

# Resumen

Esta tesis propone una metodología de análisis dinámico que mejora el diagnóstico de programas erróneos escritos en el lenguaje Maude. La idea clave es combinar técnicas de verificación de aserciones en tiempo de ejecución con la fragmentación dinámica de trazas de ejecución para detectar automáticamente errores en tiempo de ejecución, al tiempo que se reduce el tamaño y la complejidad de las trazas a analizar. En el caso de violarse una aserción, se infiere automáticamente el criterio de fragmentación, lo que facilita al usuario identificar rápidamente la fuente del error.

En primer lugar, la tesis formaliza una técnica destinada a detectar automáticamente eventuales desviaciones del comportamiento deseado del programa (síntomas de error). Esta técnica soporta dos tipos de aserciones definidas por el usuario: aserciones funcionales (que restringen llamadas a funciones deterministas) y aserciones de sistema (que especifican los invariantes de estado del sistema). La técnica de verificación dinámica propuesta es demostrablemente correcta en el sentido de que todos los errores señalados definitivamente delatan la violación de las aserciones. Tras eventuales violaciones de aserciones, se generan automáticamente trazas fragmentadas (es decir, trazas simplificadas pero igualmente precisas) que ayudan a identificar la causa del error. Además, la técnica también sugiere una posible reparación para las reglas implicadas en la generación de los estados erróneos.

La metodología propuesta se basa en (i) una notación lógica para especificar las aserciones que se imponen a la ejecución; (ii) una técnica de verificación aplicable en tiempo de ejecución que comprueba dinámicamente las aserciones; y (iii) un mecanismo basado en la generalización (ecuacional) menos general que automáticamente obtiene criterios precisos para fragmentar trazas de ejecución a partir de aserciones falsificadas.

Por último, se presenta una implementación de la técnica propuesta en la herramienta de análisis dinámico basado en aserciones ABETS, que muestra cómo es posible combinar el trazado de las propiedades asertadas del programa para obtener un algoritmo preciso de análisis de trazas que resulta útil para el diagnóstico y la depuración de programas.