

Métodos Formales para Ingeniería de Software

Ma. Laura Cobo

Modelado Estático – Modelado de Dinámica

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

Departamento de Ciencias e Ingeniería de la Computación – Universidad Nacional del Sur, Argentina

Alloy: modelado estático y dinámico

Modelado Estático

- Describe estados, no comportamientos
- Las propiedades son **invariantes**

Modelado Dinámico

- Describe transiciones entre estados
- Las propiedades son **operaciones**

Alloy: modelado estático y dinámico

Modelado Estático

- Describe estados, no comportamientos
- Las propiedades son **invariantes**

Una lista está ordenada

Cada libro tiene al menos un autor

Modelado Dinámico

- Describe transiciones entre estados
- Las propiedades son **operaciones**

Cómo funciona el algoritmo de ordenamiento

Cuál es el efecto de agregar un libro a la biblioteca

Alloy: modelado estático

Modelos estáticos

- Define los valores permitidos como componentes de los estados
 - Valores para los conjuntos
 - Valores para las relaciones
- Una instancia del modelo es un conjunto que establece el estado de los valores de las componentes que:
 - Satisfacen las restricciones definidas por multiplicidad, hechos, condiciones, etc.

Alloy: modelado estático

Esta basado en la construcción del modelo:

- Clasificar los átomos:
 - Considerar qué es relevante
 - Determinar la estructura jerárquica
 - Determinar subconjuntos y clasificación ortogonal de los mismos
- Modelar relaciones
 - Agregar relaciones y restricciones sobre las relaciones existentes
 - Restricciones basadas en la naturaleza de las abstracciones (simetría, relaciones acíclicas, ordenadas, etc)

Alloy: modelado dinámico

Los modelos estáticos permiten describir estados legales para un sistema dinámico

Pero se espera que las transiciones legales entre estados puedan describirse. Por ejemplo

- Que una persona haya nacido antes de casarse
- Que exista el libro en la biblioteca antes de sacarlo en préstamo.

Alloy: modelado dinámico

Esto implica el modelado de la idea de **transición**

Así por ejemplo la relación prestado, no será igual todo el tiempo. Habrá un momento donde la relación está vacía, un momento donde la relación contendrá a un libro que ha sido prestado y así siguiendo.

Alloy: modelado dinámico

Alloy no cuenta con una noción de embebida de estado, ni de transición entre estados.

Sin embargo hay varias maneras de modelar los aspectos dinámicos de un sistema

¿Qué podemos esperar de las operaciones?

Dado que las firmas son estáticas ...

¿cómo hablamos de las operaciones? ¿de los efectos?

¿Qué podemos esperar de las operaciones?

Dado que los átomos son estáticos

¿cómo hablamos de las operaciones? ¿de los efectos?

Alloy no considera la noción de tiempo o de estado mutable.

Resulta necesario modelar estas nociones explícitamente

¿Qué podemos esperar de las operaciones?

Dado que los átomos son estáticos

¿Cómo expresamos una transición?

Puede modelarse a través de un predicado que establezca una relación entre dos estados

- El estado **anterior** a la transición y
- El estado **siguiente**

Requiere las restricciones necesarias: pre y post condiciones para cada transición, condiciones de marco “frame”

Solución: “nuevo” átomo

Una de la estrategias es tener un nuevo átomo que refleja el cambio realizado.

De esta manera capturamos la noción de **cambio**

Solución: “nuevo” átomo

Una de la estrategias es tener un nuevo átomo que refleja el cambio realizado.

De esta manera capturamos la noción de **cambio**

La desventaja de esta aproximación está en que ya no se trata del mismo “objeto” sino de otro que en general es igual salvo por el efecto esperado de la operación

Solución: “nuevo” átomo

Modelo

```
sig Biblioteca { coleccion: set Libro}  
sig Libro { escritoPor: set Autor}  
sig Autor {}
```

Si agregamos un libro a nuestra biblioteca, el efecto es que hay una nueva biblioteca luego de cada mutación.

```
pred agregar[b, b1: Biblioteca, l: Libro] {  
  b1.coleccion = b.coleccion + l  
}
```

Solución: “nuevo” átomo

Modelo

```
sig Biblioteca { coleccion: set Libro}  
sig Libro { escritoPor: set Autor}  
sig Autor {}
```

Si agregamos un libro a nuestra biblioteca, el efecto es que hay una nueva biblioteca luego de cada mutación.

```
pred agregar[b, b1: Biblioteca, l: Libro] {  
  b1.coleccion = b.coleccion + l  
}
```

Efectivamente, ¿siempre se agrega l a la colección? ¿es b1 siempre un átomo diferente de b?

Solución: “nuevo” átomo

Modelo

```
sig Biblioteca { coleccion: Libro -> Autor}  
sig Libro { escritoPor: set Autor}  
sig Autor {}
```

Si agregamos un libro a nuestra biblioteca, el efecto es que hay una nueva biblioteca luego de cada mutación.

```
pred agregar[b,b1:Biblioteca,l:Libro,a: Autor]  
{  
  b1.coleccion = b.coleccion + l -> a  
}
```


Solución: “nuevo” átomo

Modelo

```
sig Biblioteca { coleccion: Libro -> Autor}  
sig Libro { escritoPor: set Autor}  
sig Autor {}
```

Se modela garantizando las precondiciones en el predicado y asegurando las poscondiciones o efectos de la acción dinámica que se intenta modelar.

```
pred agregar[b,b1:Biblioteca,l:Libro,a: Autor]  
{  
  no (l->a in b.colección)      (precondición)  
  b1.coleccion = b.coleccion + l -> a (postcondicion)  
}
```

ejemplo Biblioteca: agregar con la relacion ternaria

Se modela la situación de manera mas realista.

```
sig Biblioteca{coleccion: Libro -> Autor}
sig Libro{escritoPor : set Autor}
sig Autor{}

/* si el libro esta en la biblioteca esta con todos sus autores*/
fact {all b:Biblioteca, l: Libro | (l in b.coleccion.Autor) implies l.(b.coleccion) = l.escritoPor}

pred agregar[b1,b2:Biblioteca, l:Libro]{
  --precondiciones
  not ((l -> (l.escritoPor)) in b1.coleccion) and
  --poscondicion
  (l -> (l.escritoPor)) in b2.coleccion and
  -- marco o frame
  b2.coleccion = b1.coleccion + (l -> (l.escritoPor))
}

run{ some bib1,bib2:Biblioteca, lib:Libro | agregar[bib1,bib2,lib] }
```