
Resumen

Las técnicas de análisis de programas tienen una gran cantidad de aplicaciones en el mundo actual. Sin embargo, existen aún aspectos a mejorar dentro de las mismas que pueden ayudar a difundir más su uso. Esta tesis está dedicada a la mejora de dos aspectos del análisis de programas: la rigidez de sus técnicas y su complejo proceso de aprendizaje.

La rigidez de las técnicas de análisis de programas es debida a la gran complejidad de los algoritmos de análisis, que provocan que, sin un costoso entrenamiento y aprendizaje previos, los desarrolladores no puedan crear sus propios análisis para mejorar la calidad de sus programas. El análisis de programas declarativo tiene como objetivo reducir el esfuerzo en el diseño de implementación de análisis gracias al aumento del nivel de abstracción del lenguaje de especificación usado, siempre sin renunciar a ofrecer un método de ejecución del análisis comparable en términos de eficiencia a las implementaciones más tradicionales.

En esta tesis se mejora en dos aspectos la aproximación de análisis de programas JAVA basada en el lenguaje lógico de especificación Datalog. En primer lugar, se traducen especificaciones Datalog a *Sistemas de Ecuaciones Booleanas* con el fin de distribuir el cómputo de los análisis mejorando así sus tiempos de ejecución. Los *Sistemas de Ecuaciones Booleanas* están equipados con algoritmos eficientes y distribuidos para su evaluación y existen herramientas industriales que los implementan. En segundo lugar, se traducen especificaciones Datalog a teorías de *Lógica de Reescritura* con el fin de soportar la extensión del lenguaje de especificación Datalog de forma que puedan expresarse análisis más sofisticados, por ejemplo análisis que tengan en cuenta el uso de reflexión en los programas.

Otra contribución de esta tesis está relacionada con la automatización de la inferencia de especificaciones como mecanismo de apoyo a las técnicas que mejoran la calidad de los programas (análisis de programas, verificación, depuración, documentación, etc.). Podemos decir que las especificaciones son la base del análisis de programas en particular (y del resto de técnicas mencionadas en general) ya que cualquier análisis comprueba si el comportamiento del programa que está siendo analizado se corresponde con el dado por una especificación. Sin un entrenamiento previo, los desarrolladores pueden no ser capaces de formular especificaciones adecuadas que puedan ser usadas como

entrada de analizadores estáticos, herramientas de testing o verificadores de programas, o incluso como documentación o para ser estudiadas y analizadas de forma manual. La inferencia automática de especificaciones tiene como objetivo final reducir el esfuerzo necesario para escribir especificaciones de programas. Para ello computa especificaciones aproximadas sin necesidad de intervención del desarrollador. En esta tesis se mejora la inferencia automática de especificaciones para el lenguaje multiparadigma CURRY y para programas orientados a objetos en general proponiendo dos nuevas aproximaciones. Por un lado, se presenta una técnica para inferir especificaciones algebraicas para programas CURRY construyendo expresiones a partir de la signatura del programa y clasificándolas en función de la semántica asociada a dichas expresiones. En contraste con las aproximaciones existentes en la literatura, esta técnica permite distinguir entre partes de la especificación *correctas* y partes *posiblemente correctas*. Por otro lado, se presenta una técnica para inferir especificaciones de alto nivel presentadas en forma de pre/post-condiciones para lenguajes orientados a objetos. Esta técnica se formaliza en el contexto del marco de verificación de *Matching Logic* de forma que se habilita la posibilidad de verificar la especificación inferida.