspekte nicht erfaßt
z.B. gleiche Bedeutung „Morgenstern“, „Abendstern“ und „Venus“
z.B. gleiche Bedeutung von „rundes Viereck“ und „Einhorn“

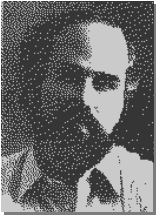


Was ist „semantisch“ an semantischen Netzen?

Was ist hier modelliert?

- John`s Finger ist Teil eines Orchesters? John kämpft mit sich selbst?
- Brathühner machen Musik? Henny Penny paßt gut zu Estragon (einem Gewürz)?
- Problem früher semantischer Netze: Keine Semantik für Konzepte und Relationen!

Vokabular für Repräsentationen



Roger Schank

- „The initial choice of [primitives] to represent knowledge in a new domain is necessarily ad hoc. We make an initial, tentative commitment to a new set ... In the process of codifying new knowledge, using the knowledge in computing programs ... we modify, change or even replace our original choice ... We believe that this method rapidly converges upon a ... set of basic units that organizes the knowledge of the domain ...“



Pat Hayes

- „The symptom of having got [the vocabulary] wrong is that it seems hard to say anything very useful about the concepts one has proposed It is easier, fortunately, to recognize when one [has got it right]: assertions suggest themselves faster than one can write them down.“

[Quelle: Charniak & McDermott, S.324]

Übersicht

- Aspekte der Wissensrepräsentation

Conceptual Primitives Ansatz

- Frame-Sprachen
- KL-ONE-artige Sprachen

„Conceptual Primitives“-Ansatz

- Idee: die Repräsentation wird aus fest vorgegebenen Grundeinheiten („konzeptuellen Primitiven“) komponiert
- Als Beispiel wird hier betrachtet: Roger Schanks Conceptual Dependency (CD)-Notation
- entwickelt für die semantische Repräsentation in Programmen zur Verarbeitung natürlicher Sprache
- Zielrichtung: Alltagswelt („everyday human actions“)
- Basiseinheit: „event“ (oder auch „conceptualization“)

Beispiel: ptrans (physischer Transfer)

event-token event-type agent action
 (event ev-29 (do john-22 (**ptrans** book-3 dunham becton)))
 (speed ev-29 slow)
 (vehicle ev-29 golfcart-103)

denotiert „physischen Transfer“: ein Handlungsprimitiv/Primitive Act!

in slot-assertion-Notation:
 (inst ev-29 **ptrans**)
 (agent ev-29 john-22)
 (patient ev-29 book-3)
 (source ev-29 dunham)
 (dest ev-29 becton)
 (speed ev-29 slow)
 (vehicle ev-29 golfcart-103)

- hypothetisches Ereignis; wird konkret mit Assertierung von (happen ev-29)
- (actual-event *token type*) bedeutet (and (event *token type*) (happen *token*))

„Primitive Acts“ (nach R. Schank)

(ptrans object place-1 place-2)
ein Objekt wird physisch transferiert

(move body-part dest)
ein Körperteil wird wohin bewegt

(atrans object owner-1 owner-2)
Kontrollübergabe (abstrakter Transfer)

(propel object place-1 place-2)
ein Objekt wird „abgestoßen“

(grasp object)
ein Objekt wird gegriffen

(ingest object orifice)

(expel object orifice)
etwas wird eingenommen / exkretiert

ein Beschreibungs-Repertoire für „everyday human actions“

(mtrans info place-1 place-2)
Informationstransfer (mental); dabei ist *info* eine Formel oder ein Term

(mbuild formula)
der Actor nimmt *formula* zu den beliefs

(speak sounds)
z.B. einen string von Phonemen (eine Möglichkeit des mtransens)

(attend sensor-type object)
ein Objekt wird perzipiert, z.B. (attend eye book-23)

Beispiele zu mtrans

```
(do john-22 (mtrans (name-of murderer-23)
                    (head john-22)
                    (head police-chief-7)))
```

„John übermittelt den Namen des Mörders an den Polizei-Chef.“

```
(actual-event ev-279
  (do jack-84
    (mtrans (actual-event ev-303
                  (do janet-50
                    (ptrans kite-3
                          house-4
                          store-5)))
            (head jack-84)
            (head sally-3))))
```

„Jack erzählte Sally, daß Janet den Drachen in den Laden gebracht hat.“

Im ersten Fall wird ein Symbol und im zweiten Fall eine Formel transferiert.

Story von John und dem Löwen

“John was walking through the woods.
A lion appeared, hungry and in a rage.
John killed it.”



“John was walking through the woods.
A lion appeared, hungry and in a rage.
John killed it.”

```
(actual-event ev-1 (do john-22 (ptrans startplace-1
                                (part middle woods-2))))
(precede ev-1 ev-2)
(actual-state ev-4 (loc lion-3 (near john-22)))
(precede ev-4 ev-2)
(actual-event ev-2
  (do john-22 (mtrans (actual-state ev-4 (loc lion-3 (near john-22)))
                    eye
                    (head john-22))))
(actual-event ev-3 (do john-22 (attend eye lion-3)))
(instr ev-2 ev-3)
(actual-event ev-5 (hungry lion-3 very))
(actual-event ev-6 (angry lion-3 very))
(same-time ev-5 ev-6)
(actual-event ev-7 (do john-22 ...))
(actual-event ev-8 (become (dead lion-3)))
(result ev-7 ev-8)
(reason ev-2 ev-7)
```

Die Geschichte in CD-Repräsentation

Bemerkungen zu „Primitives“

- ◆ Die Idee der konzeptuellen Primitive – obwohl „nett“ – hat sich für größere Domänen als zu beschränkt erwiesen (das „scaling-up“-Problem).
- ◆ Es fehlt die Möglichkeit, bei Bedarf weitere Konzepte zur Differenzierung einzuführen (das geht z.B. in KL-ONE).
- ◆ Allerdings sind konzeptuelle Primitive für den Zweck der Mensch-Maschine-Kommunikation interessant, da die Anzahl der „primitive acts“ dort überschaubar ist.

Übersicht

- ◆ Aspekte der Wissensrepräsentation
- ◆ Conceptual Primitives Ansatz
- ◆ **Frame-Sprachen**
- ◆ KL-ONE-artige Sprachen

Kernideen von Frames

(bzw. allgemein von objektzentrierten Repräsentationen)

- ◆ Vererbungshierarchien
- ◆ zugeordnete Prozeduren
- ◆ Erwartungswerte (Defaults)
- ◆ Verwendungsaspekte
 - a) Eigenschaftszuschreibung
 - b) Klassifikation anhand von Eigenschaften

Frames - Beispiel

Objekt	Eigenschaften	Werte
Säugetier :	ist_ein :	Wirbeltier
	Vermehrung :	lebendgebärend

Objekt	Eigenschaften	Werte
Elefant :	ist_ein :	Säugetier
	Farbe :	grau
	hat :	Rüssel
	Größe :	groß
	Lebensraum :	Boden

Zugeordnete Prozeduren

- ◆ (kleine) Programme, die einer Eigenschaft eines Objektes zugeordnet sind und bei jedem Lese- oder Schreibzugriff auf dessen Wert ausgeführt werden, etwa um:
 - aus vorhandenen Parametern neue zu berechnen, z.B. Alter aus Geburtsdatum
 - Werteänderungen zu überwachen, z.B. bei jedem Schreibzugriff den neuen Wert auf dem Bildschirm anzeigen (active values)
- ◆ Zugeordnete Prozeduren können in einer Objekthierarchie (ähnlich wie Werte) vererbt werden.

Erwartungswerte (Defaults)

- ◆ Vorbelegungen von Werten, die normalerweise, aber nicht immer stimmen und daher durch konkrete Informationen überschrieben werden können.
- ◆ Beispiel:
 - (Vogel Flugfähigkeit ja) (typischerweise können Vögel fliegen)
 - (Pinguin ist_ein Vogel)
 - (Pinguin Flugfähigkeit nein)
- ◆ ACHTUNG: Der Einsatz von Defaults kann erfordern, daß Schlußfolgerungen zurückgezogen werden müssen.

Beispielsprache FRL (1977)

(Frame Representation Language)

- ◆ Frame (für jeweils ein Objekt)
- ◆ Slots (für Eigenschaften)
- ◆ Facetten (für „Wertarten“)
 - \$value ; tatsächlicher Wert des Slots
 - \$require ; Wertebereich für \$value
 - \$default ; Erwartungswert (Standardannahme)
 - \$if-added ; auszuführen bei Eintrag eines Werts für \$value
 - \$if-removed ; auszuführen bei Löschen eines Werts für \$value
 - \$if-needed ; auszuführen falls \$value-Wert benötigt wird

„Werte“ können auch (Lisp-) Prozeduren sein: procedural attachment

◆ Werte

Vererbungshierarchien in FRL

```

Frame Elefant
  Slot      AKO
  Facette   $value
  Wert      Säugetier

Frame Clyde
  Slot      AKO
  Facette   $value
  Wert      Elefant
    
```

Objekt	Eigenschaften	Werte
Elefant :	ist_ein :	Säugetier
	Farbe :	grau
	hat :	...
	Größe :	...
	Lebensraum :	...

Objekt	Eigenschaften	Werte
Clyde :	ist_ein :	Elefant
	Farbe :	...
	hat :	...
	Größe :	...
	Lebensraum :	...

- ACHTUNG: Eigentlich zu unterscheiden:
- generische Objekte (Objektklassen)
 - individuelle Objekte (Instanzen)
- Manche Frame-Sprachen unterscheiden daher ISA- und AKO-Slots!



Problematik von Defaults

◆ FRAGE:

Was ist groß, grau, hat einen Rüssel und lebt auf Bäumen?

◆ ANTWORT:

Ein Elefant –

die Bäume sind eine Abweichung bezüglich des Lebensraums typischer Elefanten.

(Brachmann, 1985)

Kritik an Frame-Sprachen/FRL

- ◆ Alle Zusicherungen können durch Ausnahmen überschrieben werden.
- ◆ Eine automatische Klassifikation neuer Objekte anhand ihrer Eigenschaften ist daher nicht möglich.

◆ Radikale Konsequenz:

Erwartungswerte (Defaults) und Abweichungen bei der Objekt-Definition verbieten.

» Grundidee der KL-ONE-Sprachen

Übersicht

- ◆ Aspekte der Wissensrepräsentation
- ◆ Conceptual Primitives Ansatz
- ◆ Frame-Sprachen
- ◆ KL-ONE-artige Sprachen

Ideen von KL-ONE-Sprachen

(ab 1980: KL-ONE, KRYPTON, KL-TWO, NIKL, BACK, MESON, SB-ONE ...)

- ◆ in Tradition von Frames / semantischen Netzen
- ◆ Trennung zwischen terminologischem und sonstigem Wissen („T-Box“, „A-Box“)
- ◆ T-Box:
 - Vererbungshierarchien mit präziser Semantik
 - definierte Konzepte; keine Defaults
 - automatische Klassifikation neuer Objekte anhand ihrer Eigenschaften möglich
- ◆ A-Box:
 - Zuschreibung sonstiger (zufälliger) Eigenschaften für einzelne Individuen
- ◆ Logische Rekonstruktion; formale Semantik

T-Box:
Terminology

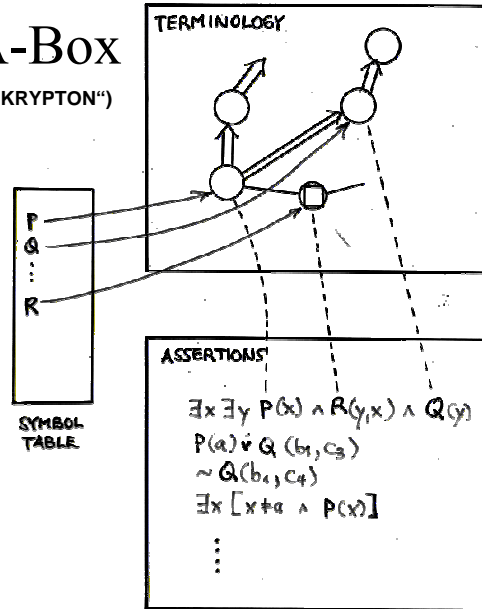
A-Box:
Assertions

T-Box, A-Box (KL-ONE-SPRACHE „KRYPTON“)

T-Box: Taxonomie strukturierter Terme

A-Box: PL-Formeln, deren Prädikate in der T-Box definiert sind

Symboltabelle: enthält die Namen der T-Box-Terme



Konzepte & Rollen in KL-ONE

Kernidee der KL-ONE-Sprachen zur Ermöglichung automatischer Klassifikation: strikte Trennung von Aussagen über Objekte in definierende und zufällige Eigenschaften.

- definierende Eigenschaften → T-Box
- zufällige Eigenschaften → A-Box

- ◆ Objekte heißen in KL-ONE Konzepte.
- ◆ Definierende Eigenschaften heißen Rollen und müssen genau präzisieren, inwiefern ein Konzept spezieller als seine Vorgänger ist.

Wichtigste Sprachstrukturen

◆ Definition von Konzepten

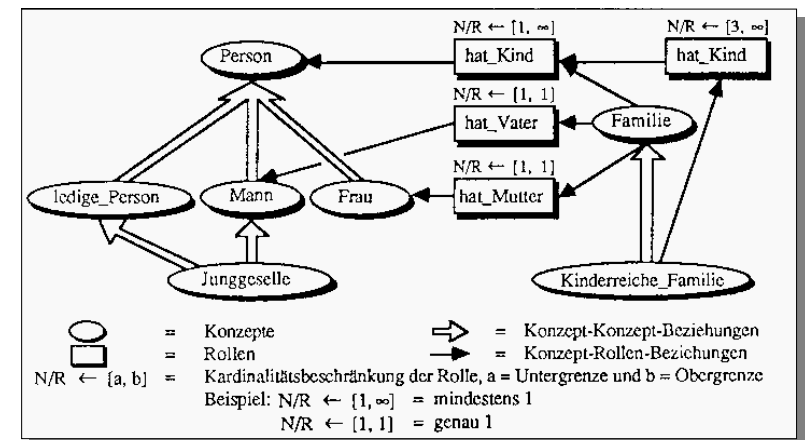
- Konjunktion Conj-Generic („Conjunction“)
- Wertbeschränkung VR-Generic („Value Restriction“)
- Zahlbeschränkung NR-Generic („Number Restriction“)
- Spezialisierung Prim-Generic

◆ Definition von Rollen

- Wert-Differenzierung VR-Diffrole
- Verkettung Role-Chain
- Spezialisierung Prim-Rolle

Beispiel zu KL-ONE

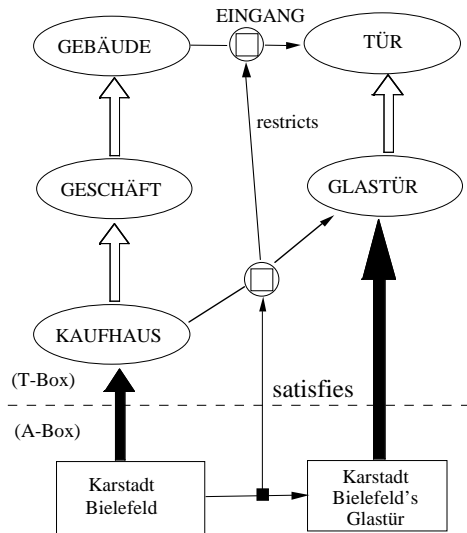
[Quelle: Puppe]



Graphische Darstellung der Konzeptdefinitionen „Junggeselle“ (mit Konjunktion) und „Kinderreiche_Familie“ (mit Zahl-/Kardinalitätsbeschränkung der „hat_Kind“-Rolle)

KL-ONE Rollenrestriktion

KL-ONE:
Erfüllung von Rollen in der A-Box



hier:
Wertbeschränkung

Für KAUFHAUS-Gebäude wird der Wert der Rolle (definierenden Eigenschaft) EINGANG auf Instanzen des Konzepts GLASTÜR restringiert.

Automatischer Classifier

besteht aus

- ◆ einem Prädikat subsumes („umfaßt“):
 - $(K \text{ subsumes } K')$ ist wahr, wenn alle Rollen eines Konzepts K die Rollen eines Konzepts K' subsumieren, das heißt: Die Wertbeschränkung und Kardinalitätsbeschränkung aller Rollen von K sind umfassender als die der Rollen von K'
- ◆ einem Suchalgorithmus, der das spezifischste Konzept SK findet, das das einzuordnende Konzept EK noch subsumiert
 - das heißt: $(SK \text{ subsumes } EK)$ wahr, aber $(SK' \text{ subsumes } EK)$ falsch für jedes Unterkonzept SK' von SK .

Bemerkungen zum Classifier

- ◆ Bei T-Box-Sprachen mit Kardinalitätsbeschränkung kann der Classifier exponentiellen Rechenaufwand erfordern.
- ◆ Der Classifier erfordert strenge Einschränkungen in der T-Box-Sprache (keine überschreibbaren Defaults).
 - Beliebige, auch der T-Box widersprechende Zusicherungen in der A-Box möglich (vom Classifier nicht berücksichtigt),
 - können z.B. in KRYPTON mit einem Theorembeweiser für die Prädikatenlogik erster Stufe verarbeitet werden.
- ◆ Für viele Anwendungen sind die für den Classifier bezahlten Einschränkungen bei der Konzeptdefinition zu rigide.

Beispiel aus L-LILOG (IBM 1987)

(KL-ONE-artige Repräsentationssprache für textverstehende Systeme)

Projekt LILOG (1986-1991)
„Linguistische und logische Methoden für das maschinelle Verstehen des Deutschen“

```
forall X : MARKIERUNG , Y : WEG , Z : MENSCH
  (( bez_objekt ( dom = X , ran = Y )
    and
    folgen ( ag = Z , obj = X )
  )
  impl
  folgen ( ag = Z , obj = Y )
)
```

„An dieser Stelle finden wir eine Hinweistafel des Vogesenklubs und die Markierung des roten Rechtecks auf weißem Grund, der wir vorerst folgen.“

„Wenn eine Markierung X einen Weg Y markiert und jemand Z folgt der Markierung X , dann folgt er Z auch dem Weg Y .“

Repräsentationswerkzeuge

(KEE, ART, GoldWorks, Babylon, NexpertObject, ProKappa, Twice ...)

- ◆ „hybrid“: mehrere Wissensrepräsentationssprachen
 - ◆ darauf abgestimmte elementare Inferenzmechanismen
 - ◆ Integration der verschiedenen Komponenten
 - ◆ fensterorientierte, grafikfähige Oberflächen
-
- um die 90er Jahre (last century) von kommerziellen Anbietern bereitgestellt, inzwischen jedoch abgelöst durch objekt-orientierte Programmierumgebungen und sog. case-tools...

Leseempfehlungen

zu conceptual primitives:

- ◆ Charniak & McDermott, Kapitel 6, Seite 326-332

zu FRL und KL-ONE:

- ◆ Puppe: Einführung in Expertensysteme, Kap.5