

Übungszettel 08

Abgabetermin: 17. Dezember 2022, 18:00 Uhr. Die Abgabe erfolgt über Ihr Ilias-Tutorium.

Aufgabe 1 (Negation as failure). In Prolog kann man eine Negation oft mithilfe der Kombination „von cut (!) und fail ausdrücken. Beispiel:

```
notMember(X, Ys) :- member(X, Ys), !, fail.  
notMember(X, Ys).
```

1. Schreiben Sie mit Hilfe von „cut/fail“ ein Prädikat **noDups**, das überprüft, dass keine Duplikate in einer Liste zu finden sind.

```
noDups([2,4,7]) % liefert true  
noDups([2,4,7,4]) % liefert no.
```

2. In manchen Prolog-Implementierungen, wie auch in 2Prolog gibt es auch einen Operator **not**, mit dem man ein Prädikat negieren kann. **not(member(X, Y))** hat dann die gleiche Semantik wie obiges **notMember(X, Y)**. Testen Sie nun, ob die doppelte Negation eines goals *g* stets das gleiche Verhalten zeigt wie das Originalgoal. Begründen Sie Ihr Ergebnis.

4

DNF $(\ldots \wedge \ldots) \vee (\ldots \wedge \ldots)$ (2+2 Punkte)

Aufgabe 2 (Subsumierung von Klauseln). Eine disjunktive Klausel κ_1 subsumiert eine Klausel κ_2 , falls jedes Literal *L* aus κ_1 auch in κ_2 vorkommt.

1. Begründen Sie: Aus einer Menge \mathcal{K} von V-Klauseln kann man jede Klausel κ_2 weglassen, die von einer anderen Klausel $\kappa_1 \in \mathcal{K}$ subsumiert wird.

2. Schreiben Sie ein Prolog Prädikat **subsumes(K1,K2)**, das testet, ob Klausel K1 die Klausel K2 subsumiert.

Bsp.: **subsumes([-2,-3,-1,-2], [-1,-6,-2,-3,42]). % yes**

3. Schreiben Sie ein Prädikat **someoneSubsumes(KSet, K)**, das testet, ob es eine Klausel in der Klauselmenge KSet gibt, die die Klausel K subsumiert.

Bsp.: **someoneSubsumes([[1,-2], [-1,-2]], [-2,-3,-1,-2]). % yes**

4. Schreiben Sie ein Prädikat **removeSubsumed(K, KSet, NewKSet)**, das alle Klauseln aus der Klauselmenge KSet entfernt, die von K subsumiert werden. Das Ergebnis ist NewKSet.

Bsp.: **removeSubsumed([1,-3], [[2,-3,1], [3,1], [2,3,1], [3,-1], [-3,2,1]], X).**
liefert: % X = [[3,1], [2,3,1], [3,-1]]

7

(1+2+2+2 Punkte)

Aufgabe 3 (Verbesserte Resolventenmethode). Im Ilias finden Sie den Quellcode der Datei **resolve.pl**, die die Resolventenmethode implementiert.

- (a) Fügen Sie Ausgabeanweisungen hinzu, so dass jeder erfolgreiche Resolutionsschritt zu einer aussagekräftigen Ausgabe führt, z.B.: **[1,-2,-3] + [1,2,5] ---> [1,-3,5]**

(b) Verbessern Sie das Prädikat **solve**, unter Zuhilfenahme von einem oder mehreren Prädikaten aus der vorigen Aufgabe.

(Falls Sie Aufgabe 1 noch nicht lösen konnten, implementieren Sie vorübergehend die dort verlangten Prädikate durch folgende dummy-Versionen:

```
subsumes(_,_)          :- fail.  
someoneSubsumes(Ks,K)  :- fail.  
removeSubsumed(_,KSet,KSet).
```

Mit diesen dummys wird das so „verbesserte“ **solve** natürlich nicht schneller sein als vorher.)

(1+4 Punkte)

