

fynn-lehrer210w-ss3q0-do10-12-ss3p-q0-q04.dff.nl

Page 1

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Aufgabenblatt 04 - SE3-LP WiSe 16/17
%
% Finn-Lasse Jørgensen 6700628 4joergen@informatik.uni-hamburg.de
% Fabian Behrendt 6534523 fabian.behrendt95@gmail.com
% Daniel Klotzsche 6535732 daniel.klotzsche@hotmail.de
%
% Wir sind bereit folgende Aufgaben zu präsentieren:
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[medien2]
%% A1 %%

```

A1: 0P

A2: 11P  
Tests? -1P

```

%% A2 %%
%% Aufgabe 2.1
% Prüft ob die gegebene Oberkategorie (spezifizierbar durch ID und / oder Name)
% in der Hierarchie über
% der gegebenen Unterkategorie steht bzw. umgekehrt.
% Kann ebenfalls Ober- bzw. Unterkategorien finden oder Paare von Ober- und Unt-
% kategorien finden.

% oberkategorie von (Oberkategorie ID, Oberkategorie Name, Unterkategorie ID), Un-
% terkategorie Name (ID, wurzelkategorie, UK_ID, UK Name) :-
%   kategorie(UK ID, UK Name, 0).

oberkategorie_vonOk ID, OK Name, UK ID, UK Name :-
%   kategorie(Ok ID, OK Name, ),
%   kategorie(UK ID, UK Name, OK ID).

oberkategorie_vonOk ID, OK Name, UK ID, UK Name :-
%   kategorie(UK ID, UK Name, X),
%   oberkategorie_vonOk ID, OK Name, X, ).

%% Aufgabe 2.2
% Gibt true aus, wenn der Pfad vom Wurzelknoten zur angegebenen Kategorie n de-
% r Pfadliste enthalten ist.

% kategorie_pfad(Kategorie ID, Kategorie Name, Liste mit Pfad in der From (0,
% wurzelkategorie), (1, buch), ...))
kategorie_pfadK ID, K Name, Pfadliste :-
%   kategorie(K ID, K Name, ),
%   findall((Ok ID, OK Name, ),
%     oberkategorie_vonOk ID, OK Name, K ID, ),
%   ReversePfadList,
%   reverse(ReversePfadList, Pfadliste).

%% Aufgabe 2.3
% Es werden alle Produkte der Kategorie und alle Produkte der Unterkategorien a-
% ufgelistet.
% Wenn die Wurzelkategorie angegeben wird, wird nicht geprüft, ob diese ex stie-
% rt.

% alle produkte in kategorie( Kategorie ID, Kategorie Name, Liste der Produkte
% in Kategorie in der Form [(34567, hoffnung, sand molly), ...])
alle_produkte_in_kategorieID, K Name, ProduktListe :-
%   findall((P ID, P Titel, P Autor, (
%     in_kategorieP ID, 0),
%     produkt(P ID, , P Titel, P Autor, , , )
%   )), ProduktListe.

alle_produkte_in_kategorieID, K Name, ProduktListe :-
%   kategorie(K ID, K Name, ),
%   findall((P ID, P Titel, P Autor, (
%     in_kategorieP ID, K ID,
%     produkt(P ID, , P Titel, P Autor, , , )
%   )), ProduktListe.

```

fynn-lehrer210w-ss3q0-do10-12-ss3p-q0-q04.dff.nl

Page 2

```

in_kategorieP ID, K ID :-
%   produkt(P ID, K ID, , , , ).

in_kategorieP ID, K ID :-
%   kategorie(UK ID, , K ID,
%   in_kategorieP ID, UK ID, ).

%% Aufgabe 2.4
% Listet die Anzahl aller Produkte in einer Kategorie einschließlich ihrer Unte-
% rkategorien auf.

% anzahl produkte in kategorie( Kategorie ID, Kategorie Name, Anzahl der P oduk-
% te in Kategorie einschließlich Unterkategorien)
anzahl_produkte_in_kategorieID, K Name, ProduktAnzahl :-
%   alle produkte in kategorieID, K Name, ProduktListe,
%   length(ProduktListe, ProduktAnzahl).

%% Aufgabe 2.5
% Es wird ermittelt wie viele Produkte aus welcher Kategorie in einem Jahr wie
% oft verkauft wurde.
% Wenn die Wurzelkategorie angegeben wird, wird nicht geprüft, ob diese ex stie-
% rt.

% verkaufte produkte in kategorie in jahr( Kategorie ID, Kategorie Name, Anzahl
% der Produkte in dem gesucht werden soll)
verkaufte_produkte_in_kategorie_in_jahr wurzelkategorie, ProduktAnzahl Jah-
r :-
%   verkauf((Jahr, , ), % finde ein jahr falls nicht angegeben
%   findall(Anzahl, (
%     in_kategorieP ID, 0,
%     verkauf(P ID, Jahr, , Anzahl)
%   ), AnzahlListe,
%   sum list(AnzahlListe, ProduktAnzahl).

verkaufte_produkte_in_kategorie_in_jahrID, K Name, ProduktAnzahl Jahr :-
%   verkauf((Jahr, , ), % finde ein jahr falls nicht angegeben
%   kategorie(K ID, K Name, ),
%   findall(Anzahl, (
%     in_kategorieP ID, K ID,
%     verkauf(P ID, Jahr, , Anzahl)
%   ), AnzahlListe,
%   sum list(AnzahlListe, ProduktAnzahl).

```

A3: 7P  
Doku? -1P  
Tests? -1P  
-1P

Symmetrie? g1-g2 / g2-g1

```

%% 1.
% Gleis hat eine direkte oder eine indirekte Verbindung zu Gleis2,
% ohne dabei die Fahrtrichtung wechseln zu müssen.
%
% verbindungA1(?Gleis1, ?Gleis2)
verbindungA1Gleis1, Gleis2 :-
%   weiche( , Gleis1, Gleis2, ).

verbindungA1Gleis1, Gleis2 :-
%   weiche( , Gleis1, GleisMitte, ),
%   verbindungA1GleisMitte, Gleis2.

%% 2.
% Das Prädikat "belegt(Gleis)" legt fest, dass ein Gleis belegt ist.
?- assert(belegt(g2)).
?- assert(belegt(g3)).
?- assert(belegt(g4)).
?- assert(belegt(g9)).

% Angepasste Definition, sodass eine Verbindung nur besteht, wenn

```

g1-g2, g2-g1, g3-g4, g4-g9, g9-g1

```

% keines der Gleise belegt ist.
%
% verbindungA2(?Gleis1, ?Gleis2)
verbindungA2(?Gleis1, Gleis2) :-
    \+ belegt(Gleis1),
    \+ belegt(Gleis2),
    weiche1( , Gleis1, Gleis2, ).

verbindungA2(Gleis1, Gleis2) :-
    weiche1( , Gleis1, GleisMitte ),
    \+ belegt(GleisMitte),
    verbindungA2(GleisMitte, Gleis2).

% Tests für A2
?- verbindungA2(a1, z5).
% true.

?- verbindungA2(z1, z5).
% true.

?- verbindungA2(a11, a5).
% false.

?- verbindungA2(z2, a4).
% false.

%% 3.

% Erweiterte Definition, sodass der Zug nur Gleise, die mindestens die Länge
% des Zuges aufweisen, befahren darf.
%
% verbindungA3(?Gleis1, -Gleis2, ?Zuglaenge)
passend(Gleis, Zuglaenge Typ) :-
    gleis(Gleis, Laenge, Typ),
    Laenge > Zuglaenge

verbindungA3(Gleis1, Gleis2, Zuglaenge) :-
    \+ belegt(Gleis1),
    \+ belegt(Gleis2),
    passend(Gleis1, Zuglaenge ),
    passend(Gleis2, Zuglaenge ),
    weiche1( , Gleis1, Gleis2, ).

verbindungA3(Gleis1, Gleis2, Zuglaenge) :-
    weiche1( , Gleis1, GleisMitte ),
    \+ belegt(GleisMitte),
    verbindungA3(GleisMitte, Gleis2, Zuglaenge).

%% 4.

% Dient zur Bestimmung eines Ankunftsgleises eines Zuges.
%
% ankunft(?Von Ort, -Gleis, ?Zuglaenge)
ankunft(Von Ort, Gleis, Zuglaenge) :-
    einfahrt(AnkunftsGleis, Von Ort,
    verbindungA3AnkunftsGleis, Gleis, Zuglaenge).

% Dient zur Bestimmung eines Abfahrtsgleises eines Zuges.
%
% abfahrt(?Nach Ort, -Gleis, ?Zuglaenge)
abfahrt(Nach Ort, Gleis, Zuglaenge) :-
    ausfahrt(AbfahrtsGleis, Nach Ort,
    verbindungA3AbfahrtsGleis, Gleis, Zuglaenge).

```