Practical 2b

Finding whether or not, a given relation is:
i.Reflexive
ii. Antisymmetric
iii. Transitive
iv. Partial order

1

A relation R on a set A is called symmetric
if (b, a) ∈ R whenever (a, b) ∈ R, for all a, b ∈ A.

A relation R on a set A such that for all a, b ∈ A,
if (a, b) ∈ R and (b, a) ∈ R, then a = b
is called antisymmetric.

1.1

```
→ kill(all);
(%o0) done
```

ex 7 : Rosen

```
→ A:makelist(k, k, 1, 4);
(%o1) [1,2,3,4]

→ R1:[[1, 1], [1, 2], [2, 1], [2, 2], [3, 4], [4, 1], [4, 4]];
(%o2) [[1,1],[1,2],[2,1],[2,2],[3,4],[4,1],[4,4]]

→ s:0;
  t:R1[1];
  if(member([t[2], t[1]], R1)) then (s:s+1);
(%o3) 0
(%o4) [1,1]
(%o5) 1

→ t:R1[2];
  if(member([t[2], t[1]], R1)) then (s:s+1);
(%o6) [1,2]
(%o7) 2
```

```
(%i8) t:R1[3];
      if(member([t[2], t[1]], R1)) then (s:s+1);
(%o8) [2,1]
(%o9) 3

(%i10) t:R1[2];
       if(member([t[2], t[1]], R1)) then (s:s+1);
(%o10) [1,2]
(%o11) 4

(%i12) checkSymmetric(A, R):=block(
         [s:0, t],
         for i:1 thru length(R) do(
             t:R[i],
             if(member([t[2], t[1]], R)) then (s:s+1)),
         if(s=length(R)) then return("symmetric") else return("Not symmetric")
       );
(%o12) checkSymmetric(A,R):=block([s:0,t],for i thru
       length(R) do (t:R_i,if member([t_2,t_1],R) then s:s+1 ),if s=
       length(R) then return(symmetric) else return(Not symmetric))

(%i13) checkSymmetric(A, R1);
(%o13) Not symmetric

(%i14) R2:[[1, 1], [1, 2], [2, 1]];
(%o14) [[1,1],[1,2],[2,1]]

(%i15) checkSymmetric(A, R2);
(%o15) symmetric

(%i16) R3:[[1, 1], [1, 2], [1, 4], [2, 1], [2, 2], [3, 3], [4, 1], [4, 4]];
(%o16) [[1,1],[1,2],[1,4],[2,1],[2,2],[3,3],[4,1],[4,4]]

(%i17) checkSymmetric(A, R3);
(%o17) symmetric

(%i18) R4:[[2, 1], [3, 1], [3, 2], [4, 1], [4, 2], [4, 3]];
(%o18) [[2,1],[3,1],[3,2],[4,1],[4,2],[4,3]]

(%i19) R5:[[1, 1], [1, 2], [1, 3], [1, 4], [2, 2], [2, 3], [2, 4], [3, 3], [3, 4], [4, 4]];
(%o19) [[1,1],[1,2],[1,3],[1,4],[2,2],[2,3],[2,4],[3,3],[
       3,4],[4,4]]

(%i20) R6:[[3, 4]];
(%o20) [[3,4]]
```

```
→ checkSymmetric(A, R4);
(%o21) Not symmetric

→ checkSymmetric(A, R5);
(%o22) Not symmetric

→ checkSymmetric(A, R6);
(%o23) Not symmetric
```

1.2

```
→ kill(all);
(%o0) done
```

ex 7 : Rosen

```
→ A:makelist(k, k, 1, 4);
(%o1) [1,2,3,4]

→ R1:[[1, 1], [1, 2], [2, 1], [2, 2], [3, 4], [4, 1], [4, 4]];
(%o2) [[1,1],[1,2],[2,1],[2,2],[3,4],[4,1],[4,4]]

→ R2:[[1, 1], [1, 2], [2, 1]];
(%o3) [[1,1],[1,2],[2,1]]

→ R3: [[1, 1], [1, 2], [1, 4], [2, 1], [2, 2], [3, 3], [4, 1], [4, 4]];
(%o4) [[1,1],[1,2],[1,4],[2,1],[2,2],[3,3],[4,1],[4,4]]

→ R4:[[2, 1], [3, 1], [3, 2], [4, 1], [4, 2], [4, 3]];
(%o5) [[2,1],[3,1],[3,2],[4,1],[4,2],[4,3]]

→ R5:[[1, 1], [1, 2], [1, 3], [1, 4], [2, 2], [2, 3], [2, 4], [3, 3], [3, 4], [4, 4]];
(%o6) [[1,1],[1,2],[1,3],[1,4],[2,2],[2,3],[2,4],[3,3],[
3,4],[4,4]]

→ R6:[[3, 4]];
(%o7) [[3,4]]

→ R7:[[1, 1], [3, 4]];
(%o8) [[1,1],[3,4]]

→ R8:cartesian_product_list(A, A);
(%o9) [[1,1],[1,2],[1,3],[1,4],[2,1],[2,2],[2,3],[2,4],[
3,1],[3,2],[3,3],[3,4],[4,1],[4,2],[4,3],[4,4]]
```

```
checkAntiSymmetric(A,  R):=block(
   [s:0,  t],
  for i:1 thru length(R) do(
      t:R[i],
      if( is(t[1]#t[2])) then(if(member([t[1],  t[2]],  R) and member([t[2],  t[1]],  R)
  if(s=0) then return("Anti symmetric") else return("Not Anti symmetric")
);
```

(%o10) checkAntiSymmetric(A,R):=block([s:0,t],for i thru length(R) do (t:R$_i$,if is(t$_1$≠t$_2$) then if member([t$_1$,t$_2$],R) ∧ member([t$_2$,t$_1$],R) then s:s+1),if s=0 then return(*Anti symmetric*) else return(*Not Anti symmetric*))

1.3

R1, R2, R3 are not anti symmetric :

This is done by finding a pair (a, b) with
a ≠ b such that (a, b) and (b, a) are both in the relation.

```
checkAntiSymmetric(A,  R1);
```

(%o11) *Not Anti symmetric*

```
checkAntiSymmetric(A,  R2);
```

(%o12) *Not Anti symmetric*

```
checkAntiSymmetric(A,  R3);
```

(%o13) *Not Anti symmetric*

1.4

R4, R5, R6 are anti symmetric :

For each of these relations there is no pair of elements
a and b with a ≠ b such that both (a, b) and (b, a)
belong to the relation.

```
checkAntiSymmetric(A,  R4);
```

(%o14) *Anti symmetric*

```
checkAntiSymmetric(A,  R5);
```

(%o15) *Anti symmetric*

```
checkAntiSymmetric(A,  R6);
```

(%o16) *Anti symmetric*

1.5

```
(%i17) checkAntiSymmetric(A, R7);
```
(%o17) *Anti symmetric*

```
(%i18) checkAntiSymmetric(A, R8);
```
(%o18) *Not Anti symmetric*