TECHNICAL UNIVERSITY OF DENMARK (DTU)

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%)

Consider the following Prolog program serving as a database of Danish names and their frequencies in a particular town (hence both `andersen` and `hansen` occur 888 times):

```
db(4,888,andersen).
db(3,888,hansen).
db(1,999,jensen).
db(12,199,johansen).
db(13,100,knudsen).
db(5,707,larsen).
db(8,404,madsen).
db(2,888,nielsen).
db(9,303,olsen).
db(7,505,petersen).
db(11,256,poulsen).
db(6,606,rasmussen).
db(10,256,thomsen).
```

The format is `db(Index,Frequency,Name)` where `Index` is `1` for the most frequent name, `2` for the second-most frequent name, and so on (names with the same frequency are indexed in an arbitrary way with respect to each other).

Question 1.1

Write a deterministic Prolog program `summarize(?Size)` that counts the number of names in the database as follows:

```
?- summarize(Size).

Size = 13

Yes
```

Question 1.2

Write a deterministic Prolog program `dump` that prints the frequency for each name in the database in sorted order as follows:

```
?- dump.
999 jensen
888 nielsen
888 hansen
888 andersen
707 larsen
606 rasmussen
505 petersen
404 madsen
303 olsen
256 thomsen
256 poulsen
199 johansen
100 knudsen

Yes
```

Question 1.3

Write a deterministic Prolog program `check1` that succeeds if and only if all names in the database are unique.

For example, changing `olsen` to `jensen` would make `check1` fail.

Question 1.4

Write a deterministic Prolog program `check2` that succeeds if and only if the indices in the database are correct in the sense that they correspond to the sorted order of the frequencies.

For example, changing `303` to `500` would make `check2` fail.

Question 1.5

Write a deterministic Prolog program `check3` that succeeds if and only if all indices in the database are unique and in sequence starting with `1`.

For example, changing `11` to `12` would make `check3` fail.

Problem 2 (25%)

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 2.1

Write a deterministic Prolog program `word(+Atom)` that succeeds if and only if `Atom` is a word in the word frequency list.

Question 2.2

Write a deterministic Prolog program `count(?Integer)` that succeeds if and only if `Integer` is the number of different categories in the word frequency list.

Question 2.3

Write a deterministic Prolog program `sum(?Integer)` that succeeds if and only if `Integer` is the sum of all frequencies in the word frequency list.

Problem 3 (25%)

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

Recall that CNF abbreviates Conjunctive Normal Form.

Question 3.1

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

Question 3.2

Which of the following statements are correct (explain why/why not)?

1. The formula p is in CNF.

2. The atoms $p(x)$ and $q(y)$ are unifiable.

3. The formula $p \vee q$ is in CNF.

4. The atoms $p(f(x), x)$ and $p(y, y)$ are unifiable.

5. The formula $(\neg p \vee \neg q) \wedge r$ is in CNF.