```
──────────────────────── MODULE RelationUtils ────────────────────────
(* Relation related operators. *)

LOCAL INSTANCE Naturals
LOCAL INSTANCE FiniteSets
LOCAL INSTANCE Sequences
LOCAL INSTANCE SequencesExt
LOCAL INSTANCE Functions
──────────────────────────────────────────────────────────────────────
(* Basic definitions. *)

Dom(R)     ≜ {a : ⟨a, b⟩ ∈ R}   (* Domain of R *)
Ran(R)     ≜ {b : ⟨a, b⟩ ∈ R}   (* Range of R *)
Support(R) ≜ Dom(R) ∪ Ran(R)     (* Support of R *)
──────────────────────────────────────────────────────────────────────
(* Basic operations. *)

Image(R, a) ≜ {b ∈ Ran(R) : ⟨a, b⟩ ∈ R}
LeftRestriction(R, a) ≜ {⟨a, b⟩ : b    ∈ Image(R, a)}

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

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

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

GT(R, a) ≜ {b ∈ Ran(R)  : ⟨a, b⟩ ∈ R}   (* ≜ Image(R, a) *)
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. *)
TC(R) ≜   (* Transitive closure of R *)
    LET S ≜ Support(R)
        RECURSIVE TCR(_)
        TCR(T) ≜ IF T = {}
                   THEN R
                   ELSE LET r  ≜ CHOOSE s ∈ T : TRUE
                            RR ≜ TCR(T \ {r})
                        IN  RR ∪ {⟨s, t⟩ ∈ S × S :
                                   ⟨s, r⟩ ∈ RR ∧ ⟨r, t⟩ ∈ RR}
    IN  TCR(S)

(* Example: SeqToRel(⟨1, 2, 3⟩) = {⟨1, 2⟩, ⟨1, 3⟩, ⟨2, 3⟩} *)
RECURSIVE Seq2Rel(_)
Seq2Rel(s) ≜   (* Transform a sequence s into a strict total order relation *)
    IF s = ⟨⟩ THEN {}
```

$\quad\quad \text{ELSE} \; \text{LET} \; h \triangleq Head(s)$

$\quad\quad\quad\quad\quad\quad t \triangleq Tail(s)$

$\quad\quad\quad\quad \text{IN} \; \{\langle h, r\rangle : r \in Range(t)\} \cup Seq2Rel(t)$

---

Basic properties.

$IsReflexive(R, S) \triangleq \forall a \in S : \langle a, a\rangle \in R$

$IsIrreflexive(R, S) \triangleq \forall a \in S : \langle a, a\rangle \notin R$

$IsSymmetric(R, S) \triangleq \forall a, b \in S : \langle a, b\rangle \in R \equiv \langle b, a\rangle \in R$

$IsAntisymmetric(R, S) \triangleq \forall a, b \in S : \langle a, b\rangle \in R \land \langle b, a\rangle \in R \Rightarrow a = b$

$IsTransitive(R, S) \triangleq$

$\quad \forall a, b, c \in S : (\langle a, b\rangle \in R \land \langle b, c\rangle \in R) \Rightarrow \langle a, c\rangle \in R$

$IsTotal(R, S) \triangleq$

$\quad \forall a, b \in S : \langle a, b\rangle \in R \lor \langle b, a\rangle \in R$

$IsSemiconnex(R, S) \triangleq$

$\quad \forall a, b \in S : a \neq b \Rightarrow (\langle a, b\rangle \in R \lor \langle b, a\rangle \in R)$

$IsPartialOrder(R, S) \triangleq$

$\quad \land IsReflexive(R, S)$

$\quad \land IsAntisymmetric(R, S)$

$\quad \land IsTransitive(R, S)$

$IsTotalOrder(R, S) \triangleq$

$\quad \land IsPartialOrder(R, S)$

$\quad \land IsTotal(R, S)$

$IsStrictPartialOrder(R, S) \triangleq$

$\quad \land IsIrreflexive(R, S)$

$\quad \land IsTransitive(R, S)$

$IsStrictTotalOrder(R, S) \triangleq$

$\quad \land IsStrictPartialOrder(R, S)$

$\quad \land IsSemiconnex(R, S)$

$Respect(R, T) \triangleq T \subseteq R$ Does $R$ respect $T$?

---

Special elements in a relation

$Minimal(R, S) \triangleq$ the set of minimal elements in relation $R$ on the set $S$

$\quad \{m \in S : \neg \exists a \in Dom(R) : \langle a, m\rangle \in R\}$

$Maximal(R, S) \triangleq$ the set of maximal elements in relation $R$ on the set $S$

$\quad \{m \in S : \neg \exists b \in Ran(R) : \langle m, b\rangle \in R\}$

---

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

See https://*en.wikipedia.org*/wiki/*Topological_sorting* $\neq$ *Kahn*'s_algorithm

```
Cyclic(R) ≜   Is R cyclic?
    LET RECURSIVE CyclicUtil(_, _)
          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 Thu Apr 22 14:56:42 CST 2021 by hengxin
\* Created Tue Sep 18 19:16:04 CST 2018 by hengxin
```