

# Einführung in die logische Programmierung

---

## Aufgabenblock V

Die nachfolgenden Aufgaben befassen sich mit dem Einsatz von Definite Clause Grammars (DCGs).

Studieren Sie die zur Verfügung gestellten Programme mit Grammatiken für Aussagenlogik (`parser_aussagenlogik.pl`) und für ein lustiges Fragment der natürlichen Sprache (`parser_fliegen.pl`). Laden Sie diese Dateien in Ihrem Prolog und probieren Sie die vorhandenen Prädikate aus. Sie werden im Code Ideen zur Lösung der Aufgaben finden.

## Aufgabe 1

Das Programm für die Fliegen besitzt noch keine Eingabeschnittstelle. Bauen Sie eine: D.h. erstellen Sie ein Programm, dass natürlichsprachliche Sätze einlesen kann (Sowohl aus Standardeingabe, als alternativ auch aus einer Datei. Letzteres wurde in der Vorlesung nicht gezeigt, somit müssten Sie sich dies selbst erarbeiten) und anschließend den Parsebaum ausgibt (Auf Standardaus, aber auch in eine Datei), bzw. registriert, wenn ein Satz nicht korrekt ist. Das Programm muss mehrere Sätze hinter einander verarbeiten können (auch bei der Dateivariante!).

Wenn Sie Spaß daran haben, dann erweitern Sie doch auch noch die Grammatik! (z.B. so, dass auch der Satz „hinter fliegen fliegen fliegen“ grammatikalisch korrekt ist (und somit einen Parsebaum hat).

## Aufgabe 2

Schreiben Sie ein Programm (sozusagen einen Taschenrechner), das Benutzer auffordert, Berechnungen in den Grundrechenarten vorzugeben, und diese dann anschließend berechnet und das Resultat davon ausgibt. Zahlen können auch Vorzeichen (z.B. „-3“) oder Fraktionen (z.B. „3.14“ haben).

Z.B.:

Bitte Formel eingeben: `31 * 5.3 +12`

Die Antwort: `31*5+12 = 176.3`

Bitte Formel eingeben:

Geben Sie anschließend mittels einer Beispielimplementierung an, wie man weitere mathematische Funktionen in die Sprache einführen kann (natürlich inkl. ihrer Berechnung).

## Aufgabe 3

Schreiben Sie eine Version des 8-ter Puzzles, die DCG verwendet.