```
library librarySet

from Basic/Numbers get Int

spec Set0[sort Elem] =
        generated type Set[Elem] ::= setVide | ajouter(Set[Elem]; Elem)
        pred appartient : Elem * Set[Elem]
        forall x, y : Elem; M, N : Set[Elem]
        . not appartient(x, setVide)
        . appartient(x, ajouter(M, y)) <=> x = y \/ appartient(x, M)
        . M = N
          <=> forall x : Elem
                        . appartient(x, M) <=> appartient(x, N)
end

spec Set[sort Elem] =
        Set0[sort Elem]
then preds estVide : Set[Elem];
                inclut : Set[Elem] * Set[Elem]
        op      enlever : Set[Elem] * Elem -> Set[Elem]
        forall x, y : Elem; M, N : Set[Elem]
        . estVide(M) <=> M = setVide
        . inclut(M, N)
          <=> forall x : Elem . appartient(x, M) => appartient(x, N)
        . enlever(setVide, y) = setVide
        . enlever(ajouter(M, x), y)
          = enlever(M, y) when x = y else ajouter(enlever(M, y), x)
end
```