

2. Übung

Abgabe in der Woche vom 8. - 12. Mai 2006

Aufgabe 5 (2+2+1+3 Punkte)

Gegeben sei ein azyklischer Graph als eine Auflistung seiner Kanten als Fakten. Eine Kante ist dargestellt durch die Relation `kante/2` und verbindet zwei Knoten. Das erste Argument in der Relation `kante/2` ist der Kantenursprung.

Folgende Datenbasis beschreibt bspw. einen solchen Graphen mit vier Kanten: `kante(1,2)`. `kante(2,3)`. `kante(1,4)`. `kante(4,5)`..

1. Definieren Sie eine Regel `erreichbar/2`, welche wahr wird, wenn es eine Kantenfolge zwischen dem ersten und dem zweiten Argument der Relation gibt. Dabei sei jeder Knoten von sich selbst erreichbar.

```
?- erreichbar(1,2).
```

```
yes.
```

```
?- erreichbar(2,X).
```

```
X=2;
```

```
X=3;
```

```
no.
```

2. Definieren Sie eine Regel `erreichbar_gerade/2`, welche wahr wird, wenn das zweite Argument von dem ersten Argument über eine gerade Anzahl von Kanten erreichbar ist.

```
?- erreichbar_gerade(1,2).
```

```
no.
```

```
?- erreichbar_gerade(1,X).
```

```
X=1;
```

```
X=3;
```

```
X=5;
```

```
no.
```

3. Definieren Sie eine Regel `erreichbar_ungerade/2`, welche wahr wird, wenn das zweite Argument von dem ersten Argument über eine ungerade Anzahl von Kanten erreichbar ist.

```
?- erreichbar_ungerade(1,2).  
yes.  
?- erreichbar_ungerade(1,X).  
X=2;  
X=4;  
no.
```

4. Wir betrachten nun ungerichtete Graphen, d.h. wenn eine Kante von Knoten a zum Knoten b existiert, dann gibt es auch eine Kante von b nach a . Die Datenbasis des oberen Beispiels ist demnach: `kante(1,2)`. `kante(2,1)`. `kante(2,3)`. `kante(3,2)`. `kante(1,4)`. `kante(4,1)`. `kante(4,5)`. `kante(5,4)`.

Zeichnen Sie den Abarbeitungsbaum für die Anfrage `?- erreichbar(1, X)`. auf und kommentieren Sie ihn.

Aufgabe 6 (2 Punkte)

1. Schreiben Sie eine Relation `kopf_schwanz/2`, die zwei Listen als Argumente erhält. Sie soll wahr werden, wenn die zweite Liste aus dem ersten Element und dem letzten Element der ersten Liste besteht. Wir betrachten hier nur Listen mit mindestens zwei Elementen.

```
?- kopf_schwanz([1,2,3], [1,3]).  
yes.  
?- kopf_schwanz([1,2,3,1], X).  
X=[1,1];  
no.
```

Aufgabe 7 (3+2+2+3 Punkte)

1. Definieren Sie eine Relation `loesche_alle/3`, die ein Element und zwei Listen als Argumente erhält. Die Relation soll wahr werden, wenn die zweite Liste aus der ersten Liste durch Löschen aller Vorkommen des Elementes entsteht.

```
?- loesche_alle(1, [1,2,3,1,2], X).  
X=[2,3,2];  
no.
```

2. Definieren Sie eine Relation `menge/2`, die zwei Listen als Argumente erhält. Sie soll wahr werden, wenn die zweite Liste aus der ersten entsteht, indem alle Duplikate aus der ersten Liste entfernt werden. Nutzen Sie hierfür die Relation `loesche_alle/3`.

```
?- menge([1,2,3,3,1], X).  
X=[1,2,3];  
no.
```

3. Definieren Sie eine Relation `vereinigung/3`, die drei Listen als Argumente erhält. Sie soll wahr werden, wenn die dritte Liste gerade die Mengenvereinigung der ersten beiden Listen ist. Nutzen hierzu die Relation `konk/3` aus der Vorlesung.

?- `vereinigung([1,2,3], [1,3,7], X)`.

`X=[1,2,3,7]`;

`no.`

4. Definieren Sie eine Relation `durchschnitt/3`, die drei Listen als Argumente erhält. Sie soll wahr werden, wenn die dritte Liste gerade der Mengendurchschnitt der ersten beiden Listen ist. Nutzen hierzu die Relation `mitglied/3` aus der Vorlesung.

?- `durchschnitt([1,2,3], [1,3,7], X)`.

`X=[1,3]`;

`no.`