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

Métodos Formales para Ingeniería de Software

Segundo Cuatrimestre de 2024

Trabajo Práctico N° 2: **Modelado Estático y Verificación Formal en Alloy**

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 validar los modelos. De manera similar, deberá verificarse que los predicados y/o funciones definidos siguen el comportamiento esperado en su definición.

Deberá dejarse constancia de los comandos utilizados (es decir, deberán quedar escritos en el archivo .als correspondiente).

Ejercicios

1. Considere en cada inciso, el modelo definido por las signaturas T, U y S. Genere con la herramienta instancias del modelo. Puede pensar en ejemplos de instancias inválidas para cada caso? Por qué serían inválidas?

   ```
   sig T {}
   sig U {}
   ```

   (a) `sig S {r: lone T}`
   (b) `sig S {r: one T}`
   (c) `sig S {r: T -> one U}`
   (d) `sig S {r: T lone -> U}`
   (e) `sig S {r: some T}`
   (f) `sig S {r: set T}`
   (g) `sig S {r: T set -> set U}`
   (h) `sig S {r: T one -> U}`

2. Considere el siguiente modelo en Alloy:

   ```
   abstract sig Target {}
   abstract sig Name extends Target { addressBook: some Target }
   sig Alias extends Name {}
   sig Group extends Name {}
   sig Addr extends Target {}
   ```

   (a) Defina un comando `run` en Alloy con las restricciones necesarias para generar la siguiente instancia del modelo

   ```
   Target = {(Addr0), (Addr1), (Addr2), (Alias0), (Alias1), (Group0)}
   Name = {(Alias0), (Alias1), (Group0)}
   Alias = {(Alias0), (Alias1)}
   ```

```
Group = {(Group0)}
Addr = {(Addr0), (Addr1), (Addr2)}
addressBook = {(Alias0,Addr0), (Alias0,Addr1), (Alias0,Addr2), (Alias0,Alias0),
(Alias0,Alias1), (Alias0,Group0), (Alias1,Addr0), (Alias1,Addr1), (Alias1,Addr2),
(Alias1,Alias0), (Alias1,Alias1), (Alias1,Group0), (Group0,Addr0), (Group0,Addr1),
(Group0,Addr2), (Group0,Alias0), (Group0,Alias1), (Group0,Group0)}
```

(b) Para cada una de las siguientes operaciones, indique el resultado de evaluar dicha operación sobre la instancia arriba utilizando el *evaluador* de Alloy para determinar el resultado de las operaciones aplicadas sobre la instancia generada.

- `Alias + Group`
- `Alias & Target`
- `Name - Alias`
- `Target in Group`
- `Alias in Name`
- `Target = Group + Alias`
- `Alias -> Addr`
- `Name -> Target`
- `Name -> Addr + Name -> Name`
- `addressBook & (Name -> Addr)`
- `addressBook.Alias`
- `addressBook[Name]`
- `Name.addressBook`
- `~addressBook`
- `Alias.~addressBook`
- `^addressBook`
- `*addressBook`
- `*addressBook - ^addressBook`
- `^(Name -> Target)`
- `Group <: addressBook`
- `Addr <: Target`
- `Name :> Group`
- `iden ++ addressBook`

3. Considere el siguiente modelo en Alloy:

```
one sig Juan, Pedro {}

sig Culpable in univ {}

fact { Juan not in Culpable }

fact { Juan in Culpable implies Pedro in Culpable }

assert conclusion { Pedro not in Culpable }
```

(a) Analice el modelo brindado. Para ello, puede utilizar la funcionalidad de visualización provista por Alloy.

(b) Describa el significado intuitivo de los hechos incluidos en el modelo.

(c) Genere instancias del modelo provisto mediante el uso del comando `run { }`. ¿Cuál es el significado intuitivo de este comando? ¿Qué características observa en las instancias generadas?

(d) Defina el comando correspondiente para chequear la aserción definida. ¿Qué observa con respecto al resultado mostrado por la herramienta? ¿Qué sucede si se vuelve a chequear la aserción, habiendo removido el primer hecho? Justifique sus respuestas.

4. Considere el siguiente modelo en Alloy:

```
sig Candidato { }

one sig Alejo, Luca, Carlos, David in Candidato { }

one sig Maria {
                alto : set Candidato,
                moreno : set Candidato,
                buenmozo : set Candidato
}
```

(a) Analice el modelo brindado. Para ello, puede utilizar la funcionalidad de visualización provista por Alloy.

(b) Genere instancias del modelo, sin restricciones adicionales, y analice sus características.

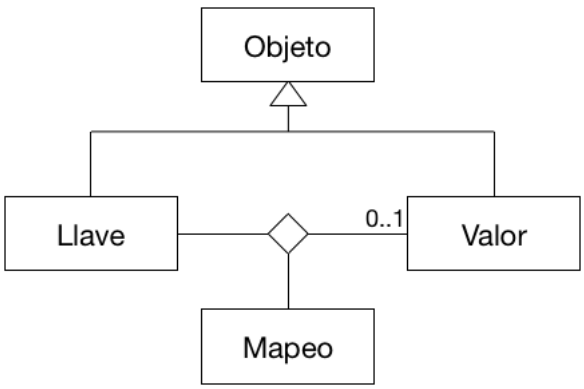
(c) Considere un escenario en el que María tiene cuatro candidatos: Alejo, Luca, Carlos y David. Dicha situación ¿se corresponde con las instancias del modelo anteriormente generadas? En caso negativo, añada las restricciones necesarias para modelar el escenario aquí planteado.

(d) El hombre ideal de María, es alto, moreno y buen mozo. Sólo uno de los cuatro hombres tiene todas las características que María desea ¿cuál es?

Hallar la respuesta a esta pregunta mediante el uso de la herramienta, agregando restricciones sobre el modelo para garantizar que:

- Sólo tres de los hombres son altos, sólo dos son morenos, y sólo uno es buenmozo.
- Cada uno de los cuatro hombres tiene al menos una de las características buscadas por María.
- Alejo y Luca tienen la misma complexión (ambos son morenos o ambos no lo son).
- Luca y Carlos tienen la misma altura.
- O bien Carlos es alto o David lo es, pero ambos no lo son.

5. Considere el siguiente diagrama correspondiente a la definición de un mapeo que relaciona objetos de diferente tipo:



Resuelva los siguientes incisos:

(a) Escriba un modelo en Alloy para representar este dominio, respetando el diagrama brindado y las siguientes consideraciones:
- No deben especificarse signaturas abstractas.
- No deben existir objetos fuera del contexto de mapeos.

(b) ¿Puede existir un objeto que no sea llave o valor? ¿Por qué? Justifique especificando el comando correspondiente en Alloy e indique la respuesta brindada por el analizador.

(c) Utilice una aserción para verificar si el mapeo define una relación funcional parcial entre llaves y valores. ¿Se verifica la aserción? En caso negativo, añada las restricciones necesarias sobre el modelo para asegurar que se cumpla esta propiedad.

(d) ¿Puede existir una llave que no pertenezca a un determinado mapeo? ¿y un valor?. Intente generar instancias en las que ocurran estas situaciones. ¿Fue posible generarlas? ¿Por qué?

(e) Genere una instancia en la que una misma llave se encuentre asociada a dos valores distintos. ¿Qué condición se verifica en tal caso?

6. En matemáticas, un conjunto es una colección de elementos considerada en sí misma como un objeto. Los elementos de un conjunto pueden ser, por ejemplo, personas, números, colores, letras, figuras, etc. Se dice que un elemento (o miembro) pertenece a un conjunto si está incluido de algún modo dentro de él. Si dos conjuntos poseen los mismos elementos, entonces son el mismo conjunto. El conjunto que no contiene elementos se llama conjunto vacío, denotado como $\emptyset$ o $\{\}$, y claramente es un conjunto.

   El siguiente modelo intenta capturar la noción básica de conjunto:

```
sig Conjunto { elementos: some Elemento }

sig Elemento {}

fact { all c1: Conjunto | some c2: Conjunto, e: Elemento |
(c1.elementos = c2.elementos + e) and (e ! in c1.elementos) }
```

   (a) Utilice el analizador para validar si el modelo es correcto. Indique cuáles son los problemas del modelo, corríjalos, y valide el modelo nuevamente. En todos los casos, explique el resultado brindado por el Analizador en respuesta a la ejecución de los comandos definidos para realizar la validación.

   (b) Considere la versión modificada del modelo, realizada en el inciso anterior. ¿Qué es posible asegurar acerca de la cantidad máxima de elementos de los conjuntos existentes en cada instancia del modelo si se considera el mismo scope para las signaturas Conjunto y Elemento? Justifique su respuesta utilizando una aserción y explique la respuesta brindada por el analizador para comando ejecutado.

   (c) Verifique mediante el uso de una aserción que la definición de conjunto dada es *cerrada*, es decir, que todo conjunto puede expresarse como la unión de dos conjuntos. Explique la respuesta brindada por el analizador para comando ejecutado durante la validación. En caso de que la propiedad antes mencionada no valga, modifique el modelo para que la satisfaga y valide nuevamente la aserción definida.

7. Las paradojas resultan buenos problemas para explorar con *model checkers*. Una paradoja muy conocida es la paradoja de Russell, que establece la siguiente situación: "*en una villa en la cual el barbero afeita a cada hombre que no se afeita a sí mismo, ¿quién afeita al barbero?*"

   El siguiente modelo es un intento de reflejar lo descripto en la paradoja:

```
sig Hombre { afeita: set Hombre }
one sig Barbero extends Hombre {}

fact { Barbero.afeita = {h: Hombre | h in h.afeita} }
```

(a) Utilice el analizador para verificar si el modelo resulta apropiado para modelar el problema, considerando una villa de tamaño pequeño. En caso que el modelo no resulte apropiado, indique los problemas existentes, realice los cambios que considere necesario y valide el modelo nuevamente.

El nuevo modelo, ¿soluciona los inconvenientes antes identificados? justifique su respuesta utilizando el analizador.

(b) Las feministas han notado que la paradoja desaparece si se toma en cuenta la existencia de mujeres en la villa.

- Defina un modelo para modelar una nueva versión de la paradoja que permita clasificar a los habitantes de la villa en hombres (que necesitan ser afeitados) y mujeres (que no son afeitadas).
- Utilice el analizador para verificar si el nuevo modelo presenta los inconvenientes del modelo anterior.
- Utilice el analizador para mostrar que el nuevo modelo admite una solución simple a la paradoja.
- Si se mantiene la restricción de que hay un solo barbero en la villa, ¿qué es posible asegurar sobre el barbero? Justifique su respuesta con el analizador mediante el uso de una aserción.

(c) Por último, considere una variante del modelo original que permite múltiples barberos, los cuales afeitan a hombres que no se afeiten a sí mismos. Utilice el analizador para verificar si el modelo permite obtener una solución a la paradoja.

8. Considere la siguiente información acerca de las bicicletas que vende un negocio. Cada bicicleta posee un rodado, el cual es chico, mediano o grande. Asimismo, toda bicicleta es de uno de los siguientes tipos: playera, BMX o mountain bike. Por último, algunas bicicletas tienen módulos de cambios, cada uno de los cuales puede tener 3, 6 o 12 cambios.

Considere el siguiente modelo, para representar la información de este dominio:

```
sig Bicicleta {  rod: one Rodado,
                 mod: some ModuloCambios  }

sig BMX, Playera, MountainBike extends Bicicleta {}
abstract sig Rodado {}
one sig Chico, Mediano, Grande in Rodado {}
abstract sig ModuloCambios {}
one sig Tres, Seis, Doce extends ModuloCambios {}
```

(a) Defina comandos para verificar que el modelo brindado cumple las restricciones brindadas en la descripción anterior. En caso que alguna restricción no se cumpla, indique los motivos por los que no se cumple explicando qué comandos fueron utilizado para validarla y la respuestas brindadas por el analizador para comando ejecutado. Asimismo, para aquellas restricciones no satisfechas, corrija el modelo y valídelo nuevamente.

(b) Añada al modelo la siguiente definición de predicado:

```
pred tiene18cambios[b: Bicicleta]{
((#b.mod = 4) and (#(b.mod & Seis) = 2) and (#(b.mod & Tres) = 2)) or
((#b.mod = 5) and (#(b.mod & Seis) = 1) and (#(b.mod & Tres) = 4)) or
((#b.mod = 6) and (b.mod in Tres))
}
```

¿Es posible que una bicicleta posea 18 cambios? Brinde un comando en Alloy que justifique su respuesta. En caso que su respuesta sea negativa, indique los motivos por los cuales no se admite dicha situación a partir de la respuesta brindada por el analizador. Realice los cambios necesarios (en el modelo, en el predicado, o ambos) para que se permita, y valide nuevamente su respuesta a la pregunta anterior.

9. Dada la siguiente definición de modelo en Alloy:

```
abstract sig Person {  children: set Person,
                       siblings: set Person  }

sig Man, Woman extends Person {}

sig Married in Person { spouse: one Married }
```

(a) Genere y analice tres instancias del modelo brindado. ¿Qué irregularidades detecta? Especifique comandos en Alloy que incluyan las restricciones necesarias para generar instancias que ilustren explícitamente cada irregularidad.

(b) ¿Cómo es posible identificar a los padres de una persona? Defina en Alloy un predicado que permita determinar si dos personas son los padres de otra persona dada, y comandos para verificar su correcta definición. Explique las respuestas brindadas por el analizador para los comandos ejecutados.

(c) Modifique el modelo brindado, añadiendo al mismo las restricciones indicadas en cada ítem:

- Ninguna persona puede ser su propio ancestro.
- Ninguna persona puede tener más de una madre, ni más de un padre.
- Los hermanos de una persona son aquellas personas que poseen algún padre en común (es decir, considere la existencia de medio-hermanos).

(d) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento, ¿la relación *siblings* es simétrica? Defina uno o más comandos en Alloy para verificar esto y explique las respuestas brindadas por el analizador.

(e) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento, ¿la relación *siblings* admite que una persona sea hermana de sí misma? Defina uno o más comandos en Alloy para verificar esto y explique las respuestas brindadas por el analizador.

¿Cómo modificaría el modelo para asegurar que la restricción antes mencionada se cumpla? Realice los cambios necesarios y defina uno o más comandos para verificar la correctitud de dicha restricción, explicando las respuestas brindadas por el analizador para cada comando ejecutado.

(f) Defina en Alloy una restricción que permita determinar si dos personas son parientes de sangre. (Observación: En general, dos personas son parientes de sangre si poseen un ancestro en común).

Defina uno o más comandos en Alloy para verificar la correctitud de la restricción previamente definida y explique las respuestas brindadas por el analizador para comando ejecutado.

(g) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento ¿es posible que dos parientes de sangre tengan hijos en común? Defina uno o más comandos en Alloy para verificar esto y explique las respuestas brindadas por el analizador para comando ejecutado.

Luego, modifique el modelo para garantizar que dicha restricción se cumpla y verifique su correctitud mediante la definición de uno o más comandos en Alloy, explicando las respuestas brindadas por el analizador para comando ejecutado.

(h) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento, ¿la relación *spouse* es simétrica? ¿Es posible que una persona esté casada con más de una persona? ¿Es posible que una persona esté casada consigo misma? Defina uno o más comandos en Alloy para verificar esto y explique las respuestas brindadas por el analizador para comando ejecutado.

¿Cómo modificaría el modelo para asegurar la primera restricción y evitar la segunda y tercera? Realice los cambios correspondientes y valide nuevamente el modelo, explicando las respuestas obtenidas para la ejecución de los comandos definidos.

(i) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento ¿es posible que una persona esté casada con un familiar de sangre? Defina uno o más comandos en Alloy para verificar esto y explique las respuestas brindadas por el analizador para comando ejecutado.

¿Cómo modificaría el modelo para evitar dicha situación? Realice los cambios correspondientes y valide nuevamente el modelo, explicando las respuestas obtenidas para la ejecución de los comandos definidos.

(j) A partir del modelo resultante de añadir las restricciones indicadas hasta el momento ¿es posible que haya personas que no posean padre ni madre?

Defina un predicado que permita determinar si una persona se encuentra en tal situación, y una función que permita obtener el conjunto de personas en tal condición.

Defina uno o más comandos en Alloy para verificar la correctitud de las restricciones previamente definidas y explique las respuestas brindadas por el analizador para comando ejecutado.

(k) Defina comandos `run` para crear, en caso de ser posible, instancias del modelo que cumplan las siguientes restricciones (OBSERVACIÓN: defina un comando para cada ítem):

- Crear una instancia en la que las relaciones *spouse* y *siblings* no sean vacías.
- Crear una instancia con dos parejas casadas *diferentes*.
- Crear una instancia con a lo sumo un átomo en cada signatura de alto nivel y con un hombre casado.
- Crear una instancia con a lo sumo dos átomos en cada signatura de alto nivel y con un hombre casado.
- Crear una instancia donde haya a lo sumo una persona, alguna mujer y ningún hombre.
- Crear una instancia donde haya exactamente 3 mujeres, 4 hombres y 2 personas casadas.
- Crear una instancia donde haya al menos 2 mujeres, entre 1 y 3 hombres, y exactamente 6 personas casadas.

Explique las respuestas obtenidas a partir de la ejecución de cada comando.