Model checking

Ma. Laura Cobo

Métodos formales para Ingeniería de Software
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Argentina

Model checking

Las técnicas de verificación basada en modelos están basadas en:

Una descripción del comportamiento del sistema

De una manera precisa, matemática y no ambigua

Algoritmos

Exploran sistemáticamente todos los estados del modelo del sistema

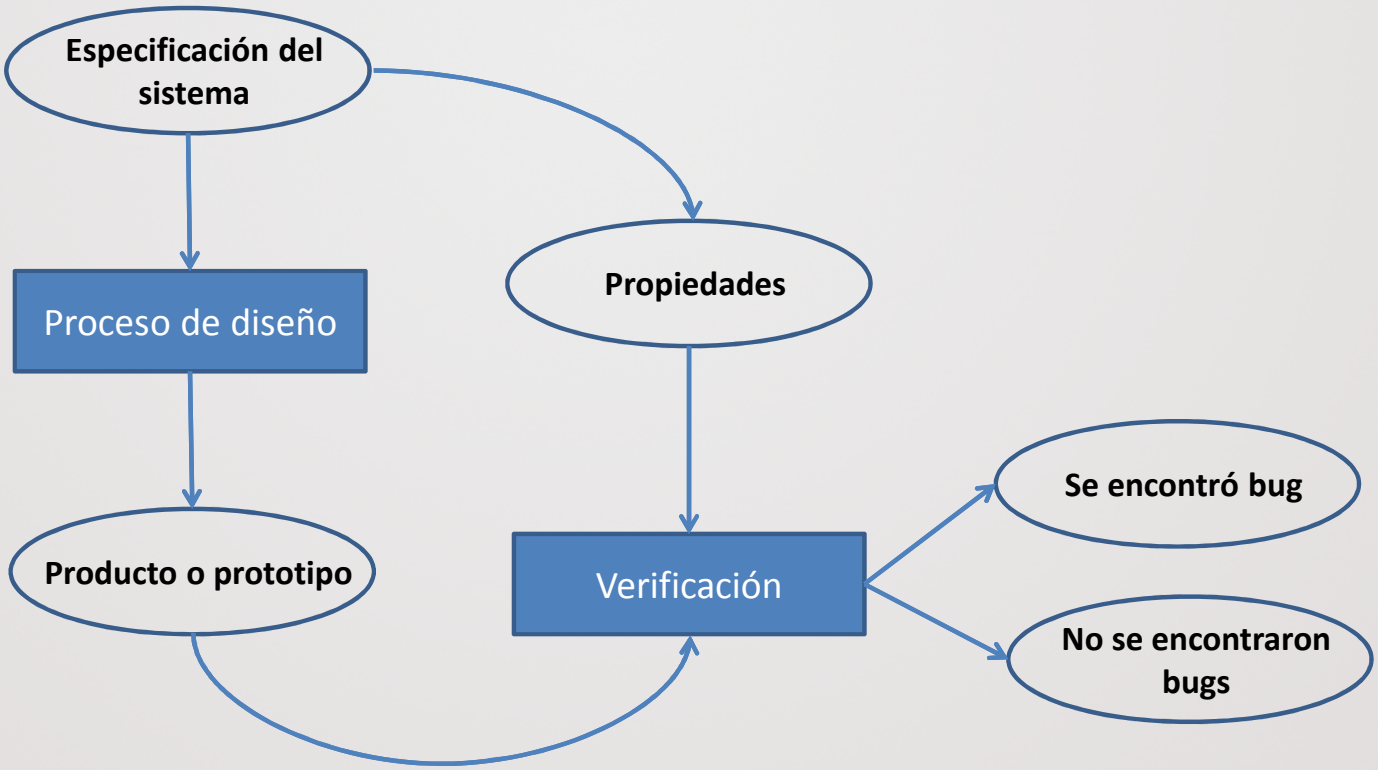
Generalmente, el modelo preciso del sistema, conduce a descubrir que el mismo es:

- Incompleto
- Ambiguo
- Inconsistente con la especificación informal del sistema

La exploración de los estados del modelo proveen las bases para el rango de técnicas de verificación:

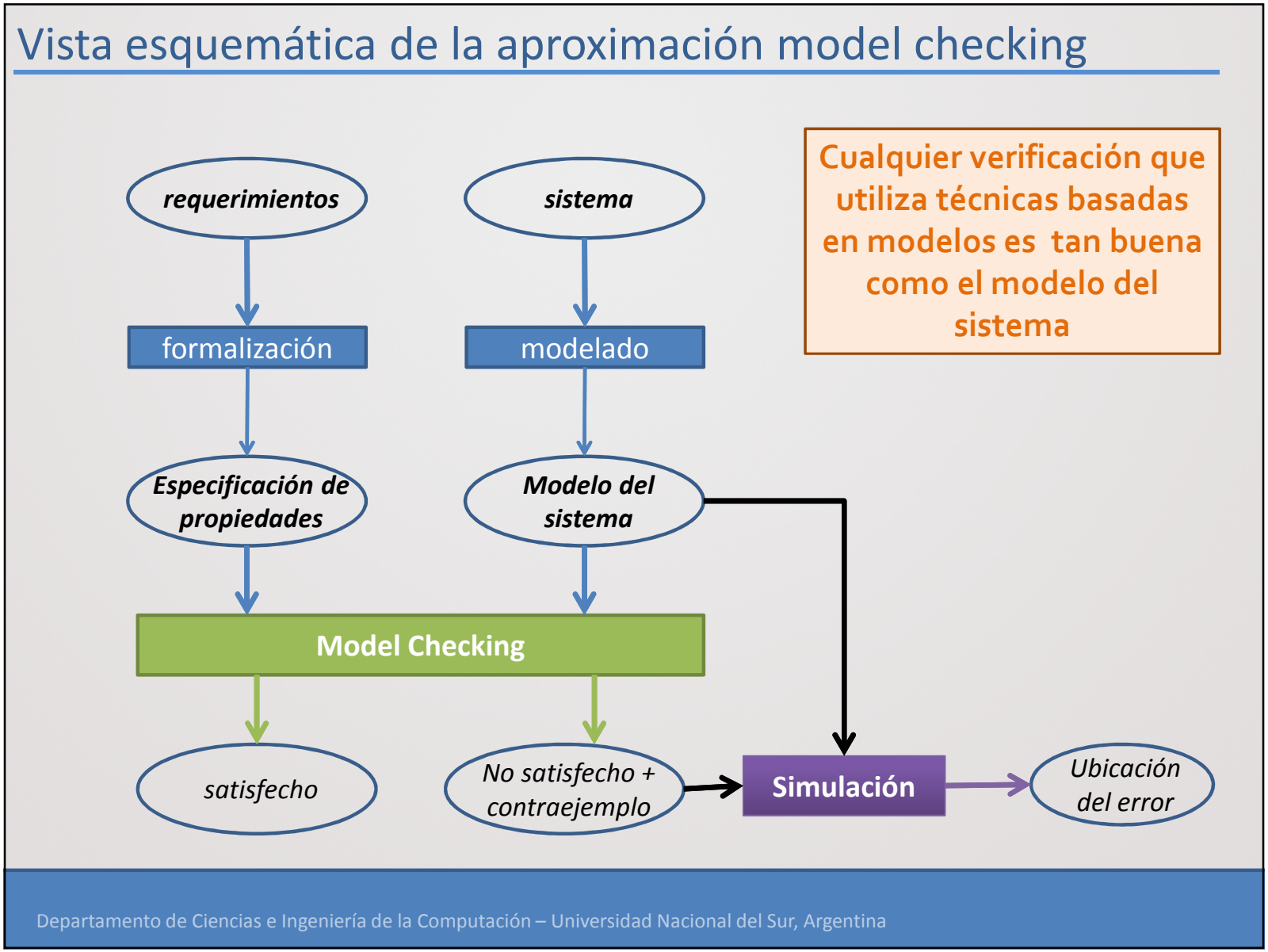
- Exhaustiva (Model Checking)
- Restringida a un conjunto de escenarios (Simulación)
- Real (Testing)

Esquema de un sistema de verificación general



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

Vista esquemática de la aproximación model checking



Características

Model checking es una técnica automatizada, dado un modelo en forma de autómata finito y una propiedad formal, chequea sistemáticamente si la propiedad de verificada para (un dado estado en) el modelo

Se pueden distinguir las siguientes fases:

- Fase de modelado
- Fase de ejecución o “running”
- Fase de análisis

Características

Model checking es una técnica automatizada, dado un modelo en forma de autómata finito y una propiedad formal, chequea sistemáticamente si la propiedad de verificada para (un dado estado en) el modelo

Se pueden distinguir las siguientes fases:

- Fase de modelado:
  - Modela el sistema bajo consideración usando el lenguaje de descripción para el mismo que provee el model checker a utilizar.
  - Se realizan algunas simulaciones para hacer un primer chequeo de sanidad
  - Formalizar la propiedad a chequear utilizando el lenguaje de especificación de propiedades

Características

Model checking es una técnica automatizada, dado un modelo en forma de autómata finito y una propiedad formal, chequea sistemáticamente si la propiedad de verificada para (un dado estado en) el modelo

Se pueden distinguir las siguientes fases:

- Fase de modelado:
  - Modela el sistema bajo consideración usando el lenguaje de descripción para el mismo que provee el model checker a utilizar.
  - Se realizan algunas simulaciones para hacer un primer chequeo de sanidad
  - Formalizar la propiedad a chequear utilizando el lenguaje de especificación de propiedades
- Fase de ejecución o "running": se ejecuta el model checker para asegurar la validez de la propiedad en el modelo del sistema

Características

Model checking es una técnica automatizada, dado un modelo en forma de autómata finito y una propiedad formal, chequea sistemáticamente si la propiedad de verificada para (un dado estado en) el modelo

Se pueden distinguir las siguientes fases:

- Fase de modelado
- Fase de corrida o "running
- Fase de análisis
  - ¿se satisfizo la propiedad? → chequear próxima (si hay)
  - ¿la propiedad no fue satisfecha? →
    1. Analizar el contraejemplo provisto por la simulación
    2. Refinar el modelo, diseño o propiedad
    3. Repetir el proceso en forma completa
  - ¿sin memoria? → tratar de reducir el modelo e intentarlo nuevamente

Características

Model checking es una técnica automatizada, dado un modelo en forma de autómata finito y una propiedad formal, chequea sistemáticamente si la propiedad de verificada para (un dado estado en) el modelo

Se pueden distinguir las siguientes fases:

- Fase de modelado
- Fase de corrida o "running
- Fase de análisis
- Organización de la verificación:

  El proceso de verificación debe ser
  1. Planificado
  2. Administrado y
  3. Organizado

Fortalezas

- Es una aproximación general a la verificación que es aplicable a un rango amplio de aplicaciones (incluyendo sistemas embebidos, ingeniería de software y diseño de hardware)
- Suporta verificación parcial (se verifica propiedad a propiedad)
- No es vulnerable a la probabilidad de exposición del error.
- Provee información de diagnóstico (particularmente útil para realizar el debugging)
- Es una potencial tecnología *push-button* (no requiere mucha interacción por parte del usuario ni nivel de experticia)
- Esta despertando del interés de la industria rápidamente
- Puede integrarse fácilmente en los ciclos de desarrollo existentes
- Posee una base sana y matemática (sustentado en algoritmos de teoría de grafos, estructuras de datos y lógica)

Debilidades

- Es más apropiado para aplicaciones centradas en control. Las aplicaciones centradas en datos trabajan típicamente sobre dominios infinitos
- Su aplicabilidad queda sujeta a aspectos de decidibilidad.
- Verifica un modelo del sistema, no el sistema en si mismo.
- No garantiza completitud, ya que no se puede afirmar nada sobre las propiedades no chequeadas
- Sufre del problema de "*explosión de memoria*". Los modelos mas reales son demasiados grandes para "caber".
- Requiere cierto grado de experiencia para encontrar abstracciones apropiadas.
- El model checker puede tener defectos de software (como cualquier software)
- No permite chequear generalizaciones (por ejemplo sistemas con tipos parametrizados)

Fortalezas

A pesar del hecho de creer que la garantía absoluta de correctitud es imposible de garantizar para un sistema de un tamaño real y de las limitaciones de la técnica se puede afirmar que:

Model checking es una técnica efectiva para exponer potenciales errores de diseño

Sistemas de transición

Se utilizan para describir el comportamiento de un sistema.

Básicamente son un grafo dirigido donde:

- los estados son *nodos* y
- las transiciones son *arcos* o eventos-acciones que producen cambios de estados

*El **estado** describe información sobre al sistema en cierto momento de su comportamiento o evolución*

*La **transición** especifica como el sistema puede evolucionar de un estado a otro*

Un ***sistema de transición (TS)*** es básicamente un autómata

Es finito si el conjunto de estado y acciones es finito

Un TS posee un **conjunto** de estados iniciales

Sistemas de transición

Un ***sistema de transición (TS)*** es una tupla: (S, Act, →, I, AP, L)

- ***S*** *es un conjunto de estados*
- ***Act*** *es un conjunto de acciones*
- → *es la relación de transición*
- **I** *es el conjunto de estados iniciales (⊆ S)*
- **AP** *es un conjunto de proposiciones atómicas*
- **L** *es una función de etiquetado (S → 2^{AP})*

OBSERVACIONES IMPORTANTES:

- La ejecución es netamente **no-determinista**, se elige de esta manera una de las transiciones de salida del estado actual.
- Si el conjunto I no es unitario, se elige uno de sus elementos en forma no-determinista para determinar el estado de inicio
- La ejecución culmina cuando se alcanza un estado que no tiene transiciones de salida.
- La función de etiquetado L(s) determina que conjunto de proposiciones de AP se satisfacen en el estado s.

Idea del model checker

¿ T |= φ ?

1. Representar el sistema T como un autómata Büchi, B_T, que acepta aquellas palabras que corresponden a ejecuciones de T
2. Construir un autómata Büchi para la negación de la fórmula φ, $B_{\neg\varphi}$
3. Si $L^{\omega}(B_T) \cap L^{\omega}(B_{\neg\varphi}) = \varnothing$
   entonces φ se verifica
   sino
   cada elemento del conjunto intersección es un contraejemplo para φ

Para verificar la condición se construye al autómata intersección y se busca un ciclo hacia el estado aceptador.

LTL Model Checking