Frage: Wir haben unsere Aufgabenstellung Ihnen nur informell vorgestellt, müssen wir die noch präzisieren?

Antwort: Nein, in dieser Veranstaltung möchte ich nur grob das fachliche Thema und die eingesetzten Technologien wissen. Sie haben so in der Ausarbeitung einige Freiheitsgrade und können gerade mit der neuen Technologie experimentieren, wie es bei Forschungsansätzen üblich ist. Die präzise Aufgabenstellung (0.5 bis eine Seite) steht im ersten Kapitel der Hausarbeit. Wenn Sie die Aufgaben detaillierter besprechen wollen, können Sie natürlich auch eine kurze schriftliche Darstellung mit mir durchsprechen.

Frage: Sind sie außerhalb der Vorlesungszeit noch erreichbar?

Antwort: Ja. Generell antworte ich auf E-Mails, falls es nicht passiert, habe ich es schlicht vergessen, dann spätestens nach 4 Tagen nochmal schicken. Per Mail können wir auch einen Zoom-Termin vereinbaren, schicken Sie mir dann möglichst große Zeitintervalle, so dass ich dann einen Termin finden kann.

Frage: Können beliebige Meta-Prädikate benutzt werden?

Antwort: Ja, wobei immer gilt, wenn es einfach ohne geht, sollte der Ansatz ohne Meta-Prädikate, wie findall genutzt werden. Da in den Praktika schon einige Varianten vorgekommen sind, hier kurze Infos dazu.

```
x(3).
x(1).
x(2).
x(1).
```

Das Prädikat setof/3 arbeitet sehr ähnlich wie findall/3 hat aber keine doppelten im Ergebnis und liefert ohne Lösung false.

```
?- setof(X,x(X),Y).
Y = [1, 2, 3].

?- setof(X,(x(X), X>3),Y).
false.

?- findall(X,(x(X), X>3),Y).
Y = [].
```

Das Prädikat bagof/3 liefert wie findall/3 alle Ergebnisse (bag = Multiset), aber false, wenn es kein Ergebnis gibt.

```
?- bagof(X,x(X),Y).
Y = [3, 1, 2, 1].

?- bagof(X,(x(X), X>3),Y).
false.
```

Das Prädikat maplist/2 ist in der Anwendung komplexer, die Idee ist, dass ein gegebenes Prädikat auf eine Sammlung von Listen angewandt wird. Dies soll mit der Matrixaufgabe verdeutlicht werden, bei der eine Matrix aus der Angabe der Zeilenanzahl, der Spaltenanzahl und einer Liste von Zeilen, die wieder jeweils eine Liste sind, besteht. Dieser Aufbau soll überprüft werden, u. a. mit den folgenden Tests.

```
testMatrixOK :-
      matrixOK([3,2,[[1,2],[3,4],[5,6]]]),
      matrixOK([0,0,[]]),
      matrixOK([1,1,[[42]]]),
      not(matrixOK([3,3,[[1,2],[3,4],[5,6]]])),
      not(matrixOK([2,4,[[1,2],[3,4],[5,6]]])),
      not(matrixOK([2,3,[[1,2],[3,4],[5,6,7]]])),
      not(matrixOK([3,3,[[1,2],[3,4],[5,6],[7,8]]])).
```

Zunächst wird ein Prädikat benötigt, dass auf Listen angewandt werden soll. Im konkreten Fall soll die Länge von Listen berechnet werden. Da die Parameter in anderer Reihenfolge benötigt werden, wird folgende Regel ergänzt.

```
laenge(Wert, Liste) :- length(Liste, Wert).

matrixOK([Zeilen, Spalten, Matrix]) :-
    maplist(laenge(Spalten), Matrix),  %1
    maplist(laenge(Zeilen), [Matrix]). %2
```

Mit maplist/2 wird das erste Prädikat auf alle Elemente der Liste angewandt, dabei erhält das Prädikat alle angegebenen Variablen ergänzt um alle Elemente der jeweiligen Liste als Parameter. Das bedeutet, dass bei der ersten Zeile des ersten Tests in Zeile %1 folgende Aufrufe durchgeführt werden:

laenge(2,[1,2]) , laenge(2,[3,4]), laenge(2, [5,6)).

Die Zeile %2 führt zu folgender Prüfung, da die Liste nur ein Element, also eine Liste, enthält; durch die zusätzlichen eckigen Klammern wird aus einer Liste mit drei Elementen eine Liste aus nur einem Element (was eine Liste aus drei Elementen ist):

laenge(3, [[1,2],[3,4],[5,6]]])

Das Zusammensetzen eines aufzurufenden Prädikats aus Teiltermen kennen wir von call.