

Información académica

Año de presentación (*)

2020

1-a-

Departamento docente que inicia el tramite:
Departamento de Computación
Nombre del curso:
Tipos comportamentales
Nombre, Cargo y Título del docente responsable:
Hernán Melgratti, Profesor Adjunto
En caso de dictarse en paralelo con una materia de grado, nombre de la misma:
Tipos comportamentales y Contratos
Nombre y Título de los docentes que colaboran con el dictado del curso (*) (*):
--
Fecha propuesta para el primer dictado luego de la aprobación:
Primer cuatrimestre 2020

Duración:

Duración total en horas	64
Duración en semanas	16

Distribución carga horaria:

Número de horas de clases teóricas	34
Número de horas de clases de problemas	30
Número de horas de trabajos de laboratorio	
Número de horas de trabajo de campo	
Número de horas de seminarios	

Forma de evaluación:

Trabajo práctico final a entregar en forma posterior al cursado

Lugar propuesto para el dictado (departamento, laboratorio, campo, etc.):

Departamento de Computación

Puntaje propuesto para la carrera de doctorado:

3

Número de alumnos:

Mínimo: 5

Máximo: 20

Audiencia a quien está dirigido el curso:

1-b-

Necesidades materiales del curso:

Programa analítico del curso con Bibliografía (puede adjuntarse en hojas separadas):

Cubriremos los siguientes temas:

1- Enfoque clásico de diseño por contrato (DbC). Principios del enfoque a la Meyer. Pre y post condiciones. Contratos a runtime y asignación de responsabilidades en la violación de contratos.

2- Cálculo lambda. Limitaciones del enfoque clásico para verificación de contratos en lenguajes de alto orden. Ejemplos. Lenguaje PCF

(Programming Computable Functions): Sintaxis y semántica operacional. Contratos como funciones PCF. Monitoreo dinámico de contratos en PCF. Enfoque a la Findler & Felleisen. PCF + Contracts : Sintaxis y semántica operacional. Contratos dependientes.

3- Fundamentos de Tipos Sesión binarios.

Motivaciones. Introducción informal. Modelado de una función. Modelado de objetos (Typestates). Sintaxis de Tipos Session Finitos (send, receive, branching, selection) con tipado explícito de sesiones. Ejemplos. Dualidad. Lenguaje de Procesos: Sintaxis y semántica operacional. Ligadores y variables libres. Equivalencia estructural. Contexto de tipado y linealidad. Juicios de tipado. Propiedades: Fidelidad y Progreso. Deadlocks. Deadlock-freedom por diseño (relación con lógica lineal). Tipos de sesiones infinitos (recursividad). Sintaxis. Dualidad (ingenua, definición clásica). Problema con la definición ingenua. Recursion en procesos: Sintaxis y semántica operacional. Tipos canales de uso ilimitados (canales a la pi-calculus). Regla de tipado para el caso recursor. Propiedades del tipado.

Dr. PABLO J. GROISMAN
Secretario Adjunto de Posgrado
FCEyN - UBA

Dr. JUAN CARLOS REBOREDA
DECANO

S
u
bt
ip
a
d
o.
V
ar
ia
n
z
a
y
c
o
v
ar
ia
n
z
a
e
n
ti
p
o
s
e
si
ó
n.
S
u
bt
ip
a
d
o
d
e
ti
p
o
s
re
c
u
rs
iv

os. Noción de simulación de tipos. Definición conductiva de subtipado y dualidad. Sustitutividad. Implementación de tipos session binarios en lenguajes de programación. API GV para sesiones binarias. Definición de GV en Ocaml. Dualidad y polimorfismo paramétrico. Primitivas para la creación de sesiones. Variantes y Variantes polimórficas en OCaml. Representación equirecursiva de tipos infinitos. Representación de tipos de sesiones à la Dardha, Giacchino & Sangiorgi. Dualidad como unificación. FuSe: Interface de una librería para comunicación con tipos sesión en OCaml. Implementación de primitivas de FuSe.

Verificación dinámica de linearidad. Subtipado de tipos session como subtipado de variantes polimórficas.

4- Contratos para tipos de sesión binarios. FuSe con servicios (canales compartidos).

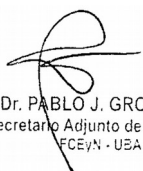
DSL para contratos de sesión en OCaml. Contratos sobre payloads. Contratos dependientes. Contratos sobre la estructura de ellas sesiones (choices). Asignación de culpa en la violación de un contrato.

LambdaCOS: sintaxis y semántica operacional. Tipado. Propiedades del sistema de tipo. Contract entailment. Locally correctness. Blame safety. Extensión de FuSe con contratos. Contratos como GADTs en OCaml. Implementación del monitoreo de contratos en ejecución.

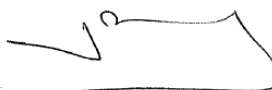
5- Tipos de sesión multiparte (MST). Grafos Globales (Coreografías). Sintaxis formal de MST finitos y de primer orden. Coherencia (o noción de bien-formado). Linearidad. Conocimientos de las elecciones. Tipos Locales. Proyecciones. Procesos: Sintaxis y semántica operacional. Tipado. Semántica de tipos Globales. Propiedades.

6- Contratos para MST. Aserciones sobre interacciones. Lenguaje de las aserciones globales.

Coherencia en aserciones globales. Consistencia de las aserciones: sensibilidad a la historia, localidad y satisfactibilidad temporal. Verificación de consistencia, sistema de tipado para sensibilidad a la historia y satisfactibilidad de formula booleana. Tipos locales con aserciones. Proyección de dependencias causales.



Dr. PABLO J. GROISMAN
Secretario Adjunto de Posgrado
FCEyN - UBA



Dr. JUAN CARLOS REBORADA
DECANO

7- Tipos sesión dependientes. Introducción a tipos dependientes (de Idris): tipos función dependiente (Producto dependiente, tipo-Pi) y pares dependientes (suma dependiente, tipo-Sigma). Tipos Label- Dependent. Tipos session Label-Dependent. Duality. Tipado. Extensión para tipos Label-Dependent recursivos. Lenguaje de procesos con aserciones. Semántica operacional con monitoreo runtime de aserciones. Sistema de tipos. Garantía de no violación de aserciones.

Bibliografía:

- Mitchell, Richard, and McKim, Jim: Design by Contract: by example, Addison-Wesley, 2002
- Hüttel, Hans, et al. Foundations of session types and behavioural contracts. ACM Computing Surveys (CSUR), 2016, vol. 49, no 1, p. 3.
- Simon J. Gay, Malcolm Hole: Subtyping for session types in the pi calculus. Acta Inf. 42(2-3): 191-225 (2005)
- Vasco T. Vasconcelos: Fundamentals of session types. Inf. Comput. 217: 52-70 (2012)
- Luca Padovani: A simple library implementation of binary sessions. J. Funct. Program. 27: e4 (2017)
- Ornela Dardha, Elena Giachino, Davide Sangiorgi: Session types revisited. Inf. Comput. 256: 253-86 (2017)
- Findler, R. B., & Felleisen, M. (2002, October). Contracts for higher-order functions. In ACM SIGPLAN Notices (Vol. 37, No. 9, pp. 48-59). ACM.
- Melgratti, H., & Padovani, L. (2017). Chaperone contracts for higher-order sessions. Proceedings of the ACM on Programming Languages, 1(ICFP), 35.
- Gommerstadt, H., Jia, L., & Pfenning, F. (2018, April). Session-typed concurrent contracts. In European Symposium on Programming (pp. 771-798). Springer.
- Honda, K., Yoshida, N., & Carbone, M. (2008). Multiparty asynchronous session types. ACM SIGPLAN Notices, 43(1), 273-284.
- Bocchi, L., Honda, K., Tuosto, E., & Yoshida, N. (2010, August). A theory of design-by-contract for distributed multiparty interactions. In International Conference on Concurrency Theory (pp. 162-176). Springer, Berlin, Heidelberg.
- Toninho, B., Caires, L., & Pfenning, F. (2011, July). Dependent session types via intuitionistic linear type theory. In Proceedings of the 13th international ACM SIGPLAN symposium on Principles and practices of declarative programming (pp. 161-172). ACM.

1-c-

Actividades prácticas propuestas (puede adjuntarse en hojas separadas):

Los alumnos deberán analizar y presentar artículos recientes relacionados con las problemáticas abordadas en la materia, en particular, artículos sobre implementaciones concretas de las técnicas estudiadas, como por ejemplo:

- Imai, K., Yoshida, N., & Yuen, S. (2019). Session-ocaml: a session-based library with polarities and lenses. Science of Computer Programming, 172, 135-159.
- Jespersen, T. B. L., Munksgaard, P., & Larsen, K. F. (2015, August). Session types for Rust. In Proceedings of the 11th acm sigplan workshop on generic programming (pp. 13-22).
- Lindley, S., & Morris, J. G. (2016). Embedding session types in Haskell. ACM SIGPLAN Notices, 51(12), 133-145.
- Kokke, W. (2019). Rusty Variation: Deadlock-free Sessions with Failure in Rust. arXiv preprint arXiv: 1909.05970.
- Orchard, D., & Yoshida, N. (2017). Session types with linearity in Haskell. Behavioural Types: from Theory to Tools, 219.
- Lindley, S., & Morris, J. G. (2017). Lightweight functional session types. Behavioural Types: from Theory to Tools, 265-286.

brary implementation of binary sessions. *Journal of Functional Programming*, 27. (Monadic implementation)

Scalas, A., & Yoshida, N. (2016). Lightweight session programming in scala.

Yoshida, N., Hu, R., Neykova, R., & Ng, N. (2013, August). The Scribble protocol language. In *International Symposium on Trustworthy Global Computing* (pp. 22-41). Springer, Cham.

Fowler, S. (2016). An Erlang implementation of multiparty session actors. arXiv preprint arXiv:1608.03321.

Scalas, A., Dardha, O., Hu, R., & Yoshida, N. (2017). A linear decomposition of multiparty sessions for safe distributed programming (artifact). In *DARTS-Dagstuhl Artifacts Series (Vol. 3)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

de Muijnck-Hughes, J., Brady, E., & Vanderbauwhede, W. (2019). Value-Dependent Session Design in a Dependently Typed Language. *Places 2019*. arXiv preprint arXiv:1904.01288.

Imai, K., Neykova, R., Yoshida, N., & Yuen, S. (2020). Multiparty Session Programming with Global Protocol Combinators. *ECOOP 2020*. arXiv preprint arXiv:2005.06333.

Lange, J., Tuosto, E., & Yoshida, N. (2015, January). From communicating machines to graphical choreographies. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (pp. 221-232).

Neykova, R., & Yoshida, N. (2014, June). Multiparty session actors. In *International Conference on Coordination Languages and Models* (pp. 131-146). Springer, Berlin, Heidelberg.

Miu, A., Ferreira, F., Yoshida, N., & Zhou, F. (2020). Generating Interactive WebSocket Applications in TypeScript. arXiv preprint arXiv:2004.01321.

Gabet, J., & Yoshida, N. (2020). Static Race Detection and Mutex Safety and Liveness for Go Programs (extended version). arXiv preprint arXiv:2004.12859.

(*) Todos los cursos tendrán una validez de 5 años

(*)(*) Las actualizaciones de los docentes colaboradores son informados por la Dirección departamental al inicio de cada dictado del curso

Firma Subcomisión Doctorado

Firma del docente responsable

A handwritten signature in black ink, appearing to read 'Hmelgra', is written inside a rectangular box.

E-mail y teléfono del docente responsable

hmelgra@dc.uba.ar, +54 11 4576 3359