Métodos Formales para Ingeniería de Software

Modelado de Dinámica

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

Alloy: modelado estático vs. modelado de dinámica

Modelado Estático

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

```
Una lista está ordenada

Cada libro tiene al menos un autor
```

Modelado de Dinámica

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

```
Cuál es el resultado de un algoritmo de
ordenamiento

Cuál es el efecto de agregar un libro a una
biblioteca
```

Alloy: modelado estático

Modelo Estático

- 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 (signaturas y relaciones) que satisfacen las restricciones definidas por las keywords de multiplicidad, los hechos, etc.

Alloy: modelado estático

Esta basado en la construcción del modelo:

- Modelar signaturas:
  - Considerar qué es relevante.
  - Determinar la estructura jerárquica.
  - Determinar subconjuntos y clasificación de los mismos.

- Modelar relaciones
  - Definir relaciones y restricciones sobre las relaciones existentes.
  - Restricciones basadas en la naturaleza de las abstracciones (simetría de relaciones, relaciones acíclicas, relaciones ordenadas, etc.).

Alloy: modelado de dinámica

Los modelos estáticos permiten describir estados legales para un sistema dinámico.

Más allá de esto, se espera que las transiciones legales entre estados, que capturan la **dinámica** del sistema, puedan describirse. Por ejemplo:

```
Una persona debe haber nacido antes de casarse.

Un libro debe estar en la biblioteca antes de otorgarlo en préstamo.
```

Alloy: modelado de dinámica

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

Así, por ejemplo, la relación *prestado* definida sobre los libros de una biblioteca no será igual todo el tiempo:

Habrá un momento en el que la relación está vacía, un momento en el que la relación indicará que hay un libro que ha sido prestado, y así siguiendo.

Alloy: modelado de dinámica

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 signaturas son estáticas …

¿cómo hablamos de las operaciones?

¿y de sus efectos?

¿Qué podemos esperar de las operaciones?

Dado que los átomos son estáticos …

¿cómo hablamos de las operaciones?

¿y de sus efectos?

Alloy no considera la noción de tiempo o de estado mutable en forma embebida en la definición de signaturas y relaciones.

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 átomos en dos estados:

- El estado **actual**, en el que se aplica la transición; y
- El estado **siguiente**, al que se arriba como resultado de aplicar la transición.

Requiere modelar ciertas restricciones: *pre* y *post condiciones* para cada transición, y condiciones de marco (*frame*).

Solución: “nuevo” átomo

Una estrategia es utilizar un “nuevo” átomo que refleja las características que tendrá el átomo “original” luego de efectuarse la transición.

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

Hablamos de un “nuevo” átomo porque se trata de un átomo ya existente en la misma instancia que el átomo “original”, dado que las instancias del modelo son inmodificables.

Solución: "nuevo" átomo

Una estrategia es utilizar un "nuevo" átomo que refleja las características que tendrá el átomo "original" luego de efectuarse la transición.

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 aplicada.

Solución: "nuevo" átomo - Ejemplo

Modelo

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

Si agregamos un libro a nuestra biblioteca, el efecto es que existe una "nueva" biblioteca luego de cada modificación.

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

Efectivamente, ¿siempre se agrega **l** a la colección?

¿**b1** siempre es un átomo diferente de **b**?

Solución: "nuevo" átomo - Ejemplo

Modelo

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

Si agregamos un libro a nuestra biblioteca, el efecto es que existe una "nueva" biblioteca luego de cada modificación.

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

Como en el caso anterior,
¿se garantiza el añadido *efectivo* del libro?

Solución: "nuevo" átomo - Ejemplo

Modelo

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

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

```
pred agregar[b,b1:Biblioteca,l:Libro,a:Autor]
{
   ((l -> a) !in b.coleccion) and  //precondición
   (b1.coleccion = b.coleccion + (l -> a)) //postcondicion
}
```

Solución: "nuevo" átomo - Ejemplo

Modelo

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

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

Consideremos que restringimos el modelo para establecer una correspondencia entre la información de libros y sus autores expresada por las relaciones *coleccion* y *escritoPor.*

¿qué impacto tiene esto en la definición del predicado *agregar?*

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] }
```