

Resumen

Vivimos en una sociedad donde la digitalización está presente en nuestro día a día. Nos despertamos con la alarma de nuestro teléfono móvil, apuntamos nuestras reuniones en nuestro calendario digital, guardamos nuestros archivos en el almacenamiento en la nube, y entramos a las redes sociales prácticamente a diario. Cada una de estas acciones se ejecuta sobre un sistema software que asegura su correcto funcionamiento. Esta digitalización masiva ha hecho que el desarrollo de software se dispare en los últimos años. Durante el ciclo de vida de un sistema software, la etapa de mantenimiento supone un gasto de billones de dólares anuales. La razón detrás de este gasto es la aparición de bugs o errores que no fueron detectados durante la fase de producción del software, y que se traducen en un mal funcionamiento del sistema. Por este motivo, las técnicas de detección y localización de errores como el testeo, la verificación o la depuración son un factor clave para asegurar la calidad del software.

Aunque son muchas las técnicas que se utilizan para la depuración, testeo y verificación de sistemas software, esta tesis se centra solo en algunas de ellas. En concreto, esta tesis presenta mejoras en la técnica de fragmentación de programas (depuración), una nueva metodología para hacer testeo de regresión (testeo) y una nueva implementación del modelo de verificación de diseño-por-contrato para el lenguaje de programación Erlang (verificación).

Las mejoras propuestas para la técnica de fragmentación de programas incluyen diversas propuestas aplicables a diferentes escenarios: (i) mejoras en la representación y fragmentación de programas orientados a objetos, (ii) mejoras en la representación y fragmentación de estructuras de datos complejas (objetos, vectores, listas, tuplas, registros, etc.), (iii) un nuevo modelo de grafo basado en una representación más detallada de programas, aumentando la expresividad del grafo y generando fragmentos de código más precisos como resultado, y (iv) una nueva técnica para calcular fragmentos mínimos de un programa dado un conjunto específico de posibles valores de entrada.

Por otro lado, la nueva metodología para hacer testeo de regresión se denomina testeo de punto de interés, e introduce la posibilidad de comparar automáticamente el comportamiento de un punto cualquiera del código dadas dos versiones del mismo sistema software.

Por último, la tesis contiene la nueva implementación del modelo de verificación de programas diseño-por-contrato para el lenguaje de programación Erlang, donde se explican en detalle los nuevos tipos de contratos diseñados para las partes secuencial y concurrente de Erlang.

Todos los análisis presentados en esta tesis han sido formalmente definidos y su corrección ha sido probada, asegurando de esta manera que los resultados tendrán el grado de fiabilidad necesario para ser aplicados a sistemas reales.

Otra contribución de esta tesis es la implementación de dos herramientas de fragmentación de programas para dos lenguajes de programación diferentes (Java y Erlang), una herramienta para realizar testeo de punto de interés para el lenguaje de programación Erlang y un sistema para ejecutar verificación de diseño-por-contrato en Erlang. Es de destacar que todas las herramientas implementadas a lo largo del desarrollo de esta tesis son herramientas de código

abierto y públicamente accesibles, de manera que pueden ser usadas o extendidas por cualquier investigador interesado en este area.