DD1361 - KS Logikprogrammering

2012-9-10

Kontrollskrivningen består av tjugo frågor där ett kortare svar krävs. För godkänt krävs att 16 av de 20 frågorna är rätt besvarade.

Namn:

Personnummer:

1 Logiken

Följande satser, satslogiska och predikatlogiska, skall skrivas i Hornklasulform, dvs den form som Prolog använder för att beskriva program. Varje sats kan resultera i en eller flera Hornklausuler, eller vara omöjlig att uttrycka på Hornklasulform. Om man inte kan utrycka satsen på Hornklasulform skall detta anges.

1. $a \Rightarrow b$

2. $a \vee b \Rightarrow c$

3. $a \vee b$

4. $\neg a \vee b$

5. $(a \wedge b) \vee c \Rightarrow d$

2 Unifiering

Utför, om möjligt, nedanstående unifieringar, var och en för sig, och skriv ner den resulterande unifieringsmängden på formen $\{Variabel \rightarrow Term, ...\}$. Om unifieringen inte kan utföras så skall resultatet "fail" skrivas.

1. $foo(X, [1|R]) = foo(b, [1, 2, 3])$

2. $[1, [2, 3]] = [H|T]$

3. $bar(X, Y, Y) = bar(Z, Z, a)$

4. $zot(X, Y, f(Y)) = zot(f(Z), 2, X)$

5. $[H|T] = [a]$

3 Prolog

3.1 foo/2

Vad gör nedanstående program, vad blir det första resultatet om man har målet `foo([1,2,1,4,2,1,5], X)`?

```
foo([], []).
foo([1|R], X) :-
  foo(R, X).
foo([2|R], [2,2|X]):-
  foo(R, X).
foo([I|R], [I|X]):-
  foo(R, X).
```

3.2 foo/2 backtracking

Om vi inte är nöjda med första resultatet i uppgiften ovan, vad blir det andra resultatet som genereras.

3.3 bar/3

Vad blir första resultatet om, vi givet programmet nedan, har målet `bar(X,[3,4],[1,2,3,4]`?

```
bar([], Y, Y).
bar([H|T], Y, [H|Z]):-
  bar(T, Y, Z).
```

3.4 bar/3 backtracking

Hur många lösningar finns det, givet programmet i föregående uppgift, till målet `bar(X, Y, [1,2,3])`?

3.5 double/2

Skriv ett predikat `double(+List, ?List)`, där första argumentet är en lista av tal och andra argumentet är motsvarande lista där varje tal har dubblerats. Om man har målet `double([1,3,2],X>)`, skall resultatet bli `{X = [2,6,4]}`.

4 Cut, negation mm

4.1 hur många lösningar

Givet programmet nedan, vilka lösningar finns till målet `a(X, Y, Z)`?

```
a(1, Y, Z) :- b(Y), !, c(Z).
a(2, 3, 4).

b(1).
b(2).
b(3).

c(1).
c(2).
```

4.2 slå upp ett värde

Givet programmet nedan, vilka lösningar finns till målet `lookup([e(17,foo), e(42,bar)|H], 42, X)`?

```
lookup([e(N,X)|_], N, X):- !.
lookup([_|T], N, X):-
  lookup(T, N, X).
```

4.3 bygg en mappning

Givet programmet ovan och anropen nedan, vad är resultatet för variabel, `Dict`?

```
:- lookup(Dict, 13, foo), lookup(Dict, 7, bar), lookup(Dict, 13, zot).
```

4.4 negation

Antag att vi implementerar negation med hjälp av "cut" som i programmet nedan. Detta tycks funger för mål som `man(fred)`, `woman(joe)` och `woman(sue)`. När fungerar det inte? Skriv ett mål där resultatet kanske inte är helt logiskt.

```
man(fred).
man(jim).
man(joe).

woman(X):- man(X), !, fail.
woman(_).
```

4.5 omvänd ordning

Om vi använder oss av "cut", så kan vi råka på problem vid den logiska tolkningen av våra program. Nedan finns två nästan identiska program, hur skilljer sig deras logiska tolkning?

```
foo :- b, !,  c.
foo :- a.

bar :- a.
bar :- b, !, c.
```