Z

Para cada uno de los ejercicios se deberá escribir las designaciones de los términos formales 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 $\approx on$.
   - La lampara esta apagada $\approx off$.
   - La lampara se encuentra en el estado $l \approx Sistema\,(l)$.
   - Se presiona alguno de los botones $\approx Presionar$.

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

$$\begin{array}{|l} Sistema \\ \hline l : lampara \\ \hline \end{array}$$

$$\begin{array}{|l} SistemaInit \\ \hline Sistema \\ \hline l = off \\ \hline \end{array}$$

$$\begin{array}{|l} PresionarEnOn \\ \hline \Delta Sistema \\ \hline l = off \\ l' = on \\ \hline \end{array}$$

$$\begin{array}{|l} PresionarEnOff \\ \hline \Delta Sistema \\ \hline l = on \\ l' = off \\ \hline \end{array}$$

$$Presionar == PresionarEnOn \lor PresionarEnOff$$

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 en el buffer $buffer \approx Despachante\,(buffer, s)$.
- 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 $\approx Canal_n\,(c_n?, e_n!)$.
- El despachante está recibiendo y su buffer se llenó $\approx bufferLleno$.
- El despachante está enviando de a uno los mensajes a través del canal $r! \approx bufferVaciando\,(r!)$.
- 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*

$\Xi Despachante$

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

$c_1? \notin paridadOk$

$e_1! = yes$

$Canal_1 == Canal_1 Ok \lor Canal_1 Error \lor BufferLleno \lor BufferVaciando \lor BufferVacio$

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

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 $\approx no$.
- La base de datos almacena las películas $peliculas \approx BaseDatos\,(peliculas)$.

- Se modifica el director $do?$ de la película $p?$ por el director $dn?$ y se informa el resultado $e!$ de la operación $\approx ModificarDirector\,(p?, do?, dn?, e!)$.
- Se agrega a la base de datos la película $p?$ dirigida por los directores $ds?$ y guionada por los guionistas $gs?$, informando el resultado $e!$ de la operación $\approx NuevaPelicula\,(d?, ds?, gs?, e!)$.
- Se borra la película $p?$ de la base de datos y se informa el resultado $e!$ de la operación $\approx BorrarPelicula\,(p?, e!)$.
- Se buscan todas las películas $r!$ dirigidas por $d?$ $\approx BuscarPorDirector\,(d?, r!)$.
- Se buscan todas las películas $r!$ guionadas por $g?$ $\approx BuscarPorDirector\,(d?, g!)$.

$[PELICULA, DIRECTOR, GUIONISTA]$

$error ::= yes \mid no$

BaseDatos

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

BaseDatosInit

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

ModificarDirectorOk

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

PeliculaInvalida

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

DirectorInvalido

$\Xi BaseDatos$
$p? : PELICULA$
$do? : DIRECTOR$
$e! : error$

$do? \notin peliculas\,(p?)\,.1$
$e! = yes$

$ModificarDirector == ModificarDirectorOk \lor PeliculaInvalida \lor DirectorInvalido$

NuevaPeliculaOk

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

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

PeliculaExistente

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

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

$NuevaPelicula == NuevaPeliculaOk \lor PeliculaExistente$

BorrarPeliculaOk

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

$p? \in dom\,(peliculas)$
$peliculas' = \{p?\} \lhd peliculas$
$e! = no$

$BorrarPelicula == BorrarPeliculaOk \lor PeliculaInvalida$

BuscarPorDirectorOk

$$\Xi BaseDatos$$
$$d? : DIRECTOR$$
$$r! : \mathbb{P} PELICULA$$
$$r! = \{p : PELICULA \mid p \in dom\,(peliculas) \land d? \in peliculas\,(p).1\}$$

BuscarPorGuionistaOk

$$\Xi BaseDatos$$
$$g? : GUIONISTA$$
$$r! : \mathbb{P} PELICULA$$
$$r! = \{p : PELICULA \mid p \in dom\,(peliculas) \land g? \in peliculas\,(p).2\}$$

Solución con subesquemas

- COMPLETAR DESIGNACIONES.

$[NOMBRE, DIRECTOR, GUIONISTA]$

$error ::= yes \mid no$

PELICULA

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

BaseDeDatos

$$bd : NOMBRE \pfun PELICULA$$

BaseDeDatosInit

$$BaseDeDatos$$
$$bd = \varnothing$$

PELICULAModificarDirector

$$\Delta PELICULA$$
$$do? : DIRECTOR$$
$$dn? : DIRECTOR$$
$$do? \in ds$$
$$ds' = ds \setminus \{do?\} \cup \{dn?\}$$
$$gs' = gs$$

BuscarPorDirectorOk

$$\Xi BaseDatos$$
$$d? : DIRECTOR$$
$$r! : \mathbb{P} NOMBRE$$
$$r! = dom\,(bd \rhd \{p : PELICULA \mid d? \in p.ds\})$$

ModificarDirectorOk

$$\Delta BaseDatos$$
$$PELICULAModificarDirector$$
$$p? : NOMBRE$$
$$e! : error$$
$$p? \in dom\,(bd)$$
$$bd\,(p?) = \Theta PELICULA$$
$$bd' = bd \oplus \{p? \mapsto \Theta PELICULA'\}$$
$$e! = no$$

NuevaPeliculaOk

$$\Delta BaseDatos$$
$$n? : NOMBRE$$
$$p? : PELICULA$$
$$e! : error$$
$$n? \notin dom\,(bd)$$
$$bd' = bd \cup \{n? \mapsto p?\}$$
$$e! = no$$

BorrarPeliculaOk

$$\Delta BaseDatos$$
$$p? : NOMBRE$$
$$e! : error$$
$$n? \in dom\,(bd)$$
$$bd' = \{p?\} \lhd bd$$
$$e! = no$$

Solución sencilla

- COMPLETAR DESIGNACIONES.

BaseDeDatos

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

ModificarDirectorOk

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

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

NuevaPeliculaOk

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

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

BuscarPorDirectorOk

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

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

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 DESIGNACIONES.

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

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

$$chekPassword : USUARIO \pfun CONTRASEÑA$$

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

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

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

$Connect == ConnectOk \lor UserConnected \lor InvalidLogin$

GetOk

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

$u? \in conectados$
$a? \in dom\,(archivos)$
$r! = archivos\,(a?)$
$e! = no$

UserNotConnected

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

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

FileNotFound

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

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

$Get == GetOk \lor UserNotConnected \lor FileNotFound$

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

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

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

$Put == PutOk \lor FileAlreadyExist \lor UserConnected$

CloseOk

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

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

$Close == CloseOk \lor UserNotConnected$

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 $[ENCABEZADO, CUERPO, NOMBRE, USUARIO]$

$error ::= yes \mid no$

$estado ::= editando \mid libre$

$as : \mathbb{P} USUARIOS$
$es : \mathbb{P} USUARIOS$
$ls : \mathbb{P} USUARIOS$

$PROGRAMA$
$e : ENCABEZADO$
$c : CUERPO$

REPOSITORIO

$his : seq PROGRAMA$
$s : estado$
$u : USUARIO$

REPOSITORIOInit

REPOSITORIO
$u? : USUARIO$
$p? : PROGRAMA$

$his = \langle p? \rangle$
$u = u?$
$s = libre$

CVS

$rs : NOMBRE \pfun REPOSITORIO$

CVSInit

CVS

$rs = \varnothing$

CreateOk

ΔCVS
REPOSITORIOInit
$n? : NOMBRE$
$e! : error$

$n? \notin dom\,(rs) \land u? \in as$
$rs' = rs \cup \{n? \mapsto \Theta REPOSITORIO\}$
$e! = no$

REPOSITORIOGetOk

$\Xi REPOSITORIO$
$u? : USUARIO$
$p! : PROGRAMA$
$e! : error$

$u? \in ls$
$p! = last\,(his)$
$e! = no$

GetOk

$$\Xi CVS$$
$$REPOSITORIOGetOk$$
$$n? : NOMBRE$$

$$n? \in dom\,(rs)$$
$$\Theta REPOSITORIO = rs\,(n?)$$

REPOSITORIOEditOk

$$\Delta REPOSITORIO$$
$$REPOSITORIOGetOk$$
$$e! : error$$

$$u? \in es$$
$$s = libre$$
$$s' = editando$$
$$u' = u?$$
$$his' = his$$
$$e! = no$$

REPOSITORIOaCVS

$$\Delta CVS$$
$$n? : NOMBRE$$

$$n? \in dom\,(rs)$$
$$\Theta REPOSITORIO = rs\,(n?)$$
$$rs' = rs \oplus \{n? \mapsto \Theta REPOSITORIO'\}$$

$$EditOk == REPOSITORIOaCVS \land RepositorioEditOk$$

REPOSITORIODeltaOk

$$\Delta REPOSITORIO$$
$$p? : PROGRAMA$$
$$u? : USUARIO$$
$$e! : error$$

$$u? \in es \land u? = u$$
$$s = editando$$
$$p? \neq last\,(his)$$
$$s' = libre$$
$$u' = u$$
$$his' = his^\frown \langle p? \rangle$$
$$e! = no$$

$$DeltaOk == REPOSITORIOaCVS \land RepositorioDeltaOk$$

$$
\begin{array}{|l}
\hline
\textit{DeleteOk} \\
\Delta CVS \\
n? : NOMBRE \\
u? : USUARIO \\
\hline
u? \in as \\
n? \in dom\,(rs) \\
rs' = \{n?\} \lhd rs \\
\hline
\end{array}
$$

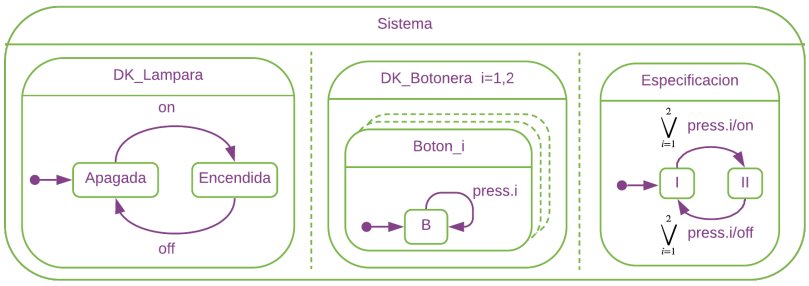
Statecharts, CSP y TLA

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. Resuelva cada problema en todos los lenguajes.

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

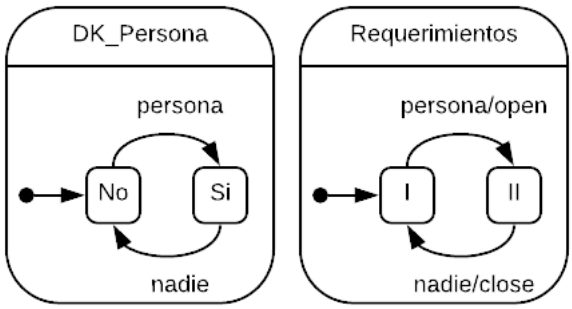
   - Se presiona el botón ≈ press. Enviroment controled. Shared.
   - Se enciende la lampara ≈ on. Machine controled. Shared.
   - Se apaga la lampara ≈ off. Machine controled. Shared.

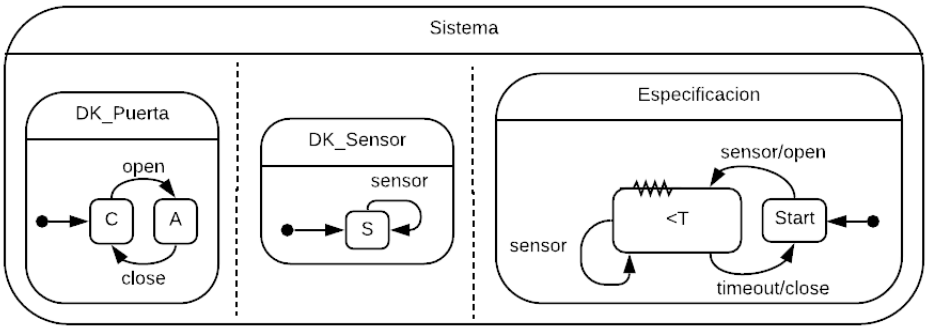


2. Las puertas de los aeropuertos están provistas de un mecanismo que permite que una persona no necesite abrirla o cerrarla cada vez que la va a atravesar. Para ello poseen detectores de ambos lados que le permiten saber cuando una persona se encuentra en una zona determinada cercana a la misma, lo cual es un indicio de que se inentará atravesarla. Una vez que la persona se retira de esta zona de influencia y transcurren T unidades de tiempo, la puerta se cierra.

Solución

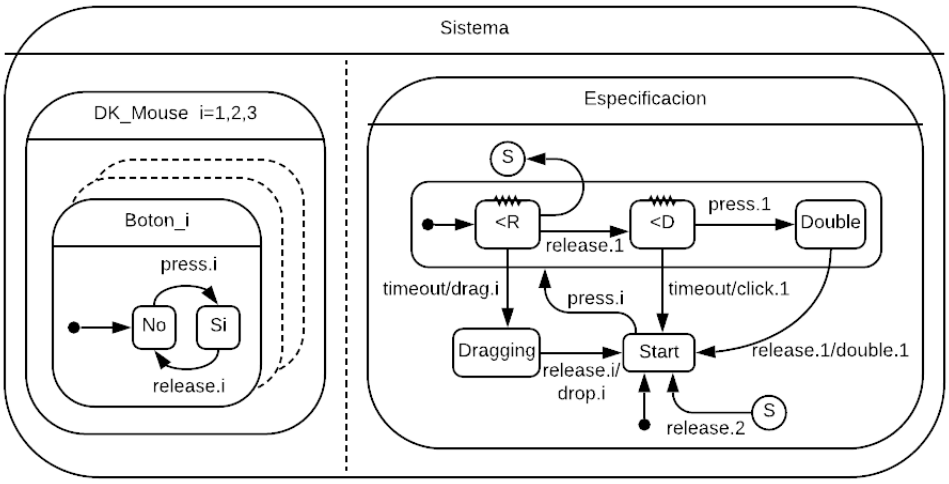
- Se abre la puerta ≈ open. Machine controled. Shared.
- Se cierra la puerta ≈ close. Machine controled. Shared.
- Hay al menos una persona en la zona ≈ persona. Enviroment controled. Unshared.
- Salen todas las personas de la zona ≈ nadie. Enviroment controled. Unshared.
- El sensor detecta una persona ≈ sensor. Enviroment controled. Shared.
- Tiempo de cortesía ≈ T.





3. Describa el funcionamiento de un mouse de tres botones. El botón central no tiene asignada ninguna función. Con el botón derecho solo se puede hacer click simple, y arrastrar. Con el botón izquierdo click simple, doble click y arrastrar. Pulsar más de un botón al mismo tiempo no tiene efecto.

Solución



4. Los pacientes de una guardia de terapia intensiva son monitoreados por dispositivos electrónicos analógicos adosados a sus cuerpos mediante sensores de varias clases. A través de los sensores los dispositivos miden los signos vitales de los pacientes: un dispositivo mide el pulso, otro la temperatura, otro la presión sanguínea.

   Se necesita un programa que lea los signos vitales, con una frecuencia especificada para cada paciente. Los factores leídos será comparados con rangos de seguridad especifica- dos para cada paciente y las lecturas que excedan dichos rangos serán reportadas con mensajes de alarma en el monitor del box de enfermería. También debe mostrarse un mensaje apropiado si falla alguno de los dispositivos.

   Solución COMPLETAR.

5. Sobre una calle de un solo sentido existe un cruce peatonal. Para mayor seguridad de las personas que lo utilizan se decide colocar un semáforo con dos grupos de botones ubicados a cada uno de los lados de la calle.

   Cuando el semáforo está en verde y se presiona uno de los botones, debe pasar T_1 unidades de tiempo para que se encienda la luz amarilla del semáforo y T_2 para que se encienda la roja. Esta última permanecerá encendida por T_3 unidades de tiempo.

   Presionar alguno de los botones no tendrá ningún efecto, hasta que hayan pasado al menos T_4 unidades de tiempo desde la última vez que se presionó uno.

   Analizar la situación cuando hay 2 semáforos (cada uno con su propia botonera), los cuales deben respetar los mismos requerimientos dados en la parte inicial del ejercicio.

   Solución COMPLETAR.

6. Una celda de producción muy simple consta de dos cintas transportadoras que funcionan concurrentemente y un robot que toma ítems de una u otra según lo que ordene la computadora que controla la celda y luego los suelta en algún lugar. Se puede detener o reanudar la marcha de cada cinta (independientemente) por medio de una única señal por cinta. Hay sensores que detectan el paso de un ítem por cada una de

las cintas. Estos sensores pueden ser configurados para que emitan las señales apropiadas en caso de que dos ítem sobre la misma o diferentes cintas estén demasiado juntos. Dos ítem están demasiado juntos si el robot no es capaz de tomarlos a ambos sin que uno se caiga. El requerimiento para el sistema de control es que no debe caerse ningún ítem de ninguna cinta.

Solución COMPLETAR.

7. Considere el problema 6. Suponga que los sensores no son tan inteligentes como para poder indicar la proximidad de dos ítem. Es decir, sólo son capaces de emitir una señal cada vez que pasa un ítem. Por otro lado, el robot demora un tiempo T_r para tomar un ítem, soltarlo y volver a estar en condiciones de tomar otro. Los requerimientos para el sistema de control son los siguientes:

   - No debe caerse ningún ítem.
   - Eventualmente todos los ítems de ambas cintas será tomados por el robot.
   - Por razones de eficiencia se espera que se tomen ítems de ambas cintas de forma más o menos regular (siempre que haya alguno disponible).

   Solución COMPLETAR.

8. Problema 5 de la sección anterior.

   Solución COMPLETAR.

9. El automóvil tiene una llave de encendido; la misma puede estar en tres posiciones, apagado, contacto o encendido. Se pasa de la primera posición a la segunda, de la segunda a la tercera y viceversa. Cuando la llave esta en contacto se pueden subir y bajar los vidrios. Cuando la llave esta en encendido se puede conducir.

   Este automóvil tiene caja de cambios automática, es decir que los cambios de marcha se efectúan automáticamente con el aumento o disminución de la velocidad. Posee 4 marchas y punto muerto. Está en punto

muerto cuando el auto esta frenado, Está en 1ra cuando su velocidad está entre 0.1 y 30km/h, 2da entre 30,1 y 50km/h, 3ra entre 50,1 y 70 km/h, y 4ta con mas de 70km/h.

Cuando la llave se apaga se tienen 20" extras por si se quieren subir o bajar los vidrios. Las luces del auto se controlan con una perilla mecánica, de tres posiciones: apagadas, posición y cortas. Las luces sólo están encendidas (en posición o cortas) cuando la llave de encendido del auto está en contacto o encendido.

Al pasar la llave a «apagado», se apagan las luces, pero la perilla mecánica se mantiene en su estado. Al volver a pasar la llave a posición, el estado de las luces dependerá del estado de la perilla. Esto implica que la perilla mecánica es independiente de la llave del auto, pero el estado de las luces no.

Solución COMPLETAR.

10. El termómetro posee una tapa que al abrirse muestra un display (apagado) más dos botones, uno para encender el termómetro y otro para seleccionar la escala de temperaturas (F° o C°). El termómetro se apaga al cerrar la tapa o cuando se termina la batería. Además, el termómetro posee una punta que se introduce en el oído del paciente y un botón que debe ser pulsado al menos un segundo para que la lectura sea exitosa. Deben mediar 10 segundos entre dos mediciones consecutivas. El display muestra un indicador de batería, la temperatura medida (o nada si recién se encendió) y la escala de temperaturas seleccionada (este dato es «recordado» por el termómetro aun luego de ser apagado). Si una medición de temperatura es fallida, se muestra una E invertida. Una vez pulsado el botón de encendido, el termómetro chequea su funcionamiento durante 5 segundos y luego emite los datos iniciales al display (cualquier acción que se intente antes de esos 5 segundos debe ser obviada). Un dato importante que se registra durante el chequeo es el estado de la batería. Esta pierde un quanto de energía cada un segundo si sólo se usa el display, 3 quantos si se cambia la escala y 5 en cada medición errónea o no de temperatura. Cuando quedan 10 quantos de energía se enciende un icono en el display.

Solución COMPLETAR.

11. El horno microondas posee un panel de control semejante a algunos modelos populares. Los botones numerados del 1 al 4 se utilizan para fijar la potencia de cocción. La perilla circular puede girar en ambos sentidos y sirve para fijar el tiempo de cocción: si lo hace en sentido horario aumenta el tiempo de a 5 segundos, en el sentido contrario disminuye en la misma cantidad.

    Normalmente, en el display se muestra la hora actual pero cuando se fija el tiempo de cocción se la reemplaza por el tiempo que el usuario selecciona a medida que hace girar la perilla. Una vez que detiene la perilla, debe seleccionar una de las potencias de cocción y finalmente pulsar el botón Start. Al hacerlo la puerta debe estar cerrada y en el display se comienza a decrementar el tiempo de a un segundo.

    - Si se pulsa Stop se borra el tiempo de cocción, se resetea la selección de la potencia y se vuelve a mostrar la hora.
    - Si durante la cocción se abre la puerta, se detiene el tiempo de cocción y si se vuelve a cerrar se deberá pulsar Start nuevamente para continuar la cocción desde el punto donde se la abandonó.
    - Se puede cambiar la potencia de cocción mientras se está cocinando.
    - Si durante la cocción el usuario hace girar la perilla, el tiempo de cocción debe actualizarse.

    La cocción se detendrá automáticamente cuando el tiempo llegue a cero, luego de lo cual se volverá a mostrar la hora y se reseteará la potencia seleccionada.

    Solución COMPLETAR.

12. Un barco de guerra cuenta con un radar principal (RP), dos cañones idénticos ($C1$ y $C2$) cada uno con su radar ($RS1$ y $RS2$) y una sala de comando desde donde se indica si un objetivo apuntado por alguno de los dos cañones debe ser abandonado o destruido

    El RP está permanentemente escrutando el área de disparo en busca de objetivos. Cada vez que RP encuentra uno se lo debe asignar a uno de los dos cañones si alguno está libre (sin objetivo en su mira); al asignarlo se le indica la posición y la trayectoria del objetivo

La posición inicial del cañón al asignársele un nuevo objetivo es conocida por el software. En función de la posición inicial y de los datos transmitidos por el RP, el software apunta el cañón y el RS al objetivo. Normalmente, un RS está inactivo hasta que le es asignado un objetivo. Desde ese momento el RS comienza a seguir su objetivo hasta que:

- Recibe la orden de disparar;
- Recibe la orden de abandonar el objetivo;
- El objetivo se va fuera de alcance.

En cualquiera de los tres casos, el RS luego pasa a estar inactivo. El seguimiento del objetivo por parte de un RS implica mover el cañón lo cual está a cargo de un motor. El software tiene a su cargo emitir las ordenes de movimiento del motor las cuales tienen la forma: $Arriba(ang)$, $Abajo(ang)$, $GiroA(ang)$ y $GiroH(ang)$, donde ang es un ángulo.

Nota Resolver en CSP y TLA.

Solución COMPLETAR.