TECHNICAL UNIVERSITY OF DENMARK (DTU)

Written Sample-Exam-2, 2021 Page 1 of 4 pages

Course: Logical Systems and Logic Programming

Course number: 02156

Exam duration: 2 hours

Aids allowed: All written works of reference

Weighting: Stated for each problem

The following basic predicates can be used when writing Prolog programs:

```
member(H,[H|_]).
member(H,[_|T]) :- member(H,T).

append([],U,U).
append([H|T],U,[H|V]) :- append(T,U,V).
```

Here `member(?Elem,?List)` succeeds if and only if `Elem` can be unified with one of the members of `List` and `append(?List1,?List2,?List3)` succeeds if and only if `List3` unifies with the concatenation of `List1` and `List2`.

Standard predicates like `is`, `fail`, `write`, `nl` and `findall` can also be used.

In the following a Prolog program is said to be deterministic if and only if it does not succeed more than once.

Assume available a deterministic predicate `sort(+List,?Sorted)` that can be used to sort a list. Duplicates are merged as shown in the following example:

```
?- sort([3,1,4,1,2],S).

S = [1, 2, 3, 4]

Yes
```

Assume also available a predicate `length(+List,?Integer)` that can be used to calculate the number of elements in a list.

Problem 1 (50%)

In the following a semicolon (;) is used to separate the solutions to a query. This corresponds to the common use of the semicolon in an interactive Prolog session.

Question 1.1

State the remaining solutions to the following query:

```
?- append(L,_,[1,2,3]), member(X,L).

L = [1]
X = 1 ;

L = [1, 2]
X = 1 ;

L = [1, 2]
X = 2 ;

...
```

Question 1.2

Consider the following Prolog program:

```
r([],[]).
r([H|T],X):- r(T,Y), append(Y,[H],X).
```

State the solutions to the following query:

```
?- member(L,[[],[1],[1,2],[1,2,3]]), r(L,R).
```

The problem continues on the next page

Question 1.3

In the following it can be assumed that all elements of the lists are integers.

Consider a Prolog program `cutoff` such that `cutoff(+List1,?List2)` succeeds if and only if `List2` is the longest prefix of `List1` without negative elements.

Sample queries:

```
?- cutoff([1,-2,3],[1]).

Yes

?- cutoff([1,-2,3],[1,-2,3]).

No
```

Write the Prolog program `cutoff` without using the cut- or if-then-else-operators.

Write another version of the Prolog program `cutoff` using the cut- or if-then-else-operators and with as few uses of the arithmetic comparison operators as possible. Call this program `cutoff2` and make sure that it behaves exactly like `cutoff` for the sample queries above.

Problem 2 (25%)

Consider the following fragment of a boxing club database:

```
beat(a,[b,c,d]). beat(b,[]). beat(c,[d]). beat(d,[b]). beat(e,[a]).
```

Hence boxer `a` has beaten `b`, `c` and `d`, whereas `b` has not beaten anyone.

Question 2.1

Write a deterministic Prolog program `fighter(+Boxer)` corresponding to the definition: X is a fighter if and only if X has been beaten by someone and X has beaten someone. So `a`, `c` and `d` are fighters but `b` and `e` are not fighters.

Question 2.2

Write a deterministic Prolog program `count` that prints the number of beaten boxers for each boxer as follows:

```
?- count.
3 a
0 b
1 c
1 d
1 e

Yes
```

Problem 3 (25%)

Consider the following formula: $\exists y \forall x (p(y) \rightarrow p(x))$

Question 3.1

Use refutation and the systematic construction of a semantic tableau. State whether this shows that the formula is valid or not.

Question 3.2

Is the formula logically equivalent to $\exists y p(y) \rightarrow \forall x p(x)$ (explain why/why not)?