moodle_2023-24 Mani Sarthak

PROGRAMMING LANGUAGES

Dashboard ▶ My courses ▶ 2302-COL226 ▶ 11 January - 17 January ▶ Assignment 1 (a) Prolog

NAVIGATION

Dashboard

Site home

Site pages

My courses

2105-CURE04

2105-CURE03

2104-POSHS4

2302-CVL100C

2301-SBL100

2301-HUL212

2301-COL351

2301-COL334

2301-COL333

2301-COD310

2302-COL226

Participants

Assignment 1 (a) Prolog

- Part A: Specification

1. Specify the reflexive-transitive closure of a relation R over S (R is a subset of S X S) in terms of membership
2. Specify the reflexive-symmetric-transitive closure of a relation R over S (R is a subset of S X S) in terms of membership

- Part B: Implementation of Sets as Lists with no duplicates.

Consider the following programs in PROLOG (these can help and/or guid you)

```
/*  del(X,L1,L2) -- delete element X from a list L1 to obtain L2 */

del(X, [ ] , [ ]) :- !.

del(X, [X|R], Z) :- del(X, R, Z), !.

del(X, [Y|R], [Y|Z]) :- del(X, R, Z), !.


/*  remdups(L, L1) remove duplicates from a list L to get L1 */

remdups([ ], [ ]) :- !.

remdups([X|R], [X|Z]) :- del(X, R, L), remdups(L, Z).


/* Assuming no duplicates in S1, S2

 here is an implementation of union of S1, S2 */

unionI([ ], S2, S2) :- !.

unionI(S1, [ ], S1) :- !.
```

Competencies
Grades
General
28 December - 3 January
4 January - 10 January
11 January - 17 January
Assignment 1 (a) Prolog
18 January - 24 January
25 January - 31 January
1 February - 7 February
8 February - 14 February
15 February - 21 February
22 February - 28

```
unionl([X|R], S2, [X|Z]) :- del(X, S2, S3),  unionl(R, S3, Z).


/* append(L1, L2, L3) -- append lis  L1 to list L2 to get list  L3 */

append( [ ], L, L).

append( [X|R], L, [X|Z]) :- append(R, L, Z).


/* mapcons(X,L1, L2) --  cons the element X to each list in L1 to get L2 */

mapcons(X, [ ], [ ]) :- !.

mapcons(X, [Y|R], [ [X|Y] | Z ]) :- mapcons(X, R, Z).
```

/* powerI(S, P1): Here is an **implementation** of powerset of S */

```
powerI([ ], [ [ ] ]) :- !.

powerI([X|R], P) :- powerI(R, P1),  mapcons(X, P1, P2), append(P2, P1, P).
```

1. Check with sufficient examples that unionI and powerI indeed implement union and power.
2. Check that union does not have duplicates.
3. Assuming no duplicates in lists representing S1 and S2, write a PROLOG program interI(S1, S2, S3) that **implements intersection of two finite sets.**
4. Assuming no duplicates in lists representing S1 and S2, write a PROLOG program diffI(S1, S2, S3) that **implements set-difference of two finite sets.**
5. Assuming no duplicates in lists representing S1 and S2, write a PROLOG program cartesianI(S1, S2, S3) that **implements cartesian of two finite sets.**
6. Provide sufficient test cases examples to demonstrate your implementations are correct.
7. Suggest a way to check that the powersets obtained from the implementation of two different valid representations of a set (elements given in different order) are equal.

Submission status

Submission status	No attempt
Grading status	Not graded

- February
- 29 February - 6 March
- 7 March - 13 March
- 14 March - 20 March
- 21 March - 27 March
- 28 March - 3 April
- 4 April - 10 April
- 11 April - 17 April
- 18 April - 24 April
- 25 April - 1 May
- 2 May - 8 May
- 9 May - 15 May
- 16 May - 22 May
- 23 May - 29 May
- 30 May - 5 June
- 6 June - 12 June
- 13 June - 19 June
- More...

Due date	Thursday, 18 January 2024, 11:59 PM
Time remaining	6 days 15 hours
Last modified	-
Submission comments	▸ Comments (0)

Add submission

You have not made a submission yet

◀ Notes about Sets

Jump to...

You are logged in as Mani Sarthak (Log out)
2302-COL226

Get the mobile app