

Akzeptoren in PROLOG

Grammatiken

4-Tupel mit $G = (N, T, P, S)$

Variable	Bedeutung/Beispiele
Nichtterminalsymbole/Variablen	<Satz>, <Nebensatz>, <Subjekt>, <Prädikat>
Terminalsymbole	Auto, der, die, Haus, laufen, lief, Textes, Duden
Menge von Produktionsregeln	verbindet man zwei Sätze durch die Zeichenfolge „und“, so erhält man wieder einen Satz. <Satz> \rightarrow <Satz> und <Satz>
Startsymbol	Satz (ein ausgewähltes Nichtterminalsymbol)

Aufgabe (Grammatik einer natürlichen Sprache in PROLOG)

1. Analyse und Erweiterung eines Aussagesatzes (gemeinsame Erarbeitung!)
 - a) Öffnen Sie die Datei „Grammatik.pl“ im SWI-PROLOG-Editor. Analysieren Sie den Inhalt der Datei und beschreiben Sie den Aufbau der abgebildeten Grammatik.
 - b) Konsultieren Sie die Datei durch Drücken der Taste. Damit wird der PROLOG-Quelltext auf Korrektheit geprüft und dem PROLOG-Fragesystem bekannt gemacht.
 - c) Geben Sie im PROLOG-Fragesystem im unteren Fensterbereich hinter „?-“ ein:


```
phrase('Satz', [kuehe, fressen, graeser]).
```

 und beschreiben Sie die Reaktion. Drücken Sie die ;-Taste bis „?-“ erscheint.
 - d) Geben Sie im PROLOG-Fragesystem hinter „?-“ ein: `phrase('Satz',Vorschlag).` und beschreiben Sie die Reaktion. Drücken Sie mehrfach die ;-Taste. Mit der ENTER-Taste können Sie die Suche abbrechen.
 - e) Erweitern Sie die vorhandene Grammatik um das Nichtterminalsymbol `Artikel` und das Terminal `die` sowie eine Regel zur korrekten Einbindung des Artikels. Konsultieren Sie anschließend die Datei erneut und prüfen Sie durch geeignete Fragestellung mit `phrase(,Satz',..., ob die Grammatik korrekt arbeitet.`
 - f) Erweitern Sie die vorhandene Grammatik um die Attribute `schoene`, `grosse` und `saftige`. Binden Sie alles in das Regelwerk ein und testen Sie erneut.
 - g) Erweitern Sie die vorhandene Grammatik um die Verschachtelung der Attribute (Adjektive).

2. Entwickeln Sie eine neue Grammatik zur Bildung einfacher Fragesätze nach dem Schema 'Frage' → 'Fragewort', 'Prädikat', 'Subjekt'.
Nutzen Sie die Terminale wo, wie, was, hund, mensch, erich, trinkt, geht, ist, ein, der, schoene, knackige, kleine, weiche.

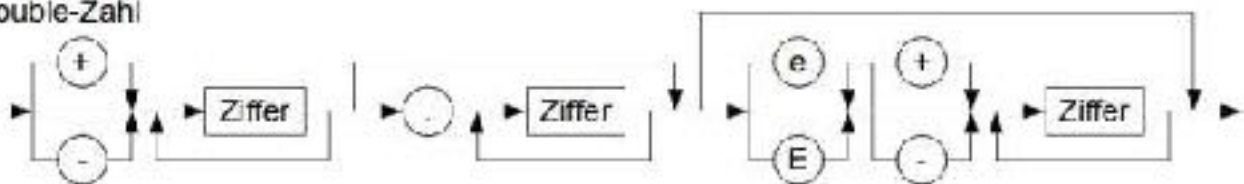
Aufgabe (Grammatik formaler Sprachen in PROLOG)

Hinweise:

Im Gegensatz zur Grammatik einer natürlichen Sprache ist die Grammatik formaler Sprachen rein auf die Syntax beschränkt. Eine Wortgrammatik (Deklination, Konjugation, ...) wie in natürlichen Sprachen existiert in der Informatik nicht. Eine Grammatik enthält mit Nichtterminalsymbolen und Metazeichen Zeichen, die nicht Bestandteil der von ihr festgelegten Sprache sind.

3. Öffnen Sie die Datei „07_Arithmetik.pl“ im SWI-PROLOG-Editor, konsultieren Sie die Datei und lösen Sie folgende Anfragen/Fragen:
- `phrase('Ausdruck', [a, '+', b]).`
 - `phrase('Ausdruck', [a, '*', b, '/', '(', b, '-', a, ')', '+', b]).`
 - Ist $b/(c-b)/b+a$ ein Ausdruck?
 - Ist $((b+a)*a)/c-d$ ein Ausdruck?
4. Öffnen Sie die Datei „Double-Zahl.pl“ im SWI-PROLOG-Editor und konsultieren Sie die Datei.
- Prüfen Sie ob `-12.34` und `1e4` erkannt werden.
 - Erweitern Sie die Grammatik um die fehlenden Regeln. Nachfolgend zur Erinnerung das Syntaxdiagramm für Java-Double-Zahlen. Das Nichtterminal `Ziffer` wird durch eine der Ziffern `0 ... 9` ersetzt.

Double-Zahl



Deterministischer Endlicher Automat

5-Tupel mit $A = (X, Z, \delta, z_0, Z_E)$

Variable	Bedeutung	Bedingung
X	Eingabealphabet	nichtleere, endliche Menge
Z	Zustandsmenge	nichtleere, endliche Menge
δ	Überföhrungsfunktion	$\delta : X \times Z \rightarrow Z$ jedem Paar (Eingabezeichen, Zustand) wird ein Folgezustand zuordnet
z_0	ist der Anfangszustand	$z_0 \in Z$
Z_E	ist die Menge der Endzustände	$Z_E \subseteq Z$

Sprache des Automaten

Gelangt der Automat in den Akzeptierzustand, so wird die dazu notwendige Eingabefolge als ein Wort des Automaten bezeichnet. Die Menge aller akzeptierten Eingabefolgen, die man Wörter nennt, bezeichnet man als Sprache des Automaten $L(A)$.

$L(A) = \{w \mid w \in X^* \text{ und } \delta^*(w, z_0) \in Z_E\}$, wobei gilt:

- w ist ein Eingabewort über dem Alphabet X ,
- δ^* ist eine Folge von Überföhrungsfunktionen die beginnend im Startzustand z_0 mit dem Eingabewort w den Automaten in einen Endzustand überföhren.

Aufgabe (Lachautomat)

Es ist ein Akzeptor zu konstruieren, der folgendes erkennt: ha!, haha!, hahaha! usw.

- Modellieren Sie diesen Akzeptor an Hand eines Zustandsgraphen.
- Öffnen Sie die Datei „Lachautomat.pl“ konsultieren Sie sie und lösen Sie folgende Anfragen/Fragen:
 - `akzeptiert([h,a,h,a,'!'])`.
 - `akzeptiert([])`.
- Analysieren Sie das Programm.
 - Identifizieren Sie Start- und Endzustand/-zustände sowie die Überföhrungsfunktionen. Kommentieren Sie den Quelltext an den entsprechenden Stellen sinnvoll.
 - Beschreiben Sie den Ablauf des tatsächlichen Prüfens als Kommentar im Quelltext.

Aufgabe (Blumenautomat)

Gegeben ist der folgende Mealy-Automatengraph mit $A = (X, Y, Z, \delta, \lambda, z_0)^1$ eines Blumen-Automaten, der zwei Sorten Blumen anbietet.

8. Öffnen Sie die zugehörige Datei in PROLOG. Machen Sie sich mit dem Aufbau des Programms vertraut. Testen Sie das Programm.
9. Bestimmen Sie das Eingabealphabet X , das Ausgabealphabet Y und die Zustandsmenge Z .
10. Ermitteln Sie den Preis für die Blumen.
11. Beschreiben Sie das Bedienkonzept des Automaten.
12. Zeichnen Sie einen Zustandsgraphen für den Blumenautomat.
13. Welche Erweiterung könnte an dem Programm vorgenommen werden? Implementieren Sie diese.

Aufgabe (Parkscheinautomat)

Auf einem Parkplatz kostet das Parken ohne Zeitbegrenzung 2,00 €. Nach Einwurf der korrekten Geldsumme soll sich eine Schranke öffnen. Der Automat wechselt nicht, gibt zu viel gezahltes Geld nicht zurück und hat keine Möglichkeit des Abbruchs.

14. Öffnen Sie die zugehörige Datei in PROLOG. Machen Sie sich mit dem Aufbau des Programms vertraut. Testen Sie das Programm.
15. Bestimmen Sie das Eingabealphabet X , das Ausgabealphabet Y und die Zustandsmenge Z .
16. Beschreiben Sie das Bedienkonzept des Automaten.
17. Zeichnen Sie einen Zustandsgraphen für den Parkscheinautomat.
18. Welche Erweiterung könnte an dem Programm vorgenommen werden? Implementieren Sie diese

¹ Y = Ausgabealphabet, λ = Ausgabefunktion