Simulación y modelos

2017

Fernando Asteasuain (fasteasuain@undav.edu.ar)
Ingeniería en Informática
Universidad Nacional de Avellaneda

Herramientas

GOAL:

http://goal.im.ntu.edu.tw/wiki/doku.php?id=goal:download

-LTSA

http://www.doc.ic.ac.uk/~jnm/book/ltsa/download.html

Santo Grial

- Santo Grial En Software:

Ausencia de errores y bugs

Testing

- ¿Testing es el lado oscuro del software?

Testing: sus propósitos

- Programadores invierten mucho tiempo en testear y debuguear software.

- Plétora de aplicaciones, herramientas, etc, enfocadas en el proceso de Testing.

Tipos de Testing

Testing: Sus limitaciones

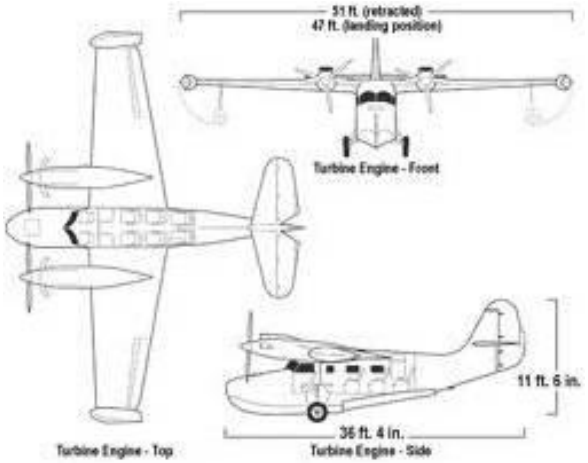
- **Correcto pero no completo**

- **No garantiza la ausencia de errores**.
- **Pueden todavía ocurrir fallas graves en los sistemas**.

- Se buscaron alternativas en métodos formales y verificación.

Verificación formal

Cumple



Sistema **Especificación**

Queremos ver si el sistema se comporta como esperado

Los 80's

- Verificación formal: Pruebas formales manuales usando axiomas y reglas de inferencia

¿Problemas?

*The building has to be at **least**… three times bigger than this!*

Había que especificar TODO el sistema

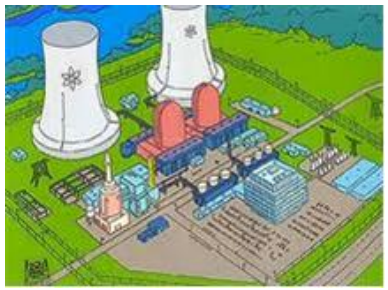
Model Checking

- *Proceso **automático** para comprobar el comportamiento de un modelo del sistema respecto de ciertas propiedades.*

- *Cada propiedad se enfoca en un aspecto en particular del comportamiento*

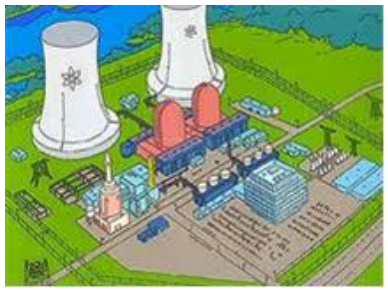
- *Tools: Bandera, Blast, LTSA, SPIN, UPPAAL, VinTime, Zeus, Kronos.*

Transferencia a la industria



- ¿Qué es transferencia?
- El proceso de transformar creaciones del mundo académico al mundo industrial.
- ¿Ejemplos?
  - Internet
  - Google Maps
  - Forall, foreach: constructores en lenguajes de programación, etc.

Transferencia a la industria



- ¿Cuál es la mayor dificultad para que estas técnicas de model checking se adopten en la industria?

Escribir las Propiedades!

Verificación

Herramienta Automática

(Escrita en un lenguaje formal)

¿Cómo escribir las propiedades en un lenguaje formal?

Escribiendo propiedades

- Del comportamiento expresado en lenguaje natural a la propiedad escrita en un lenguaje formal hay un largo trecho.
- Haciendo una analogía con la vida

Lenguaje Natural

Lenguaje Formal

Lenguaje Natural

- ¿Es posible utilizarlo?

- Complica los procesamientos automáticos
- Especificaciones ambiguas, redundantes, etc.

Confusiones

- El lenguaje natural puede ser confuso

Messi erra su primer penal con la camiseta Argentina

1. Es el primer penal que erra en su vida, y tenía puesta la remera argentina
2. Había errado antes penales, pero el primero jugando para la selección

Confusiones

- El lenguaje natural puede ser confuso

Ví las montañas yendo a Bariloche

1. En el viaje a Bariloche vi montañas al costado del camino
2. Había montañas literalmente viajando a Bariloche…

Escribir en lenguajes formales

- Se trata de algo más complejo que sólo escribir…

1. **Escribir la propiedad en el lenguaje formal**

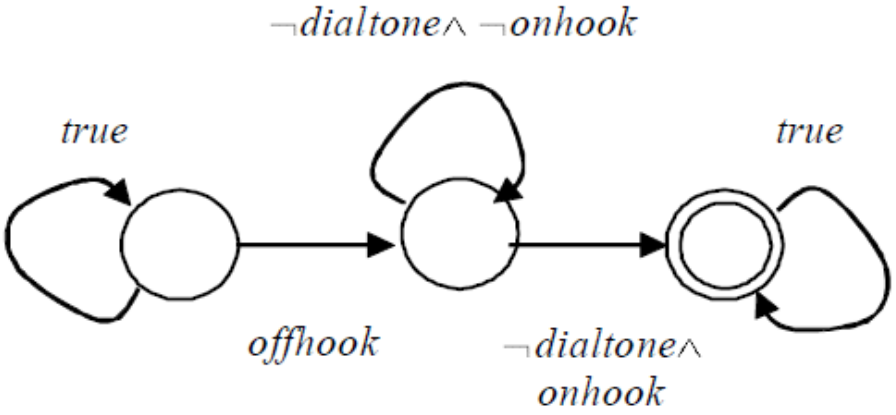
2. **Chequear que lo escrito se corresponda con lo que se desea expresar.**

Qué necesito

- Necesito **entender** la fórmula
- Especificaciones pequeñas, fáciles de comparar, modificar y manipular.
- Facilidades para razonar por comportamiento complementario

Lenguajes Formales

- Notaciones Operacionales: Autómatas



- **Notaciones Declarativas: Lógicas Temporales:**
  - **Ejemplo Fórmula LTL: □(S → ◊R)**

LTL: Linear Temporal Logics

- Predica sobre una traza de eventos
- Permite describir qué eventos ocurren y cuando
- Operadores:
  - Negación,
  - Implicación
  - AND
  - OR
- Modalidad:
  - Siempre
  - Eventualmente
  - Next
  - Until

Sintaxis y Semántica LTL

LTL

Syntax:

$\phi ::= P \mid \neg\phi \mid \phi \wedge \phi \mid F\phi \mid G\phi \mid \phi \; U \; \phi \mid O\phi$

Intuitive semantics:

- P: Propositional constant P holds now/at the current time instant
- $F\phi$: At *some future* time instant ϕ is true
- $G\phi$: *For all future* time instants ϕ is true
- $\phi \; U \; \psi$: ϕ is true *until* ψ becomes true
- $O\phi$: ϕ is true at the *next* time instant

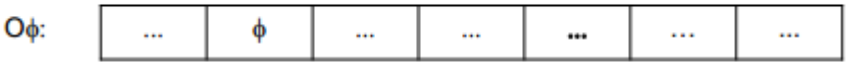
Gráficamente

- Eventualmente:

- Siempre:

- Until

- Next



Algunos ejemplos

- G (($\neg$ Pasaporte $\vee$ $\neg$ ticket) $\rightarrow$ O $\neg$ Abordar_Vuelo)
- G(Pedido $\rightarrow$ F recibido)
- G(recibido $\rightarrow$ O Procesado)
- Empieza_Charla $\rightarrow$ Hablar **U** finCharla
- Nacer $\rightarrow$ Vivir **U** Morir

Ejemplos

- "Entre que se llama al ascensor en un piso, y el ascensor llega, el mismo podrá pasar como máximo dos veces por ese piso"
- *En LTL:*

$$\Box((\text{call} \wedge \Diamond\text{open}) \rightarrow ((\neg\text{atfloor} \wedge \neg\text{open}) \; \mathcal{U} \; (\text{open} \vee ((\text{atfloor} \wedge \neg\text{open}) \; \mathcal{U} \; (\text{open} \vee ((\neg\text{atfloor} \wedge \neg\text{open}) \; \mathcal{U} \; (\text{open} \vee ((\text{atfloor} \wedge \neg\text{open}) \; \mathcal{U} \; (\text{open} \vee (\neg\text{atfloor} \; \mathcal{U} \; \text{open})))))))))$$

Difícil de:
Escribir
Entender
Modificar
Manipular

Ejemplo 2

- “When the subscriber picks up the phone, dial-tone is always generated”
- En LTL

$$\Box\,(offhook \rightarrow \Diamond\ dialtone)$$

- ¿Qué pasa si el usuario cuelga y vuelve a levantar el teléfono?
- Reescribo la propiedad:

$$\Diamond\,(\,offhook \rightarrow (\,\neg\ dialtone\ \mathbf{U}\ onhook\,)).$$

Entonces...?

- ¿Cumple $\Diamond\,(\,offhook \rightarrow (\,\neg\, dialtone\ \mathbf{U}\ onhook\,))$.

  lo esperado?

- La implicación lógica no es un operador temporal

  $$(p \rightarrow q) \equiv (\neg p \vee q)$$

- Luego, la fórmula anterior puede satisfacerse en una ejecución sin *offhook (ie, offhook es false)* o también si en una ejecución ocurre el evento *onhook*.

finalmente

- Y así hasta:

$$\Diamond\ (\ offhook \wedge \mathbf{X}\ (\ (\neg\ dialtone \wedge \neg\ onhook)\ \mathbf{U}\ (\neg\ dialtone \wedge onhook\ )).$$