Recuperatorio de Imperativo - Algoritmos y Estructuras de Datos I 16 dic. 2005

Aclaraciones: El parcial NO es a libro abierto. Cualquier decisión de interpretación que se tome debe ser aclarada y justificada. Para aprobar el parcial se requieren al menos 60 puntos. Indicar el número de orden, LU y la cantidad total de hojas entregadas. Entregar cada ejercicio en hoja separada.

Ejercicio 1. **[20 p.]** Implemente en imperativo el siguiente problema:

problema buscarSumasDeADos($l : [\mathbb{Z}], n : \mathbb{Z}, r : [\mathsf{Bool}]$) {
  modifica r;
  requiere $|l| == n \land n > 0$;
  requiere $(\forall x \leftarrow l)\ x \geq 0$;
  requiere $|r| = 1 + 2 \cdot \mathsf{max}(l)$;
  asegura $|r| == |\mathsf{pre}(r)|$;
  asegura $r == [\mathsf{hayDosQueSuman}(l, i) \mid i \leftarrow [0..|r|)]$;
  aux $\mathsf{max}(z : [\mathbb{Z}]) : \mathbb{Z} = [x \mid x \leftarrow z\ , (\forall y \leftarrow z)\ x \geq y]_0$;
  aux $\mathsf{hayDosQueSuman}(z : [\mathbb{Z}], y : \mathbb{Z}) : \mathsf{Bool} = (\exists j \leftarrow [0..|z|), k \leftarrow (j..|z|)) z_k + z_j == y$;
}

Ejercicio 2. Dada la siguiente especificación:

problema modificaRaro($l : [\mathbb{Z}], n : \mathbb{Z}$) {
  modifica l;
  requiere $|l| == n$;
  requiere $n \mod 2 == 0$;
  asegura $|l| == |\mathsf{pre}(l)|$;
  asegura $l == [\mathsf{modif}(\mathsf{pre}(l)_i, \mathsf{pre}(l)_{n-1-i}) \mid i \leftarrow [0..|l|)]$;
  aux $\mathsf{modif}(a, b : \mathbb{Z}) : \mathbb{Z} = \beta(a == b) \cdot 2 \cdot a + \beta(a \neq b) \cdot b$;
}

1. **[15 p.]** Implemente la función en imperativo.
2. Para cada ciclo utilizado en la implementación del item anterior, proponga:
   a) **[5 p.]** una poscondición para el ciclo
   b) **[15 p.]** un invariante
   c) **[5 p.]** una expresión variante

   que permitan demostrar correctitud utilizando el Teorema del Invariante (¡no se pide demostrar nada!).

¡Sigue del otro lado! ↻

Ejercicio 3. Dada la siguiente especificación y su correspondiente implementación:

problema multiplicar($a : [\mathbb{Z}], x : \mathbb{Z}, n : \mathbb{Z}$) = {
  modifica a;
  requiere $|a| == n \land n > 0$;
  requiere $a_0 == 0$;
  requiere todosDígitos(a) $\land\, x \in [0..9]$;
  asegura $|a| == n$;
  asegura todosDígitos(a);
  asegura dec(a) $==$ dec(pre(a)) $\cdot\, x$;
  aux dec($l : [\mathbb{Z}]$) : $\mathbb{Z} = \sum[10^{|l|-i-1} l_i \mid i \leftarrow [0..|l|)]$;
  aux todosDígitos($l : [\mathbb{Z}]$) : Bool $= (\forall x \leftarrow l)\ x \in [0..9]$;
}

```
int multiplicar (int a[], int x, int n) {
    int mellevo = 0;
    int cuentaActual;
    int i = n - 1;
      // estado antesC;
      // vale i == n − 1 ∧ mellevo == 0;
    while ( i >= 0 ) {
            // invariante I : dec([mellevo] + +a(i..n)) == dec(pre(a)(i..n)) · x ∧ −1 ≤ i < n∧
            //                todosDígitos(a(i..n)) ∧ a[0..i] == pre(a)[0..i] ∧ mellevo ∈ [0..9]∧
            //                i == −1 → mellevo == 0
            // variante (−1) v: i;
        cuentaActual = a[i] * x + mellevo;
        a[i] = cuentaActual % 10;
        mellevo = cuentaActual / 10;
        i = i - 1;
    }
      // estado despuésC;
      // vale Q : dec(a) == dec(pre(a)) · x ∧ todosDígitos(a) ∧ i == −1
}
```

Se desea completar la demostración de correctitud de esta implementación. Para ello se pide demostrar algunos puntos del Teorema del Invariante:

1. **[10 p.]** El invariante y la negación de la guarda garantizan que vale la poscondición del ciclo: $(I \land \neg B) \rightarrow Q$.

2. **[30 p.]** El cuerpo del ciclo preserva el invariante: `/*` vale $I \land B$; `*/ cuerpo /*` vale I; `*/`.

¡Empieza del otro lado! ↻