

```

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

pred enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos (original: señal, reconstruida: señal) {
  ( $\forall i : \mathbb{Z} \ 0 \leq i < |original| \rightarrow_L$ 
  (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)  $\vee$  indiceSiguienteNoNulo(s, i)
    fi
  fi
;

fun dist (x:  $\mathbb{Z}$ , y:  $\mathbb{Z}$ ) :  $\mathbb{Z} = \text{abs}(x - y)$ ;

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:  $\mathbb{Z}$ ) {( $\forall j : \mathbb{Z} \ p \leq j < i \rightarrow_L (s[j] = 0) \wedge_L (s[p] \neq 0)$ )}

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:  $\mathbb{Z}$ ) {( $\forall j : \mathbb{Z} \ i \leq j < p \rightarrow_L (s[j] = 0) \wedge_L (s[p] \neq 0)$ )}

fun el2doIndiceNoNuloMasCercano (s: señal, i:  $\mathbb{Z}$ ) :  $\mathbb{Z} =$ 
  elIndiceNoNuloMasCercano(setAt(s, elIndiceNoNuloMasCercano(s, i), 0));

pred enDondeNoSeaCeroDebenCoincidir (original: señal, reconstruida: señal) {
  ( $\forall i : \mathbb{Z} \ 0 \leq i < |original| \rightarrow_L (\text{original}[i] \neq 0) \wedge_L (\text{original}[i] = \text{reconstruida}[i])$ )
}

```