COMP348/1 – Sections AA – Summer 2019

ASSIGNMENT #1

Due: Sunday, May 26th, 2019 by midnight 11:59 PM

Important Note

- Submit only one file of code containing all these clauses and rules
- Your program should be compiled, executed and return the expected results; otherwise a mark 0 (zero) will be assigned.

Processing relational model in Prolog

A relational model of the data is based on table representation.

A data set can be represented as a list of items. Each item includes a certain number of fields: each field is identified by its name, a string of characters, and the value of a field can be typically of type: Boolean, String, Integer or Real.

Here are two examples of relations presented under form of tables:

"Name"	"Course"
"John"	"COMP232"
"Louise"	"COMP248"
"William"	"COMP348"
"William"	"COMP232"

Relation Registration

"Course"	"Prof"	"Local"
"COMP232"	"Tim W."	"H6010"
"COMP248"	"Louise L."	"H5605"
"COMP348"	"Mohamed T."	"H7610"

Relation Assignment

The objective is to represent and manipulate these relations in Prolog.

We admit to represent a relation with a predicate name relation, and having as arguments a list of column headers and a list of elements representing the contents of the table. The list of elements corresponds in fact to values of the fields of the table. Each element is represented by the predicate element with as argument a value of type integer, real or string of characters.

In addition, we admit that in valid relation, headers of the column do not contain repetitions (otherwise there would be a risk of ambiguity), each row has the same length of the list of headers of columns and the data type in a column is the same of one row to another. The predicate **verifRelation** presented in the file **C348SA1.pl** verifies whether a relation is well-formed.

REQUESTED WORK

You need to implement in Prolog two operations on relations, namely:

1. **projection(list of columns to remember, relation)** that produces a new relation by containing only the specified columns of the original relation. For example:

   relation1(Assignment), projection(["Course", "Local"], Assignment, RelationR).

   produces the relation "RelationR" with following values:

"Course"	"Local"
"COMP232"	"H6010"
"COMP248"	"H5605"
"COMP348"	"H7610"

2. **join(list of columns to be matched, relation1, relation2)** that produces a new relation obtained by welding rows together from the relation1 and relation2 that have identical values of fields in specified columns by taking care to remove columns that are repeated. For example:

   relation1(Registration), relation2(Assignment), join(['Course'], Registration, Assignment, RelationR).

   produces the relation "RelationR" with following values:

"Name"	"Course"	"Prof"	"Local"
"John"	"COMP232"	"Tim W."	"H6010"
"Louise"	"COMP248"	"Louise L."	"H5605"
"William"	"COMP348"	"Mohamed T."	"H7610"
"William"	"COMP232"	"Tim W."	"H6010"

```
/*******************************************************************************************/
/*                         COMPILATION - EXECUTION - RESULTS                               */
/*******************************************************************************************/
?- compile('prolog/C348SA1.pl').

% compiling file/usagers/prolog/C348SA1.pl

% C348SA1.pl compiled in module user, 1.490 sec 4,940 bytes

yes
```

?- **relation2(Assignment), projection(['Course', 'Local'], Assignment, RelationR).**

<u>RESULT:</u>

```
Assignment = relation(['Course', 'Prof', 'Local'],
                      [['COMP232','Tim W.','H6010'],
                      ['COMP248','Louise L.','H5605'],
                      ['COMP348','Mohamed T.','H7610']]),

RelationR = relation( ['Course', 'Local'],
                      [['COMP232','H6010'],
                      ['COMP248','H5605'],
                      ['COMP348', 'H7610']]).
```

?- **relation1(Registration), relation2(Assignment), join(['Course'], Registration, Assignment, RelationR).**

<u>RESULT:</u>

```
Registration = relation(['Name', 'Course'],
                      [['John', 'COMP232'],
                      ['Louise','COMP248'],
                      ['William','COMP348'],
                      ['William','COMP232']]),

Assignment = relation(['Course', 'Prof', 'Local'],
                      [['COMP232','Tim W.','X6010'],
                      ['COMP248','Louise L.','H5605'],
                      ['COMP348','Mohamed T.','H7610']]),

RelationR = relation(['Name', 'Course', 'Prof', 'Local'],
                      [['John', 'COMP232','Tim W.','H6010'],
                      ['Louise','COMP248','Louise L.','H5605'],
                      ['William','COMP348','Mohamed T.','H7610'],
                      ['William','COMP232','Tim W.','H6010']]).
```

Evaluation Criteria or Assignment #1 (7.5 points)

Assignment (7.5 points)	
Finding the arguments of clause Projection	2 pts.
Implementing the clause Projection	2 pts.
Finding the arguments of clause Join	1 pt.
Implementing the clause Join	1 pt.
Testing and validating the clause Projection	0.5 pt.
Testing and validating the clause Join	0.5 pt.
General correctness of code	0.5 pt.

EDUCATIONAL GUIDELINES

- The file **C348SA1.pl** contains a description of the structure and the main data to be manipulated in this assignment. It serves as a basis working and must be completed with the requested clauses.

- You must respect strictly the format of the requested predicates, including the number and names of their arguments.

- Your work is not to develop new rules in the program but rather to reuse rules that are provided in the program. Therefore, no new rule will be accepted.

- The work is realized in a team members 4 students.

- The delivery must be made no later than **Sunday, May 19th, 2019 by midnight 11:59 PM** using the Website submission as mentioned in the course outlines.