Nombre y apellido: Eugenio Bergia Legajo: B-6379/7

Ingeniería de Software 1
LCC – FCEIA - UNR

Cristiá – Colussi – Krapf – Scandolo
Lunes 22 de julio de 2024

Examen Final

Nota: La interpretación de las consignas es parte del examen. **Cada parte** se aprueba con al menos dos problemas bien y no menos del 65 % de la puntuación de esa parte. Problemas parcialmente correctos no necesariamente incrementan la puntuación.

Primera parte

Duración aproximada 2 hs.

Se evalúa: Temporal Logic of Actions.

1. Escriba las designaciones y modele en TLA el conocimiento de dominio y la especificación de los requerimientos que se enuncian a continuación.

   Un sistema de memoria consiste en cierta cantidad de procesadores que se comunican con la memoria física a través de cierta interfaz. Esta interfaz posee una operación por medio de la cual un procesador puede requerir a la memoria una lectura o escritura, y otra operación por medio de la cual la memoria envía cierto valor a un procesador. La interfaz está dada, no debe ser programada; se debe programar el funcionamiento o acceso a la memoria física.

   Los procesadores pueden escribir un valor en una celda de memoria o solicitar el valor almacenado en una celda. Cada procesador efectúa un pedido a la vez y espera la respuesta de la memoria antes de hacer el siguiente pedido. La respuesta a un pedido de lectura es el valor almacenado en la celda solicitada y la respuesta a un pedido de escritura es un código especial que indica que la operación ha concluido.

   Claramente, ni la interfaz ni la memoria física pueden controlar cuándo un procesador hará una solicitud. Por lo tanto, se espera que el sistema esté preparado para recibir pedidos en cualquier momento y que utilice los períodos ociosos para completar las operaciones de acceso a la memoria física (que son las más lentas).

   Se espera que todo pedido efectuado por algún procesador eventualmente reciba una respuesta proveniente de la memoria.

2. Explique la incidencia del teorema de Alpern-Schneider en el lenguaje de especificación TLA.

Eugenio Bergia

① MODULE Memoria

EXTENDS Naturals, Sequences, Booleans

CONSTANTS N, CPU, Write(p, n, i), Read(p, i), Reply(p, n)

ASSUME ∧ N ∈ Nat ∧ N > 0

∧ ∀ p ∈ CPU, n ∈ Nat, i ∈ (1..N):

∧ Write(p, n, i) ∈ BOOLEAN

∧ Read(p, i) ∈ BOOLEAN

∧ Reply(p, n) ∈ BOOLEAN

VARIABLES mem, reqs

TypeInv ≜ ∧ mem ∈ [(1..N) → Nat]

∧ reqs ∈ Seq([op: {"wr", "rd"}, p: CPU, n: Nat, i: (1..N)])

Init ≜ ∧ mem = [i ∈ (1..N) ↦ 0]

∧ reqs = ⟨⟩

Req ≜ ∨ ∃ p ∈ CPU, n ∈ Nat, i ∈ (1..N):

∧ Write(p, n, i)

∧ reqs' = Append(reqs, [op ↦ "wr", p ↦ p, n ↦ n, i ↦ i])

∧ UNCHANGED mem

∨ ∃ p ∈ CPU, i ∈ (1..N)

∧ Read(p, i)

∧ reqs' = Append(reqs, [op ↦ "rd", p ↦ p, n ↦ 0, i ↦ i])

∧ UNCHANGED mem

```
Rep ≜ ∧ reqs ≠ ⟨⟩
      ∧ ∨ ∧ Head(reqs).op = "wr"
          ∧ mem' = [mem EXCEPT ![Head(reqs).i] = Head(reqs).n]
          ∧ Reply(Head(reqs).p, CHOOSE x ∈ Nat)
          ∧ reqs' = Tail(reqs)
        ∨ ∧ Head(reqs).op = "rd"
          ∧ Reply(Head(reqs).p, mem[(Head(reqs).i)])
          ∧ reqs' = Tail(reqs)
          ∧ UNCHANGED mem

Next ≜ Req ∨ Rep

Spec ≜ Init ∧ □[Next]_⟨mem,reqs⟩ ∧ WF_⟨mem,reqs⟩(Rep)

THEOREM Spec ⇒ □TypeInv
```

ES EC.

Tamaño máximo de la memoria ≈ N

El procesador p solicita escribir el valor n en la celda número i de la memoria ≈ Write(p, n, i) E_c, S

El procesador p solicita leer el valor de la celda número i de la memoria ≈ Read(p, i) E_c, S

Se ordena responder al procesador p con el valor n ≈ Reply(p,n) M_c, S

EC.

FcHc DK.

Eugenio Bergia

② El teorema de Alpern-Schneider dice que toda propiedad P sobre un conjunto de ejecuciones puede ser expresada como la conjunción entre una propiedad de seguridad y una propiedad de vitalidad. Las propiedades de seguridad indican que nada malo puede suceder durante la ejecución de un programa, y comunmente se expresan indicando cuales son los pasos permitidos, prohibiendo implicitamente todos los demás. Las propiedades de vitalidad indican que algo bueno eventualmente sucederá durante la ejecución de un programa, y para evitar que estas agreguen restricciones por sobre las de seguridad, se utilizan unas propiedades de vitalidad especiales llamadas de equidad.

Yaciden sobre el lenguaje TLA ya que en este las especificaciones tienen comunmente la siguiente forma:

$$Spec \triangleq Init \wedge \Box[Next]_{vars} \wedge Fairness_{vars}(A)$$

donde $Init \wedge \Box[Next]_{vars}$ es una propiedad de seguridad ya que indica el estado inicial y los únicos pasos que puede tomar la ejecución, y $Fairness_{vars}(A)$ es una propiedad de equidad débil o fuerte que indica que la acción A debe suceder eventualmente. El subíndice vars indica que puede haber pasos repetitivos donde dicho conjunto de variables no cambie.