Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Métodos Formales para Ingeniería de Software

Segundo Cuatrimestre de 2024

Trabajo Práctico Nº 4: **Alloy: Modelado de Dinámica con Relaciones**

Importante: Para la resolución de los ejercicios de este trabajo práctico, es imprescindible el uso del *analizador* de Alloy. A tal efecto, deberá definir comandos para generar instancias significativas del modelo y verificar las restricciones impuestas sobre el mismo, o bien chequear propiedades satisfechas por toda instancia del modelo. De manera similar, deberá verificarse que los predicados y/o funciones definidos siguen
el comportamiento esperado en su definición.

Deberá dejarse constancia de todos los comandos utilizados para la resolución de cada ejercicio.

Ejercicios

1. El problema *Misioneros y Caníbales* es un problema lógico clásico. Involucra a N misioneros y N caníbales que deben cruzar un río utilizando un bote que solo puede trasladar a dos personas a la vez. El número de caníbales que se encuentran cada orilla del río en ningún momento puede superar al número de misioneros en la misma orilla.

   El bote no puede cruzar el río por sí mismo, es decir, sin personas en él.

   (a) Definir un modelo, estableciendo los hechos y/o predicados que sean necesarios para modelar el problema. El modelo debe tener en cuenta en qué orilla se encuentra cada persona. Considere que el efecto del cruce de personas de una orilla a la otra utilizando el bote es inmediato (es decir, no deberán considerarse estados intermedios en los que hay personas arriba del bote que se están trasladando de una orilla a la otra).

   (b) Definir un predicado para modelar las condiciones bajo las cuales un estado dado es válido:

   ```
   pred esValido[e:Estado] { ...  }
   ```

   Para que un estado sea válido, cada persona debe estar en alguna de las dos orillas del río.

   Observación: Recuerden que el nombre de la signatura que modela los estados queda a elección suya, no necesariamente debe llamarse "*Estado*".

   (c) Añadir hechos y/o predicados que especifiquen:

   - La situación del estado inicial.
   - La evolución de los estados.

   Generar instancias para validar el modelo.

   (d) Utilice el analizador para determinar si es posible resolver el problema para 3 misioneros y 3 caníbales. Explique la respuesta obtenida.

2. Considere el siguiente modelo, que corresponde a una simplificación del modelo visto en prácticos anteriores, limitando la atención a un solo libro y sus marcadores.

   (a) Escriba el siguiente modelo en Alloy, correspondiente al dominio antes mencionado:

```
sig Libro { paginas: some Pagina,
            marcadores: set Marcador }

sig Marcador { pag: Pagina }

sig Pagina { }

fact {all l: Libro | #l.marcadores >= 0}

fact {all l: Libro | #l.marcadores < 4}

fact {no m: Marcador, l: Libro | (m in l.marcadores) and (m.pag !in l.paginas) }
```

   (b) Modifique el modelo para poder realizar dinámica basada en relaciones para los marcadores de un libro. Valide el modelo estáticamente.

   (c) Añada los predicados necesarios para modelar dinámica sobre los marcadores de un libro. Los cambios de estado tendrán lugar por el *agregado de un marcador*, la *eliminación de un marcador* y la *modificación de la página marcada por un marcador del libro.*

   Valide el comportamiento del modelo dinámico.

3. Considere un puente levadizo sobre el que pasan autos cuando se encuentra bajo, permitiendo también el paso de barcos (por debajo de él) cuando se encuentra elevado. Considere que en toda instancia deben existir al menos dos barcos y al menos dos autos. Por cuestiones de seguridad, en todo momento puede haber como máximo un auto sobre el puente.

   (a) Evalúe si el modelo dado a continuación es adecuado en un contexto de modelado estático:

```
open util/ordering[Estado] as ord

sig Barco { }

sig Auto { }

abstract sig EstadoPuente { }

one sig Elevado, Bajo extends EstadoPuente { }

sig Estado {
              puente: one EstadoPuente,
              barcosEnEspera: some Barco,
              autosEnEspera: set Auto,
              autosEnPuente: one Auto
}
```

   Deje comentarios sobre los comandos utilizados para validar el modelo y los resultados obtenidos/problemas encontrados. **No modifique el modelo en este inciso**.

(b) Corrija los errores detectados en el inciso anterior y verifique que el modelo sea correcto estáticamente. Puede modificar el modelo de base (definición de signaturas y/o relaciones) y/o agregar hechos, según considere necesario.

Deje expresado en comentarios las modificaciones realizadas y las razones que motivaron los cambios, referenciando explícitamente a los comentarios asociados del inciso anterior (es decir, vincular cada cambio a los problemas detectados previamente que intenta solucionar).

(c) Agregar los predicados y/o facts necesarios para modelar la dinámica del problema. Considere que los cambios de estados se producen porque:

- llega **un** barco al puente; o
- llega **un** auto al puente; o
- sube **un** auto al puente (el puente debe estar bajo); o
- baja **un** auto del puente (el auto de debe estar en el puente); o
- cruza **un** barco (el puente debe estar elevado y el barco debe estar esperando); o
- cambia la posición del puente (cambia de elevado a bajo o de bajo a elevado).

Estas acciones deben ser disjuntas, es decir los cambios de estados se producen por la ocurrencia de solo una de ellas. Escriba la especificación para que así sea.

Entre los **predicados** que defina deberá considerar necesariamente predicados para:

- Expresar cuándo un estado es válido. (Observación: En caso de optar por no definir un predicado para esto deberá dejarse registro, explícitamente, de qué parte del modelo captura la especificación de las condiciones que caracterizan estados válidos).
- Expresar cada acción posible.

Genere instancias para validar el modelo.

(d) Utilice el analizador para validar si es posible que ocurra lo siguiente:

Si hay un auto en el puente, puede mantenerse sobre el puente en forma indeterminada.

Dejar expresado en un comentario su respuesta, haciendo referencia al resultado obtenido por el analizador.

(e) Utilice el analizador para verificar si se cumplen las siguientes propiedades:

- *El puente siempre estará bajo en el estado inmediato siguiente de aquel en el cual hay algún auto sobre el puente.*
- *En todos los estados, si hay un auto en el puente entonces en el próximo estado ya no lo está.*

Deje expresado en un comentario cuál es la respuesta del analizador en cada caso.

(f) Responda y justifique sus respuestas:

- Si se garantiza la prioridad del descenso de un auto por sobre el resto de las operaciones que provocan un cambio de estado[1] ¿se mantiene la respuesta sobre el cumplimiento o no de la propiedad enunciada en el segundo ítem del inciso anterior?
- ¿Qué ocurre con dicha propiedad si se modifica el modelo para permitir que haya más de un auto sobre el puente en forma simultánea? Considere el escenario con y sin prioridad de la operación de descenso por sobre el resto de las operaciones.

[1] Si hay algún auto en el puente, la única operación que puede realizarse es la de descenso de un auto; en caso contrario, puede realizarse cualquiera de las otras.

Modifique el modelo para considerar las situaciones descriptas en cada ítem, respectivamente, y determine sus respuestas utilizando el analizador.

4. Considere el clásico problema de sincronización "*Los filósofos comensales*" o "*Dining philosophers problem*".

   El problema consiste de n filósofos, los cuales están sentados alrededor de una mesa redonda. Cada filosofo tiene su plato y necesita dos tenedores para comer, pudiendo utilizar solo los que están a su alcance. Los tenedores están ubicados a la derecha e izquierda de cada filósofo respectivamente. Cada filósofo comparte cada tenedor con uno de sus vecinos, siendo estos los únicos tenedores que el filósofo en cuestión puede utilizar. La siguiente imagen muestra la disposición de la mesa para 5 filósofos:

   

   La idea es que todos los filósofos puedan comer cuando tienen hambre. La dinámica del problema se establece de la siguiente manera. Al comienzo todos los filósofos están pensando, por lo que todos los tenedores están sobre la mesa. Los cambios de estado están vinculados al efecto de alguna de las siguientes acciones:

   - Tomar **un** tenedor
   - Comer/Pensar (Comer y pasar a pensar)

   IMPORTANTE: Un filósofo solo puede tomar un tenedor por vez (es decir, ninguna acción le permite tomar ambos tenedores al mismo tiempo).

   Considere el siguiente modelo en Alloy para el problema planteado:

```
sig Filosofo {  derecha: lone Tenedor,
                izquierda: Tenedor }

sig Tenedor { }

sig Situacion {  comoEstan: Filosofo -> some Tenedor }

// Todos los filosofos y los tenedores estan ubicados en una sola mesa redonda
fact { all f: Filosofo | #Filosofo = #(f.^(izquierda.(~derecha))) }
```

   (a) Analice si el modelo planteado resulta adecuado. Recuerde tener en cuenta las condiciones que debería tener la "situación" para ser considerada válida: los tenedores no pueden estar en posesión de más de un filósofo a la vez, y los filósofos solo pueden sostener los tenedores que estén a su alcance.

   Además, toda instancia debe contar con al menos dos filósofos. En consecuencia, en cada instancia, la cantidad de tenedores debe coincidir con la cantidad de filósofos.

   Deje comentarios sobre los comandos utilizados para validar el modelo y los resultados obtenidos/problemas encontrados. **No modifique el modelo en este inciso**.

(b) Corrija los errores detectados en el inciso anterior y verifique que el modelo sea correcto estáticamente. Puede modificar el modelo de base (definición de signaturas y/o relaciones) y/o agregar hechos, según considere necesario.

Deje expresado en comentarios las modificaciones realizadas y las razones que motivaron los cambios, referenciando explícitamente a los comentarios asociados del inciso anterior (es decir, vincular cada cambio a los problemas detectados previamente que intenta solucionar).

(c) Agregue hechos y/o predicados (según corresponda) que especifiquen lo siguiente:

i. Un filósofo f puede *comer/pensar* en el estado s. Para poder comer, el filósofo debe poseer los dos tenedores que están a su alcance. Luego de comer pasa a pensar, soltando los dos tenedores.

ii. Un filósofo f puede *tomar uno de los dos tenedores* que se encuentran a su alcance en el estado s. Para que la acción sea posible el tenedor debe estar disponible.

iii. Determinar la situación del estado inicial.

iv. Determinar la evolución de los estados. Estos cambios solo son posibles a través de las operaciones que pueden efectuar los filósofos.

Genere instancias para validar el modelo. Recuerde identificar qué parte de su modelo se corresponde a cada uno de los aspectos mencionados en la lista anterior. Puede agregar cualquier otra restricción que considere necesaria.

(d) Defina un predicado que determine si en un estado dado ningún filosofo puede comer.

(e) Definir una aserción que determine si es posible que se produzca *deadlock*. Verifique la misma y explique el resultado obtenido por el analizador.

Opcional: Se quiere encontrar un criterio para evitar los deadlocks, una manera simple es contar con un "*monitor*".

Solución "monitor": hay un monitor que controla a los filósofos. El monitor controla que no todos los tenedores se tomen al mismo tiempo (es decir, permite a un filósofo tomar un tenedor si tiene otro tenedor en la mano o hay más de un tenedor disponible en la mesa). Definir un predicado que modele el comportamiento del monitor para un estado dado.

Escriba una aserción que permita verificar si la solución "monitor" previene deadlocks. Explique los resultados mostrados por el analizador.