

Logik-Programmierung und Prolog für IMP-Wechselnde

Wintersemester 2025/2026

Übungsblatt 11

Abgabe: bis 19. Januar 2026, 13.00 Uhr

Aufgabe 1:

(20 Punkte)

Lesen Sie Kapitel 7 aus dem Buch „Learn Prolog Now!“.

Achtung: Die Bearbeitung der Aufgabe ist unter Beachtung der bekannten Abgabehinweise über Moodle abzugeben! Analog zu früheren Blättern finden Sie die benötigten Dateien auf der Seite zur Prolog-Übung. Machen Sie sich auch mit den neuen Prolog-Modulen *unit_propagation* und *pure_literal* vertraut.

- (a) Importieren Sie im Modul `dp11` Ihrer Abgabe `blatt11.pl` die Prädikate aus den Prolog-Modulen, die Sie für die Lösung der folgenden Teilaufgabe benötigen.
- (b) Wir kodieren Klauselmengen wie gewohnt als Listen von Listen von Literalen. Implementieren Sie das Prädikat `dp11/1`, sodass eine Anfrage

`?- dp11(KM).`

für eine Klauselmenge `KM` genau dann erfolgreich ist, wenn die Klauselmenge erfüllbar ist.

Beispielsweise sollte die Anfrage für die Klauselmenge

```
KM = [[x1,~x5,~x6,x7], [~x1,x2,~x5], [~x1,~x2,~x3,~x5,~x6],  
      [x1,x2,~x4,x7], [~x4,~x6,~x7], [x3,~x5,x7], [x3,~x4,~x5],  
      [x5,~x6], [x5,x4,~x8], [x1,x3,x5,x6,x7], [~x7,x8],  
      [~x6,~x7,~x8]]
```

erfolgreich sein. Es macht hierbei nichts, wenn die Antwort `true.` durch das Backtracking mehrfach ausgegeben werden kann. Für die Klauselmenge

```
KM = [[~r,t,w], [~r,~s,~w], [~r,~t], [~q,s,t], [~q,r,~s],  
      [r,s,w], [r,~t,~w], [q,u], [s,~u,~w], [q,w], [q,~s,~u]]
```

sollte dieselbe Anfrage jedoch scheitern.

Hinweise: Implementieren Sie dazu den *DPLL-Algorithmus*, wie er auf den Seiten 93/94 des Skripts beschrieben ist. Definieren Sie geeignete Hilfsprädikate. Nutzen Sie insbesondere die bereits auf Blatt 9 und 10 implementierten Vereinfachungsheuristiken *Unit Propagation* und *Pure Literal Rule*, die Sie aus den Modulen des entsprechenden Namens importieren können. Die Streichung von Klauseln, die Obermengen von anderen Klauseln sind, müssen Sie nicht implementieren.

- (c) Gegeben sei die kontextfreie Grammatik $G = (\Sigma, V, S, P)$ mit den Terminalsymbolen $\Sigma := \{\text{if, then, else, e1, e2, s1, s2}\}$, den Nichtterminalsymbolen $V := \{\text{stmt, expr}\}$, dem Startsymbol $S := \text{stmt}$, und den Produktionen $P :=$

$$\left\{ \begin{array}{ll} \text{stmt} \longrightarrow \text{if expr then stmt}, & \text{stmt} \longrightarrow \text{if expr then stmt else stmt}, \\ \text{stmt} \longrightarrow \text{s1}, & \text{stmt} \longrightarrow \text{s2}, \quad \text{expr} \longrightarrow \text{e1}, \quad \text{expr} \longrightarrow \text{e2} \end{array} \right\}$$

Bilden Sie für die kontextfreie Grammatik G eine *Definite Clause Grammar (DCG)*, sodass die Anfrage

`?- stmt(X, []).`

genau dann erfüllt wird, wenn X eine Liste von Terminalsymbolen aus Σ ist, die einem Wort der durch G beschriebenen Sprache entspricht. Dies gilt beispielsweise für die Liste

`X = [if, e1, then, if, e2, then, s1, else, s2] .`

Fügen Sie Ihre Definite Clause Grammar der Datei `blatt11.pl` hinzu.