TECHNICAL UNIVERSITY OF DENMARK (DTU)

Written Sample-Exam-8, 2021

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,[1,2,3,4,5]), append(_,[X,Y|_],L).

L = [2, 3, 4, 5]
X = 2
Y = 3 ;

L = [2, 3, 4, 5]
X = 3
Y = 4 ;

...
```

Question 1.2

State the solutions to the following query:

```
?- member(X,[1,2,3]), append(L,L,[X,X]), length(L,N), \+ member(N,L).
```

Question 1.3

Consider the following Prolog program:

```
main(N) :- member(L,[[2,1],[2],[],[1],[1,2]]), fun(L,N).

fun(L,N) :- member(N,L), !.
fun(_,-1).
```

State the solutions to the following query:

```
?- main(N).
```

The problem continues on the next page

Question 1.4

Write a deterministic Prolog program `expel(+N,+List,?Term,?Rest)` such that `Rest` is the list that remains when the `N`'th element `Term` is removed from `List` (the first element in the list is number 1):

```
?- expel(2,[a,b,c,d],X,R).

X = b
R = [a, c, d]

Yes

?- expel(5,[a,b,c,d],X,R).

No

?- expel(-1,[a,b,c,d],X,R).

No
```

Problem 2 (25%)

Consider the following formula: $\forall x(\neg p(a,b,x) \rightarrow \neg(\neg\exists x p(x,b,c) \wedge \forall x p(a,x,c)))$

Question 2.1

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

Question 2.2

Use refutation, skolemization and the general resolution procedure. State whether this shows that the formula is valid or not.

Problem 3 (25%)

Consider the following fragment of a database for items and names:

```
entry(item1,name1).
entry(item1,name2).
entry(item1,name3).
entry(item2,name4).
entry(item3,name1).
entry(item4,name1).
entry(item4,name4).
```

Question 3.1

Write a deterministic Prolog program `collect(?List)` that succeeds if and only if `List` is the sorted list of names in the database, hence for the above fragment for example:

```
?- collect(S).

S = [name1, name2, name3, name4]

Yes
```

Question 3.2

Write a deterministic Prolog program `details(+List)` that prints the full entry (both item and name) in the database for all items with a name not in `List`, hence for the above fragment for example:

```
?- details([name1,name2,name3]).
item2 name4
item4 name4

Yes

?- details([]).
item1 name1
item1 name2
item1 name3
item2 name4
item3 name1
item4 name1
item4 name4

Yes
```