\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod NATURAL is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

sort N .

sort NoN .

subsort NoN < N .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

op cero : -> N .

op sucesor : N -> N .

op suma : N N -> N .

op esCero : N -> Bool .

op esIgual : N N -> Bool .

op esDistinto : N N -> Bool .

op NODEFINIDO : -> NoN .

op INFINITO : -> NoN .

op NEGATIVO : -> NoN .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var n m : N .

eq suma(cero, n) = n .

eq suma(sucesor(m), n) = sucesor(suma(m, n)) .

eq esCero(cero) = true .

eq esCero(sucesor(n)) = false .

eq esIgual(cero, n) = esCero(n) .

eq esIgual(sucesor(n), cero) = false .

eq esIgual(sucesor(n), sucesor(m)) = esIgual(n, m) .

eq esDistinto(n, m) = not esIgual(n, m) .

endfm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod VOCAL is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

sort V .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ops A E I O U : -> V .

op esIgual : V V -> Bool .

op esDistinta : V V -> Bool .

op esMenor : V V -> Bool .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var v w : V .

eq esIgual(v, v) = true .

eq esIgual(v, w) = false .

eq esDistinta(v, w) = not esIgual(v, w) .

eq esMenor(v, v) = false .

eq esMenor(A, E) = true .

eq esMenor(A, I) = true .

eq esMenor(A, O) = true .

eq esMenor(A, U) = true .

eq esMenor(E, I) = true .

eq esMenor(E, O) = true .

eq esMenor(E, U) = true .

eq esMenor(I, O) = true .

eq esMenor(I, U) = true .

eq esMenor(O, U) = true .

eq esMenor(U, O) = false .

eq esMenor(U, I) = false .

eq esMenor(U, E) = false .

eq esMenor(U, A) = false .

eq esMenor(O, I) = false .

eq esMenor(O, E) = false .

eq esMenor(O, A) = false .

eq esMenor(I, E) = false .

eq esMenor(I, A) = false .

eq esMenor(E, A) = false .

endfm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod CONJUNTO is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

protecting NATURAL .

protecting VOCAL .

sort C .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

op cjtoVacio : -> C .

op esVacio : C -> Bool .

op insertar : V C -> C .

op esMiembro : V C -> Bool .

op tamano : C -> N .

op union : C C -> C .

op interseccion : C C -> C .

op diferencia : C C -> C .

op eliminar : V C -> C .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var v w t r : V .

var c d f : C .

eq esVacio(cjtoVacio) = true .

eq esVacio(insertar(v, c)) = false .

eq esMiembro(v, cjtoVacio) = false .

eq esMiembro(v, insertar(w, c)) = esIgual(v, w) or esMiembro(v, c) .

eq tamano(cjtoVacio) = cero .

eq tamano(insertar(v, c)) = suma(tamano(c), if esMiembro(v, c) then cero else sucesor(cero) fi ) .

eq union(cjtoVacio, cjtoVacio) = cjtoVacio .

eq union(cjtoVacio, c) = c .

eq union(insertar(v, c), d) = insertar(v, union(c, d)) .

eq interseccion(cjtoVacio, c) = cjtoVacio .

eq interseccion(insertar(v, c), d) = if esMiembro(v, d) == true then insertar(v, interseccion(c, d)) else interseccion (c, d) fi .

eq diferencia(cjtoVacio, cjtoVacio) = cjtoVacio .

eq diferencia(c, cjtoVacio) = c .

eq diferencia(c, insertar(v, d)) = diferencia(eliminar(v, c), d) .

eq eliminar(v, cjtoVacio) = cjtoVacio .

eq eliminar(v, insertar(w, c)) = if esIgual(v, w) then eliminar(v, c) else insertar(w, eliminar(v, c)) fi .

endfm