Ejercicio Cola Circular

Algoritmos y Estructuras de Datos - FCEyN - UBA

Fecha de compilación: 21 de octubre de 2024

```
Modulo ColaCircular<T> implementa ColaAcotada<T> {
    var datos: Array<T>
    var inicio: int
    var fin: int

    pred InvRep(c : ColaCircular<T>) {
        (c.datos ≠ null)
        ∧ ((inicio = 0 ∧ fin = -1 )
            ∨ (0 ≤ inicio < datos.length
                ∧ 0 ≤ fin < datos.length  ))
    }

    pred Abs(c: ColaCircular<T>, c': ColaAcotada<T>) {
        (c'.capacidad = c.datos.length)
        ∧ (c.fin = -1  ⟹  c'.s = <>)
        ∧ (c.fin ≠ -1 ∧ c.inicio < c.fin
            ⟹  c'.s = subseq(c.datos, c.inicio, c.fin) )
        ∧ (c.fin ≠ -1 ∧ c.inicio ≥ c.fin
            ⟹  c'.s = subseq(c.datos, c.inicio, c.datos.length) ++ subseq(c.datos, 0, c.fin)  )


    }

    proc nuevaCola(in cap: int) : ColaCircular<T> {
        c := new ColaCircular<T>
        c.datos := new Array<T>[cap]
        c.inicio := 0
        c.fin := -1
    }

    proc encolar(inout c:ColaCircular<T>, in e: T) {
        if (c.fin = -1) {
            c.datos[c.inicio] := e
            c.fin := 1
        } else {
            c.datos[c.fin] := e
            c.fin := (c.fin + 1) % c.datos.length
        }
    }
```

```
    proc desencolar(inout c: ColaCircular<T>): T {
        res := c.datos[c.inicio]
        c.inicio := (c.inicio + 1) % c.datos.length
        if (c.inicio = c.fin) {
            c.inicio := 0
            c.fin := -1
        }

        return res
    }
}
```