Ingeniería de Software 1:
enunciados de ejercicios de examen

Estudiantes de LCC*

junio de 2015

Este documento contiene enunciados de ejercicios que se tomaron en algún examen de Ingeniería de Software 1 entre 2007 y 2015. Fue armado en base a una colección de escaneos de enunciados que hemos ido generando por medio del aporte de diversos compañeros a lo largo de los años.

1 Teoría

1.1 Teoría general

Ejercicio 1 Explique la diferencia entre requerimientos y especificaciones.

*Este documento fue hecho por Mariano Street, pero no es más que un ordenamiento de todo el contenido de la colección de enunciados de exámenes. La misma es producto del aporte de muchos compañeros.

1.2 Teoría de Z

Ejercicio 1 Tomado en: 2010.07.02(r1), 2015.04.29(p1).

Considere la siguiente especificación Z.

$$A \mathrel{\widehat=} [...]$$
$$Op_A^1 \mathrel{\widehat=} [\Delta A...]$$
$$Op_A^2 \mathrel{\widehat=} [\Delta A...]$$
$$E \mathrel{\widehat=} [f : \mathbb{N} \pfun A]$$

Especifique una operación sobre E, utilizando promoción y composición de operaciones, que tome dos A, aplique Op_A^1 sobre la primera instancia, Op_A^2 sobre la segunda y agregue a f las instancias de A luego de haberle aplicado las operaciones respectivas.

Ejercicio 2 Tomado en: 2012.04.13(p1).

En Z no existe el tipo booleano (es decir el tipo que contiene solo los dos valores de verdad, *true* y *false*) y no se puede definir. Más aun, se desaconseja utilizar tipos similares (es decir, tipos que intentan capturar el valor de verdad de cierto predicado). Mediante un ejemplo explique cómo lo reemplazaría. El ejemplo debe mostrar primero una especificación que use un tipo equivalente al booleano y luego la misma especificación pero sin utilizarlo.

Ejercicio 3 Indique la razón por la cual Z no incluye y desalienta el uso del tipo *Bool*, entendido como el tipo que tiene solo dos valores cada uno de los cuales representa uno de los valores de verdad de la lógica clásica.

Ejercicio 4 Describa cómo puede reemplazarse el tipo *bool* en Z.

Ejercicio 5 Explique la semántica del operador θ de Z.

Ejercicio 6 Determine el tipo de la expresión $f \cup \{x \mapsto y\}$ siendo $f : X \pfun Y;\ x : X;\ y : Y$; justifique su respuesta.

Ejercicio 7 Defina formalmente el operador Z de composición ($\fatsemi$).

Ejercicio 8 Sean A y B dos esquemas Z y ♣ un conector lógico básico. Explique formalmente el significado del esquema $A \clubsuit B$.

Ejercicio 9 El siguiente modelo Z tiene un error de estilo: detéctelo y corríjalo.

$[X, Y]$

$Bool ::= F \mid V$

Estado

$existen : X \pfun Bool$

*Oper*1

$\Delta Estado$

$x? : X$

$x? \notin \mathrm{dom}\ existen$

$existen' = existen \cup \{x? \mapsto F\}$

*Oper*2

$\Delta Estado$

$x? : X$

$x? \in \mathrm{dom}\ existen$

$existen' = existen \cup \{x? \mapsto F\}$

Ejercicio 10 Explique y ejemplifique las dos formas vistas en clase para formalizar los invariantes de estado de una especificación Z. Analice las ventajas y desventajas de ambas formas.

Ejercicio 11 Determine si las funciones parciales en Z constituyen un tipo o no. En cualquier caso justifique la respuesta.

1.3 Teoría de Statecharts y CSP

Ejercicio 1 Tomado en: 2009.08.07(r2):a, 2010.06.15(p2):b, 2015.06.10(p2):b.

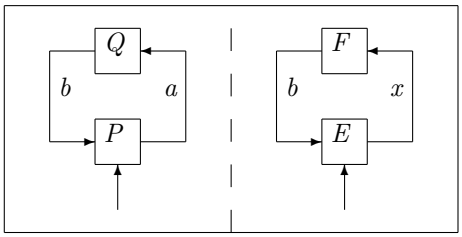
Para cada uno de los procesos siguientes, encuentre el proceso secuencial equivalente, justificando cada paso de su cálculo.

a) $P = a \to b \to STOP \| (a \to a \to STOP \sqcap a \to c \to STOP)$

b) $Q = (b \to b \to STOP \Box a \to b \to STOP) \| a \to c \to STOP$

c) $R = c \to a \to STOP \| (a \to b \to STOP \Box b \to b \to STOP)$

Ejercicio 2 Muestre un Statechart equivalente al que se muestra en la figura que no haga uso de los estados AND.



Ejercicio 3 Tomado en: 2015.06.10(r2).

Traduzca el Statechart de la figura anterior en un proceso CSP equivalente. Si no puede hacerlo, justifique su respuesta.

Ejercicio 4 Explique la diferencia entre los operadores $\Box$ y ∇. Luego muestre cómo representar ∇ en Statecharts.

Ejercicio 5 Tomado en: 2010.06.04.

Modele un proceso CSP que debe recibir cien señales en cualquier orden y, una vez que las ha recibido a todas, debe comportarse como el proceso W.

Ejercicio 6 Tomado en: 2010.06.04.

Explique en pocas líneas el concepto semántico utilizado en CSP que permite formalizar la idea de no determinismo.

Ejercicio 7 Explique las diferencias y semejanzas entre los operadores CSP: $|$, $\Box$ y $\sqcap$.

Ejercicio 8 Explique con la debida precisión la semántica del modelo de concurrencia de CSP.

Ejercicio 9 Muestre un ejemplo sencillo donde se pueda apreciar la diferencia de los modelos de concurrencia de Statecharts y CSP.

Ejercicio 10 Escriba en CSP y explique cada una de las siguientes leyes: "Interleaving", "Deadlock", Sincronización y "IEOF".

Ejercicio 11 Determine si la siguiente proposición es correcta o no (es decir si la igualdad es correcta o no) y justifique su respuesta.

$$c?x \to P;\ Q = c?x \to (P;\ Q)$$

Ejercicio 12 Describa el modelo de fallas y divergencias de CSP, explicando y definiendo los conceptos de traza, rechazo, falla y divergencia.

Ejercicio 13 Explique y ejemplifique el concepto de pasos intrascendentes.

1.4 Teoría de TLA$^+$

Ejercicio 1 Tomado en: 2007.03.09.

Explique y ejemplifique las condiciones bajo las cuales equidad débil no es equivalente a equidad fuerte.

Ejercicio 2 Tomado en: 2008.08.08.

Explique brevemente las ventajas de tener un lenguaje de especificación no tipado.

Ejercicio 3 Tomado en: 2010.02.12.

Explique y justifique el significado de la fórmula final de una especificación TLA.

Ejercicio 4 Tomado en: 2010.05.28.

Explique y ejemplifique la forma de tener una cantidad ilimitada de instancias de un módulo en TLA$^+$. Luego explique el significado formal de una expresión de la forma $...H(i)!Action...$ donde $H(i)$ es una instancia del módulo H y $Action$ es una de las acciones definidas en H.

Ejercicio 5 Tomado en: 2010.08.06.

Explique la contribución del concepto de máquina cerrada a la teoría de especificaciones de sistemas concurrentes.

Ejercicio 6 Tomado en: 2010.12.10.

Explique los conceptos de vitalidad, seguridad y equidad según se estudiaron en clase.

Ejercicio 7 Tomado en: 2011.02.25.

Describa formalmente con cierto detalle la semántica de una especificación TLA de la forma:

$$Init \wedge \Box[Op_1 \vee Op_2]_v \wedge WF_v(Op_1)$$

Ejercicio 8 Tomado en: 2011.09.09.

a) Respecto del lenguaje TLA escriba el significado formal de $\Box[A]_v$ con respecto a una ejecución, donde A es una acción y v es una variable.

b) Explique la razón por la cual Lamport sugiere escribir la vitalidad de un sistema con fórmulas de equidad.

Ejercicio 9 Explique la incidencia del teorema de Alpern-Schneider en el lenguaje de especificación TLA.

Ejercicio 10 Tomado en: 2012.02.10, 2015.07.10.

Indique, justificando su respuesta, todos los conceptos teóricos importantes que utiliza Lamport en TLA$^+$para definir el lenguaje. Por ejemplo, el teorema de Alpern-Schneider.

Ejercicio 11 Explique y ejemplifique la diferencia entre *EXTENDS* e *INSTANCE* en TLA$^+$.

Ejercicio 12 Defina formalmente el concepto de estado y explique por qué en TLA los estados son estados del universo y cuál es su utilidad en las especificaciones.

1.5 Teoría de Z/EVES

Ejercicio 1 Explique lo más formalmente posible las situaciones en las cuales Z/EVES genera obligaciones de prueba automáticamente.

Ejercicio 2 Explique formalmente el efecto que tiene sobre la prueba $C \models P$ el comando *split* A, donde A es un predicado.

2 Práctica

2.1 Práctica de Z

Ejercicio 1 Tomado en: 2010.05.04(p1).

Un proceso puede estar en tres estado: activo, pasivo o muerto. La señal *activate* pasa el proceso de pasivo a activo; la señal *kill* lo pasa de pasivo a muerto; la señal *suspend* lo pasa de activo a pasivo. El estado inicial de un proceso es pasivo.

En un cierto sistema cada proceso se identifica por un *identificador de proceso*. El sistema presenta una interfaz que permite suscribir un proceso a una de las señales mencionadas en el párrafo anterior. Es decir, si una de las señales aparece, el sistema la comunica a todos los procesos suscritos a esa señal (lo que implica que todos ellos cambian de estado de acuerdo a las reglas mencionadas más arriba). Puede suscribirse un proceso a más de una señal simplemente utilizando la interfaz varias veces. El usuario de la interfaz es quien determina el identificador para el proceso que se esté suscribiendo; el sistema rechazará la suscripción si el identificador ya está usado para la misma señal.

El sistema establece que uno de los procesos es el primario; cuando aun no se ha suscrito ningún proceso el primario es un proceso especial llamado *idle*. Cada vez que llega una señal *activate*, el sistema elige aleatoriamente entre los procesos afectados al que será el nuevo proceso primario.

a) Especifique en Z usando promoción de operaciones los requerimientos anteriores.
Nota: para modelar la aleatoriedad es suficiente la parte más básica de la teoría de conjuntos.

b) Escriba los invariantes de estado del problema.

c) La señal *killnow* hace que un proceso activo pase a estar muerto. Modele esta operación, usando la técnica más adecuada de Z, tanto a nivel de proceso individual como a nivel de sistema de procesos.

Ejercicio 2 Tomado en: 2010.07.02(r1).

El equipo de diseño ha definido la siguiente interfaz para un módulo que representa una *lista de control de acceso* (ACL).

- $AddUserRight(u, r)$: agrega al usuario u en la ACL con el permiso r.
- $AddGrpRight(g, r)$: agrega al grupo g en la ACL con el permiso r.
- $IsReader(u)$: determina si u es un lector de la ACL, es decir si u tiene permiso de lectura o pertenece a un grupo que tiene permiso de lectura.
- $IsWriter(u)$: determina si u es un escritor de la ACL, es decir si u tiene permiso de escritura o pertenece a un grupo que tiene permiso de escritura.
- Los permisos posibles son *Read* y *Write*.
- La notación refiere al uso habitual en DOO, es decir si $a : ACL$ y $u1$ es un usuario, entonces $a.AddUserRight(u1, Read)$ agregará $u1$ con permiso *Read* en la ACL a.
- La función que asocia grupos con usuarios está disponible en el sistema.
- Se requiere un adecuado tratamiento de los errores.

Especifique en Z el problema anterior.

Ejercicio 3 Tomado en: 2012.04.13(p1).

Un hotel tiene N habitaciones con diversas capacidades (entre 2 y 8 camas) para huéspedes. A las habitaciones se accede utilizando una llave electrónica (con forma de tarjeta de crédito) que se configura en el momento en que el huésped hace el *check-in*. El conserje inserta la llave en un dispositivo conectado a una computadora y luego debe usar el sistema de gestión del hotel para configurar la llave. La operación de *check-in* incluye, entonces, tomar los datos del huésped (entre ellos el período en que permanecerá en el hotel), asignarle una habitación y configurarle la llave de la habitación. Si bien la habitación se le asigna a un huésped, esta puede ser ocupada por otras personas (familiares, amigos, etc.) y es importante registrar ese

número para que sea tenido en cuenta en la logística general del hotel. Obviamente no se puede asignar una habitación que ya está ocupada (total o parcialmente) dentro del período en que permanecerá el huésped, ni se pueden registrar más personas en la habitación que la capacidad de esta.

a) Usando promoción de operaciones especifique en Z la asignación de una habitación a un huésped.

b) Especifique la configuración de una llave para una habitación.

c) Especifique la operación de *check-in* usando de la forma más conveniente el lenguaje de especificación.

d) Especifique una operación que liste todas las habitaciones ocupadas y la fecha en que se desocuparán.

Ejercicio 4 Tomado en: 2012.04.13(p1).

En un sistema operativo existe un conjunto de procesos que esperan ser ejecutados por el *scheduler*. Este debe seleccionar uno de ellos y, al hacerlo, lo debe eliminar de ese conjunto.

Suponga que no desea especificar en detalle la política de *scheduling*. Especifique en Z la operación que describe cómo el *scheduler* selecciona el proceso.

Ejercicio 5 Tomado en: 2015.04.29(p1).

La interfaz de administración de usuarios de un sistema operativo tipo UNIX tiene los siguientes requerimientos:

- Se deben especificar las operaciones: creación de un usuario, eliminación de un usuario, "login" de un usuario presentando su contraseña, "logout" de un usuario, consulta de usuarios "logueados", agregar un usuario a su grupo de usuarios.
- Cada usuario puede estar en exactamente un grupo de usuarios. Los nombres de los grupos de usuarios son fijos.
- Tener en cuenta que un usuario siempre debe tener una contraseña la cual se guarda cifrada (piense cuidadosamente cómo especificar esto sin describir un algoritmo criptográfico particular).
- Existe un usuario especial, *root*, que es el único que puede agregar y borrar usuarios y asignar un usuario a su grupo, pero para ello debe estar "logueado".

a) Especifique en Z. Especificar solo los casos exitosos excepto para la operación "login", de la cual se deben especificar todos los casos.

b) Especifique los invariantes de estado del problema y escriba la obligación de prueba correspondiente a la operación que representa el "logout" del usuario.

Ejercicio 6 Un club está formado por socios y una comisión encargada de tomar las decisiones y mantener un estatuto. En el estatuto se describen un conjunto de reglas que los socios deben cumplir. Esta medida debe quedar registrada en un acta indicando el artículo del estatuto que fue violado por el socio. Por ejemplo: un socio tiene una cierta cantidad de cuotas atrasadas, entonces la comisión decide suspenderlo hasta que regularice su situación. De esta manera constará en acta dicha decisión haciendo referencia al artículo correspondiente.

Los miembros de la comisión deben ser socios del club y uno de ellos debe ser el presidente.

Las cuotas del club deben ser liquidadas una vez por mes. El club debe llevar un registro del estado de cuenta de cada socio, donde se identifique las cuotas pagas y las cuotas impagas.

El club tiene la posibilidad de reestructurar el estado de cuentas de un socio. Esto quiere decir que se pueden cancelar y crear nuevas deudas mensuales para conformar un nuevo plan de pago. Esta decisión es facultad de la comisión y debe quedar registrado en acta con el mismo procedimiento anteriormente mencionado.

Por último, cada socio puede consultar su estado de cuenta y cuál es el mínimo de cuotas que debe adeudar para hacer uso de los beneficios del club. Para esto se deberá consultar el estatuto y las actas correspondientes para verificar si el socio no tiene suspensiones efectivas.

Especifique en Z el problema.

Ejercicio 7 Una empresa posee una balanza para pesar camiones cargados con materia prima. El camión debe ubicarse más o menos sobre el centro de la balanza para que la pesada sea correcta. Con este fin la empresa instaló cuatro sensores en los vértices de un rectángulo imaginario de forma tal que cuando detectan que el camión está dentro de ese rectángulo, se debe bajar una barrera detrás del camión. Si el camión rebasa alguno de los laterales del rectángulo se enciende una (de dos) luz ubicada delante del camión que indica qué lado está rebasado.

Una vez que el camión está correctamente ubicado y se bajaron las barreras, el chofer debe deslizar una tarjeta magnética que lo identifica. Si la tarjeta es válida, se activa la balanza. Cuando el pesaje finaliza, se debe imprimir un tique con los datos del conductor y el peso. Luego se levantan las barreras.

Modele en Z las siguientes operaciones relacionadas con los requerimientos enunciados arriba:

a) Ingresa el camión a la balanza.

b) Un par de sensores emite la señal que indica que el camión está mal ubicado. Especificar ambos pares.

c) Encender una luz para indicarle al chofer qué lado está rebasado. Especificar ambas luces.

d) Un par de sensores emite la señal que indica que ese lado no está rebasado. Esta señal se emite si previamente se rebasó ese lado. Especificar ambos pares.

e) Apagar una luz para indicarle al chofer que ese lado ahora está bien ubicado.

f) Pasan 5 segundos desde que el camión ingresó a la balanza o desde que se detectó el último "lado no rebasado", en ambos casos sin que se haya detectado un "lado rebasado". En este caso se debe bajar la barrera.

g) Validar la tarjeta recibiendo su número.

h) Activar la balanza.

i) Imprimir el tique.

Ejercicio 8 Tomado en: 2010.06.04.

Un archivo consta de un nombre que lo identifica y de un contenido que consiste en una secuencia de caracteres. Existen operaciones para agregar un caracter al final de un archivo y borrar el último caracter. Un sistema de archivos consta de un conjunto de archivos. Debe ser posible cambiar el nombre de un archivo del sistema de archivos por otro que no exista en el sistema. Dando el nombre de un archivo, es posible agregar un caracter al final de aquel solo si el tamaño del archivo no supera cierta cota máxima permitida para el sistema de archivos. De igual forma debe ser posible eliminar el último caracter de un archivo dado.

Modele en Z utilizando promoción de operaciones los requerimientos anteriores. Designe los fenómenos de interés.

Ejercicio 9

- Una operación carga en la memoria de una computadora un programa que le es enviado desde otra computadora.
- Como el programa puede ser muy largo, puede ocurrir que sea enviado en más de un paquete de datos, lo que implica que la operación a especificar será invocada varias veces. Esto se denomina secuencia de comandos.
- La interfaz de la operación consta de:

  Entradas:
  La secuencia de bytes del programa (posiblemente incompleto, dada su longitud), la dirección de memoria a partir de la cual debe cargarse el programa y el número de secuencia de paquetes de datos (CSC). Todos estos elementos forman un paquete de datos.

  Salidas:
  Un código de error y, posiblemente, un número natural.
- El CSC inicial debe ser mayor que cero; la carga finaliza cuando el CSC que se recibe es cero. El CSC de cada paquete de datos que se recibe debe ser una unidad menor al anterior.

- La dirección inicial debe estar dentro del sector de memoria disponible para programas y la carga no puede desbordar dicho sector.
- Se debe controlar que un paquete de datos no pise un área de memoria modificada durante la misma secuencia de comandos.
- En caso de que todo vaya bien la operación debe retornar el mensaje "Comando recibido" y modificar la memoria.
- En caso de error se debe retornar "Error de carga" más un dato según se indica a continuación. Si el CSC de un paquete está fuera de secuencia debe retornar el CSC. Si la dirección inicial implica que se sobrescribirá un área de memoria modificada en la misma secuencia de paquetes o si no está dentro del sector adecuado, se debe retornar la dirección recibida. Si se desborda la memoria se debe retornar la longitud de la secuencia de bytes.

Especifique un esquema de estado y una operación Z que cumplan con los requerimientos anteriores.

Ejercicio 10 Un juego de computadora permite crear diferentes monstruos para luego enfrentarse a ellos. Para crear un monstruo se le debe seleccionar una cabeza, un cuerpo y las extremidades. El jugador puede probar tantas veces como quiera los diferentes tipos de partes disponibles hasta obtener un monstruo de su agrado (o de su desagrado). Cuando un monstruo está completo se puede guardar en un repositorio de monstruos ya creados, del cual se pueden ir seleccionando monstruos para enfrentarse.

Especifique en Z las diferentes etapas de la creación/selección de un monstruo y el repositorio de monstruos según se describe arriba.

Ejercicio 11 Se debe armar el "fixture" de partidos de una liga de equipos de fútbol. Los equipos inscriptos juegan todos contra todos una sola vez. Se juega solo los domingos. Los partidos se juegan siempre en cancha neutral. Se puede suponer que cada equipo posee una cancha propia. Cada equipo juega a lo sumo un partido por día. Cada equipo puede indicar en forma previa al comienzo del torneo a lo sumo 5 domingos en los cuales no puede intervenir.

Especifique en Z las operaciones (totales) que se listan a continuación, en relación a los requerimientos que se enuncian arriba.

a) Inscripción de un nuevo equipo.

b) Inclusión en el "fixture" de un nuevo partido entre dos equipos.

c) Eliminación de un equipo de la lista con la consiguiente eliminación de todos los partidos en los que participara.

Ejercicio 12 Un edificio contará con varios ascensores controlados desde un programa. No puede haber más de dos ascensores moviéndose al mismo tiempo. Cada ascensor puede ser llamado desde cada piso mediante un botón. Dentro de cada ascensor hay una botonera para dirigirlo hacia el piso al que se desee ir. En cada piso y en cada túnel hay un sensor que avisa del paso del ascensor por ese piso. Las puertas son manuales.

a) Especifique en Z las siguientes operaciones básicas referidas al problema de más abajo: detectar que se haya pulsado un botón dentro de un ascensor, ordenar a un ascensor moverse hacia abajo o arriba, detectar que el ascensor haya pasado por un piso, detener el ascensor.

b) Promover las operaciones anteriores a nivel de un edificio que cuente con p pisos y n ascensores.

Ejercicio 13 Una solicitud de póliza de seguro de una cierta compañía de seguros debe pasar por varias etapas antes de ser comunicada la decisión de la empresa al solicitante. La solicitud consta de varios datos que debe llenar el solicitante y de otros que son completados por las diferentes áreas de la empresa a medida que la solicitud es procesada. Las etapas son las siguientes:

a) La solicitud es recibida y se controla que los datos consignados por el solicitante sean coherentes. Si lo son se acepta la solicitud y pasa a la siguiente etapa; caso contrario la solicitud se rechaza.
Los datos son coherentes si se solicita un seguro para un auto y se consigna la patente o si se solicita un seguro para una casa y se consigna el domicilio.

b) En esta etapa las solicitudes rechazadas no sufren cambios y pasan a la siguiente etapa. Las solicitudes aceptadas son evaluadas y se le asigna un precio anual por el seguro solicitado o se las puede rechazar, en cuyo caso no se asigna un valor. En cualquier caso pasan a la siguiente etapa.

c) En esta etapa las solicitudes rechazadas en las etapas anteriores no sufren cambio y pasan a la siguiente etapa. Las solicitudes aceptadas en ambas etapas son analizadas por el departamento de legales y este les adjunta un contrato, en cuyo caso es aceptada de forma definitiva.

Modele en Z utilizando composición de operaciones el proceso anterior.

Ejercicio 14 Una medida judicial puede referirse a uno o más gravámenes. Cada medida judicial se identifica por el número de juez que la emite, las dos partes intervinientes (personas físicas o jurídicas) y la fecha de emisión de la medida. Se espera que los jueces o sus ayudantes puedan ingresar una medida o modificarla según se explica más abajo. Las medidas no pueden darse de baja.

Cada gravamen es de un cierto tipo (por ejemplo, embargo o inhibición) del cual dependen el código, el nombre y la caducidad del gravamen (es decir, una vez transcurrido el tiempo de caducidad desde la fecha de emisión de la medida que lo contiene, cesa el efecto del gravamen, lo que produce un cambio de estado en el gravamen). Además, el tipo del gravamen determina si este afecta o bien a una persona o bien a algunos o todos los inmuebles que posea una persona. Cada gravamen puede estar en uno de varios estados según se describe a continuación: activo, desde el momento en que se carga la medida en el sistema; cancelado, desde el momento en que la persona afectada toma alguna acción judicial que el juez interviniente entienda que justifica cancelarle el gravamen que pese sobre aquella o alguno de sus inmuebles; caduco, desde el momento en que, no habiendo sido cancelado, ha transcurrido el período de caducidad del mismo (si bien este cambio de estado no es iniciado por un usuario, puede suponerse que hay un componente externo que emite una señal cada vez que transcurre un día).

Modele en Z utilizando promoción de operaciones los requerimientos anteriores. Designe los fenómenos de interés.

Ejercicio 15 Para que una persona pueda utilizar un sistema debe completar un formulario web donde se le pide nombre completo y dirección de correo electrónico. Luego el sistema agrega a la persona una base temporal. Después alguien autoriza a la persona cargando en el sistema una contraseña para ese usuario. Una vez autorizado, el sistema le comunica a la persona la contraseña por correo electrónico y desde ese momento pasa a la base definitiva.

Utilice composición de operaciones para modelar en Z todo el proceso de alta de un usuario.

Ejercicio 16 Un documento tiene un nombre que lo identifica, un cuerpo de texto y una fecha de última modificación. Al modificar el documento se debe actualizar la fecha correspondiente. Además, puede buscarse una cadena de caracteres dada en el cuerpo de un documento comunicándole al usuario todos los números de línea donde se encuentra.

Modificar un documento significa cambiar la línea por otra o agregar o borrar una; las líneas están numeradas. Si el documento tiene las líneas número i y j con $i > j + 1$, entonces debe tener todas las líneas numeradas entre i y j. Si el usuario no las agregó, el sistema debe poner líneas en blanco. Cualquier modificación se lleva a cabo dando el número de línea y la cadena de caracteres para esa línea.

Un sistema de documentación consiste en un conjunto de documentos y un conjunto de usuarios que pueden modificar o buscar en los documentos. Si un usuario registrado quiere modificar un documento debe dar su nombre, número de línea y cadena de reemplazo (si es necesario). Si un usuario quiere buscar una cadena en la base de documentación, el sistema

le responderá dándole el nombre de cada documento donde la cadena aparece junto a todos los números de línea donde se encuentra la cadena.

Modele en Z utilizando promoción de operaciones los requerimientos anteriores.

2.2 Práctica de Statecharts y CSP

Ejercicio 1 Tomado en: 2009.08.07(r2).

Un sistema debe controlar un aparato para efectuar electroencefalogramas simples. El análisis consiste en estudiar el voltaje que emiten 10 electrodos que permiten conocer la actividad bioeléctrica cerebral (cada uno comunica un valor al sistema). Es necesario tomar 5 muestras por segundo, espaciadas uniformemente. En cada una de las 5 muestras se lee el valor de los 10 electrodos. Notar que si el cerebro del paciente no presenta actividad en las cercanías de un electrodo, este no emitirá señal alguna. Por lo tanto, el sistema no puede esperar indefinidamente por la señal del electrodo. Finalmente, el sistema debe enviar secuencialmente a una impresora el valor obtenido en cada electrodo (que haya retornado uno o nada en caso de que no se haya registrado ninguno).

a) Modele en Statecharts la especificación y el conocimiento del dominio del problema anterior.

b) Escriba las designaciones y modele en CSP el conocimiento de dominio y la especificación de los requerimientos enunciados en el problema. Para las cuestiones temporales puede asumir la existencia de un temporizador como el que se mostró en clase.

Ejercicio 2 Tomado en: 2010.06.15(p2).

Un sistema distribuido consta de N computadoras cada una de las cuales ejecuta el mismo programa; estas computadoras pueden estar encendidas o apagadas. Este programa recibe dos eventos desde el exterior (en cada computadora), a y b. Cuando el programa en una computadora recibe a, debe comunicarles a todas las otras computadoras, por medio de un evento interno, que lo ha recibido. Luego de comunicar esta situación esa computadora debe esperar a que todas le respondan. Cada computadora encendida responde con un evento diferente pero solo si está en su estado inicial. Como puede haber computadoras apagadas o que estén haciendo otra cosa, el programa esperará un tiempo B cada respuesta. Cuando recibe todas las respuestas o han transcurrido B unidades de tiempo, debe emitir el evento c y volver al estado inicial. Si el programa recibe el evento b debe esperar un tiempo t hasta poder recibir el evento a a menos que se dé otro evento b luego de T unidades de tiempo, en cuyo caso debe reiniciar el tiempo de espera.

a) Designe los términos más importantes del enunciado.

b) Describa en Statecharts la especificación del programa mencionado en el problema.

c) Describa en CSP la especificación del programa mencionado en el problema.

Ejercicio 3 Tomado en: 2015.06.10(p2).

Un sistema de alarma hogareño simple consta de un teclado (contiene los 10 dígitos y las dos primeras letras del abecedario), dos sensores de movimiento y dos magnéticos (para aberturas) y una bocina. Todo el sistema será controlado por un programa.

El sistema puede estar armado o no. Armado significa que si se detecta una intrusión debe sonar la bocina que se teclee el código de seguridad. Si no está armado no se responderá a las intrusiones. Una intrusión se detecta cuando alguno de los sensores envía una señal. Si está armado, al teclear el código de seguridad, el sistema se desarma; si está desarmado, al teclear el código de seguridad, se arma.

El código de seguridad viene de fábrica y consta de 2 dígitos.

El sistema agrupa un sensor de movimiento y uno magnético en el sector A y a los sensores restantes en el sector B. Cada sector se puede armar independientemente del otro. Para hacerlo se teclea su letra y el código de seguridad. Si un sector no está armado no se responderá a las intrusiones en ese sector. Si solo se teclea el código de seguridad se arman todos los sectores; si algún sector está armado y se teclea el código de seguridad, se desarman todos los sectores.

a) Designe los fenómenos de interés, escribir la tabla de control y modelar en Statecharts el conocimiento de dominio y la especificación de los requerimientos anteriores, abstrayendo convenientemente el teclado y el código de seguridad.

b) Utilizando las mismas designaciones modele en CSP el conocimiento de dominio y la especificación de los requerimientos enunciados en el problema.

Ejercicio 4 Tomado en: 2015.06.17(r2).

Se cuenta con un proceso S que envía mensajes, un medio E que los transmite y un proceso R que los recibe. E puede almacenar solo un mensaje a la vez, puede corromper a lo sumo uno de cada tres mensajes consecutivos y, por simplicidad, se supone que cada mensaje es un 0 o un 1; E está fuera del control del sistema, es decir es parte del entorno. Se espera que S y R colaboren para evitar la corrupción producida por E. Para ello utilizarán un mecanismo simple: S enviará cada mensaje por triplicado y R decidirá cuál es el válido de acuerdo a una elección por mayoría simple.

Especifique en Statecharts los procesos S, E y R descriptos arriba.

Ejercicio 5 Tomado en: 2015.06.17(r2).

Se requiere el software de control para un robot industrial con las siguientes características. El robot puede moverse hacia la izquierda o derecha una cierta distancia y consta de un rociador y termómetro. El robot está ubicado al borde de una cinta transportadora que acarrea piezas metálicas. El requerimiento básico es que si el sensor detecta que la temperatura de la pieza que está en su cercanía es superior a *maxtemp*, el robot debe abrir el rociador y debe moverse en el sentido de la cinta hasta que la temperatura de la pieza ea inferior a *maxtemp*. Si el robot llega hasta su límite de desplazamiento debe cerrar el rociador. En cualquier caso debe retornar a la posición inicial.

Tener en cuenta lo siguiente:

- El termómetro envía una señal con la temperatura medida.
- El motor del robot envía una señal indicando que se ha alcanzado el extremo derecho o izquierdo.
- El software de control puede encender el motor en uno u otro sentido y detenerlo.
- No se puede poner en funcionamiento el motor en el mismo paso en que se recibe una medición de temperatura.

a) Describa en Statecharts la especificación y el conocimiento del dominio del problema que se enuncia arriba. Abstraiga convenientemente las diferentes temperaturas que puede comunicar el termómetro.

b) Describa en CSP la especificación y el conocimiento del dominio de los requerimientos enunciados en el problema.

Ejercicio 6 Un productor produce de a un mensaje a la vez a razón de 3 segundos cada uno. Existen otros dos productores que se comportan de la misma forma pero trabajan con tiempos de 1 y 4 segundos, respectivamente. Todos los productores tienen un error en sus relojes de $\pm\delta$ segundos con $0 < \delta < 1$.

Los mensajes enviados por los tres productores son recibidos por un componente que puede recibir de a un mensaje a la vez. El componente cuenta con un "buffer" con capacidad para *MB* mensajes. Poner o sacar un mensaje del "buffer" le toma 0,1 segundos.

Por otro lado, hay dos consumidores de mensajes que pueden consumir de a un mensaje a la vez cada uno, a razón de 2 segundos por mensaje. El componente comunica los mensajes según una política FIFO.

Tener en cuenta que los productores y consumidores son las entidades activas.

a) Escriba las designaciones y la tabla de control y visibilidad del problema anterior.

b) Modele en Statecharts el conocimiento del dominio (productores y consumidores) y la especificación del componente de almacenamiento.

c) Modele en CSP el conocimiento del dominio (productores y consumidores) y la especificación del componente de almacenamiento.

d) Explique claramente (mostrando un ejemplo si es necesario) si el modelo semántico "broadcast" de Statecharts hace que el sistema completo se comporte de forma diferente a como lo hace la especificación CSP, que responde a un modelo semántico sincrónico.

Ejercicio 7 Una empresa posee una balanza para pesar camiones cargados con materia prima. El camión debe ubicarse más o menos sobre el centro de la balanza para que la pesada sea correcta. Con este fin la empresa instaló cuatro sensores en los vértices de un rectángulo imaginario de forma tal que cuando detectan que el camión está dentro de ese rectángulo, se debe bajar una barrera detrás del camión. Si el camión rebasa alguno de los laterales del rectángulo se enciende una (de dos) luz ubicada delante del camión que indica qué lado está rebasado.

Una vez que el camión está correctamente ubicado y se bajaron las barreras, el chofer debe deslizar una tarjeta magnética que lo identifica. Si la tarjeta es válida, se activa la balanza. Cuando el pesaje finaliza, se debe imprimir un tique con los datos del conductor y el peso. Luego se levantan las barreras.

Modele en Statecharts el conocimiento de dominio y la especificación de los requerimientos anteriores. Abstraiga adecuadamente la validación de la tarjeta y la impresión del tique con sus datos.

Ejercicio 8 Tomado en: 2010.06.04.

Una habitación posee varias puertas de ingreso/egreso. En cada puerta se han instalado dos sensores ópticos uno al lado del otro (cada uno es semejante a los que se instalan en las puertas de los ascensores para evitar que se cierren si hay alguien entrando o saliendo). Estos sensores envían una señal cada vez que algo interrumpe el haz de luz que se establece entre cada extremo. Como hay dos por puerta es posible determinar si una persona ingresa a la habitación o sale de ella. Notar que esta inteligencia *no* es parte de la funcionalidad de los sensores.

La habitación cuenta con un sistema de ventilación electrónico. Es posible aumentar o disminuir discretamente la cantidad de aire que el sistema hace circular.

Se requiere un programa que aumente/disminuya paulatinamente la circulación de aire desde cero hasta el máximo posible, cada vez que ingresan/egresan tres personas a la sala. Es decir la potencia de circulación debe ser la misma, por ejemplo, si hay 3, 4 o 5 personas en la sala. Además, si la habitación permanece vacía por más de 12 horas se debe encender el sistema de ventilación por 1 hora.

Modele en Statecharts el conocimiento de dominio y la especificación de los siguientes requerimientos.

Ejercicio 9 Un edificio contará con varios ascensores controlados desde un programa. No puede haber más de dos ascensores moviéndose al mismo tiempo. Las puertas de los ascensores solo deben abrirse cuando estos están detenidos en un piso. Cada ascensor puede ser llamado desde cada piso mediante un botón. Dentro de cada ascensor hay una botonera para dirigirlo hacia el piso al que se desee ir. En cada piso y en cada túnel hay un sensor que avisa el paso del ascensor por ese piso. Las puertas son manuales.

a) Describa en Statecharts el conocimiento de dominio y la especificación del sistema de ascensores que se enuncia arriba, para un edificio de p pisos y n ascensores. No tener en cuenta los botones de los pisos. Designe los fenómenos de interés.

b) Modele en CSP el conocimiento de dominio y la especificación del problema.

Ejercicio 10 Una máquina expendedora de bebidas funciona de la siguiente manera. La máquina está inactiva hasta que se libera una traba de seguridad. Los clientes pueden seleccionar una bebida pulsando el botón correspondiente. Hay cinco bebidas diferentes. Cuando la máquina no está inactiva se pueden insertar monedas mientras se muestra en una pequeña pantalla el total hasta el momento. (Se aceptan monedas de 25 y 50 centavos y de 1 peso. La máquina detecta el valor de la moneda.) Si se selecciona una bebida y el total de dinero es igual al precio del producto, entonces la máquina entrega la bebida correspondiente. Cada bebida

puede tener un precio diferente. Si el total es mayor que el precio del producto, se entrega la bebida y la pantalla se actualiza poniendo como nuevo total la diferencia. La máquina no puede ponerse en modo inactivo si el total mostrado no es cero. El usuario puede pulsar un botón para que la máquina le retorne el total mostrado en pantalla, en cuyo caso el total mostrado se pone a cero.

a) Modele en Statecharts el conocimiento de dominio y la especificación de los requerimientos anteriores. Aunque no es necesario mostrar el total de dinero ingresado por el usuario, el modelo debe contemplar una abstracción adecuada. También abstraiga adecuadamente el tema de los precios de las bebidas.

b) Modele en CSP el conocimiento de dominio y la especificación de los requerimientos enunciados en el problema.

Ejercicio 11 Se trata de mantener cierta temperatura dentro de un automóvil. La temperatura solo puede ser regulada ajustando el funcionamiento del acondicionador de aire (AA); el AA solo se apaga o se prende, no es posible hacer que el aire salga a mayor o menor temperatura. Entonces, cuanto más funciona el AA más se enfría el habitáculo. El conductor puede seleccionar la temperatura que desee mantener por medio de una perilla giratoria. La temperatura deseada puede estar entre 18 y 30 °C; si la temperatura deseada es de 18 °C, el AA no se detiene nunca. El sistema estará en funcionamiento solo si el conductor ha pulsado un botón de activación; el sistema sale de funcionamiento cuando el botón se vuelve a pulsar o el motor se detiene. Existen dos termómetros dentro del habitáculo (uno en la parte delantera y el otro en la parte trasera) que, una vez encendido el sistema, envían señales de incremento o decremento de la temperatura sensada. Los termómetros inicialmente se encuentran a 18 °C; los termómetros no envían señales si la temperatura excede su rango de funcionamiento.

Designe los fenómenos de interés, armar la tabla de visibilidad y control y modele en CSP el conocimiento de dominio y la especificación de los requerimientos anteriores.

Ejercicio 12 Un software controlará el funcionamiento de un minicomponente musical. El equipo consta de reproductor de CD y radio AM/FM. El panel de control cuenta con la interfaz de usuario habitual: encendido/apagado, “play”, “prev”, “next”, “stop”, “pause”, “eject” y selector CD-radio. Los botones como “play”, “prev”, etc. sirven para todos los componentes dependiendo del selector (por ejemplo, “prev” sirve para mover el sintonizador de la radio hacia las estaciones más bajas). Si el CD está seleccionado y se pulsa de forma continua por 2 segundos o más cualquiera de los botones “prev” o “next”, entonces se debe avanzar dentro de la misma canción y no saltar de canción. En la misma situación si se pulsan dos veces con diferencia de menos de 1 décima de segundo, entonces se pasa a la última o primera pista. Además, hay un sensor que detecta la presencia de un CD. Si se pulsa el botón de apagado mientras el CD está en funcionamiento primero se debe apagar el CD y luego el equipo. El minicomponente cuenta con una memoria que permite que al re-encenderse se active el último componente usado, en particular si es la radio se encenderá AM o FM.

Las órdenes recibidas desde los botones, si el CD está activado, se deben traducir en órdenes al motor (giro normal o giro rápido hacia atrás o adelante, parar, arrancar, expulsar) o el lector láser (encender, apagar, leer).

Las órdenes recibidas desde los botones, si la radio está activada, se deben traducir en órdenes a un sintonizador (aumentar/disminuir frecuencia y cambiar rango de frecuencia).

a) Liste las designaciones y confeccione la tabla de control y visibilidad para los fenómenos de interés de los requerimientos anteriores.

b) Describa en CSP un modelo para el conocimiento de dominio del sistema de control.

c) Describa en CSP un modelo para el sistema de control.

d) Describa en Statecharts un modelo para el conocimiento de dominio del sistema de control.

e) Describa en Statecharts un modelo para el sistema de control.

Ejercicio 13 Un proceso, B, debe almacenar elementos en dos *buffers* de la misma capacidad finita y conocida, N. Por otro lado, existen procesos (llamados productores) que envían datos a B para que este los almacene en los /emphbuffers, y existen procesos (llamados consumidores) que le piden a B los datos que tiene almacenados (lo que hace que los *buffers* se vayan vaciando).

Cuando un consumidor quiere un dato que B tiene, el proceso le debe indicar el *buffer* del cual lo quiere; en cambio los productores no pueden seleccionar el *buffer*.

Obviamente B no puede poner elementos en un *buffer* lleno y no puede sacar elementos de un *buffer* vacío. Lo que sí debe hacer B es balancear el uso de los *buffers*: cuando almacena datos lo debe hacer en el *buffer* más vacío y si un *buffer* se está vaciando más rápido que el otro, debe pasar elementos del último al primero.

Describa en CSP la especificación y el conocimiento de dominio de los requerimientos anteriores. Designe los términos básicos de su modelo.

Ejercicio 14 Una cinta transportadora acarrea ítems de dos tipos, A y B, que luego son capturados por las máquinas M_A y M_B, respectivamente. Un software controla la cinta y ambas máquinas.

El software debe encender cada una de las máquinas M_A y M_B, mediante los eventos *turnona* y *turnonb*, respectivamente. Cada máquina necesita r unidades de tiempo para estar operativa.

Cada vez que en la cinta se detecta un ítem de un tipo, el software debe emitir el evento *takea* o *takeb* para que la máquina correspondiente capture el ítem en cuestión, a menos que a continuación se indique otra cosa.

Si entre la aparición de dos o más ítems transcurren menos de n unidades de tiempo, no se debe emitir la orden de captura correspondiente a los ítems posteriores al primero; se deberá emitir las órdenes de captura cuando aparezca un ítem n unidades de tiempo después del anterior.

Si transcurren más de t unidades de tiempo entre dos ítems, se debe apagar todo el sistema mediante el evento *abort*.

Especifique en Statecharts el conocimiento de dominio y la especificación de los requerimientos que se enuncian arriba.

Ejercicio 15 **Protocolo CSMA/CD**. Para transmitir datos entre terminales de trabajo conectadas en red se debe hacer uso de algún protocolo. En algunas redes *broadcast* con un único bus la clave está en cómo asignar el uso de este cuando varias terminales compiten por él.

Uno de los protocolos que resuelven esta cuestión es el *Carrier Sense, Multiple Access with Collision Detection*, o simplemente CSMA/CD. Una breve descripción del funcionamiento de este protocolo es la siguiente.

Una terminal transmite al sistema (es decir a la implementación del protocolo) un mensaje que debe ser enviado por la red. El sistema, si el bus está disponible (esto es, no hay otra terminal transmitiendo), comienza a enviar su mensaje. Sin embargo, si detecta que el bus está ocupado, espera un tiempo aleatorio y vuelve a intentar transmitir el mensaje. Esto lo hará tantas veces como sea necesario hasta que pueda empezar a transmitir.

Aun tomando estas precauciones puede ocurrir que dos terminales usen el bus al mismo tiempo, lo que da lugar a una *colisión*. Cuando una colisión ocurre el bus comunica esta situación a todas las terminales. Esto implica que todas las terminales abortan inmediatamente las transmisiones y, nuevamente, eperan un tiempo aleatorio para empezar a transmitir de nuevo.

Los mensajes que colisionan se pierden. Una vez que el mensaje ha sido transmitido, la terminal que inició la transmisión es notificada.

Se supone que todos los mensajes (sin importar tamaño) demoran exactamente λ unidades de tiempo en ser transmitidos.

En resumen, en cada terminal corre una implementación del protocolo y todas las terminales comparten el bus. La interfaz que provee el bus consta de: determinar si el bus está libre o no, enviar un mensaje, comunicar que un mensaje se transmitió, comunicar a todas las terminales que hay colisión.

Modele la especificación del protocolo CSMA/CD que se describe arriba en:

a) Statecharts

b) CSP

Ejercicio 16 Se trata del software que controla un teléfono celular muy simple. El teléfono cuenta con un teclado numérico, una tecla "enviar" y otra "cortar". Además posee una bocina que puede ser encendida o apagada y una pantalla que puede mostrar una cadena de dígitos; existe una operación que borra el contenido de la pantalla. El teléfono se enciende si se pusa la tecla "cortar" durante más de 2 segundos. Una vez encendido el teléfono espera que el usuario teclee un número o que llegue una llamada.

Los números a los cuales se puede llamar poseen una cantidad de dígitos variable. Cada vez el usuario pulsa un dígito el número correspondiente debe mostrarse en la pantalla. Una vez que el usuario pulsa "enviar" el sistema debe esperar a que se pulse "cortar" (no se modela la comunicación en sí). Mientras tanto no se pueden recibir llamadas, es decir se pierden. Cuando se pulsa "cortar" se debe borrar la pantalla.

Si el teléfono puede recibir una llamada y esto ocurre, entonces el sistema debe hacer sonar la bocina de manera intermitente hasta que el usuario pulse "enviar" o "cortar". La intermitencia es: 1 segundo de ruido seguido de un segundo de silencio. Si el usuario pulsa "cortar" el teléfono podrá recibir o hacer nuevas llamadas. Si el usuario pulsa "enviar" no se podrán hacer ni recibir nuevas llamadas hasta que pulse "cortar". Si el número que se recibe coincide con uno que se mantiene en la agenda del teléfono, entonces se debe mostrar en la pantalla el nombre que le corresponda.

Escriba las designaciones y modele en CSP el conocimiento de dominio y la especificación de los requerimientos que se enuncian arriba.

Nota: puede utilizar la especificación del temporizador visto en clase.

Ejercicio 17 Tomado en: 2016.07.01 (r2).

Designar los fenómenos de interés de los siguientes requerimientos, construir la tabla de control y visibilidad y modelar en CSP el conocimiento de dominio (incluir el mecanismo de comunicación entre los sensores) y la especificación. Considerar que la máquina a construir es el software que controla a D.

El dispositivo D debe recibir datos de N sensores conectados en serie, y con esos datos realizar diversas estadísticas. El mecanismo debe ser el siguiente: D debe pedir al primer sensor que le envíe el dato que tenga disponible, este sensor, a su vez, luego de enviar el dato le "avisa" al siguiente sensor que es su turno, y así sucesivamente hasta el sensor N. D espera un cierto tiempo a que arribe el siguiente dato, tras el cual considera que no existen más sensores (el hecho de que estén en serie y que se "avisen" entre ellos hace que N sea transparente para D). Cuando D termina de recibir los valores debe enviarle el promedio de los valores recibidos a otro dispositivo, y comenzar el ciclo nuevamente en cuto caso se reinicia el arreglo de sensores de manera tal que el primero es el que posiblemente enviará el primer valor en el nuevo ciclo. En caso de no haber recibido ningún valor luego de efectuar el primer pedido (porque N es cero o porque el primer sensor se demoró más de lo esperado), debe enviarle al otro dispositivo un mensaje de error.

2.3 Práctica de TLA$^+$

Ejercicio 1 Tomado en: 2007.03.09.

Un brazo mecánico controlado digitalmente demora T_{ir} unidades de tiempo en ir de un extremo a otro pero el sistema debe detenerlo porque de lo contrario se dañaría el motor. Si durante su movimiento un operario pulsa un botón el sistema debe interrumpir el movimiento; si se vuelve a pulsar se lo debe reanudar, pero si no se pulsa el botón por segunda vez pasadas T_{det} unidades de tiempo desde la detención se lo debe mover en sentido contrario por el mismo tiempo que se lo movió hasta la detención. Cuando el brazo llega a cualquiera de los extremos debe permanecer allí a lo sumo T_{ext} unidades de tiempo luego de lo cual debe regresar (esto mismo vale para el estado inicial del sistema).

Especifique en TLA$^+$el conocimiento de dominio y la especificación de los requerimientos anteriores. Escriba las designaciones correspondientes.

Ejercicio 2 Tomado en: 2008.08.08, 2010.07.30.

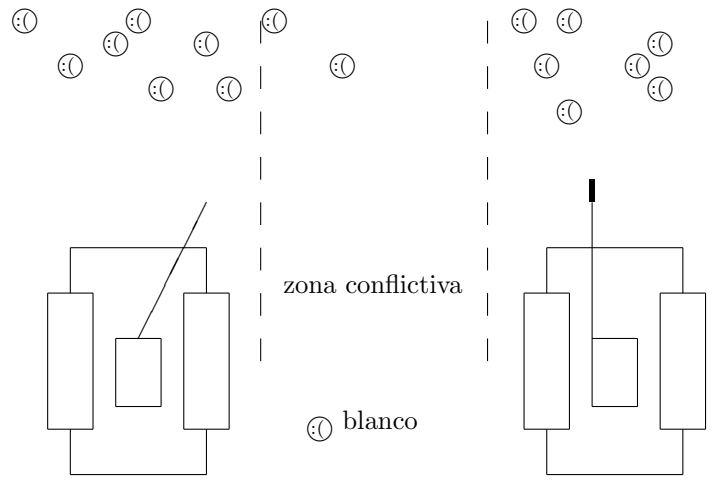
Se trata básicamente de que dos tanques en la misma zona de combate actúen coordinadamente en la selección y destrucción de blancos fijos. Cada tanque cuenta con un cañón, un sensor para detección de blancos por temperatura y una computadora de abordo. El programa de abordo, que consta de dos módulos funcionales A y B, se encarga de seleccionar blancos en base a los datos proporcionados por el sensor y a la cooperación con la computadora del otro tanque.

El módulo A comunica al módulo B la posición del blanco a partir de los datos crudos provistos por el sensor; la posición es el identificador universal para un blanco en la zona de combate. El módulo B se encarga de seleccionar los blancos en base a la posición y de mostrar a la tripulación los próximos blancos a destruir en el orden adecuado. La tripulación, leyendo estos datos, manejará el tanque y el cañón (es decir el sistema *no* conduce el tanque *ni* realiza los disparos).

Los dos tanques son jerárquicamente iguales; ninguno tiene prioridad sobre el otro para determinar la selección de blancos; ambos poseen una copia del módulo B en su sistema. En cada comunicación podrán determinar quién seleccionará un blanco en la zona conflictiva (ver la figura para mayores detalles).

La comunicación consta de dos mensajes. El iniciador de la comunicación le avisa al receptor que *no* puede encargarse del blanco en cuestión; el receptor puede aceptar el blanco o rechazarlo por tener más de $MAXTARGET$ blancos seleccionados. Si el receptor acepta envía la respuesta correspondiente y luego lo ubica en su lista; si el receptor no acepta el blanco ambos lo reportan a la tripulación.

La funcionalidad se debe dividir en las siguientes acciones: recepción del blanco, clasificación del blanco (conflictivo o no), incorporación del blanco a la lista, actualización de la lista de blancos en pantalla, reporte de blanco no aceptado. La comunicación con el otro tanque es asíncrona.



Especifique en TLA$^+$el módulo funcional B del sistema que se describe arriba.

Ejercicio 3 Tomado en: 2010.02.12, 2015.07.10.

Cuando un cliente web solicita un archivo HTML a un servidor web la comunicación es asíncrona, es decir: un navegador solicita un archivo HTML, se corta la conexión y finalmente el servidor se comunica con el cliente, cuando puede satisfacer el pedido, para enviarle el archivo solicitado o el error apropiado.

También es asíncrona la comunicación cuando el cliente se autentica ante servidor: se envían los datos, se corta la conexión, el servidor verifica los datos, si son correctos registra al cliente y retorna el error apropiado.

Los archivos HTML que almacena el servidor pueden estar restringidos a ciertos clientes. Si este es el caso, el cliente que solicita uno de estos archivos debe estar autenticado y debe ser uno de los clientes autorizados a ver el archivo.

Por el contrario, cuando un cliente web desea enviarle un archivo al servidor la comunicación es asíncrona, aunque el servidor puede atender varios pedidos a la vez.

Un servidor web puede atender hasta M clientes simultáneamente.

Especifique en TLA$^+$el servidor mencionado en el problema anterior.

Ejercicio 4 Tomado en: 2010.05.28.

Protocolo CSMA/CD. Para transmitir datos entre terminales de trabajo conectadas en red se debe hacer uso de algún protocolo. En algunas redes *broadcast* con un único bus la clave está en cómo asignar el uso de este cuando varias terminales compiten por él.

Uno de los protocolos que resuelven esta cuestión es el *Carrier Sense, Multiple Access with Collision Detection*, o simplemente CSMA/CD. Una breve descripción del funcionamiento de este protocolo es la siguiente.

Una terminal transmite al sistema (es decir a la implementación del protocolo) un mensaje que debe ser enviado por la red. El sistema, si el bus está disponible (esto es, no hay otra terminal transmitiendo), comienza a enviar su mensaje. Sin embargo, si detecta que el bus está ocupado, espera un tiempo aleatorio y vuelve a intentar transmitir el mensaje. Esto lo hará tantas veces como sea necesario hasta que pueda empezar a transmitir.

Aun tomando estas precauciones puede ocurrir que dos terminales usen el bus al mismo tiempo, lo que da lugar a una *colisión*. Cuando una colisión ocurre el bus comunica esta situación a todas las terminales. Esto implica que todas las terminales abortan inmediatamente las transmisiones y, nuevamente, eperan un tiempo aleatorio para empezar a transmitir de nuevo.

Los mensajes que colisionan se pierden. Una vez que el mensaje ha sido transmitido, la terminal que inició la transmisión es notificada.

Se supone que todos los mensajes (sin importar tamaño) demoran exactamente λ unidades de tiempo en ser transmitidos.

En resumen, en cada terminal corre una implementación del protocolo y todas las terminales comparten el bus. La interfaz que provee el bus consta de: determinar si el bus está libre o no, enviar un mensaje, comunicar que un mensaje se transmitió, comunicar a todas las terminales que hay colisión.

Describa en TLA la especificación del protocolo CSMA/CD que se describe arriba.

Ejercicio 5 Tomado en: 2010.08.06.

Varios procesos compiten por la utilización de un cierto recurso. Cada proceso, en el momento en que desea utilizar el recurso, pregunta al sistema si el recurso está libre o no. Si lo está el sistema lo reserva para ese proceso, le comunica esta decisión y luego el proceso puede ejecutar una de tres operaciones diferentes sobre el recurso.

Si no lo está, el sistema le comunica al proceso esta situación; el proceso espera una unidad de tiempo y vuelve a preguntar, si la respuesta es la misma, espera dos unidades de tiempo y reitera la pregunta; esto se repite hasta que el recurso esté libre.

Si un proceso recibe el OK para utilizar el recurso pero pasan más de T unidades de tiempo sin que lo use, el sistema lo libera para que lo utilice otro proceso. Luego de que un proceso

utiliza el recurso, no podrá volver a utilizarlo hasta que lo utilice otro proceso diferente, a menos que el primer proceso sea el único en el sistema.

Describa un modelo TLA para los requerimientos enunciados.

Ejercicio 6 Tomado en: 2010.12.10.

Un sistema debe controlar un aparato para efectuar electroencefalogramas simples. El análisis consiste en estudiar el voltaje que tiene un conjunto de 10 electrodos que permiten conocer la actividad bioeléctrica cerebral (cada uno comunica un valor al sistema). Es necesario tomar 5 muestras por segundo, espaciadas uniformemente. En cada una de las 5 muestras se lee el valor de los 10 electrodos. Notar que si el cerebro del paciente no presenta actividad en las cercanías de un electrodo, este no emitirá señal alguna. Por lo tanto, el sistema no puede esperar indefinidamente por la señal del electrodo. Finalmente, el sistema debe enviar secuencialmente a una impresora el valor obtenido en cada electrodo (que haya retornado uno o nada en caso de que no se haya registrado ninguno).

Escriba las designaciones y modele en TLA el conocimiento de dominio y la especificación de los requerimientos que se enuncian arriba. Se premiará el uso correcto del carácter no tipado de TLA. Para las cuestiones temporales puede asumir la existencia de un temporizador como el que se mostró en clase.

Ejercicio 7 Tomado en: 2011.02.25.

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 de 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 a 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.

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

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

Ejercicio 8 Tomado en: 2011.09.09.

Un proceso, B, debe almacenar elementos en dos *buffers* de la misma capacidad finita y conocida, N. Por otro lado, existen procesos (llamados productores) que envían datos a B para que este los almacene en los /emphbuffers, y existen procesos (llamados consumidores) que le piden a B los datos que tiene almacenados (lo que hace que los *buffers* se vayan vaciando).

Cuando un consumidor quiere un dato que B tiene, el proceso le debe indicar el *buffer* del cual lo quiere; en cambio los productores no pueden seleccionar el *buffer*.

Obviamente B no puede poner elementos en un *buffer* lleno y no puede sacar elementos de un *buffer* vacío. Lo que sí debe hacer B es balancear el uso de los *buffers*: cuando almacena datos lo debe hacer en el *buffer* más vacío y si un *buffer* se está vaciando más rápido que el otro, debe pasar elementos del último al primero.

Describa en TLA$^+$la especificación y el conocimiento de dominio de los requerimientos anteriores. Designe los términos básicos de su modelo. Debe poner especial cuidado en determinar cuáles de las operaciones que implícitamente se describen son internas y cuáles son externas.

Ejercicio 9 Tomado en: 2012.02.10.

Una caldera consiste en un tanque de agua y un quemador. El quemador calienta el agua y el agua caliente viaja por tuberías para calefaccionar un edificio. El tanque se llena con agua de la red pero fundamentalmente se retroalimenta con el agua que circula por las tuberías del edificio (es decir, el agua sale muy caliente del tanque, viaja por las tuberías, pierde calor en el trayecto y vuelve a ingresar al tanque). Un termómetro mide la temperatura del agua en el tanque y un barómetro la presión.

El quemador, las válvulas de entrada de agua de la red y las de entrada o salida de las cañerías pueden ser controladas digitalmente. El termómetro y el barómetro son sensores electrónicos activos.

El sistema debe controlar que la presión y la temperatura del agua en el tanque se mantengan dentro de ciertos parámetros. Cuando la temperatura sube (baja) se debe bajar (subir) el quemador; cuando la presión aumenta se debe liberar el agua a las cañerías, pero si disminuye se debe suministrar agua de la red.

Liste las designaciones y confeccione la tabla de control y visibilidad para los fenómenos de interés del problema anterior. Luego modele en TLA el conocimiento de dominio y la especificación del software de control que se mencionan.

Ejercicio 10 Tomado en: 2017.08.4.

(Segun lo recuerdo)

Un celular consiste de una bocina que puede ser prendida o apagada, un display y un teclado numerico con dos botones de "cortar" y "enviar".

El celular se enciende luego de presionar el boton de "cortar" dos segundos. Mientras no se este realizando ninguna llamada se puede presionar numeros del teclado y estos apareceran en el display. Si se presiona el boton de "enviar" se llama al numero marcado, si se presiona "cortar" se borra el display.

Dada una llamada entrante sonara la bocina intermitentemente de la siguiente forma: 1 segundo de sonido seguido de 1 sonido de silencio. Si se presiona el boton de "cortar" se podra volver a marcar numeros, si se presiona "enviar" se entra en la llamada. No se modelaran las llamadas.

Por ultimo, si el numero de una llamada entrante se encuentra en la agenda se mostrara el nombre del mismo.

2.4 Práctica de Z/EVES

Ejercicio 1 Tomado en: 2007.03.09, 2010.05.28, 2011.09.09.

Considere la siguiente especificación.

Set

$$A : \mathbb{P}\,\mathbb{Z}$$

Add

$$\Delta Set$$
$$a? : \mathbb{Z}$$
$$A' = A \cup \{a?\}$$

Remove

$$\Delta Set$$
$$a? : \mathbb{Z}$$
$$A' = A \setminus \{a?\}$$

$$Total \mathrel{\widehat=} Add \fatsemi Remove$$

a) Encuentre la mínima condición bajo la cual *Total* deja a A sin cambios.

b) Pruebe la conjetura del punto anterior utilizando Z/EVES.

c) Reformule la especificación de forma tal que $Total \Rightarrow \Xi Set$.

d) Pruebe usando Z/EVES que la especificación del punto anterior efectivamente verifica la propiedad en cuestión.

Ejercicio 2 Tomado en: 2008.08.08.

Un banco ha adquirido otro banco. Por cuestiones del Ministerio de Trabajo se deben mantener dos nóminas de empleados: una en la que están únicamente los empleados del banco comprador y la otra donde están todos los empleados (los de ambos bancos). La nómina relaciona el empleado con su sueldo. En Z se ha modelado de la siguiente forma:

$$[DNI]$$
$$DINERO == \mathbb{N}$$

NóminasBanco

$$comprador, todos : DNI \pfun DINERO$$

Existe una operación para modificar el sueldo de un empleado del banco comprador.

ModificarSueldoCompradorOk

$$\Delta NóminasBanco$$
$$d? : DNI$$
$$m? : \mathbb{Z}$$
$$d? \in \mathrm{dom}\ comprador$$
$$comprador' = comprador \oplus \{d? \mapsto max\{comprador\ d? + m?, 0\}\}$$
$$todos' = todos \oplus \{d? \mapsto max\{comprador\ d? + m?, 0\}\}$$

La segunda poscondición garantiza que ambas nóminas se mantengan consistentes; es decir, que cualquier empleado visto como miembro del banco comprador tenga el mismo sueldo que visto como miembro del banco actual.

a) Defina el invariante de estados que captura lo dicho en el párrafo anterior.

b) Pruebe usando Z/EVES que *ModificarSueldoCompradorOk* preserva el invariante. Ayuda: es posible que necesite usar explícitamente el teorema *inPower* del *toolkit* matemático.

c) Intente descargar la obligación de prueba que le plantea Z/EVES en el esquema *ModificarSueldoCompradorOk*. Si no puede descargarla, explique las razones que lo impiden. Finalmente, sugiera la modificación a la especificación necesaria para poder hacerlo.

Ejercicio 3 Tomado en: 2010.02.12.

La siguiente especificación describe la operación que borra un símbolo de una tabla que relaciona símbolos con valores.

$[SYM, VAL]$
$REPORT ::= ok \mid symbolNotPresent$

ST

$$st : SYM \pfun VAL$$

DeleteOk

$$\Delta ST$$
$$s? : SYM$$
$$rep! : REPORT$$

$$s? \in \dom st$$
$$st' = \{s?\} \rhd st$$
$$rep! = ok$$

$DeleteE \mathrel{\widehat=} LookUpE$
$Delete \mathrel{\widehat=} DeleteOk \lor DeleteE$

Una herramienta de "testing" que extrae casos de prueba analizando la especificación sugiere que la implementación se testee en las siguientes condiciones:

- que la tabla de símbolos no esté vacía;
- que el dominio de la tabla esté estrictamente incluido en el conjunto formado por el símbolo que se va a eliminar.

Asumiendo el siguiente teorema:

theorem Teorema 1
$axiomRelDomEmpty[X, Y]$
$\forall R : X \rel Y \bullet (\dom R = \{\} \Rightarrow R = \{\}) \land (R = \{\} \Rightarrow \dom R = \{\})$

pruebe, usando Z/EVES, que las condiciones mencionadas para el caso de prueba son insatisfactibles.

Ejercicio 4 Tomado en: 2010.07.30, 2015.07.10.

(Arranca igual que el 3.)

Una herramienta de "testing" que extrae casos de prueba analizando la especificación sugiere el siguiente caso de prueba:

- que el elemento a borrar no esté en la tabla de símbolos;
- que el dominio del conjunto formado únicamente por el par (x, y), donde x es el elemento a borrar e y es un elemento cualquiera, sea igual al dominio de la tabla de símbolos.

Pruebe, usando Z/EVES, que las condiciones mencionadas para el caso de prueba son insatisfactibles.

Ejercicio 5 Tomado en: 2010.08.06.

La especificación que se incluye a continuación describe mínimamente el estado de un sistema de archivos y la operación de cambio de información de control de acceso. En este sistema de archivos cada archivo se representa por un entero y tiene asociado un conjunto de *categorías* que representa de una forma u otra la información de control de acceso (llamada *ac*).

La política de seguridad usada en este sistema de archivos requiere que en todo momento los archivos abiertos en modo de lectura tengan una *ac* inferior a los archivos abiertos en modo de escritura. La relación de orden (superior e inferior) está dada por $\subseteq$.

La variable de estado *ac* representa la asociación entre nombres de archivo e información de control de acceso. La variable *secmat* representa los archivos abiertos así como también en qué modo han sido abiertos.

$[CATEGORY]$
$MODE ::= READ \mid WRITE$

State

$$ac : \mathbb{Z} \pfun \mathbb{P}\, CATEGORY$$
$$secmat : \mathbb{Z} \pfun MODE$$

*ChangeObjectAttrConfOk*1

$$\Delta State$$
$$o? : \mathbb{Z}$$
$$c? : \mathbb{P}\, CATEGORY$$

$$o? \in \operatorname{dom} ac$$
$$o? \notin \operatorname{dom} secmat$$
$$secmat' = secmat$$
$$ac' = ac \oplus \{o? \mapsto c+\}$$

*ChangeObjectAttrConfOk*2

$$\Delta State$$
$$o? : \mathbb{Z}$$
$$c? : \mathbb{P}\, CATEGORY$$

$$o? \in \operatorname{dom} ac$$
$$o? \in \operatorname{dom} secmat$$
$$secmat\ o? = READ$$
$$\forall o : \operatorname{dom} secmat \mid secmat\ o = WRITE \bullet c? \subseteq ac\ o$$
$$secmat' = secmat$$
$$ac' = ac \oplus \{o? \mapsto c?\}$$

*ChangeObjectAttrConfOk*3

$$\Delta State$$
$$o? : \mathbb{Z}$$
$$c? : \mathbb{P}\, CATEGORY$$

$$o? \in \operatorname{dom} ac$$
$$o? \in \operatorname{dom} secmat$$
$$secmat\ o? = WRITE$$
$$\forall o : \operatorname{dom} secmat \mid secmat\ o = READ \bullet ac\ o \subseteq c?$$
$$secmat' = secmat$$
$$ac' = ac \oplus \{o? \mapsto c?\}$$

$$ChangeObjectAttrConfOk \mathrel{\widehat=} ChangeObjectAttrConfOk1 \lor ChangeObjectAttrConfOk2 \lor ChangeObje$$

a) Defina formalmente la política de seguridad como un invariante.

b) Pruebe que *ChangeObjectAttrConfOk* preserva el invariante definido en el punto anterior.
Ayuda: pruebe que cada uno de los términos de la disyunción verifica el invariante; la prueba que involucra al primero debería ser casi automática; las otras dos son más laboriosas pero estructuralmente idénticas.

Ejercicio 6 Tomado en: 2010.12.10.

Considere la siguiente especificación en la que dos procesos compiten por un conjunto de recursos. Se espera que los recursos utilizados por ambos coincidan siempre con el conjunto de recursos utilizado por el proceso pB.

$[RESOURCES]$

Processes

$$pA, pB : \mathbb{P}\, RESOURCES$$

La operación *Open* permite que los procesos utilicen un nuevo recurso bajo ciertas restricciones.

*OpenOk*1

$$\Delta Processes$$
$$r? : RESOURCES$$
$$pA \subseteq pB$$
$$pA' = pA \cup \{r?\}$$
$$pB' = pB \cup \{r?\}$$

*OpenOk*2

$$\Delta Processes$$
$$r? : RESOURCES$$
$$pA \cap pB = \emptyset$$
$$pA' = pA \cup \{r?\}$$
$$pB' = pA \cup \{r?\}$$

$$Open \mathrel{\widehat{=}} OpenOk1 \lor OpenOk2$$

a) Formalice el invariante del sistema.

b) Pruebe que *Open* lo satisface.
Sugerencia: haga una prueba por casos según la estructura de la operación. El caso en que $pA \subseteq pB$ es más o menos simple utilizando un par de teoremas del *toolkit* matemático. El otro caso, si bien es simple, requiere ayudar a Z/EVES a demostrar algo trivial. Para hacerlo recurra a la asociatividad de la unión, al hecho de que $A \cup A = A$ y use normalización.

Ejercicio 7 Tomado en: 2011.02.25.

La especificación que se incluye a continuación describe mínimamente el estado de un sistema de archivos y la operación de cambio de información de control de acceso. En este sistema de archivos cada archivo se representa por un entero y tiene asociado un conjunto de *categorías* que representa de una forma u otra la información de control de acceso (llamada *ac*).

La política de seguridad usada en este sistema de archivos requiere que en todo momento los archivos abiertos en modo de lectura tengan una *ac* inferior a los archivos abiertos en modo de escritura. La relación de orden (superior e inferior) está dada por $\subseteq$.

La variable de estado *ac* representa la asociación entre nombres de archivo e información de control de acceso. La variable *secmat* representa los archivos abiertos así como también en qué modo han sido abiertos.

$[CATEGORY]$
$MODE ::= READ \mid WRITE$

State

$ac : \mathbb{Z} \pfun \mathbb{P}\, CATEGORY$
$secmat : \mathbb{Z} \pfun MODE$

OpenObjectWriteConf

$\Delta State$
$o? : \mathbb{Z}$

$o? \in \mathrm{dom}\, ac$
$\forall\, o : \mathrm{dom}\, secmat \mid secmat\ o = READ \bullet ac\ o \subseteq ac\ o?$
$secmat' = secmat \oplus \{o? \mapsto WRITE\}$
$ac' = ac$

a) Defina formalmente la política de seguridad como un invariante.
Ayuda: tenga en cuenta que todo archivo en el dominico de *secmat* tiene que ser, también, un archivo del dominio *ac*.

b) Pruebe que *OpenObjectWriteConf* preserva el invariante definido en el punto anterior.

Ejercicio 8 Tomado en: 2012.02.10.

Considere el siguiente fragmento de una especificación de un sistema de votación electrónico.

$GAC ::= cen \mid caim \mid nil$
$[INVEST]$

Avales

$cand : INVEST \pfun GAC$
$avales : INVEST \pfun \mathbb{P}\, INVEST$

EmitirAvalOk

$\Delta Avales$
$c?, e? : INVEST$

$c? \in \mathrm{dom}\, cand$
$e? \notin \bigcup(\mathrm{ran}\, avales)$
$avales' = avales \oplus \{c? \mapsto avales\ c? \cup \{e?\}\}$
$cand' = cand$

La variable de estado *cand* almacena los candidatos que se presentan a una elección específica (la imagen da la función indica la elección) y la variable *avales* registra los votantes que avalan a un candidato (esto es un paso previo para que un candidato realmente se pueda presentar a la elección). Es decir, *avales c* es el conjunto de votantes que avaló al candidato c.

a) Claramente, el dominio de ambas funciones debe ser siempre el mismo. Escriba esto como un invariante de estado.

b) Pruebe, utilizando Z/EVES, que la operación *EmitirAvalOk* preserva ese invariante.