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) {
$(\forall i : \mathbb{Z})\ 0 \leq i < |original| \ \longrightarrow_L$
(original[i] $\neq 0) \wedge_L$
(original[i] = reconstruida[i]) }
pred enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos (original: señal, reconstruida: señal) {
$(\forall i : \mathbb{Z})\ 0 \leq i < |original| \ \longrightarrow_L$
(original[i] = 0) $\wedge_L$
reconstruida[i] = promedioDeVecinosNoNulos(original[i], reconstruida[i]) }
fun promedioDeVecinosNoNulos (s: señal, i: $\mathbb{Z}$) : $\mathbb{Z} = \frac{(s[obtenerIndiceAnteriorNoNulo(i)] + s[obtenerIndiceSiguienteNoNulo(i)])}{2}$;
fun obtenerIndiceAnteriorNoNulo (s: señal, i: $\mathbb{Z}$) : $\mathbb{Z}$ =
$\sum_{p=0}^{|s|-i}$ if $esElPrimerAnteriorNoNulo(s, i, p)$ then $s[p]$ else 0 fi ;
pred esElPrimerAnteriorNoNulo (s: señal, i: $\mathbb{Z}$, p: $\mathbb{Z}$) $\{(\forall j : \mathbb{Z})\ p \leq j < i \ \longrightarrow_L (s[j] = 0) \wedge_L (s[p]! = 0)\}$
fun obtenerIndiceSiguienteNoNulo (s: señal, i: $\mathbb{Z}$) : $\mathbb{Z} = \sum_{p=i}^{|s|-2}$ if $esElPrimerSiguienteNoNulo(s, i, p)$ then $s[p]$ else 0 fi ;
pred esElPrimerSiguienteNoNulo (s: señal, i: $\mathbb{Z}$, p: $\mathbb{Z}$) $\{(\forall j : \mathbb{Z})\ i \leq j < p \ \longrightarrow_L (s[j] = 0) \wedge_L (s[p]! = 0)\}$