

Prologrepetition

- Alla prologuttryck avslutas med punkt (.)
– *man(bertil).*
- Inget mellanslag mellan före parenteserna
– ~~*man(ber til).*~~ → *man(bertil).*
- *man(bertil).* Är ett predikat, inleds med gemen
- Om namnet på *Bertil* skall inledas med versal
– *man('Bertil').*

Predikat har ställighet

- Enställt – *man/1*
– *man(bertil).*
- Tvåställt – *pappa/2*
– *pappa(hilding,ber til).*
- Precis som i Perl, kommentera väl era anrop
– kommentarer i Prolog inleds med %

Att köra ett Prologprogram

- Prologprogram körs genom att man i *tolken* skriver (om programmet heter 'släkt.rul'):
– *consult('släkt.rul').*
- Om allt gick OK (Prolog kollar filen) så får man upp ett *yes*
- En redan inläst fil kan konsulteras igen med:
– *reconsult('släkt.rul').*

Att dra slutsatser

- Om vi vid frågeprompten i tolken skriver:
?- pappa(X,ber til).
- och sedan tidigare har predikatet
– *pappa(hilding,ber til).*
- Så kommer *X* att unifieras, bindas, till *hilding*, varpå vi får svaret:
X = hilding
yes

Testa att bygga ett släkträd

- Bygg en databas med predikaten
– *mor/2*
– *far/2*
– *son/2*
– *dotter/2*
– *man/1*
– *kvinn/1*
- och pröva med att ställa lite frågor till den!

Predikatet son/2

- Skulle t.ex. kunna se ut så här:

```
son(X,Y) :-  
    man(X),  
    far(Y,X).
```

```
son(X,Y) :-  
    man(X),  
    mor(Y,X).
```

Om predikatet *bror/2* är:

```
bror(X,Y) :-  
  man(X),  
  förälder(Z,X),  
  förälder(Z,Y).
```

- Hur skulle då predikatet *förälder/2* se ut?

Listor!

- Vi har en lista
[1,2,3,4,5,6].
- Som vi vill kunna skriva ut med
?- *skriv_ut_lista([1,2,3,4,5,6]).*
- Då måste vi rekursera genom listan!

Rekursion!

- Programmet *skriv_ut_lista/1* ser ut så här:
skriv_ut_lista([]). % Basfall som avslutar rekursionen
skriv_ut_lista([FörstaElementet|RestenAvListan]) :-
 write(FörstaElementet), nl,
 skriv_ut_lista(RestenAvListan).
- *nl* skriver ut ett nyradstecken (\n i Perl)

Mera listor!

- Listor i Prolog kan innehålla andra listor
[1,2,3,[4,5,6],7,8,9].
- Titta på hur *skriv_ut_lista/2* funkar m.h.a.
trace/0 och *step/0*.
- Nu kan ni se på när Prolog kör programmet!
- Trace'ning slås sedan av med *notrace/0*.

I/O

- *write/1* skriver till skärmen
test :- write('Kalle gick på bio!').
- *read/1* läser från tangentbordet
hej :-
 read(Namn),
 write('Hej på dig '),
 write(Namn), nl.

Mer I/O!

- Öppnar filer för läsning gör man med *see/1*
see('Kalle.txt').
- Stänger gör man sedan med *seen/0*
seen.
- Öppnar filer för skrivning gör man med *tell/1*
tell('Ut.txt').
- Stänger gör man sedan med *told/0*
told.