

Hinweis: Die Lösungen sind per Email an ha-lp.mmis@uni-rostock.de bis spätestens **21. Juni 2019 um 23:59 Uhr (CEST)** abzugeben.

- Geben Sie Ihren Quelltext als Prolog-Datei mit dem Namen (Die Gruppennummer immer entsprechend anpassen) **Gruppe-01-HA3.pl** ab.
- Zusätzliche Erläuterungen können als PDF (maschinengeschrieben) abgegeben werden.
- Stellen Sie alle anderen Lösungen kurz und präzise in einem PDF dar.
- Achten Sie darauf, dass Ihre Prolog-Datei erfolgreich in SWI-PROLOG (wie im Rechnerpool installiert) geladen werden kann. Lösungsideen können Sie im Kommentar oder im PDF skizzieren.
- Die Bearbeitung muss in Gruppen von 3 bis 5 Personen erfolgen.
- Für alle Gruppenmitglieder ist der Name, die Matrikelnummer und der Studiengang in allen Dokumenten anzugeben, d.h. sowohl in der Email, dem PDF-Dokument als auch als Kommentar im Quelltext.

1. (8 Punkte) Gegeben sei folgende Definition des Prädikates **blup**/3:

```
1 blup(L, E, R) :-
2   append(L1, [E|L2], L), append(L1, L2, R).
```

- In welchem Zusammenhang müssen die Argumente von **blup**/3 stehen, so dass das Prädikat wahr ist?
- Geben Sie eine alternative Implementierung, ohne die Nutzung von **append**/3 (oder anderen Prädikaten der Standardbibliothek) an.
- Geben Sie für beide Definitionen den SLD-Baum für folgende Anfragen an:
 - ?- **blup**([1,2,3], 2, R).
 - ?- **blup**([1,2], E, R).

Geben Sie dabei für alle Zwischenschritte den Unifikator und die vollständige Antwortsubstitution an.

2. (20 Punkte) Definieren Sie ein Prädikat **simplify**(X,Y), welches wahr ist, wenn Y eine vereinfachte Form des Ausdrucks X ist. Mögliche Ausdrücke (E) sind durch folgende EBNF-Regeln beschrieben:

$$E ::= INT \mid FLOAT \mid -E \mid E + E \mid E - E \mid E * E \mid E / E$$

$$VAR ::= x \mid y \mid z \mid \dots$$

wobei *FLOAT* den gebrochenen Zahlen in Prolog entspricht. Implementieren Sie folgende Vereinfachungen:

- $-(-E) = E$
- $-(E + F) = (-E) + (-F)$
- $-(E * F) = (-E) * F$
- $0 + E = E + 0 = E$
- $0 * E = E * 0 = 0$
- $1 * E = E * 1 = E$
- $E/1 = E$
- $E - 0 = E$
- $E - E = 0$
- Summe, Differenz, Produkt und Quotient von Zahlen sollen dabei direkt berechnet werden (via [is](#)).

3. (5 Punkte) Wenden Sie ihre Implementierung auf die Ergebnisse des Ableitungsprädikates der Hausaufgabenserie 2 an. D.h. geben Sie die vereinfachten Ausdrücke für die Ergebnisse von Aufgabe 2.b) der Hausaufgabenserie 2 an.

Hinweis: Die Lösungen sind

- $3*1+0$
- $3*1*x+3*x*1$
- 0
- $(3*1+0)*(2*x)+(3*x+2)*(2*1)$
- $((1*x+x*1+2*1+0)*(3*x)-(x*x+2*x+3)*(3*1))/(3*x*(3*x))$

4. (27 Punkte) Laut dem *Vier-Farben-Satz* (<http://de.wikipedia.org/wiki/Vier-Farben-Satz>), lässt sich jede beliebige Landkarte mit vier verschiedenen Farben so einfärben, dass keine zwei benachbarten Länder die gleiche Farbe haben. Gegeben sei folgende Karte von Mecklenburg-Vorpommern:



Abbildung 1: Quelle – TUBS (<https://commons.wikimedia.org/wiki/User:TUBS>)

- a) Definieren Sie die Nachbarschaftsbeziehungen der Landkreise indem Sie die Karte als Fakten der Form `border/2` angeben, wobei die Landkreise durch ihr KFZ-Kennzeichen repräsentiert werden sollen.

- b) Machen Sie sich mit den unter Kapitel 4.30 („Finding all Solutions to a Goal“) des SWI-Prolog Manuals <http://www.swi-prolog.org/pldoc/refman/> und unter Kapitel 4.31 („Forall“) vorgestellten Prädikaten vertraut. Implementieren Sie anschließend folgende Prädikate:
- `counties(CS)`, welches für die erstellte Karte, `CS` mit der Liste aller Landkreise unifiziert. Stellen Sie sicher, dass Landkreise nicht mehrfach enthalten sind.
 - `neighbours(C,N)`, welches für einen gegebenen Landkreis `C`, das Argument `N` mit der Liste aller Nachbarn unifiziert. Bitte beachten Sie dabei, dass beide Argumente auch Variablen sein können. In diesem Fall sollen durch Backtracking alle Lösungen erzeugt werden.
- c) Gegeben sei folgende Liste von Farben:

```
1 colours([red,yellow,blue,green]).
```

Eine Farbzuordnung von Kreisen zu Farben ist gültig, wenn keinen benachbarten Landkreisen dieselbe Farbe zugeordnet wird. Farbzuordnungen sollen als Liste repräsentiert werden: `[hro=green,nvp=blue,...]`. Definieren Sie folgende Prädikate:

- `genColouring(Counties, Colouring)`, welches für eine gegebene Liste von Kreisen (`Counties`) jede möglich Farbzuordnung erzeugt (welche nicht notwendigerweise gültig ist).
- `validColouring(Colouring)`, welches wahr ist, wenn die übergebene Farbzuordnung bzgl. der Karte gültig ist.
- `colouring(Counties, Colouring)`, welches für eine gegebene Liste von Landkreisen (`Counties`), das Argument `Colouring` mit einer gültigen Farbzuordnung instanziiert. Dabei sollen alle gültigen Färbungen durch Backtracking erzeugbar sein. (Wie viele Landkreise können Sie mit ihrer Implementation in < 5 min Laufzeit einfärben?)