

```

proc esSeñal (in s: seq(Z), in prof: Z, in freq: Z, out result: Bool) {
  Pre { |s| ≥ 0 ∧ prof > 0 ∧ freq > 0 }
  Post {
    result = frecuenciaEnRango(freq) ∧L
    profundidadCorrecta(s) ∧L
    ningunaMuestraSuperaLaProfundidad(s, prof) ∧L
    duraMasDeUnSegundo(s, freq) }
}

pred profundidadCorrecta (prof: Z) { freq ∈ [8, 16, 32] }
pred frecuenciaEnRango (freq: Z) { freq ∈ [8, 32] }
pred ningunaMuestraSuperaLaProfundidad (s: seq(Z), p: Z) {
  (∀0 : Z) i ≤ 0 < |s|
  →L (-2)p-1 ≤ s[i] ≤ 2p-1 - 1
}
pred duraMasDeUnSegundo (s: seq(Z), freq: Z) {  $\frac{|s|}{(freq \cdot 1000)} \geq 1$  }
pred esSeñalAux (s: seq(Z), prof: Z, freq: Z) {
  |s| ≥ 0 ∧L
  frecuenciaEnRango(freq) ∧L
  profundidadCorrecta(s) ∧L
  ningunaMuestraSuperaLaProfundidad(s, prof) ∧L
  duraMasDeUnSegundo(s, freq)
}

proc seEnojó? (in s: señal, in umbral: Z, in prof: Z, in freq: Z, out result: Bool) {
  Pre { umbral > 0 ∧L esSeñalAux(s, prof, freq) }
  Post {
    result = umbralEnRango(umbral, prof) ∧L
    existeUnaSubsecuenciaQueSuperaUmbral(s, freq, umbral) }
}

pred umbralEnRango (umbral: Z, p: Z) { umbral ≥ 2p-1 - 1 }
pred existeUnaSubsecuenciaQueSuperaUmbral (s: señal, freq: Z, u: Z) {
  (∃d, h : Z) 0 ≤ d, h < |s| ∧ (d < h) ∧ ((d + freq * 1000 * 5) < h) ∧L (
  (∀i : Z) 0 ≤ i < |subseq(s, d, h)| →L abs(subseq(s, d, h)[i]) > umbral) }
fun abs (x: Z) : Z = if x > 0 then x else -x fi;

proc esReuniónVálida? (in r: reunion, in prof: Z, in freq: Z, out result: Bool) {
  Pre { |r| > 0 ∧ prof > 0 ∧ freq > 0 }
  Post {
    result = contieneSeñalesValidas(r, prof, freq) ∧L
    lasLongitudesDeSeñalSonIguales(r) ∧L
    todosHablantesDistintos(r) ∧L
    losHablantesEstanEnRangosDe0ANMenos1(r) }
}

pred contieneSeñalesValidas (r: reunion, prof: Z, freq: Z) { (∀i : Z) 0 ≤ i < |r| →L esSeñalAux(r[i]0, prof, freq) }
pred lasLongitudesDeSeñalSonIguales (r: reunion) { (∀i, j : Z) 0 ≤ i, j < |r| ∧ i ≠ j →L (|r[i]0| = |r[j]0|) }
pred todosHablantesDistintos (r: reunion) { (∀i, j : Z) 0 ≤ i, j < |r| ∧ i ≠ j →L (r[i]1 ≠ r[j]1) }
pred losHablantesEstanEnRangosDe0ANMenos1 (r: reunion) { (∀i : Z) 0 ≤ i < |r| →L 0 ≤ r[i]1 < |r| }
pred esReuniónVálidaAux (r: reunion, prof: Z, freq: Z) {
  contieneSeñalesValidas(r, prof, freq) ∧L
  lasLongitudesDeSeñalSonIguales(r) ∧L
  todosHablantesDistintos(r) ∧L
  losHablantesEstanEnRangosDe0ANMenos1(r) }

proc acelerar (inout r: reunion, in prof: Z, in freq: Z) {
  Pre { esReuniónVálidaAux(r, prof, freq) ∧ r0 = r }
  Post {
    esReuniónVálidaAux(r, prof, freq) ∧L
    |r| = |r0| ∧L

```

```

    lasSeñalesTieneLaMitadDeMuestras(r, r0) ∧L
    losImpares(r, r0)}
}

pred lasSeñalesTieneLaMitadDeMuestras (r: reunion, r0 : reunion){
  (∀i : ℤ) 0 ≤ i < |r| →L if esPar(|r[0]0|) then |r[i]0| =  $\frac{|r[0]_0|}{2}$  else |r[i]0| =  $\frac{|r[0]_0|-1}{2}$  fi
}

pred losImpares (r: reunion, r0 : reunion){(∀i : ℤ) 0 ≤ i < |r| →L (
  (∃j : ℤ) 0 ≤ j < |r| ∧L (r[i]1 = r[j]1) ∧L (
  (∀q : ℤ) 0 ≤ q < |r0[i]0| ∧ (¬esPar(q)) →L (r0[i]0[q] = r[j]0[ $\frac{q-1}{2}$ ]))))}

proc ralentizar (inout r: reunion, in prof: ℤ, in freq: ℤ) {
  Pre {esReuniónVálidaAux(r, prof, freq) ∧ r0 = r}
  Post {
    esReuniónVálidaAux(r, prof, freq) ∧L
    |r| = |r0| ∧L
    lasSeñalesTienenElDobleDeMuestras(r, r0) ∧L
    promedioEntrePares(r, r0)}
}

pred lasSeñalesTienenElDobleDeMuestras (r: reunion, rv : reunion){
  (∀i : ℤ) 0 ≤ i < |rv| →L (2 · |rv[i]0| = (|r[i]0| + 1))}
pred promedioEntrePares (r: reunion, rv : reunion){
  (∀i : ℤ) 0 ≤ i < |r| →L (
  (∃j : ℤ) 0 ≤ j < |r0| ∧L (r[i]1 = rv[j]1) ∧L (
  (∀q : ℤ) 0 ≤ q < |r[i]0| →L
  if esPar(q) then r[j]0[q] = rv[i]0[ $\frac{q}{2}$ ] else r[j]0[q] =  $\frac{r_v[i]_0[\frac{q-1}{2}] + r_v[i]_0[\frac{q+1}{2}]}{2}$  fi ) ) }

proc tonosDeVozElevados (inout r: reunion, in freq: ℤ, in prof: ℤ, out hablantes: seq(hablante)) {
  Pre {esReuniónVálidaAux(r, prof, freq)}
  Post {
    siPertenecenAHablantesElPromedioDeAmplitudEsMasGrandeOIgualQueElResto(r, hablantes) ∧L
    losHablantesPertenecenALaReunión(r, hablantes) ∧L
    losHablantesNoSeRepiten(hablantes)}
}

pred siPertenecenAHablantesElPromedioDeAmplitudEsMasGrandeOIgualQueElResto (r: reunion, hs: seq(hablante)) {
  (∀i : ℤ) 0 ≤ i < |hs| →L
  (r[i]1 ∈ hs ∧ elPromedioDeAmplitudEsMasGrandeOIgualQueElResto(r, r[i]0))
  ∨
  (r[i]1 ∉ hs ∧ ¬elPromedioDeAmplitudEsMasGrandeOIgualQueElResto(r, r[i]0))
}

pred losHablantesPertenecenALaReunión (r: reunion, hs: seq(hablante)) {
  (∀i : ℤ) 0 ≤ i < |hs| →L ((∃j : ℤ) 0 ≤ j < |r| ∧L (hs[i] = r[j]1))}
pred losHablantesNoSeRepiten (r: reunion, hs: seq(hablante)) {
  (∀i : ℤ) 0 ≤ i < |hs| →L (#apariciones(hs, hs[i]) = 1)}
pred elPromedioDeAmplitudEsMasGrandeOIgualQueElResto (r: reunion, s: señal) {
  (∀i : ℤ) 0 ≤ i < |r| →L (tonoDeVoz(s) ≥ tonoDeVoz(r[i]0))}
fun tonoDeVoz (s: señal) : ℤ = sumaDelValorAbsolutoDeAmplitudes(s) div |s|;
fun sumaDelValorAbsolutoDeAmplitudes (s: señal) : ℤ =  $\sum_{i=0}^{|s|} abs(s[i])$ ;

proc ordenar (inout r: reunion, in freq: ℤ, in prof: ℤ) {
  Pre {esReuniónVálidaAux(r, prof, freq) ∧ r0 = r}
  Post {
    esReuniónVálidaAux(r, prof, freq) ∧L
    ordenadaDeMayorAMenorPorTonoDeVoz(r) ∧L
    esUnaPermutación(r0, r)}
}

pred ordenadaDeMayorAMenorPorTonoDeVoz (r: reunion) {
  (∀i : ℤ) 1 ≤ i < |r| →L tonoDeVoz(r[i - 1]0) ≥ tonoDeVoz(r[i]0)
}
pred esUnaPermutación (x: reunion, y: reunion) {

```

```

 $|x| = |y| \wedge_L$ 
 $(\forall i : \mathbb{Z}) 0 \leq i < |x| \longrightarrow_L ($ 
 $(\exists j : \mathbb{Z}) 0 \leq j < |y| \wedge_L (x[i]_1 = y[j]_1) \wedge_L (x[i]_0 = y[j]_0)))$ 
fun tonoDeVoz (s: señal) :  $\mathbb{Z} = \text{sumaDelValorAbsolutoDeAmplitudes}(s) \text{div} |s|$ ;
fun sumaDelValorAbsolutoDeAmplitudes (s: señal) :  $\mathbb{Z} = \sum_{i=0}^{|s|} \text{abs}(s[i])$ ;

proc silencios (in s: señal, in freq:  $\mathbb{Z}$ , in prof:  $\mathbb{Z}$ , out intervalos:  $\text{seq}(\text{intervalo})$ ) {
  Pre {esSeñalAux(s, prof, freq)  $\wedge$  (umbral > 0)}
  Post {
    noHayIntervalosRepetidos(intervalos)  $\wedge_L$ 
     $(\forall i : \mathbb{Z}) 0 \leq i < |\text{intervalos}| \longrightarrow_L ($ 
    esSilencio(s, umbral, freq, intervalos[i]))
  }
}

pred noHayIntervalosRepetidos (ins:  $\text{seq}(\text{intervalo})$ ) { $(\forall i : \mathbb{Z}) 0 \leq i < |\text{ins}| \longrightarrow_L (\#apariciones(\text{ins}, i) = 1)$ }
pred finEsMayorQueInicio (inicio:  $\mathbb{Z}$ , fin:  $\mathbb{Z}$ ) {fin > inicio}
pred estaDentroDeLaSeñal (s: señal, inicio:  $\mathbb{Z}$ , fin:  $\mathbb{Z}$ ) {(inicio  $\geq$  0)  $\wedge$  (fin < |s|)}
pred esAlMenosUnDecimoDeSegundo (freq:  $\mathbb{Z}$ , inicio:  $\mathbb{Z}$ , fin:  $\mathbb{Z}$ ) {(fin - inicio + 1)  $\geq$  (frecuencia * 100)}
pred losAdyacentesSuperanElUmbral (s: señal, umbral:  $\mathbb{Z}$ , inicio:  $\mathbb{Z}$ , fin:  $\mathbb{Z}$ ) {
  (inicio = 0)  $\vee_L ((\text{inicio} - 1 \geq 0) \wedge_L (s[\text{inicio} - 1] \geq \text{umbral})) \wedge$ 
  (fin = |s| - 1)  $\vee_L ((\text{fin} + 1 < |s|) \wedge_L (s[\text{fin} + 1] \geq \text{umbral}))$ 
}
pred entreIndicesNoPasaCiertUmbral (s: señal, umbral:  $\mathbb{Z}$ , inicio:  $\mathbb{Z}$ , fin:  $\mathbb{Z}$ ) {
   $(\forall i : \mathbb{Z}) \text{inicio} \leq i < \text{fin} + 1 \longrightarrow_L (\text{abs}(s[i]) \leq \text{umbral})$ 
}
pred esSilencio (s: señal, umbral:  $\mathbb{Z}$ , freq:  $\mathbb{Z}$ , in: intervalo) {
  finEsMayorQueInicio(in0, in1)  $\wedge_L$ 
  estaDentroDeLaSeñal(s, in0, in1)  $\wedge_L$ 
  esAlMenosUnDecimoDeSegundo(freq, in0, in1)  $\wedge_L$ 
  entreIndicesNoPasaCiertUmbral(s, umbral, in0, in1)  $\wedge_L$ 
  losAdyacentesSuperanElUmbral(s, umbral, in0, in1)
}

proc hablantesSuperpuestos (in r: reunion, in prof:  $\mathbb{Z}$ , in freq:  $\mathbb{Z}$ , in umbral:  $\mathbb{Z}$ , out result: Bool) {
  Pre {esReuniónVálidaAux(r, prof, freq)}
  Post {result =  $\neg$ noHayHablantesSuperpuestos(r, freq, umbral)}
}

pred haySilencio (s: señal, umbral:  $\mathbb{Z}$ , freq:  $\mathbb{Z}$ ) { $(\exists i, j : \mathbb{Z}) 0 \leq i, j < |s| \wedge (i < j) \wedge_L \text{esSilencio}(s, \text{umbral}, (i, j))$ }
pred noHayHablantesSuperpuestos (r: reunion, freq:  $\mathbb{Z}$ , umbral:  $\mathbb{Z}$ ) {
   $(\forall i, j : \mathbb{Z}) 0 \leq i, j < |r| \wedge (i \neq j) \longrightarrow_L$ 
   $(\forall k, l : \mathbb{Z}) 0 \leq k, l < |r[i]_0| \wedge k < l \longrightarrow_L$ 
   $\neg \text{haySilencio}(\text{subseq}(r[i]_0, k, l), \text{umbral}, \text{freq}) \longrightarrow_L \text{esSilencio}(r[j]_0, \text{umbral}, \text{freq}, (k, l))$ 
}

proc reconstruir (in s: señal, in prof:  $\mathbb{Z}$ , in freq:  $\mathbb{Z}$ , out señal: Bool) {
  Pre {esSeñalAux(s, prof, freq)}
  Post {esSeñalAux(result)  $\wedge_L$ 
  |s| = |result|  $\wedge_L$ 
  enDondeNoSeaCeroDebenCoincidir(s, result)  $\wedge_L$ 
  enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos(s, result) }
}

pred enDondeNoSeaCeroDebenCoincidir (original: señal, reconstruida: señal) {
   $(\forall i : \mathbb{Z}) 0 \leq i < |\text{original}| \longrightarrow_L$ 
  (original[i]  $\neq$  0)  $\wedge_L$ 
  (original[i] = reconstruida[i]) }
pred enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos (original: señal, reconstruida: señal) {
   $(\forall i : \mathbb{Z}) 0 \leq i < |\text{original}| \longrightarrow_L$ 
  (original[i] = 0)  $\wedge_L$ 
  reconstruida[i] = promedioDeVecinosNoNulos(original[i], reconstruida[i]) }
fun promedioDeVecinosNoNulos (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z} = \frac{(s[\text{elIndiceNoNuloMasCercano}(s, i)] + s[\text{el2doIndiceNoNuloMasCercano}(s, i)])}{2}$ ;
fun elIndiceNoNuloMasCercano (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$ 
if dist(i, indiceSiguienteNoNulo(s, i)) < dist(i, indiceAnteriorNoNulo(s, i)) then
indiceSiguienteNoNulo(s, i) else
if dist(i, indiceSiguienteNoNulo(s, i)) > dist(i, indiceAnteriorNoNulo(s, i)) then

```

```

indiceAnteriorNoNulo(s, i)  else
indiceAnteriorNoNulo(s, i)  ∨ indiceSiguienteNoNulo(s, i)  fi  fi ;
fun dist (x:  $\mathbb{Z}$ , y:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  = abs(x − y) ;
fun el2doIndiceNoNuloMasCercano (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =
elIndiceNoNuloMasCercano(setAt(s, elIndiceNoNuloMasCercano(s, i), 0)) ;
fun indiceAnteriorNoNulo (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =  $\sum_{p=0}^{i-1}$  if esElPrimerAnteriorNoNulo(s, i, p) then p else 0 fi ;
pred esElPrimerAnteriorNoNulo (s: señal, i:  $\mathbb{Z}$ , p:  $\mathbb{Z}$ ) { (∀ j :  $\mathbb{Z}$ ) p ≤ j < i  $\longrightarrow_L$  (s[j] = 0) ∧L (s[p] ≠ 0) }
fun indiceSiguienteNoNulo (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  =  $\sum_{p=i+1}^{|s|-1}$  if esElPrimerSiguienteNoNulo(s, i, p) then p else 0 fi ;
pred esElPrimerSiguienteNoNulo (s: señal, i:  $\mathbb{Z}$ , p:  $\mathbb{Z}$ ) { (∀ j :  $\mathbb{Z}$ ) i ≤ j < p  $\longrightarrow_L$  (s[j] = 0) ∧L (s[p] ≠ 0) }

```