DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TPE - Reuniones Remotas

Grupo 6 - Alrescate.com

Integrante	LU	Correo electrónico
Nicolas Andres Kinaschuk	348/15	nicolaskinaschuk@gmail.com
Maria Vanesa Reyes Reyes	398/05	vanereyes22@gmail.com
Camilo Manuel DÁloisio	800/18	camilodaloisio@gmail.com
Sebastian E C Speranza	120/19	sebacagnoni@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Índice

1. Ejercicio 1

```
proc esSeñal (in s: seq⟨ℤ⟩, in prof: ℤ, in freq: ℤ, out result: Bool) {
        Pre {prof > 0 ∧ freq > 0}
        Post {result = true ↔ esSeñalAux(s, prof, freq)}
}
```

```
pred esSeñalAux (s: seq⟨ℤ⟩, prof: ℤ, freq: ℤ) {
  |s| ≥ 0∧
    frecuenciaEnRango(freq)∧
    profundidadCorrecta(s)∧
    duraMasDeUnSegundo(s, freq)∧
    ningunaMuestraSuperaLaProfundidad(s, prof)
}
```

```
pred frecuenciaEnRango (freq: ℤ) {freq ∈ [8, 32]}
```

```
pred profundidadCorrecta (prof: ℤ) {prof ∈ [8, 16, 32]}
```

```
pred duraMasDeUnSegundo (s: seq⟨ℤ⟩, freq: ℤ) {duraciónEnSegundos(s, freq) > 1}
```

```
pred ningunaMuestraSuperaLaProfundidad (s: seq⟨ℤ⟩, p: ℤ) {
  (∀i : ℤ) 0 ≤ i < |s|  ⟶_L (−2)^(p−1) ≤ s[i] ≤ 2^(p−1) − 1
}
```

```
fun duraciónEnSegundos (s: seq⟨ℤ⟩, freq: ℤ) : ℤ = |s| / (freq·1000) ;
```

2. Ejercicio 2

```
proc seEnojó? (in s: señal, in umbral: ℤ, in prof: ℤ, in freq: ℤ, out result: Bool) {
        Pre {umbral > 0 ∧ esSeñalAux(s, prof, freq)}
        Post {
            result = umbralEnRango(umbral, prof) ∧
            existeUnaSubsecuenciaQueSuperaUmbral(s, freq, umbral)}
}
    pred umbralEnRango (umbral: ℤ, p: ℤ) {umbral ≤ 2^(p−1) − 1}

    pred existeUnaSubsecuenciaQueSuperaUmbral (s : señal, freq: ℤ, umbral: ℤ) {
      (∃d, h : ℤ) 0 ≤ d, h < |s| + 1  ∧(h > (d + freq ∗ 1000 ∗ 5)) ∧_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;
```

3. Ejercicio 3

proc esReunión**Válida?** (in r: reunion, in prof: $\mathbb{Z}$, in freq: $\mathbb{Z}$, out result: Bool) {
  Pre $\{prof > 0 \land freq > 0\}$
  Post {result = esReuniónVálidaAux(r, prof, freq)}
}

pred esReuniónVálidaAux $(r : reunion, prof : \mathbb{Z}, freq : \mathbb{Z})$ {
  $contieneSeñalesValidas(r, prof, freq) \land$
  $lasLongitudesDeSeñalSonIguales(r) \land$
  $todosHablantesDistintos(r) \land$
  $losHablantesEstanEnRangosDe0ANMenos1(r)$
}

pred contieneSeñalesValidas $(r : reunion, prof : \mathbb{Z}, freq : \mathbb{Z})$ $\{(\forall i : \mathbb{Z})\ 0 \leq i < |r| \ \longrightarrow_L esSeñalAux(r[i]_0, prof, freq)\}$

pred lasLongitudesDeSeñalSonIguales $(r : reunion)$ $\{(\forall i, j : \mathbb{Z})\ 0 \leq i, j < |r| \land i \neq j \ \longrightarrow_L (|r[i]_0| = |r[j]_0|)\}$

pred todosHablantesDistintos $(r : reunion)$ $\{(\forall i, j : \mathbb{Z})\ 0 \leq i, j < |r| \land i \neq j \ \longrightarrow_L (r[i]_1 \neq r[j]_1)\}$

pred losHablantesEstanEnRangosDe0ANMenos1 $(r : reunion)$ $\{(\forall i : \mathbb{Z})\ 0 \leq i < |r| \ \longrightarrow_L 0 \leq r[i]_1 < |r|\}$

4. Ejercicio 4

```
proc acelerar (inout r: reunion, in prof: ℤ, in freq: ℤ)  {
        Pre {
          esReuniónVálidaAux(r, prof, freq) ∧_L
            lasSeñalesDuranMásDe2Segundos(r) ∧
            r_0 = r
        }
        Post {
          |r| = |r_0| ∧_L
            lasSeñalesTieneLaMitadDeMuestras(r, r_0)∧_L
            losImpares(r, r_0)
        }
}
```

pred lasSeñalesDuranMásDe2Segundos ($r : reunion, freq : \mathbb{Z}$) $\{duraciónEnSegundos(r[0]_0, freq) > 2\}$

pred lasSeñalesTieneLaMitadDeMuestras (r: reunion, $r_v : reunion$){
$(\forall i : \mathbb{Z})\ 0 \leq i < |r| \ \longrightarrow_L$ if $esPar(|r_v[i]_0|)$ then $|r[i]_0| = \frac{|r_v[i]_0|}{2}$ else $|r[i]_0| = \frac{|r_v[i]_0|-1}{2}$ fi
}

pred losImpares (r: reunion, $r_v : reunion$){
$(\forall i : \mathbb{Z})\ 0 \leq i < |r_v| \ \longrightarrow_L ($
$(\exists j : \mathbb{Z})\ 0 \leq j < |r| \ \wedge_L (r_v[i]_1 = r[j]_1) \wedge_L ($
$(\forall q : \mathbb{Z})\ 0 \leq q < |r_v[i]_0| \wedge (\neg esPar(q)) \ \longrightarrow_L (r_v[i]_0[q] = r[j]_0[\frac{q-1}{2}])))$
}

5. Ejercicio 5

```
proc ralentizar (inout r: reunion, in prof: ℤ, in freq: ℤ)  {
        Pre {esReuniónVálidaAux(r, prof, freq) ∧ r_0 = r}
        Post {
           |r| = |r_0| ∧_L
              lasSeñalesTienenElDobleDeMuestras(r, r_0)∧_L
              promedioEntrePares(r, r_0)
        }
}
```

```
pred lasSeñalesTienenElDobleDeMuestras (r : reunion, r_v : reunion) {
   (∀i : ℤ) 0 ≤ i < |r_v|  ⟶_L (2 · |r_v[i]_0|) = (|r[i]_0| + 1)
}
```

```
pred promedioEntrePares (r : reunion, r_v : reunion) {
   (∀i : ℤ) 0 ≤ i < |r|  ⟶_L (
      (∃j : ℤ) 0 ≤ j < |r_v|  ∧_L (r[i]_1 = r_v[j]_1) ∧_L (
         (∀q : ℤ) 0 ≤ q < |r[i]_0|  ⟶_L
            if esPar(q) then r[i]_0[q] = r_v[j]_0[q/2] else r[i]_0[q] = (r_v[j]_0[(q−1)/2] + r_v[j]_0[(q+1)/2]) / 2 fi ) )
}
```

6. Ejercicio 6

```
proc tonosDeVozElevados (inout r: reunion, in freq: ℤ, in prof: ℤ, out hablantes: seq⟨hablante⟩) {
        Pre {esReuniónVálidaAux(r, prof, freq)}
        Post {
          |h| ≥ |hs| ∧_L
            siPertenecenAHablantesElPromedioDeAmplitudEsMasGrandeOIgualQueElResto(r, hablantes) ∧
            losHablantesPertenecenALaReunión(r, hablantes) ∧
            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 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 = ∑_{i=0}^{|s|} abs(s[i]) ;

    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)
    }
```

7. Ejercicio 7

```
proc ordenar (inout r: reunion, in freq: ℤ, in prof: ℤ)  {
        Pre {esReuniónVálidaAux(r, prof, freq) ∧ r_0 = r}
        Post {
          esReuniónVálidaAux(r, prof, freq) ∧
            ordenadaDeMayorAMenorPorTonoDeVoz(r) ∧
            esUnaPermutación(r_0, 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| ∧_L
        (∀i : ℤ) 0 ≤ i < |x|  ⟶_L(
          (∃j : ℤ) 0 ≤ j < |y|  ∧_L(x[i]_1 = y[j]_1 ∧ x[i]_0 = y[j]_0))
    }
```

8. Ejercicio 8

```
proc silencios (in s: señal, in freq: ℤ, in prof: ℤ, out intervalos: seq⟨intervalo⟩)  {
        Pre {esSeñalAux(s, prof, freq) ∧ (umbral > 0)}
        Post {
           noHayIntervalosRepetidos(intervalos) ∧
              (∀i : ℤ) 0 ≤ i < |intervalos|  ⟶_L(
                 esSilencio(s, umbral, freq, intervalos[i]) )
        }
}
```

```
   pred esSilencio (s: senal, umbral: ℤ, freq: ℤ, in: intervalo) {
     (finEsMayorQueInicio(in_0, in_1)∧
        estaDentroDeLaSeñal(s, in_0, in_1))∧_L
        esAlMenosUnDecimoDeSegundo(freq, in_0, in_1)∧
        entreIndicesNoPasaCiertoUmbral(s, umbral, in_0, in_1)∧
        losAdyacentesSuperanElUmbral(s, umbral, in_0, in_1)
   }

   pred noHayIntervalosRepetidos (ins: seq⟨intervalo⟩) {(∀i : ℤ) 0 ≤ i < |ins|  ⟶_L (#apariciones(ins, e) = 1)}

   pred finEsMayorQueInicio (inicio: ℤ, fin: ℤ) {fin > inicio}

   pred estaDentroDeLaSeñal (s: señal, inicio: ℤ, fin: ℤ,) {(inicio ≥ 0) ∧ (fin < |s|)}

   pred esAlMenosUnDecimoDeSegundo (freq: ℤ, inicio: ℤ, fin: ℤ,) {(fin − inicio + 1) ≥ (frecuencia ∗ 100)}

   pred losAdyacentesSuperanElUmbral (s: señal, umbral: ℤ, inicio: ℤ, fin: ℤ) {
     ( (inicio = 0) ∨((inicio − 1 ≥ 0) ∧_L (s[inicio − 1] ≥ umbral))) ∧ (
        (fin = |s| - 1) ∨((fin + 1 < |s|) ∧_L (s[fin + 1] ≥ umbral)))
   }

   pred entreIndicesNoPasaCiertoUmbral (s: señal, umbral: ℤ, inicio: ℤ, fin: ℤ) {
     (∀i : ℤ) inicio ≤ i < fin + 1  ⟶_L(abs(s[i]) < umbral)
   }
```

9. Ejercicio 9

`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 noHayHablantesSuperpuestos` (r: reunion, freq: $\mathbb{Z}$, umbral: $\mathbb{Z}$) {

$(\forall i, j : \mathbb{Z})\ 0 \leq i, j < |r| \wedge (i \neq j) \ \longrightarrow_L$

$(\forall k, l : \mathbb{Z})\ 0 \leq k, l < |r[i]_0| \ \wedge k < l \longrightarrow_L$

$\neg haySilencio(subseq(r[i]_0, k, l), umbral, freq) \longrightarrow_L esSilencio(r[j]_0, umbral, freq, (k, l))\}$

`pred haySilencio` (s: señal, umbral: $\mathbb{Z}$, freq: $\mathbb{Z}$) $\{(\exists i, j : \mathbb{Z})\ (0 \leq i, j < |s| \wedge (i < j)) \ \wedge_L esSilencio(s, umbral, (i, j))\}$

10. Ejercicio 10

```
proc reconstruir (in s: señal, in prof: ℤ, in freq: ℤ, out señal: Bool) {
        Pre {esSeñalAux(s, prof, freq)}
        Post {
          esSeñalAux(result) ∧
            (|s| = |result| ∧_L
            enDondeNoSeaCeroDebenCoincidir(s, result) ∧_L
            enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos(s, result))
        }
}
```

```
pred enDondeEsCeroDebeSerElPromedioDeSusVecinosNoNulos (original: señal, reconstruida: señal) {
  (∀i : ℤ) 0 ≤ i < |original| ⟶_L
    (original[i] = 0) ∧_L reconstruida[i] = promedioDeVecinosNoNulos(original[i], reconstruida[i])
}

fun promedioDeVecinosNoNulos (s: señal, i: ℤ) : ℤ = (s[elIndiceNoNuloMasCercano(s,i)] + s[el2doIndiceNoNuloMasCercano(s,i)]) / 2 ;

fun elIndiceNoNuloMasCercano (s: señal, i: ℤ) : 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: ℤ, y: ℤ) : ℤ = abs(x − y) ;

fun indiceAnteriorNoNulo (s: señal, i: ℤ) : ℤ = Σ_{p=0}^{i−1} if esElPrimerAnteriorNoNulo(s, i, p) then p else 0 fi ;

pred esElPrimerAnteriorNoNulo (s: señal, i: ℤ, p: ℤ) {(∀j : ℤ) p ≤ j < i ⟶_L (s[j] = 0) ∧_L (s[p] ≠ 0)}

fun indiceSiguienteNoNulo (s: señal, i: ℤ) : ℤ = Σ_{p=i+1}^{|s|−1} if esElPrimerSiguienteNoNulo(s, i, p) then p else 0 fi ;

pred esElPrimerSiguienteNoNulo (s: señal, i: ℤ, p: ℤ) {(∀j : ℤ) i ≤ j < p ⟶_L (s[j] = 0) ∧_L (s[p] ≠ 0)}

fun el2doIndiceNoNuloMasCercano (s: señal, i: ℤ) : ℤ =
  elIndiceNoNuloMasCercano(setAt(s, elIndiceNoNuloMasCercano(s, i), 0)) ;

pred enDondeNoSeaCeroDebenCoincidir (original: señal, reconstruida: señal) {
  (∀i : ℤ) 0 ≤ i < |original| ⟶_L (original[i] ≠ 0) ∧_L (original[i] = reconstruida[i])
}
```