```
1  ---------- MODULE RelationUtils ----------
   Relation related operators.
5  LOCAL INSTANCE Naturals
6  LOCAL INSTANCE FiniteSets
7  LOCAL INSTANCE Sequences
8  LOCAL INSTANCE SequencesExt
9  LOCAL INSTANCE Functions
10 ----------
   Basic definitions.
14 Dom(R) ≜ {a : ⟨a, b⟩ ∈ R}  Domain of R
15 Ran(R) ≜ {b : ⟨a, b⟩ ∈ R}  Range of R
16 Support(R) ≜ Dom(R) ∪ Ran(R)  Support of R
17 ----------
   Basic operations.
21 Image(R, a) ≜ {b ∈ Ran(R) : ⟨a, b⟩ ∈ R}
22 LeftRestriction(R, a) ≜ {⟨a, b⟩ : b ∈ Image(R, a)}

24 InverseRelation(R) ≜ {⟨b, a⟩ : ⟨a, b⟩ ∈ R}
25 InverseImage(R, b) ≜ {a ∈ Dom(R) : ⟨a, b⟩ ∈ R}

27 R | S ≜ R ∩ (S × S)  Restriction of R on S

29 R ∗∗ T ≜  Composition of R and T
30     LET SR ≜ Support(R)
31         ST ≜ Support(T)
32     IN  {⟨r, t⟩ ∈ SR × ST : ∃ s ∈ SR ∩ ST : (⟨r, s⟩ ∈ R) ∧ (⟨s, t⟩ ∈ T)}

34 GT(R, a) ≜ {b ∈ Ran(R) : ⟨a, b⟩ ∈ R}  ≜ Image(R, a)
35 LT(R, b) ≜ {a ∈ Dom(R) : ⟨a, b⟩ ∈ R}  ≜ InverseImage(R, b)
   The following definition is from https://github.com/jameshfisher/tlaplus/blob/master/examples/TransitiveClosure/TransitiveClos
   It also contains several other methods for computing TC.
41 TC(R) ≜  Transitive closure of R
42     LET S ≜ Support(R)
43         RECURSIVE TCR(_)
44         TCR(T) ≜ IF T = {}
45                  THEN R
46                  ELSE LET r ≜ CHOOSE s ∈ T : TRUE
47                           RR ≜ TCR(T \ {r})
48                       IN  RR ∪ {⟨s, t⟩ ∈ S × S :
49                                ⟨s, r⟩ ∈ RR ∧ ⟨r, t⟩ ∈ RR}
50     IN  TCR(S)
   Example: SeqToRel(⟨1, 2, 3⟩) = {⟨1, 2⟩, ⟨1, 3⟩, ⟨2, 3⟩}
54 RECURSIVE Seq2Rel(_)
55 Seq2Rel(s) ≜  Transform a sequence s into a strict total order relation
56     IF s = ⟨⟩ THEN {}
```

```
57      ELSE  LET h ≜ Head(s)
58                t ≜ Tail(s)
59            IN  {⟨h, r⟩ : r ∈ Range(t)} ∪ Seq2Rel(t)
60 ----------------------------------------------------------------------------
```

Basic properties.

```
64  IsReflexive(R, S) ≜ ∀ a ∈ S : ⟨a, a⟩ ∈ R
65  IsIrreflexive(R, S) ≜ ∀ a ∈ S : ⟨a, a⟩ ∉ R

67  IsSymmetric(R, S) ≜ ∀ a, b ∈ S : ⟨a, b⟩ ∈ R ≡ ⟨b, a⟩ ∈ R
68  IsAntisymmetric(R, S) ≜ ∀ a, b ∈ S : ⟨a, b⟩ ∈ R ∧ ⟨b, a⟩ ∈ R ⇒ a = b

70  IsTransitive(R, S) ≜
71      ∀ a, b, c ∈ S : (⟨a, b⟩ ∈ R ∧ ⟨b, c⟩ ∈ R) ⇒ ⟨a, c⟩ ∈ R

73  IsTotal(R, S) ≜
74      ∀ a, b ∈ S : ⟨a, b⟩ ∈ R ∨ ⟨b, a⟩ ∈ R

76  IsPartialOrder(R, S) ≜
77          ∧ IsReflexive(R, S)
78          ∧ IsAntisymmetric(R, S)
79          ∧ IsTransitive(R, S)

81  IsTotalOrder(R, S) ≜
82          ∧ IsPartialOrder(R, S)
83          ∧ IsTotal(R, S)

85  IsStrictPartialOrder(R, S) ≜
86          ∧ IsIrreflexive(R, S)
87          ∧ IsTransitive(R, S)

89  IsStrictTotalOrder(R, S) ≜
90          ∧ IsStrictPartialOrder(R, S)
91          ∧ IsTotal(R, S)

93  Respect(R, T) ≜ T ⊆ R  Does R respect T?
94 ----------------------------------------------------------------------------
```

Special elements in a relation

```
98  Minimal(R, S) ≜  the set of minimal elements in relation R on the set S
99      {m ∈ S : ¬∃ a ∈ Dom(R) : ⟨a, m⟩ ∈ R}
100 Maximal(R, S) ≜  the set of maximal elements in relation R on the set S
101     {m ∈ S : ¬∃ b ∈ Ran(R) : ⟨m, b⟩ ∈ R}
102 ----------------------------------------------------------------------------
```

A variant of *Kahn*'s algorithm for topological sorting

See https://*en.wikipedia.org*/wiki/*Topological_sorting* ≠ *Kahn*'s_algorithm

```
108 Cyclic(R) ≜  Is R cyclic?
109     LET RECURSIVE CyclicUtil(_, _)
110             CyclicUtil(rel, set) ≜  remaining relation; set: remaining set
```

```
                IF set = {} THEN FALSE
                 ELSE LET mins ≜ Minimal(rel, set)
                          IN IF mins = {} THEN TRUE
                                ELSE LET m ≜ CHOOSE x ∈ mins : TRUE
                                         IN CyclicUtil(rel \ LeftRestriction(R, m), set \ {m})
    IN  CyclicUtil(R, Support(R))
```

Kahn's algorithm for topological sorting.

See https://*en.wikipedia.org*/wiki/*Topological_sorting* ≠ *Kahn*'s_algorithm

```
AnyLinearExtension(R, S) ≜  return an arbitrary linear extension of R on the set S
    LET RECURSIVE LinearExtensionUtil(_, _)
        LinearExtensionUtil(rel, set) ≜  rel: remaining relation; set: remaining set
            IF set = {} THEN ⟨⟩
             ELSE LET m ≜ CHOOSE x ∈ Minimal(rel, set) : TRUE
                      IN ⟨m⟩ ∘ LinearExtensionUtil(rel \ LeftRestriction(R, m), set \ {m})
    IN  LinearExtensionUtil(R, S)
```

A variant of *Kahn*'s algorithm for topological sorting

See https://*en.wikipedia.org*/wiki/*Topological_sorting* ≠ *Kahn*'s_algorithm

For some TLA+ issue, see https://*groups.google.com*/g/tlaplus/*c*/*mtyEmqhlRVg*

```
AllLinearExtensions(R, S) ≜  return all possible linear extensions of R on the set S
    LET RECURSIVE LinearExtensionsUtil(_, _)
        LinearExtensionsUtil(rel, set) ≜
            IF set = {} THEN {⟨⟩}
             ELSE LET Extend(m) ≜ {⟨m⟩ ∘ l :  extend recursively by the minimal element m
                                  l ∈ LinearExtensionsUtil(rel \ LeftRestriction(R, m), set \ {m})}
                      IN UNION {Extend(m) : m ∈ Minimal(rel, set)}  for each minimal element
    IN  LinearExtensionsUtil(R, S)

LinearExtensions(R, S) ≜  return the set of all possible linear extensions of R on the set S
    {l ∈ TupleOf(S, Cardinality(S)) : Respect(Seq2Rel(l), R)}
```

```
\* Modification History
\* Last modified Mon Apr 19 19:48:29 CST 2021 by hengxin
\* Created Tue Sep 18 19:16:04 CST 2018 by hengxin
```