

# COMP2212 Programming Language Concepts Coursework

## Submission 2

Deadline: Thursday May 3, 4pm

### Additional Problems

#### Problem 6 - Check for emptiness

Assume two unary relation symbols  $R$  and  $S$  and compute

$$x_1 \vdash R(x_1) \wedge \exists z. S(z)$$

Example 1			Example 2		
R.csv	S.csv	Expected output	R.csv	S.csv	Expected output
Hello	World	Hello	1		
			2	empty	empty

#### Problem 7 - Paths of length three

Assume a single binary relation symbol  $R$  and compute

$$x_1, x_2 \vdash \exists z_1. \exists z_2. R(x_1, z_1) \wedge R(z_1, z_2) \wedge R(z_2, x_2)$$

Example 1		Example 2	
R.csv	Expected output	R.csv	Expected output
Southampton , Romsey	Romsey , Westbury Southampton , Warminster	Foo , Foo	Foo , Foo
Romsey , Salisbury			
Salisbury , Warminster			
Warminster , Westbury			

#### Problem 8 - Cycles of length 4

Assume a single binary relation symbol  $R$  and compute

$$x_1, x_2, x_3, x_4, x_5 \vdash R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_4) \wedge R(x_4, x_5) \wedge x_1 = x_5$$

Example 1		Example 2	
R.csv	Expected output	R.csv	Expected output
A,B	A,B,C,D,A	Foo , Foo	Foo , Foo , Foo , Foo , Foo
B,C	B,C,D,A,B		
C,D	C,D,A,B,C		
D,A	D,A,B,C,D		

### Problem 9 - Triple composition

Assume three binary relation symbols  $A$ ,  $B$  and  $C$  and compute

$$x_1, x_2 \vdash \exists z_1. \exists z_2. A(x_1, z_1) \wedge B(z_1, z_2) \wedge C(z_2, x_2)$$

Example 1 A.csv	B.csv	C.csv	Expected output
A,C B,C	C,D	D,E D,F	A,E A,F B,E B,F

### Problem 10 - Check for pairs

Assume a single tertiary relation symbol  $S$  and compute

$$x_1, x_2, x_3 \vdash S(x_1, x_2, x_3) \wedge (\exists z_1. \exists z_2. S(z_1, z_1, z_2)) \wedge (\exists z_1. \exists z_2. S(z_1, z_2, z_2))$$

Example 1 S.csv	Expected output	Example 2 S.csv	Expected output
A,A,B A,B,B A,B,C	A,A,B A,B,B A,B,C	A,B,B B,C,D	empty

## Submission instructions

You are required to submit two separate files.

First, you need to submit a zip file containing programs (`pr6.cql`, `pr7.cql`, `pr8.cql`, `pr9.cql`, `pr10.cql`) written in your language that solve the problems described above. We will run our tests on your solutions and award marks for solving these additional problems correctly. This will form 50% of the total coursework mark (10% each for the five problems). You have the option of resubmitting the code for your interpreter together with a makefile (see instructions for submission 1) for a 50% penalty on this component. Thus, if you decide to resubmit your interpreter in the this second submission the maximum possible total coursework mark is capped at 75%.

Second, you are required to submit a 3 page user manual for your language in pdf format that explains the syntax, and describes any additional features (e.g. comments, programmer convenience features, useful sugar, type checking, informative error messages, etc.). Please write the manual as if you were writing a tutorial for a programmer who wants to use your language: **do not** write a dry reference document. **Please include all 10 programs (from `pr1.cql` to `pr10.cql`) in the first appendix to your manual.** If there are any technical features that you would like to highlight in more detail, you can do so in additional, clearly marked appendices. There are no limits on the number and length of the appendices.

This report, together with your programs will be evaluated qualitatively for:

- expressivity of your language (25%)
- conciseness of your programs (25%)
- writing quality and clarity of exposition of your manual (25%)
- any additional programming language features (25%).

These qualitative aspects together are worth 30% of the total coursework mark.

As part of the second submission we require a declaration of how marks are to be distributed amongst the members of your group (e.g. 50-50, 40-60, etc.).