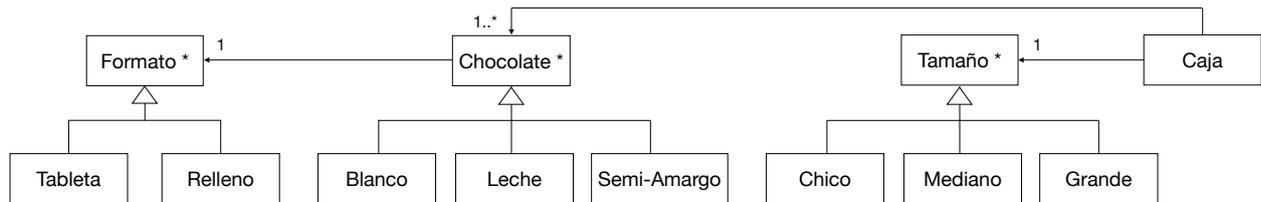


MÉTODOS FORMALES PARA INGENIERÍA DE SOFTWARE

Evaluación - Modelado Estático, Modelado de Dinámica y Verificación Formal

Segundo Cuatrimestre de 2023

El siguiente diagrama modela información acerca de las características de los chocolates que elabora una empresa artesanal, así como también información sobre las cajas en las que los vende:



Cada chocolate es de un tipo: blanco, con leche o semi-amargo. Asimismo, cada chocolate tiene un formato: tableta o relleno. Los chocolates se venden en cajas de los siguientes tamaños: chico, mediano o grande. Las cajas chicas tienen entre 1 y 3 chocolates, las medianas entre 4 y 5, y las grandes entre 6 y 7 chocolates.

Considere el siguiente modelo en Alloy, correspondiente al dominio antes presentado. Asuma que el mismo es correcto y fue validado en un contexto de modelado estático:

```

abstract sig Chocolate { formato: FormatoChoco }

sig ChocoBlanco, ChocoLeche, ChocoSemiAmargo extends Chocolate {}

abstract sig FormatoChoco {}

one sig Tableta, Relleno extends FormatoChoco {}

abstract sig Caja {
    tam: Tamano,
    chocolates: some Chocolate
}

abstract sig Tamano {}

one sig Chico, Mediano, Grande extends Tamano {}

fact { all c: Caja | (c.tam = Chico) implies (#c.chocolates =<3) }

fact { all c: Caja | (c.tam = Mediano) implies (#c.chocolates >=4 and #c.chocolates =<5) }

fact { all c: Caja | (c.tam = Grande) implies (#c.chocolates >=6 and #c.chocolates =<7) }

fact { no disj c1, c2: Caja, c: Chocolate | (c in c1.chocolates) and (c in c2.chocolates) }
    
```

- a) Extienda el modelo brindado incorporando las siguientes restricciones:
- Los chocolates dentro de una misma caja son del mismo tipo o poseen el mismo formato.
 - Las cajas grandes no pueden tener más de 3 chocolates rellenos.

Brinde comandos para validar el modelo resultante de añadir dichas restricciones. Para cada comando, deberá explicarse el propósito del mismo y el resultado obtenido al ejecutarlo. En particular, para aquellos comandos que no generen instancias, deberán explicarse los motivos por los cuales no lo hacen.

- b) Realice una copia del archivo `.als` elaborado hasta el momento, y trabaje sobre el nuevo archivo. Incorpore al modelo resultante del inciso anterior el siguiente predicado, el cual modela el comportamiento de *reemplazar un chocolate de una caja por otro chocolate*. Esta acción es posible siempre y cuando la caja original posea al menos un chocolate relleno y a lo sumo un chocolate blanco:

```
pred reemplazarChocolate[ch1, ch2: Chocolate, ca1, ca2: Caja]{
    (ch2 ! in ca1.chocolates) and
    (#(ca1.chocolates & formato.Relleno) >=1) and
    (ch2 in ca2.chocolates)
}
```

Utilice el analizador para validar si la definición del predicado `reemplazarChocolate` es correcta, considerando el modelo resultante del inciso *a)* y la descripción brindada para el predicado. **IMPORTANTE:** En este inciso no deberán realizarse modificaciones sobre el archivo resultante del inciso *a)* más allá del añadido del predicado `reemplazarChocolate` y los comandos utilizados para validarlo.

Deberá dejarse registro de cada comando utilizado en el proceso de verificación, teniendo en cuenta las siguientes consideraciones:

- Para cada comando, deberá indicarse el *propósito* y el *resultado esperado* de su ejecución (considerando la descripción brindada para el predicado) y el *resultado obtenido* al ejecutarlo.
 - Para aquellos comandos que no generen instancias, deberán explicarse los motivos por los cuales no lo hacen.
 - Para aquellos comandos que generen instancias en las que se observan irregularidades, deberá dejarse registro de dicha instancia (por ejemplo, mediante una captura de pantalla), describiendo cuáles son las irregularidades o problemas allí observados e indicando cuál fue el comando utilizado para generar dicha instancia; en caso de ser necesario, deberá indicarse también el número de instancia generada por dicho comando (por ejemplo, si se trata de la primera instancia, de la segunda instancia, etc.).
- c) Realice una copia del archivo `.als` elaborado hasta el momento, y trabaje sobre el nuevo archivo. Realice los cambios necesarios en el predicado `reemplazarChocolate` y/o en el modelo de manera tal que la nueva definición del predicado respete la descripción brindada anteriormente.

IMPORTANTE: La versión modificada del predicado deberá tener el siguiente encabezado:

```
reemplazarChocolateV2[ch1, ch2: Chocolate, ca1, ca2: Caja]
```

Una vez efectuados los cambios necesarios, valide el predicado resultante considerando al menos 6 casos significativos. En particular, para cada comando definido, deberá indicarse el *propósito* y el *resultado esperado* de su ejecución (es decir, si se espera que el predicado tenga éxito o no, considerando la descripción brindada para el mismo) y el *resultado obtenido* al ejecutarlo.

- d) Realice una copia del archivo `.als` elaborado hasta el momento, y trabaje sobre el nuevo archivo. Defina un predicado que modele el comportamiento de *añadir un chocolate a una caja de tamaño chico o mediano*:

```
agregarChocolate[ch: Chocolate, ca1, ca2: Caja]
```

Si la caja original ya poseía la cantidad máxima de chocolates permitidos por su tamaño, la operación genera el traspaso de su contenido a una caja del tamaño inmediato siguiente, a la que se añade también el nuevo chocolate. (OBSERVACIÓN: el tamaño inmediato siguiente a *chico* es *mediano*, y el tamaño inmediato siguiente a *mediano* es *grande*).

Deberá explicitarse toda pre y post-condición asociada a la operación, incluyendo las indicadas en la descripción brindada, así como también las condiciones de marco y cualquier otra condición que sea necesaria.

Valide el predicado definido, considerando al menos 4 *casos de éxito significativos* y al menos 2 *casos de no éxito significativos*. En particular, para cada comando definido, deberá indicarse claramente el *propósito*, el *resultado esperado* de su ejecución (es decir, si se espera que el predicado tenga éxito o no, considerando la descripción brindada para el mismo) y el *resultado obtenido* al ejecutarlo.

Observaciones Generales:

- Puede brindarse cualquier otra especificación (hecho, aserción, predicado, función, etc.) adicional que considere necesaria y sea sensata.
- Como se mencionó anteriormente, deberá dejarse registro de todo comando utilizado para validar el modelo, las restricciones, predicados y/o funciones definidos.
- En caso de utilizar el evaluador para complementar la validación del modelo, deberá dejarse registro (mediante capturas de pantalla) de la verificación realizada con el evaluador, indicando el comando utilizado para generar la instancia sobre la cual se utilizó el mismo.