Einführung in die Computerlinguistik – Endliche Automaten und Transduktoren in Prolog

Dozentin: Wiebke Petersen

Foliensatz 8

Prolog: Bausteine

- Fakten: drücken aus, was bedingungslos wahr ist (bzw. als wahr deklariert wird) in der Domäne.
  `human(sokrates).`
- Regeln: stellen den Bezug zwischen Fakten durch logische Implikationen her.
  `mortal(X) :- human(X).`
  - Kopf: linke Regelseite.
  - Körper: rechte Regelseite.
  - Klausel: Aussage – Regel oder Fakt.
  - Prädikat:Sammlung von Klauseln mit gleichen Köpfen (Fakten können als Regeln "ohne Bedingung" gelesen werden.
- Wissensbasis: Menge von Fakten und Regeln.
- Anfragen: die Prolog-Inferenzmaschine versucht, Anfragen aus der Information der Wissensbasis deduktiv abzuleiten.
  `?- mortal(sokrates).`

Prolog: Syntax

- Fakten: `fact.`
- Regeln: `head :- body.`
- Konjunktion in Regeln: `head :- info1 , info2.`
- Atome beginnen mit Kleinbuchstaben
- Variablen beginnen mit Großbuchstaben (Vorsicht: alles was großgeschrieben wird, ist automatisch eine Variable)

Übung: Schreiben Sie eine Prologtheorie, über die Personen Hans, Marie, Otto, Lisa und Klaus. Hans und Marie sind die Eltern von Otto. Otto und Lisa sind die Eltern von Klaus. Verwenden sie Regeln der folgenden Form, um Anfragen nach dem Vater und der Mutter einer Person zu ermöglichen:
`father(X,Y) :- parent(X,Y), male(X).`

Übungsmaterial: `http://www.learnprolognow.org/`
Auf der Homepage von Herrn Rumpf finden Sie sehr viel Material zu Prolog.

Listen in Prolog

- Listen sind rekursive Datenstrukturen:
  1. die leere Liste ist eine Liste
  2. ein komplexer Term ist eine Liste, wenn er aus zwei Teilen besteht, wovon der erste ein Term (first), und der zweite eine Liste (rest).
- `[mary|[john|[alex|[tom|[]]]]]`
- einfachere Notation: `[mary,john,alex,tom]`
- Übung: Schreiben Sie ein Prädikat `member/2`, das prüft, ob ein Term ein Element einer Liste ist.

Algorithmus für endliche Automaten

```
function D-RECOGNIZE (tape, machine) returns accept or reject
  index ← Beginning of tape
  current-state ← Initial state of machine
  loop
    if End of input has been reached then
      if current-state is an accept state then
        return accept
      else
        return reject
    elsif transition-table [current-state, tape[index]] is empty then
      return reject
   else
     current-state ← transition-table [current-state, tape[index]]
     index ← index + 1
end
```

Algorithmus für endliche Automaten

function D-RECOGNIZE (*tape, machine*) **returns** accept or reject
  index ← Beginning of tape
  current-state ← Initial state of machine
  loop
    if End of input has been reached **then**
      if current-state is an accept state **then**
        return accept
      else
        return reject
    elsif *transition-table [current-state, tape[index]]* is empty **then**
      return reject
   else
     current-state ← *transition-table [current-state, tape[index]]*
     index ← *index* + 1
end

```
% Finite state automaton.

fsa(Tape):-
    initial(S),
    fsa(Tape,S).

fsa([],S):- final(S).

fsa([H|T],S):-
    trans_tab(S,H,NS),
    fsa(T,NS).

% FSA transition table:
% trans_tab/3
% trans_tab(State, Input, New State)

trans_tab(1,a,1).
trans_tab(1,b,2).
trans_tab(2,a,2).

initial(1).
final(2).
```

Beispielprogramme (Homepage)

- `fsa.pl` Endlicher Automat, der die Sprache a^*ba^* erkennt. Aufruf mit `fsa(Eingabeliste).` (Bsp. `fsa([a,b,b,b,a]).`).
- `fst.pl` Endlicher Transduktor, der in Wörtern über dem Alphabet $\{a, b\}$ alle a's durch b's und alle b's durch a's ersetzt. Aufruf mit `fst(Eingabeliste,Ausgabelist).` (Bsp. `fsa([a,b,b],[b,a,a]).`, oder `fsa(L,[b,a,a]).` oder `fsa([a,b,b],L).`). Das Programm erlaubt weder ε zu schreiben, noch ε zu lesen. Nichtdeterministische Transduktoren sind aber möglich. Erweitern Sie den Transduktor zum Beispiel um den Übergang `trans_tab_fst(1,a,c,1).`.
- `fst_2.pl` Endlicher Transduktor von Folie 17 vom Foliensatz 7. Aufruf mit `fst(Eingabeliste,Ausgabeliste).` (Bsp. `fst([f,o,x,;,s,#],L).`). Dieses Programm ist in der Lage auch Transduktoren mit ε-Übergängen zu verarbeiten. Versuchen Sie den Transduktor von Folie 18 von Foliensatz 7 zu programmieren. Sie müssen nur die Faktenbasis ändern.

Hausaufgaben (BN: 1 von 2)

1. Erweitern Sie die Familiendatenbank um mindestens 5 weitere Personen und um die Prädikate `child/2`, `grandmother/2`, `brother/2`.
2. Implementieren Sie Ihren Transduktor aus der Hausaufgabe zu Foliensatz 7 in `fst2.pl`.