**Capitulo 28 - Métodos formales**

**Conceptos Básicos**

* Un método es formal si tiene solidas bases matemáticas. Esta base ofrece los medios para definir con precisión nociones como consistencia, completud, especificación, implementación y corrección.
* La sintaxis de un lenguaje formal de especificación permite que los requisitos y el diseño se interpreten solo en una forma, lo que elimina la ambigüedad.
* Las facilidades descriptivas de la teoría de conjuntos y la notación lógica permiten un planteamiento claro de los hechos.

**Diferencias con enfoques menos formales**

Los métodos formales eliminan:

* Contradicciones: conjunto de planteamientos que divergen unos con otros.
* Ambigüedades: planteamientos que se interpretan en varias formas.
* Vaguedades: ocurren porque la especificación de un sistema es muy extensa.
* Incompletud: cuando se dejan de lado detalles importantes.
* Grados mixtos de abstracción: ocurren cuando planteamientos muy abstractos se mezclan al azar con planteamientos que tienen un grado mucho menor de detalle.

A través de las matemáticas se puede describir exactamente una situación física, un objeto o el resultado de una acción.

En los métodos formales, es posible aplicar una prueba matemática para demostrar que un diseño encaja en una especificación y que el código del programa es un reflejo correcto de un diseño.

**Conceptos de métodos formales**

Principales conceptos involucrados en la especificación matemática de los sistemas software:

* Invariante de datos: es una condición verdadera a lo largo de la ejecución del sistema. Describe lo que está garantizado que no cambiará.
* Estado: representa un modo de comportamiento observable externamente.
* Operación: es una acción que ocurre dentro del sistema y lee o escribe datos.

Las operaciones se relacionan con 3 condiciones:

* Invariante.
* Precondiciones: define las circunstancias en las cuales es válida una operación particular.
* Pos condiciones: define lo que está garantizado que será cierto hasta completar una operación.

**Preliminares matemáticos**

Conjunto: es una colección de objetos o elementos. Éstos son únicos (sin duplicados). Los conjuntos con un número pequeño de elementos se escriben dentro de llaves, con los elementos separados por comas (ej.: {A,B,C}).

Cardinalidad: el número de elementos de un conjunto. El operador # proporciona la cardinalidad de un conjunto (ej.: #{A,B,C} = 3).

**Especificación constructiva de conjuntos**

Es otra forma de definir conjuntos, en la cual la forma general de un conjunto se especifica empleando una expresión booleana. Define explícitamente la regla que se aplico al construir el conjunto.

Consiste en tres componentes:

Firma: especifica el intervalo de valores que se consideraran cuando se forme el conjunto.

Predicado: define como se restringirá el conjunto.

Término: brinda la forma general del elemento del conjunto.

**Operadores de conjuntos**

* ∈ : indica la pertenecía de un conjunto. Ejemplo : x ∈ X
* ∉ : indica la NO pertenencia de un conjunto. Ejemplo : x ∉ X
* ⊂ : indica que los elementos de un conjunto están en otro conjunto. Ejemplo : {1,2} ⊂ {4,3,2,1}
* ⊆ : indica que los elementos de un conjunto son iguales a los contenidos por otro conjuntos. Ejemplo : {1,2,3} ⊆ {1,2,3}
* ∪ : indica la unión de 2 conjunto. Ejemplo : {1,2} ∪ {3,4,1} = {1,2,3,4}
* ∩ : indica la intersección de 2 conjuntos. Ejemplo : {1,2,3} ∩ {2,3,6} = {2,3}
* \ : indica la diferencia entre conjuntos, en otras palabras, elimina los elementos iguales en los conjuntos analizados. Ejemplo : {A,B,C,D} \ {A,C} = {B,D}
* x : producto cartesiano. El conjunto que se forma, es el equivalente a las combinaciones de los elementos del primer conjunto con cada elemento del segundo. Ejemplo : {1,2} x {3,4} = {(1,3),(1,4),(2,3),(2,4)}
* ℙ : conjunto potencia. Colección de todos los posibles subconjuntos del conjunto original. Ejemplo : ℙ{1,2,3} = {∅,{1}, {2}, {3}, {1,2},{1,3}, {2,3}, {1,2,3}}

**Conjunto vacio ∅**

Es un conjunto especial que corresponde a cero en las matemáticas normales. Tiene la propiedad de ser un subconjunto de cualquier otro conjunto.

Dos identidades que involucran al conjunto vacio:

* ∅ ∪ A = A
* ∅ ∩ A = ∅

**Operadores lógicos**

Los operadores matemáticos equivalentes:

* ∧ = y
* ∨ = o
* ¬ = no (negación)
* ⇒ = implica

Cuantificación universal: es una forma de elaborar un planteamiento acerca de los elementos de un conjunto que resulta verdadero para cualquier miembro del conjunto. Se utiliza el símbolo ∀

Ejemplo : ∀ij,ℕ•i > j ⇒ i² > j² para cada par de valores en el conjunto de los naturales, si i es mayor a j, entonces i² es mayor que j².

**Sucesiones**

* Es una estructura matemática que modela el hecho de que sus elementos están ordenados.
* Una sucesión s es un conjunto de pares cuyos elementos varían de 1 al elemento de mayor número: {(1,A),(2,F),(3,B)}
* La colección de los primeros elementos de los pares se conoce como dominio de la sucesión.
* La colección de los segundos elementos se conoce como intervalo de la sucesión.
* Otra forma de representar una sucesión: <A,F,B>
* A diferencia de los conjuntos, la duplicación se permite en una secesión, por lo que su orden es importante: <A,F,B> ≠ <A,B,F>
* La sucesión vacía se representa: <>
* Cuando se emplea una sucesión en un estado, se debe designar mediante la palabra seq.

**Operadores de Sucesión**

Los operadores son:

* ⌢ : concatenación.
  + <1,6,2> ⌢ <5,2,9> = <1,6,2,5,2,9>
* Cabeza : extrae el primer elemento de una sucesión
  + Cabeza <1,6,2,5,2,9> = 1
* Cola : proporciona los últimos n-1 elementos
  + Cola <1,6,2,5,2,9> = <6,2,5,2,9>
* Último : extrae el elemento final de la sucesión
  + Último <1,6,2,5,2,9> = 9
* Frente : proporciona los primeros n-1 elementos
  + Frente <1,6,2,5,2,9> = <1,6,2,5,2>

**Lenguajes formales de especificación**

Están compuestos usualmente de tres componentes principales:

* Una sintaxis, que define la notación especifica con la que se representa la especificación.
* Semántica, para ayudar a definir un “universo de objetos” que se empleara para describir el sistema.
* Un conjunto de relaciones que definen las reglas que indican cuales objetos satisfacen adecuadamente la especificación.

**OCL**

El lenguaje restringido a objetos (OCL) es una notación formal desarrollada de modo que los usuarios de UML puedan conferirle mayor precisión a sus especificaciones. Se agregan expresiones OCL a los diagramas (por ejemplo de clases) para establecer hechos sobre los elementos de los mismos. Estas expresiones se denominan restricciones, y cualquier implementación derivada del modelo debe garantizar que cada una de ellas siempre permanezca verdadera.

Una expresión OCL involucra operadores que operan sobre objetos. El resultado de una expresión completa siempre debe ser booleana es decir verdadero o falso. Los objetos pueden ser instancias de la clase OCL colección, de la cual conjunto y sucesión son dos subclases.

**Lenguaje de especificación Z**

Aplica conjuntos tipificados, relaciones y funciones dentro del contexto de predicados lógicos de primer orden para construir esquemas.

Esquema: un medio para estructurar una especificación formal. Describe los datos almacenados a los que un sistema accede y altera (el estado).

El esquema identifica las operaciones que se aplican para cambiar el estado y las relaciones que ocurren dentro del sistema. Un esquema está compuesto por:

* Nombre del esquema.
* Declaraciones, que identifican las variables que comprenden el estado del sistema.
* Invariante, que impone restricciones en la forma en la que el estado puede evolucionar.

**Los 10 mandamientos de los métodos formales**

1. Se debe elegir una notación apropiada.
2. Se debe formalizar pero no en exceso.
3. Se estimaran costos.
4. Se debe tener a un experto en métodos formales a disposición.
5. No se debe abandonar los métodos tradicionales de desarrollo.
6. Se debe de documentar suficientemente.
7. No se debe comprometer los estándares de calidad.
8. No se debe ser dogmatico.
9. Se llevaran a cabo pruebas extensivas bien planeadas.
10. Se debe apuntar a la reutilización.