

0.4pt=0pt

LABORATION I DD1351, del av Logik för dataloger*Fredag 28 September 2018, 8 – 10**delkursansvarig:* Thomas Sjöland, sjoland@kth.se, 08 - 790 4113

Här är några generellt användbara definitioner och inbyggda predikat som du kan använda:

```
append([], L, L).
append([H|T], L, [H|R]) :- append(T, L, R).

appendEl(X, [], [X]).
appendEl(X, [H | T], [H | Y]) :-
    appendEl(X, T, Y).

nth(N, L, E) :- nth(1, N, L, E).
nth(N, N, [H|_], H).
nth(K, N, [_|T], H) :- K1 is K+1, nth(K1, N, T, H).

subset([], []).
subset([H|T], [H|R]) :- subset(T, R).
subset(_|T, R) :- subset(T, R).

select(X, [X|T], T).
select(X, [Y|T], [Y|R]) :- select(X, T, R).

member(X, L) :- select(X, L, _).

memberchk(X, L) :- select(X, L, _), !.
```

Uppgifterna 1, 2, 3, 4, 5 skall lösas! För bonus till tentan skall minst en av uppgifterna 6 eller 7 lösas!

UPPGIFT 1

Betrakta denna fråga till ett Prologsystem:

```
?- Z=f(a,Y,Y), Z=f(X,X,b).
```

Vilka bindningar presenteras som resultat?

Ge en kortfattad förklaring till ditt svar!

UPPGIFT 2

I Prolog hanteras negation oftast med "negation-as failure". Förklara denna princip genom några enkla exempel!

UPPGIFT 3

En lista är en representation av sekvenser där den tomma sekvensen representeras av symbolen `[]` och en sekvens bestående av tre heltal `1 2 3` representeras av `[1,2,3]`.

Den exakta definitionen av en lista är:

```
list().  
list([H|T]) :- list(T).
```

Vi vill definiera ett predikat som givet en lista som representerar en sekvens skapar en annan lista där sista elementet är borttaget. Elementet skall returneras i det tredje argumentet. Till exempel:

```
?- findlast([1,2,3], R,E).
```

skall generera `R=[1,2]` och `E=3`

Definiera alltså predikatet `findlast/3`!

Visa exekveringen av

```
?- findlast([1,2,3], R,E).
```

UPPGIFT 4

Definiera predikatet `partstring/2` som givet en lista som första argument genererar en lista av ett antal element som man finner konsekutivt i den första listan!

Till exempel:

```
?- partstring([ 1, 2 , 3 , 4 ], F).
```

genererar t.ex. `F=[4]` eller `F=[1,2]` eller också `F=[1,2,3]` eller `F=[2,3]` osv.

UPPGIFT 5

Definiera predikatet `permute/2` som givet en lista som första argument genererar en lista av alla elementen man finner i den första listan ordnad på något av de möjliga sätten!

Till exempel:

```
?- permute([ 1, 2 , 3 , 4 ], F).
```

genererar t.ex. `F=[3, 2, 1, 4]` eller också `F=[4, 1, 2, 3]` osv.

Visa exekveringen av

```
?- permute([ 1, 2 ], F).
```

UPPGIFT 6

Du skall definiera ett program som arbetar med grafer.

Föreslå en representation av grafer så att varje nod har ett unikt namn (en konstant) och grannarna finns indikerade.

Definiera ett predikat som med denna representation och utan att fastna i en loop tar fram en väg som en lista av namnen på noderna i den ordning de passeras när man utan att passera en nod mer än en gång går från nod A till nod B!

UPPGIFT 7

Landet Filurien har åtta partier i parlamentet.

Senaste valet blev rafflande. Ingen vann.

Partierna fick dessa mandat: p1 38 , p2 17, p3 51, p4 23, p5 27, p6 35, p7 18, p8 25

Varje parti fick bedöma hur mycket de har gemensamt med de andra partierna på en skala från -10 till +10, Vi har alltså en matris med 8*8 signifikanta värden. (Hitta på lämpliga värden själva! Sätt värdet 0 som partiets självvärdering.)

En *stabil* regering uppfyller två villkor:

1. summan av antalet mandat för de valda partierna är minst hälften av totalantalet.
2. summan av gemensam-index för de valda partierna sinsemellan är positivt. (De har totalt mer gemensamt än motsatsen.)

Skriv ett Prolog-program som föreslår en *stabil* regering i Filurien!

Lycka till! *Thomas*