Resolución Consejo Directivo

Número: RESCD-2023-2127-E-UBA-DCT#FCEN

CIUDAD DE BUENOS AIRES
Viernes 29 de Septiembre de 2023

Referencia: EX-2023-03840101- -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: Fuzzing para Testing de Compiladores para el año 2023,

CONSIDERANDO:

lo actuado por la Comisión de Doctorado,

lo actuado por este Cuerpo en la sesión realizada en el día de la fecha 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: Fuzzing para Testing de Compiladores de 15 horas de duración, que será dictado por el Dr. Guillermo Polito con la colaboración del Dr. Sergio Abriola.

ARTÍCULO 2°: Aprobar el programa del curso de posgrado Temas de Posgrado: Fuzzing para Testing de Compiladores 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) punto 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:

El testing de compiladores es hoy en día un área crítica para el desarrollo de software profesional, dada la complejidad y constante evolución de los lenguajes de programación. El testing automático debe validar la interrelación entre nuevos lenguajes, nuevas representaciones intermedias de código, nuevas fases de optimización y transformación.

Dentro del área de testing automatizado de compiladores, las técnicas de fuzzing proponen la creación automática de forma más o menos aleatoria, de valores de input para testear un programa. Este curso presenta técnicas maduras de testing automático de compiladores. El curso se basa en tres ejes. En una primera parte, el curso presenta buenas prácticas de testing automatizado, aplicables no solo a compiladores.

En una segunda parte, el curso estudia técnicas para generar automáticamente inputs para compiladores, y cómo guiarlos para testear distintas partes de un compilador. En tercer lugar, el curso presenta el problema del oráculo de testing y cómo este problema se instancia en el caso de los compiladores.

En cada uno de esos ejes se estudian técnicas del estado del arte, acompañados de casos de estudio y artículos de investigación que ilustran los distintos casos. El curso es teórico y práctico. La práctica tiene como objetivo la implementación de fuzzers automáticos y ponerlos en práctica para testear lenguajes de programación.

Temario

1. Testing automatizado. Buenas prácticas y metodologías de testing.

2. Generación automática de datos de testing con fuzzing. Fuzzing aleatorio y guiado. Fuzzing guiado por gramáticas.

3. Fuzzing concólico. Fuzzing como mutación de inputs existentes.

4. Mutaciones guiadas por la cobertura de código. Oráculos de test y su automatización. Programas auto-validados.

5. Testing basado en propiedades. Testing diferencial. Testing metamórfico.

6. Testing de compiladores. Generación de programas y variación de programas existentes. Testing de subcomponentes de compilador. Testing entre compiladores.

Bibliografía

1. A. Zeller, R. Gopinath, M. Böhme, G. Fraser and C. Holler. (2021). The Fuzzing Book. CISPA Helmholtz Center for Information Security. https://www.fuzzingbook.org/.

2. Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., & Yoo, S. (2014). The oracle problem in software testing: A survey. IEEE transactions on software engineering, 41(5), 507-525.

3. Claessen, K., & Hughes, J. (2000, September). QuickCheck: a lightweight tool for random testing of Haskell programs. In Proceedings of the fifth ACM SIGPLAN international conference on Functional programming (pp. 268-279).

4. Polito, G., Tesone, P., Ducasse, S., Fabresse, L., Rogliano, T., Misse-Chanabier, P., & Hernandez Phillips, C. (2021, September). Cross-ISA testing of the Pharo VM: lessons learned while porting to ARMv8. In Proceedings of the 18th ACM SIGPLAN International Conference on Managed Programming Languages and Runtimes (pp. 16-25).

5. Polito, G., Ducasse, S., & Tesone, P. (2022, June). Interpreter-guided differential JIT compiler unit testing. In Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation (pp. 981-992).

6. Nagy, S., & Hicks, M. (2019, May). Full-speed fuzzing: Reducing fuzzing overhead through coverage-guided tracing. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 787-802). IEEE.

Digitally signed by MARTI Marcelo Adrian
Date: 2023.09.29 10:50:48 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 14:55:12 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 14:55:14 -03:00