Resolución Consejo Directivo

Número: RESCD-2023-2131-E-UBA-DCT#FCEN

CIUDAD DE BUENOS AIRES
Viernes 29 de Septiembre de 2023

Referencia: EX-2023-03831534- -UBA-DMESA#FCEN - POSTGRADO - Sesión 25/09/2023

VISTO:

La nota presentada por la Dirección del Departamento de Computación, mediante la cual eleva la información del curso de posgrado **Temas de Posgrado: Behavioral Types para Lenguajes de Programación de uso Frecuente** para el año 2023**,**

CONSIDERANDO:

lo actuado por la Comisión de Doctorado,

lo actuado por este Cuerpo en la sesión realizada el día 25 de septiembre de 2023,

en uso de las atribuciones que le confiere el Artículo 113º del Estatuto Universitario,

EL CONSEJO DIRECTIVO DE LA FACULTAD

DE CIENCIAS EXACTAS Y NATURALES

R E S U E L V E:

ARTÍCULO 1º: Aprobar el nuevo curso de posgrado **Temas de Posgrado: Behavioral Types para Lenguajes de Programación de uso Frecuente** de 15 horas de duración, que será dictado por el Dr. Antonio Ravara con la colaboración del Dr. Sergio Abriola.

ARTÍCULO 2°: Aprobar el programa del curso de posgrado **Temas de Posgrado: Behavioral Types para Lenguajes de Programación de uso Frecuente** que como anexo forma parte de la presente Resolución, para su dictado en el período de invierno de 2023.

ARTÍCULO 3°: Aprobar un puntaje máximo de medio (0,5) puntos para la Carrera del Doctorado.

ARTÍCULO 4°: Establecer que el presente curso no será arancelado (**CATEGORÍA 1)** .

ARTÍCULO 5°: Disponer que, de no mediar modificaciones en el programa, la carga horaria y el arancel, el presente Curso de Posgrado tendrá una vigencia de cinco (5) años a partir de la fecha de la presente Resolución.

ARTÍCULO 6º: Comuníquese a todos los Departamentos Docentes, a la Dirección de Estudiantes y Graduados, a la Biblioteca de la FCEyN y a la Secretaría de Posgrado con copia del programa incluida. Cumplido, pase a COMPUTACION#FCEN y resérvese.

ANEXO

Programa:

Los lenguajes de programación modernos como Kotlin o Rust tienen funciones avanzadas para garantizar estáticamente la seguridad de los datos y la memoria, como los tipos anulables y de propiedad. Otros lenguajes como C, Erlang, Go, Haskell, Java o Scala tienen herramientas de análisis estático para ayudar al desarrollador a detectar posibles violaciones de seguridad en tiempo de compilación que podrían generar errores en tiempo de ejecución. Sin embargo, la mayoría de estos obligan al desarrollador a adoptar un estilo de programación defensivo, tratando de cubrir todos los escenarios posibles en los que las cosas podrían salir mal. Sin embargo, todavía suceden cosas malas, siendo un testimonio de esto el error Jedis descubierto después de 9 años en enero de 2018 (el problema se cerró 14 meses después), o la vulnerabilidad del contrato inteligente MonoX Finance explotada en diciembre de 2021. Dado que muchas aplicaciones implementan protocolos, como no todas las funcionalidades están disponibles todo el tiempo (por ejemplo, solo se puede extraer de una pila que no está vacía; escribir en un búfer no lleno), los tipos de comportamiento son una forma natural de modelar y validar el código que exige que solo las secuencias correctas de las acciones se permitan. Los sistemas de tipo estático aseguran el cumplimiento del protocolo y, en algunas condiciones, la finalización. Hay herramientas para usarlos con algunos lenguajes funcionales u orientados a objetos, pero no han sido todavía ampliamente adoptadas.

El objetivo de este curso es presentar los conceptos clave sobre los tipos de comportamiento, su aplicación a lenguajes de programación, y las principales herramientas disponibles. El recorrido irá desde la teoría hasta el apoyo lingüístico, proponiendo a los alumnos un proyecto a realizar en media jornada al final de la semana.

Temario

- Motivación: modelando entidades computacionales con máquinas de estados, bugs notorios por violaciones de protocolo, observación de que el testing no es suficiente.
- Type-safety: de qué trata, cómo conseguirla, estado de lenguajes conocidos como C, Go, Haskell, Java, Rust.
- Desde data-safety a protocol-safety: sistemas de transiciones etiquetadas (labelled transition systems, LTS), usando LTS para modelar entidades computacionales,

nociones de tipos de comportamiento (session types, contracts, typestates), tipos de comportamiento para lenguajes de programación convencionales.
- Sobre session types: sintaxis, semántica, subtipado, ejemplos; session types (ST) para lenguajes simples (funcionales y orientados a objetos).
- STs en lenguajes de programación convencionales: C, Erlang, Haskell, OCaml, Java, Scala, Rust.
- Multiparty ST: ejemplos y conceptos, herramientas MST.
- Mini-proyecto: un sistema de subasta.

Bibliografía

- Davide Ancona, Viviana Bono, Mario Bravetti, Joana Campos, Giuseppe Castagna, Pierre-Malo Deniélou, Simon J Gay, Nils Gesbert, Elena Giachino, Raymond Hu, Einar Broch Johnsen, Francisco Martins, Viviana Mascardi, Fabrizio Montesi, Rumyana Neykova, Nicholas Ng, Luca Padovani, Vasco T Vasconcelos, Nobuko Yoshida: *Behavioral types in programming languages*. Foundations and Trends in Programming Languages 3 (2-3), 95-230 (2016)
- Simon Gay and António Ravara (editors): *Behavioral Types: from Theory to Tools*. CRC Press (2017).
- Hans Hüttel, Ivan Lanese, Vasco T. Vasconcelos, Luís Caires, Marco Carbone, Pierre-Malo Deniélou, Dimitris Mostrous, Luca Padovani, António Ravara, Emilio Tuosto, Hugo Torres Vieira, Gianluigi Zavattaro: *Foundations of Session Types and Behavioural Contracts*. ACM Comput. Surv. 49(1): 3:1-3:36 (2016)

Digitally signed by MARTI Marcelo Adrian
Date: 2023.09.29 10:50:19 ART
Location: Ciudad Autónoma de Buenos Aires

Marcelo Marti
Secretario
Secretaría de Posgrado
Facultad de Ciencias Exactas y Naturales

Digitally signed by DURAN Guillermo Alfredo
Date: 2023.09.29 14:55:51 ART
Location: Ciudad Autónoma de Buenos Aires

Guillermo Alfredo Duran
Decano
Decanato
Facultad de Ciencias Exactas y Naturales

Digitally signed by GDE UBA
Date: 2023.09.29 14:55:52 -03:00