

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. Se trata de especificar en TLA+ una versión simplificada de un algoritmo para compromiso atómico distribuido (*distributed atomic commitment*) publicado por Babaoglu y Toueg. El sistema consiste de un proceso coordinador y varios procesos participantes. El coordinador debe decidir si los participantes aceptan el compromiso en base a una votación. Entonces el coordinador le pide un voto a cada participante y cada uno responde por sí o por no. Si todos responden sí, entonces el coordinador acepta el compromiso, caso contrario el coordinador aborta el compromiso. En ambos casos el coordinador comunica su decisión a todos los participantes. El problema surge cuando uno de los participantes o el mismo coordinador sufren una falla y no pueden seguir participando de la votación. Es decir cualquiera de los procesos puede estar vivo o no. Un proceso puede participar en tanto esté vivo (tener en cuenta que puede estar vivo al inicio pero morir en algún punto intermedio).

Concretamente se deben especificar las siguientes operaciones:

- a) El coordinador solicita un voto a un participante (lo cual se hace únicamente si aun no le solicitó el voto).
 - b) El coordinador recibe el voto de un participante (lo cual se hace una vez que le solicitó el voto a todos los participantes y aun hay alguno que no ha respondido, y ese participante ha enviado su voto).
 - c) El coordinador detecta una falla en un participante (luego de que le ha solicitado el voto a todos los participantes el participante en cuestión muere sin votar).
 - d) Un participante envía su voto al coordinador (lo hace solo si el coordinador se lo ha solicitado).
 - e) Un participante decide unilateralmente abortar la votación, luego de que ha votado por el 'no' pero antes de que el coordinador comunique su decisión final.
 - f) Un participante decide unilateralmente abortar la votación, porque el coordinador murió sin enviar el pedido de voto.
2. Explique la incidencia del teorema de Alpern-Schneider en el lenguaje de especificación TLA.

1-

Module DAG

CONSTANTS Participante

VARIABLES coord, part, votacion, voto, votando, compromiso

TypeInv $\equiv$

- $\wedge \text{coord} \in \{\text{"vivo"}, \text{"muerto"}\}$
- $\wedge \text{part} \in [\text{Participante} \rightarrow \{\text{"vivo"}, \text{"muerto"}\}]$
- $\wedge \text{votacion} \in [\text{pendiente}; \text{BOOLEAN}, \text{activa}; \text{BOOLEAN}]$
- $\wedge \text{compromiso} \in \text{BOOLEAN}$
- $\wedge \text{voto} \in [\text{Participante} \rightarrow \{\text{"si"}, \text{"no"}, \text{"none"}\}]$
- $\wedge \text{votando} \in [\text{Participante} \rightarrow \text{BOOLEAN}]$

Init $\equiv$

- $\wedge \text{coord} \in \{\text{"vivo"}, \text{"muerto"}\}$
- $\wedge \text{part} \in [\text{Participantes} \rightarrow \{\text{"vivo"}, \text{"muerto"}\}]$
- $\wedge \text{votando} = [\text{p} \in \text{Participantes} \mapsto \text{FALSE}]$
- $\wedge \text{votacion} = [\text{pendiente} \mapsto \text{true}, \text{activa} \mapsto \text{FALSE}]$
- $\wedge \text{compromiso} = \text{true}$
- $\wedge \text{voto} = [\text{p} \in \text{Participantes} \mapsto \text{"none"}]$

Solicitar(p) $\equiv$

- $\wedge \text{part}[\text{p}] = \text{"vivo"}$
- $\wedge \text{coordinador} = \text{"vivo"}$
- $\wedge \neg \text{votando}[\text{p}]$

? Solicitar $\equiv$

- $\wedge \text{coordinador} = \text{"vivo"}$

- $\wedge \text{votacion.pendiente}$
- $\wedge \neg \text{votacion.activa}$
- $\wedge \forall p \in \text{Participante};$

- $\vee \text{Solicitar}(p)$

- $\vee \neg \text{votando}(p)$

- $\wedge \text{votacion}' = [\text{pendiente} \mapsto \text{true}, \text{activa} \mapsto \text{true}]$

No se
pide q'
voten todos
simultaneamente

$\text{Recibir}(p) \triangleq$
 $\wedge \text{coord} = \text{"vivo"}$
 $\wedge \text{part}[p] = \text{"vivo"}$
 $\wedge \text{votacion} : \text{activa}$
 $\wedge \text{votando}[p]$
 $\wedge \text{voto}[p] \neq \text{"none"}$
 $\wedge \text{votado}' = \text{[votado except !cps = FALSE]}$
 $\wedge \text{compromiso}' = \text{compromiso} \wedge \text{voto}[p] = \text{"si"}$
 $\wedge \text{UNCHANGED} < \text{coord}, \text{part}, \text{votacion}, \text{voto} >$

$\text{Falló}(p) \triangleq$
 $\wedge \text{coordinador} = \text{"vivo"}$
 $\wedge \text{votacion} : \text{activa}$
 $\wedge \text{part}[p] = \text{"muerto"}$
 $\wedge \text{votando}[p]$
 $\wedge \text{votado}' = \text{[votado except !cps = FALSE]}$
 $\wedge \text{UNCHANGED} < \text{coord}, \text{part}, \text{votacion}, \text{voto},$
 $\text{compromiso} >$

$\text{Votar}(p, v) \triangleq$
 $\wedge \text{part}[p] = \text{"vivo"}$
 $\wedge \text{votacion} : \text{activa}$
 $\wedge \text{votando}[p]$
 $\wedge \text{voto}[p] = \text{"none"}$
 ~~$\wedge v \in \{\text{"si"}, \text{"no"}\}$~~
 $\wedge \text{voto}' = \text{[voto except !cps = v]}$
 $\wedge \text{UNCHANGED} < \text{coord}, \text{part}, \text{votacion}, \text{votando}$
 $\text{compromiso} >$

$\text{Abortar}(p) \triangleq \neg \text{part}[p] = \text{"vivo"}$

$\wedge \text{votación} = \text{activa}$

$\wedge \text{voto}[p] = \text{"no"}$

$\wedge \text{votación}' = [\text{pendiente} \mapsto \text{FALSE}, \text{activa} \mapsto \text{FALSE}]$

$\wedge \text{UNCHANGED} \langle \text{coord}, \text{part}, \text{voto}, \text{votando}, \text{compromiso} \rangle$

$\text{Abortar_coord}(p) \triangleq \neg \text{part}[p] = \text{"vivo"}$

$\wedge \neg \text{votación} = \text{activa}$

$\wedge \text{votación} = \text{pendiente}$

$\wedge \text{coord} = \text{"muerto"}$

$\wedge \text{votación}' = [\text{pendiente} \mapsto \text{FALSE}, \text{activa} \mapsto$

$\text{FALSE}]$

$\text{Muere_coord} \triangleq \neg \text{coord} = \text{"vivo"}$

$\wedge \text{UNCHANGED} \langle \text{coord}, \text{part}, \text{voto}, \text{votando}, \text{compromiso} \rangle$

$\wedge \text{coord}' = \text{"muerto"}$

$\wedge \text{UNCHANGED} \langle \text{part}, \text{voto}, \text{votando}, \text{votación},$

$\text{compromiso} \rangle$

$\text{Muere_part}(p) \triangleq \neg \text{part}[p] = \text{"vivo"}$

$\wedge \text{part}'[p] = [\text{part} \text{ except } !\text{part}[p] = \text{"muerto"}]$

$\text{Next} \triangleq \text{Solicitar} \vee \exists p \in \text{Participante} :$

$\vee \text{Recibir}(p)$

$\vee \text{Fallar}(p)$

$\vee \exists v \in \{\text{"si"}, \text{"no"}\} : \text{votar}(p, v)$

$\vee \text{Abortar}(p)$

$\vee \text{Abortar_coord}(p)$

$vars \triangleq coord, part, vote, voted, votacion, compromiso$

$Spec \triangleq Init \wedge D \sqsubseteq Next \wedge vars \wedge WFvars(Solicitor)$

$\wedge WFvars$

$\wedge \exists p \in Participante :$

~~$\wedge \exists p \in Participante : WFvars(Fallz(p))$~~

~~$\wedge \exists p \in Participante : WFvars(Retirar(p))$~~

~~$\wedge \exists v \in \{ "si", "no" \} : WFvars(Votar(p, v))$~~

~~$\wedge \exists p \in Participante : WFvars(Anotar(p))$~~

~~$\wedge \exists p \in Participante : WFvars(Anotar(coord(p)))$~~

$\wedge \exists v \in \{ "si", "no" \} : WFvars(Votar(p, v))$

$theorem Spec \Rightarrow \Box \neg \text{falla}$

Comentarios:

- Considerar el uso de Timers para detectar fallas en los procesos pero me pareció que esto implicaría sobreespecificar. Considero que existe una acción del sistema que da la señal de la muerte de algún proceso (coordinador o participante). Internamente, el sistema podrá utilizar timers o algún otro método de sincronización por hardware o software. La especificación es transparente a esta propiedad del sistema.
- No se especifica la acción de comunicar los resultados y2 que no lo pide el enunciado. Esto deberá tener la forma:

$Comunicar \triangleq \wedge coord = "vivo"$

$\wedge \text{votación activa} \wedge \forall p \in Participante : \text{votación}(p)$

$\wedge \text{información} = \dots$

- No considere la acción de abortar unilateralmente luego de votar en las propiedades de equidad ya que el enunciado no indica que debería serlo. Entiendo que podría favorecer el buen funcionamiento del algoritmo, pero decidí no imponer más restricciones sobre el sistema sin que fuera completamente necesario.

2- El teorema de Alpern-Schneider indica que toda propiedad de un sist. está dada por la conjunción de una propiedad de utilidad y una de seguridad. Por otro lado, por un resultado de Abadi-Lamport, sabemos que existen ciertas propiedades de utilidad, llamadas de "equidad", que no restringen el sistema. Es decir, si S es de seguridad y E de equidad, entonces $(S \wedge E)$ es una máquina cerrada (i.e., $C(S \wedge E) = S$). TLA se aprovecha de este resultado ya que la especificación de un sistema suele tener la forma:

$$\text{Spec} \triangleq \text{Init} \wedge \Box \text{Next} \wedge \text{Equidad}$$

donde equidad es la conjunción de fórmulas de Fairness $WF(P)$ o $SF(P)$, las cuales representan equidad débil y fuerte. Particularmente:

$$WF(P) = \Box \Diamond \text{ENABLED}(P) \Rightarrow \Box \Diamond P$$

$$SF(P) = \Box \Diamond \text{ENABLED}(P) \Rightarrow \Box \Diamond P$$

donde P es una acción del sistema.

De este manera, las restricciones de utilidad no imponen restricciones de seguridad en el sistema. ~~no imponen~~