Grundläggande logik och modellteori
Höstterminen 2013

Laboration 3: Världen och Predikatlogik.

Deadline: 2014-10-10, 12:00

Inlämning: Skicka din lösning till Niklas Zechner (zechner@cs.umu.se). Ärendemeningen ska vara

[5dv102] namn-3

där 'namn' är ditt cs-användarnamn. Din *väl kommenterade* lösning ska finnas i en bifogad fil som heter

namn-3.pl

där 'namn' är ditt användarnamn. Glöm inte att klistra in dina quiz-svar i en kommentar högst upp i filen.

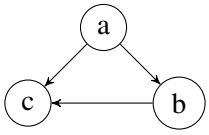
Prolog

Del 1: Uppvärmning med grafer

Antag att vi har en graf representerad som en serie predikat. T.ex.

```
edge(a,b).
edge(a,c).
edge(b,c).
```

för grafen

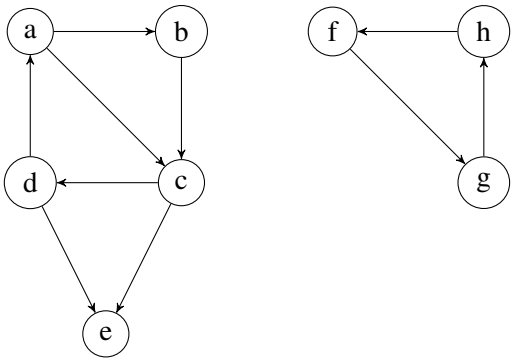


getallways

Skriv predikatet `getallways(X,Y)` som skriver ut (med predikatet `write`) *alla* vägar från hörnet `X` till hörnet `Y` i en *acyklisk* graf.

findway

Skriv ett predikat `findway(X,Y)` som för hörn `X` och `Y` är sant om det finns en väg från `X` till `Y` i en *riktad cyklisk* graf. T.ex. nedanstående.



Predikatet kommer att måsta fungera för t.ex. `findway(a,e)` och `findway(a,g)` (i det senare fallet är rätt svar '`false`'). Predikatet kommer att bli tvunget att internt (genom ett hjälppredikat med fler argument) hålla reda på besökta noder så att de inte upprepas.

Del 2: Världen

Filstruktur

I denna del arbetar vi med fler predikat i Prolog, och med det lite mer avancerade queries. Som exempeldata använder vi en geografisk databas, MONDIAL[1], som i en lite förenklad form kan hämtas här: `http://www.cs.umu.se/~mbe/mondial.pl`.

Gör hela laborationen i en egen fil, modifiera *inte* `mondial.pl`. För att ladda en fil i Prolog-interpretatorn skriver man `consult(mondial)`. Till exempel, där vi laddar filen och frågar några frågor om finland:

```
?- consult(mondial).
% mondial compiled 0.25 sec, 3,732,864 bytes
true.

?- country(finland, Capital, Area, Population).
Capital = 'Helsinki',
Area = 337030,
Population = 5105230.

?- borders(X,finland,BorderLength).
X = norway,
BorderLength = 729 ;
X = russia,
BorderLength = 1313 ;
X = sweden,
BorderLength = 586 ;
false.
```

För att i en fil ladda en annan fil skriver man `:- consult(mondial).`, påbörja din laboration i `world.pl` med denna rad högst upp.

Innehåll i MONDIAL

Mondial-filen innehåller ett stort antal fakta, över 22000 rader totalt. Exempelvis finns följande predikat:

[1] Gjord tillgänglig gratis för utbildningssyfte av institutionen för Informatik på Georg-August-Universitetet i Göttingen.

- `country(N,C,Ar,Pop)` säger att landet `N` har huvudstaden `C`, area `Ar` och populationen `Pop`.
- `borders(X,Y,L)` säger att land `X` gränsar till land `Y`, där gränsen är `L` kilometer.
- `city(N,C,R,Pop)` säger att staden `N` ligger i landet `C`, i regionen `R` och har populationen `Pop`.

Samt ett otal andra. För att lösa alla uppgifter kommer ni måsta läsa igenom filen lite, varje predikat har en förklarande kommentar.

Notera att predikatet `borders` bara har t.ex. `borders(norway, sweden, 1619)`, inte `borders(sweden, norway, 1619)`, du kan vilja skriva ett nytt predikat som är symmetriskt.

Frågor att besvara

Besvara följande frågor. Strukturera ditt program med delpredikat för varje sak du vill kontrollera, och med ett predikat som besvarar den faktiska frågan.

1. Lista de länder som innehåller en stad med en population som uppgår till 75% eller mer av landets befolkning. Notera att en del populationer inte är ifyllda, utan är satta till `null`, dessa måste filtreras bort! Använd t.ex. predikatet `integer/1`.
2. Vilket land i Europa har störst omkrets utan att gränsa till något hav? (summera gränserna till samtliga andra länder)
3. Samma fråga fast i Afrika, Amerika, Asien och Australien (`'Australia/Oceania'` i MONDIAL).

Hjälpsamma predikat för de här frågorna kan vara `setof/3`, som skapar en lista av alla svar på en viss fråga. T.ex., om vi skapar predikatet `borders/2` (där vi tar bort längden på gränsen då vi bara kan kolla en sak åt gången):

```
?- setof(Nb, borders(germany, Nb), Nbs).
Nbs = [austria, belgium, czech_republic, denmark, france, luxembourg,
netherlands, poland, switzerland].
```

Notera att `setof/3` har lite knepig semantik närhelst man har fler variabler i anropet än den man försöker "samla". T.ex. fungerar inte anropet `setof(Nb, borders(germany, Nb, _), Nbs).` som man skulle vänta sig (`Nbs` blir inte en lista med alla svar). Den enklaste lösningen är att skapa ett nytt predikat som inte har oanvända argument (som `borders/2` ovan).

Ett annat viktigt predikat är icke, som blir sant om argumentet misslyckas att satisfieras. Det har lite knepig semantik i Prologs backtracking, så använd det med stor försiktighet.

```
?- border(germany, C), \+ border(C, france).
C = denmark ;
C = france ;
C = netherlands ;
C = poland ;
C = austria ;
C = czech_republic.
```

Notera att Frankrike är en granne till Tyskland men är inte en granne med sig självt, så svaret är korrekt. Queryn `\+ border(C,france), border(germany, C)` fungerar *inte* (eller, rättare sagt, säger något

helt annat), Prolog kommer att för första delen hitta ett `C` som satisfierar `border(C,france)`, och negationen rapporterar omedelbart att helheten är falsk, utan att någonsin titta på kravet att `C` måste vara en granne till Tyskland.

Inlämning

- Dubbelkolla att ni lämnar in rätt kod.
- Dubbelkolla att den fungerar på SWI-Prolog som installerad på Datavenskap.
- Kommentera grundligt.
- Kom ihåg att lämna in era quiz-lösningar som en kommentar högst upp i filen (quizzen publiceras något senare än Prolog-delen denna gång).

Lycka till!