Attribut-Werte-Paare

- Eine Eigenschaft kann beschrieben werden durch ein Paar [a,w]. Dabei bezeichnet a das Attribut und w den konkreten Wert aus dem Wertebereich W_a des Attributs.
- Die Eigenschaften eines Objekts werden insgesamt durch eine Menge $\{ [a_1,w_1] ,..., [a_n,w_n] \}$ beschrieben.
- Bei festgelegter Reihenfolge der Attribute kann das Objekt durch das n-Tupel $[w_1, ..., w_n] \in W_1 \times ... \times W_n$ beschrieben werden.

Fragen zum Nachdenken: Wie könnte man das Fehlen eines Attribut-Wertes interpretieren?

Relationen

- Eine n-stellige Relationen ist definierbar als
  - Teilmenge $R \subseteq W_1 \times ... \times W_n$ bzw.
  - Prädikat $W_1 \times ... \times W_n \rightarrow$ {wahr,falsch}
- Endliche Relationen können als Tabellen dargestellt werden.
- Relationen können z.B.
  - Eigenschaften von Objekten (z.B. Datenbank) oder
  - Beziehungen zwischen Objekten
  - Beschränkungen von Kombinationen (Constraints)
- beschreiben

Fragen zum Nachdenken: Wie kann man die Einträge aus einer Relation identifizieren? Wie findet kann man die Einträge mit bestimmten Werten aus einer Datenbank (z.B. bestimmte Fakten in Prolog)?

Relationen

- Relationen können kombiniert werden z.B.
  - Mengentheoretisch (Durchschnitt, Komplement, Projektion...)
  - Logisch (Konjunktion,Quantifizierung, ...)
- Funktionen werden kombiniert z.B. durch Einsetzung.

Fragen zum Nachdenken: Wie werden Relationen durch die Klauseln definiert (z.B. bei mengentheoretischer Betrachtung).

Funktionen

- Eine n-stellige Funktion W_1 x ...x $W_n \rightarrow W$ kann aufgefasst werden als n+1 stellige Relation $R \subseteq W_1$ x ...x W_n x W mit $[w_1, ..., w_n, w] \in R \leftrightarrow f(w_1, ..., w_n) = w$
- Im PK1 werden Funktionen syntaktisch durch Terme beschrieben.Sie können Argumente von Relationen oder Funktionen sein.
- In Prolog können Funktionen durch Relationen definiert werden. Sie erlauben auch das Erfragen von Argumentwerten.
- Es gibt eingebaute Funktionen z.B. für Arithmetik.

Fragen zum Nachdenken: Können Relationen auch Argumente von Funktionen sein? Welche Typen unterscheidet Prolog?

Arithmetik

- Man kann in Prolog die primitiv-rekursiven Funktionen gemäß Schema definieren.
- Dabei kann nach beliebigen Argumenten gefragt werden (logische Sicht).
- Aus Effizienzgründen gibt es eine eingebaute Arithmetik, die aber instantiierte Eingangsparameter verlangt.
- Es gibt dazu entsprechende Funktionen und Vergleichsprädikate.
- Operatoren können in Prolog in verschiedener Weise deklariert werden (Stellung, Priorität).

Fragen zum Nachdenken: Welchen Zahlbereich betrachtet die Prolog-Arithmetik?

Rekursion, Standard-Prolog

- Prolog-Prädikate können rekursiv definiert werden.
- Entsprechend der Verarbeitungsstrategie des Interpreters spielt die Reihenfolge eine wesentliche Rolle.
- Die im Interpreter für Standardprolog begründete „prozedurale Semantik“ kann in verschiedenen Situationen von der eigentlich gewollten „deklarativen Semantik“ (logische Sicht) abweichen.

Fragen zum Nachdenken: Wann kann es Schwierigkeiten (welche?) bei „ungeschickter Reihenfolge“ in einer rekursiven Definition geben? Warum nimmt man die Unterschiede zwischen deklarativer und prozeduraler Semantik in Kauf. Wie kann man zeigen, was der Prolog-Interpreter tatsächlich leistet?

Cut: Eingriff in den Beweisverlauf

- Das Prädikat !/0 (cut) verändert die Suchstrategie durch Löschen gewisser Backtrack-Punkten. Dadurch werden die entsprechenden alternativen Beweisversuche verhindert.
- Grüner Cut: keine Veränderung der Semantik.
- Roter Cut: veränderte Semantik.
- Mit Cut kann die Effizienz erhöht werden.

Fragen zum Nachdenken: Wie kann der Cut im Interpreter realisiert werden?

Closed World Assumption

- Wann gibt Prolog keine positive Antwort?
- Annahme einer abgeschlossenen Welt: *Nur was ich **beweisen kann**, ist wahr.*
- Das heißt für Prolog: Antwort „no“ erfolgt, wenn alle Beweisversuche für Q tatsächlich fehlgeschlagen sind („Negation by finite failure“).
- Man kann einen Operator not einführen: not(goal) gelingt, wenn goal misslingt. Er kann implementiert werden mittels cut und fail.

Fragen zum Nachdenken: Vergleichen Sie die frühere Diskussion zur CWA. Wieweit entspricht „not“ der Negation?

Syntax von Prolog

- Die Prolog-Syntax wird im wesentlichen über Strukturen (Terme) definiert. Prädikate und komplexere Argumente sind syntaktisch von gleicher Form. Das erlaubt die Analyse und Synthese von Prädikaten in Prolog-Programmen.
- Elementare Terme (atomare Typen) werden anhand ihrer Schreibweise unterschieden.

Fragen zum Nachdenken: Gibt es Typ-Beschränkungen für die Prolog-Prädikate? Welche Vorteile haben Typ-Beschränkungen in Programmiersprachen?

Listen

- Listen können als spezielle Strukturen mit vorgegebenem Funktor „.“ dargestellt werden. Daneben sind Schreibweisen mit „[“ und „]“ zugelassen.
- Für Listen sind spezielle Prädikate definiert („Datenstruktur“).

Fragen zum Nachdenken: Müssen díe Argumente für append/3 instantiiert sein?