FACULTAD DE INFORMÁTICA
UNLP

TESINA DE LICENCIATURA

TITULO: Lógica de Pruebas para Certificación de Computación Móvil

AUTORES: Federico Feller

DIRECTOR: Eduardo Bonelli

CODIRECTOR: Gabriel Baum

CARRERA: Licenciatura en Informática

Resumen

En este trabajo se presenta un modelo para computaciones móviles que incluye la generación de certificados al estilo PCC (proof carrying code). El modelo consiste en un lenguaje de programación recortado, un sistema de tipos y una semántica basada en una máquina abstracta. El cálculo es obtenido a partir de una técnica inspirada en el isomorfismo de Curry-DeBruijn-Howard, en donde las proposiciones y pruebas de una lógica son interpretadas como los tipos y términos de un lenguaje. En este caso la lógica elegida es ILPnd, una representación en deducción natural de la versión intuicionista de la lógica de pruebas LP. Estas lógicas son lógicas modales con la característica especial que contienen el operador modal de la forma [s]A, que se interpreta como “s es una prueba A”. La interpretación computacional de este operador es el de código móvil que computa un valor de tipo A con certificado s. A esta combinación de código y certificado se la denomina unidad móvil. A partir de la definición formal del cálculo se estudian un conjunto de propiedades sobre el mismo que incluyen seguridad de tipos y normalización fuerte. Adicionalmente, se presenta una implementación del cálculo en un lenguaje funcional.

Líneas de Investigación

* Computación móvil
* Proof carrying code
* Lambda Cálculo
* Lógicas Modales
* Lógica de Pruebas

Conclusiones

Se presentó un cálculo para modelar computaciones móviles que incluye la generación de certificados. La interpretación computacional que se le dio al constructor modal [s]A es la de una unidad móvil, una expresión que incluye tanto el código como el certificado. El sistema de tipos obtenido constituye una teoría unificada para la construcción correcta de tanto código como certificados. Cuando se construye una unidad móvil a partir de otras unidades móviles, el sistema de tipos no solo asegura que la nueva unidad móvil no dependa de recursos locales, sino que también verifica que el certificado de esta se construya a partir de los certificados de sus componentes. El hecho de haber definido el cálculo y su semántica de manera formal permitió hacer un estudio y demostración de sus propiedades más importantes. Se demostró la seguridad de tipos, que garantiza, entre otras cosas, que la ejecución de un programa bien tipado no puede fallar debido a una unidad móvil que tiene un certificado que no se corresponde a su código y normalización fuerte, que asegura que los programas bien tipados siempre terminan. Finalmente, se realizaron algunas extensiones a la definición original del cálculo que aportaron mayor riqueza a la posterior implementación del prototipo del lenguaje.

Trabajos Realizados

* Estudio de las lógicas de pruebas (LP, ILP y ILPnd).
* Definición del cálculo junto al sistema de tipos para construcción de certificados.
* Definición de la máquina abstracta.
* Demostraciones completas de las propiedades del cálculo.
* Implementación de un prototipo del cálculo en Haskell

Trabajos Futuros

* Inclusión del operador de posibilidad diamante, interpretando a ◊A como el valor de un término en un nodo remoto.
* Extender la definición del lenguaje para incluir referencias y recursión (utilizando el operador de punto fijo fix).
* Agregar polimorfismo sobre las variables de certificado y sobre las variables de función.

Fecha de la presentación: Agosto 2009