Formal Methods in Software Engineering

David Martinez Diaz -- DSD

Indice de la charla:

---

1. Metodos formales

Son tecnicas sustentandas en las matematicas para analizar el software para extraer conclusiones que sean ciertas sobre este.

Los primeros que intentaron formalizar lo que ellos sabian en lo que en ese momento era secuencial, estos eran Dijsktra y Hoare.

La tecnologia no era muy potente, pese a ser la base, todo esto paso gracias al metodo del "Model Checking", y partir de ahi, surgieron una gran cantidad de tecnicas y organismos.

Por otro lado, se encuentran los mostradores de los Teoremas, que son herramientas complementarias, ya que son metodos mas deductivos.

- Abstract Interpretation.
- Model Based Testing.
- Runtime Verification.

2. Model Checking

Model Checking es una técnica para analizar sistemas de software y hardware. Sirve para asegurarnos de que el sistema cumpla con las especificaciones y propiedades deseadas.

Pasos del proceso de Model Checking:

Modelado del sistema:

Verificación: Para compara el modelo con las propiedades y determinar si se cumplen o no.

Contraejemplo: Si el sistema no cumple alguna propiedad.

El analisis de la parte critica del software es muy dificil de analizarla, y tenemos que realizarlo de forma exhaustiva.

Hay dos necesidades del software:

- Cada vez se necesita aun mas codigo para poder realizar las distintas operaciones, como pueden ser, el software que gestiona los vuelos o incluso las mismas televisiones.
- La concurrencia, es muy dificil analizar cuando alguno de ello va a dar un error, es muy dificil a priori saber que va a hacer (no determinista).

Los sistemas reactivos proceden en un ciclo continuo, es decir, esperan para el siguiente movimiento del entorno y reaccionan propiamente.

Los padres de esta tecnica son Clarke, Emerson y Sifakis, los cuales recibieron el premio Turing en 2007.

Descripcion del Model Checking

Metodo automatico para nalizar estados de modelo, teniendo en cuenta varios descriptores, como por ejemplo, ver si encaja o no encaja, ademas si un problema coincide en estado dando coincidencias, se dice que se cumple la propiedad. En caso de no ser asi, se denomina contraejemplo, esto nos puede ayudar en el futuro para depurar.

Estas propiedades se suele observar justo al reves, es decir, comenzamos diciendo que no satisface el modelo y vamos corrigiendo y comprobando poco a poco los estados.

Propiedades

- Logica Temporal Lineal, que se construyen utilizando proposiciones atomicas de "AP" y "Operadores Temporales".
- Elasticidad.

Introduccion a Uppaal:

Es una herramienta capaz de analizar las transicioens con tiempo real y con este se introducen los automatas con tiempo, utilizando las variables "clock" y el sistema para representar el tiempo.

POdemos decorar los sistemas de transcion con guardas de las transiciones, es decir, a los valores de la variable "clock" para que se ejecute las operaciones, los relojes son las variables especificas del automata.