TECHNICAL UNIVERSITY OF DENMARK (DTU)

Written Sample-Exam-7, 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 (40%)

Consider the following fragment of a word frequency list for a large English text:

```
w(5,2186369,a,det).
w(2107,4249,abandon,v).
w(5204,1110,abbey,n).
w(966,10468,ability,n).
w(321,30454,able,a).
w(6277,809,abnormal,a).
w(3862,1744,abolish,v).
w(5085,1154,abolition,n).
w(4341,1471,abortion,n).
w(179,52561,about,adv).
w(69,144554,about,prep).
w(3341,2139,above,a).
w(942,10719,above,adv).
w(786,12889,above,prep).
w(2236,3941,abroad,adv).
w(5106,1146,abruptly,adv).
```

The format is: `w(SortOrder,Frequency,Word,WordClass)`

`SortOrder` is `1` for the most frequent word. `WordClass` is the category: `det` for a determiner, `v` for verb, `n` for a noun, `a` for adjective, and so on.

Question 1.1

Write a deterministic Prolog program `extra(+WordClass1,+WordClass2)` that finds all words that are both of category `WordClass1` and category `WordClass2`, and for each such word prints a line for each additional category for the word (the line should consist of just the particular word and category), hence for the above fragment for example:

```
?- extra(adv,prep).
above a

Yes

?- extra(a,prep).
above adv

Yes

?- extra(a,det).

Yes
```

Question 1.2

A word is ambiguous if it appears in the word frequency list with more than one category.

Write a deterministic program `multi(?WordList)` that succeeds if and only if `WordList` is the sorted list of ambiguous words in the word frequency list, hence for the above fragment for example:

```
?- multi(S).

S = [about, above]

Yes
```

Question 1.3

Write a deterministic Prolog program `clist(?WordClassList)` that succeeds if and only if `WordClassList` is the sorted list of categories in the word frequency list, hence for the above fragment for example:

```
?- clist(S).

S = [a, adv, det, n, prep, v]

Yes
```

Problem 2 (25%)

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

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 (35%)

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 3.1

State the remaining solutions to the following query:

```
?- append([X|_],[X|_],[1,2,_,_,_]), length([X],X).

X = 1 ;

...
```

Question 3.2

State the solutions to the following query:

```
?- append(X,Y,[[1],[1,2]]), append(X,Y,_), member(Z,X).
```

Question 3.3

Consider the following Prolog program:

```
p(A,[A,A]) :- member(A,[0,1,2,3,4,5,6,7,8,9]).
p([],[]).
p([H|T],Z) :- p(H,X), p(T,Y), append(X,Y,Z).
```

State the solutions to the following query:

```
?- p([1,[4,2,[],5],[[3]]],L).
```