Z

Para cada uno de los ejercicios se deberá escribir las designaciones de los términos for- males primitivos que se utilizarán en el modelo (según se explicó en el apunte mencionado) y luego se deberá escribir un modelo Z de los requerimientos.

1. Describa un sistema de iluminación consistente en una lámpara y dos botones. Si la lámpara está apagada, al pulsar cualquiera de los dos botones se enciende y viceversa.

   Solución

   - La lampara esta encendida $\approx on$.
   - Se presiona alguno de los botones $\approx Presionar$.
   - La lampara se encuentra en el estado $l \approx Sistema\,(l)$.

$$lampara ::= on \mid off$$

Sistema

$$l : lampara$$

SistemaInit

$$Sistema$$
$$l = off$$

PresionarEnOn

$$\Delta Sistema$$
$$l = off$$
$$l' = on$$

PresionarEnOff

$$\Delta Sistema$$
$$l = on$$
$$l' = off$$

$$Presionar \triangleq PresionarEnOn \lor PresionarEnOff$$

2. El despachante recibe mensajes provenientes de dos canales diferentes. Externamente existe un servicio que chequea la paridad de cada mensaje. Si la paridad es errónea, el despachante envía un «error» a través de un canal de retorno (hay un canal de retorno por cada canal de entrada); si la paridad es correcta, el despachante pone el mensaje recibido en un buffer. El buffer puede almacenar 10 mensajes. Cuando el buffer está lleno, el despachante envía el contenido completo (de a un mensaje a la vez) del buffer a una unidad de procesamiento a través de otro canal. No se puede poner un mensaje si el buffer está lleno.

Solución

- m es un mensaje $\approx m \in MSG$.
- El despachante se encuentra enviando mensajes $\approx enviando$.
- El despachante se encuentra recibiendo mensajes $\approx recibiendo$.
- No hubo error al recibir el mensaje $\approx no$.
- Hubo error al recibir el mensaje $\approx yes$.
- El mensaje m tiene paridad correcta $\approx m \in paridadOk$.
- El despachante se encuentra en el estado s y almacena los mensajes $buffer \approx Despachante\,(buffer, s)$.
- Mientras el despachante esta recibiendo, se recibe por el canal n un mensaje $c_n?$ y se enviá por el canal de retorno n el resultado $e_n!$ de la operación $\approx Canal_n$.
- El despachante está recibiendo y su buffer se llenó $\approx bufferLleno$.
- El despachante está enviando de a uno los mensajes a través del canal $r! \approx bufferVaciando$.
- El despachante está enviando y su buffer esta vacío $\approx bufferVacio$.

$$[MSG]$$

$$estado ::= enviando \mid recibiendo$$

$$error ::= yes \mid no$$

$$\mid paridadOk : \mathbb{P} MSG$$

Despachante

$buffer : seq\ MSG$

$s : estado$

DespachanteInit

Despachante

$buffer = \langle\rangle$

$s = recibiendo$

Canal$_1$*Ok*

$\Delta Despachante$

$c_1? : MSG;\ \ e_1! : error$

$\# buffer \leq 9$

$s = recibiendo$

$c_1? \in paridadOk$

$buffer' = buffer^\frown \langle c_1? \rangle$

$s' = s$

$e_1! = no$

Canal$_1$*Error*

$\Xi Despachante$

$c_1? : MSG;\ \ e_1! : error$

$\# buffer \leq 9$

$s = recibiendo$

$c_1? \notin paridadOk$

$buffer' = buffer$

$s' = s$

$e_1! = yes$

$Canal_1 \triangleq Canal_1 Ok \lor Canal_1 Error$

$Canal_2 \triangleq Canal_1\ [c_2?/c_1?, e_2!/e_1!]$

BufferLleno

$\Delta Despachante$

$\# buffer = 10$

$s = recibiendo$

$buffer' = buffer$

$s' = enviando$

BufferVaciando

$\Delta Despachante$

$r! : MSG$

$\# buffer > 0$

$s = enviando$

$s' = s$

$buffer' = tail\,(buffer)$

$r! = head\,(buffer)$

BufferVacio

$\Delta Despachante$

$\# buffer = 0$

$s = enviando$

$s' = recibiendo$

$buffer' = buffer$

3. Especifique una base de datos para almacenar información sobre películas. La base de datos debe ser capaz de mantener información sobre quien dirigió un film y sobre el guionista. Todo film en la base debe tener asociado un director y un guionista. Debe contemplarse el caso de películas dirigidas por varias personas y/o escritas por varios guionistas. Especifique dos operaciones que consulten la base de datos con el fin de encontrar todas las películas dirigidas por una persona en particular y lo mismo para un guionista. Especifique una operación que permita modificar el nombre del director de una película ya existente en la base. Especifique operaciones para efectuar altas y bajas de películas.

Solución sin subesquemas

- p es una película $\approx p \in PELICULA$.
- d es un director $\approx p \in DIRECTOR$.
- g es un guionista $\approx p \in GUIONISTA$.
- Hubo un error en la operación $\approx yes$.
- No hubo un error en la operación $\approx no$.
- COMPLETAR.

$[PELICULA, DIRECTOR, GUIONISTA]$

$error ::= yes \mid no$

BaseDatos

$$peliculas : PELICULA \pfun (\mathbb{P}_1 DIRECTOR \times \mathbb{P}_1 GUIONISTA)$$

BaseDatosInit

$$BaseDatos$$
$$peliculas = \varnothing$$

ModificarDirectorOk

$$\Delta BaseDatos$$
$$p? : PELICULA$$
$$do? : DIRECTOR$$
$$dn? : DIRECTOR$$
$$rep! : error$$
$$p? \in dom\,(peliculas)$$
$$do? \in peliculas\,(p?)\,.1$$
$$peliculas' = peliculas \oplus \{p? \mapsto (peliculas\,(p?)\,.1 \setminus \{do?\} \cup \{dn?\}, peliculas\,(p?)\,.2)\}$$
$$rep! = no$$

ModificarDirectorError1

$$\Xi BaseDatos$$
$$p? : PELICULA$$
$$rep! : error$$
$$p? \notin dom\,(peliculas)$$
$$rep! = yes$$

ModificarDirectorError2

$$\Xi BaseDatos$$
$$p? : PELICULA$$
$$do? : DIRECTOR$$
$$rep! : error$$
$$do? \notin peliculas\,(p?)\,.1$$
$$rep! = yes$$

$ModificarDirector \triangleq ModificarDirectorOk \lor ModificarDirectorError1 \lor ModificarDirectorError2$

NuevaPeliculaOk

$\Delta BaseDatos$
$p? : PELICULA$
$ds? : \mathbb{P}_1 DIRECTOR$
$gs? : \mathbb{P}_1 GUIONISTA$
$rep! : error$

$p? \notin peliculas$
$peliculas' = peliculas \oplus \{p? \mapsto (ds?, gs?)\}$
$rep! = no$

NuevaPeliculaError

$\Xi BaseDatos$
$p? : PELICULA$
$rep! : error$

$p? \in peliculas$
$rep! = yes$

$NuevaPelicula \triangleq NuevaPeliculaOk \lor NuevaPeliculaError$

BorrarPeliculaOk

$\Delta BaseDatos$
$p? : PELICULA$
$rep! : error$

$p? \in peliculas$
$peliculas' = \{p?\} \lhd peliculas$
$rep! = no$

BorrarPeliculaError

$\Xi BaseDatos$
$p? : PELICULA$
$rep! : error$

$p? \notin peliculas$
$rep! = yes$

$BorrarPelicula \triangleq BorrarPeliculaOk \lor BorrarPeliculaError$

BuscarPorDirectorOk

$\Xi BaseDatos$
$d? : DIRECTOR$
$rep! : \mathbb{P} PELICULA$

$rep! = \{p : PELICULA \mid p \in dom\,(peliculas) \land d? \in peliculas\,(p).1\}$

BuscarPorGuionistaOk

$\Xi BaseDatos$
$g? : GUIONISTA$
$rep! : \mathbb{P} PELICULA$

$rep! = \{p : PELICULA \mid p \in dom\,(peliculas) \land g? \in peliculas\,(p)\,.2\}$

Solución con subesquemas

- COMPLETAR.

$[NOMBRE, DIRECTOR, GUIONISTA]$

$error ::= yes \mid no$

PELICULA

$ds : \mathbb{P}_1 DIRECTOR$
$gs : \mathbb{P}_1 GUIONISTA$

BaseDeDatos

$bd : NOMBRE \pfun PELICULA$

BaseDeDatosInit

$BaseDeDatos$

$bd = \varnothing$

ModificarDirectorOk

$\Delta BaseDatos$
$p? : NOMBRE$
$do? : DIRECTOR$
$dn? : DIRECTOR$
$rep! : error$

$p? \in dom\,(bd)$
$bd\,(p?) = \Theta PELICULA$
$do? \in \Theta PELICULA.ds$
$\Theta PELICULA'.ds' = \Theta PELICULA.ds \setminus \{do?\} \cup \{dn?\}$
$\Theta PELICULA'.gs' = \Theta PELICULA.gs$
$rep! = no$

NuevaPeliculaOk

$\Delta BaseDatos$
$n? : NOMBRE$
$p? : PELICULA$
$rep! : error$

$n? \notin dom\,(bd)$
$bd' = bd \oplus \{n? \mapsto p?\}$
$rep! = no$

BorrarPeliculaOk

$\Delta BaseDatos$
$p? : NOMBRE$
$rep! : error$

$n? \in dom\,(bd)$
$bd' = \{p?\} \lhd bd$
$rep! = no$

BuscarPorDirectorOk

$\Xi BaseDatos$
$d? : DIRECTOR$
$rep! : \mathbb{P} NOMBRE$

$rep! = dom\,(bd \rhd \{\Theta PELICULA : PELICULA \mid d? \in \Theta PELICULA.ds\})$

Solución sencilla

BaseDeDatos

$ds : PELICULA \pfun \mathbb{P}_1 DIRECTOR$
$gs : PELICULA \pfun \mathbb{P}_1 GUIONISTA$

ModificarDirectorOk

$\Delta BaseDatos$
$p? : PELICULA$
$do? : DIRECTOR$
$dn? : DIRECTOR$
$rep! : error$

$p? \in dom\,(ds)$
$do? \in ds\,(p?)$
$ds' = ds \oplus \{p? \mapsto ds\,(p?) \setminus \{do?\} \cup \{cn?\}\}$
$gs' = gs$
$rep! = no$

NuevaPeliculaOk

$\Delta BaseDatos$
$p? : PELICULA$
$ds? : \mathbb{P}_1 GUIONISTA$
$gs? : \mathbb{P}_1 GUIONISTA$
$rep! : error$

$p? \notin dom\,(ds) \land p? \notin dom\,(gs)$
$ds' = ds \oplus \{p? \mapsto ds?\}$
$gs' = gs \oplus \{p? \mapsto gs?\}$
$rep! = no$

BuscarPorDirectorOk

$\Xi BaseDatos$
$d? : DIRECTOR$
$rep! : \mathbb{P} PELICULA\ \{x \in ds \mid d? \in x.2 \bullet x.1\}$

$rep! = dom\,(ds \rhd \{x : \mathbb{P} DIRECTOR \mid d? \in x\})$

4. Un servidor FTP provee de cuatro operaciones fundamentales:

   connect, que recibe un usuario y una contraseña y, si la información es correcta, permite que ese usuario utilice las restantes operaciones;

   get, que recibe un nombre de archivo y envía su contenido al usuario que lo solicitó;

   put, que también recibe un nombre de archivo pero en este caso lo agrega al sistema archivos local;

   close, que cierra la conexión abierta por un usuario.

Pensar detenidamente cuáles son los parámetros de entrada de cada operación más allá de los que son obvios.

Especificar los casos de error de cada operación seleccionada. Tener en cuenta que el servidor FTP puede manejar varias conexiones al mismo tiempo.

Solución

- COMPLETAR.

$$[USUARIO, CONTRASEÑA, ARCHIVO, CONTENIDO]$$

$$error ::= yes \mid no$$

$$chekPassword : USUARIO \pfun CONTRASEÑA$$

Servidor

$$conectados : \mathbb{P} USUARIO$$
$$archivos : ARCHIVO \pfun CONTENIDO$$

ServidorInit

$$Servidor$$

$$conectados = \varnothing$$
$$archivos = \varnothing$$

ConnectOk

$$\Delta Servidor$$
$$u? : USUARIO$$
$$p? : CONTRASEÑA$$
$$e! : error$$

$$u? \notin conectados$$
$$u? \in dom\,(checkPassword)$$
$$checkPassword\,(u?) = p?$$
$$conectados' = conectados \cup \{u?\}$$
$$archivos' = archivos$$
$$e! = no$$

UserConnected

$\Xi Servidor$
$u? : USUARIO$
$e! : error$

$u? \in conectados$
$e! = yes$

InvalidLogin

$u? : USUARIO$
$p? : CONTRASEÑA$
$e! : error$

$u? \in dom\,(checkPassword) \lor checkPassword\,(u?) = p?$
$e! = yes$

$Connect \triangleq ConnectOk \lor UserConnected \lor InvalidLogin$

GetOk

$\Xi Servidor$
$u? : USUARIO$
$a? : ARCHIVO$
$rep! : CONTENIDO$
$e! : error$

$u? \in conectados$
$a? \in archivos$
$rep! = getFile\,(a?)$
$e! = no$

UserNotConnected

$\Xi Servidor$
$u? : USUARIO$
$rep! : error$

$u \notin conectados$
$rep! = yes$

FileNotFound

$\Xi Servidor$
$a? : ARCHIVO$
$rep! : error$

$a? \notin dom\,(archivos)$
$rep! = yes$

$$Get \triangleq GetOk \lor UserNotConnected \lor FileNotFound$$

PutOk

$$\Delta Servidor$$
$$u? : USUARIO$$
$$a? : ARCHIVO$$
$$c? : CONTENIDO$$
$$e! : error$$

$$u? \in conectados$$
$$a? \notin dom\,(archivos)$$
$$archivos' = archivos \cup \{a? \mapsto c?\}$$
$$conectados' = conectados$$
$$e! = no$$

FileAlreadyExist

$$\Xi Servidor$$
$$a? : ARCHIVO$$
$$rep! : error$$

$$a? \in dom\,(archivos)$$
$$rep! = yes$$

$$Put \triangleq PutOk \lor FileAlreadyExist \lor UserConnected$$

CloseOk

$$\Delta Servidor$$
$$u? : USUARIO$$
$$e! : error$$

$$u? \in conectados$$
$$conectados' = conectados \setminus \{u?\}$$
$$archivos' = archivos$$
$$e! = no$$

$$Close \triangleq CloseOk \lor UserNotConnected$$

5. Un museo de bellas artes desea instalar un sistema de iluminación automático que mantenga la cantidad de luz dentro de cierto rango que permite a los visitantes apreciarlas obras y al mismo tiempo la luz no les produce daños serios. El museo cuenta con ventanas e iluminación artificial. A las ventanas se les colocan persianas tipo americanas que pueden ser movidas por motores eléctricos de la siguiente manera:

- Las varillas horizontales pueden moverse desde su posición inicial (ventana cerrada) de a un grado hasta que el motor emite una señal que indica que ya no es posible seguir girando, y viceversa.
- La persiana puede elevarse o bajarse de a un centímetro hasta que el motor emite una señal que indica que no es posible seguir bajando o subiendo.

Por otro lado, los artefactos de iluminación artificial son modificados de manera tal que es posible regular su intensidad (aumentándola o dismunuyendola) discretamente.

Al mismo tiempo cerca de cada obra de arte se ha instalado un sensor que mide la cantidad de luz que llega. Este emite una señal cada vez que la cantidad de luz aumenta o disminuye en cierta medida fija. El sensor puede ser inicializado desde el exterior lo que hace que este emita varias señales consecutivas desde su estado de máxima obscuridad hasta medir la luz real en ese momento.

En primer lugar, el sistema de software deberá mantener la cantidad de luz sobre cada obra dentro de ciertos límites; y en segundo lugar, el sistema deberá optar primero por utilizar la luz proveniente de las ventanas y si esta no es suficiente deberá hacer uso de iluminación artificial.

Solución COMPLETAR.

6. Modele un CVS (Control Version System) con las siguientes características:

   a) Se administran programas que constan de un nombre, un encabezado y un cuerpo. La dos últimas partes son modificables por un usuario. Deben especificarse sendas operaciones. Todas las partes excepto el nombre pueden estar vacías.

   b) Todos los programas yacen en un repositorio común denominado línea base.

   c) Un cierto conjunto de usuarios son los únicos autorizados a trabajar con los programas almacenados en la línea base.

   d) Cada usuario de dicho conjunto debe tener uno de los siguientes roles: Lector, Editor o Autor. Autor implica Editor implica Lector.

e) El CVS lleva un registro de todos los programas almacenados en la línea base y los usuarios y roles autorizados a trabajar. Además lleva la historia de todas las versiones de cada uno de los programas. El CVS identifica un programa por su nombre.

Las operaciones (totales) del CVS a especificar son:

Create: un Autor agrega a la línea base un nuevo programa.

Get: un Lector toma de la línea base la última versión de un programa pero no puede devolverla. (En cualquier operación «tomar» significa que se hace una copia del programa fuera de la línea base pero en ningún caso se elimina el programa de aquella)

Edit: un Editor toma de la línea base la última versión de un programa y puede retornarla con modificaciones. Al retornarla está generando una nueva versión del programa. Una vez que un programa fue editado por un usuario ningún otro usuario (incluido el mismo) podrá editarlo o crear otro con el mismo nombre hasta que se efectúe el Delta correspondiente.

Delta: el Editor que realizó un Edit de cierto programa lo retorna a la línea base. Una devolución se registrará como nueva versión siempre y cuando difiera de la anterior.

Delete: un Autor elimina un programa de la línea base. Al hacerlo el CVS pierde toda la historia de las modificaciones efectuadas.

Solución COMPLETAR.

Statecharts, CSP y TLA

Para cada uno de los ejercicios se deberá escribir las designaciones de los fenómenos de interés y luego se deberá escribir un modelo Statechart para el DK, R y S. En todos los problemas el objetivo es programar un software que controle los dispositivos físicos mencionados según los requerimientos que se expresan en cada caso. En cada problema fijarse con qué lenguajes debe resolverse.

1. Describa un sistema de iluminación consistente en una lámpara y dos botones. Si la lámpara está apagada, al pulsar cualquiera de los dos botones se enciende y viceversa. Resolver en: Todos los lenguajes.