

Doctorados



Miguel Campusano

Título tesis: Mapping State Machines to Developers' Mental Model: Fast Understanding of Robotic Behaviors in the Real World

Profesores guías: Alexandre Bergel - Johan Fabry

Cuando hice mi pregrado nunca realicé ningún tipo de investigación ni se me pasó por la mente hacer un doctorado. Sin embargo, al tiempo de trabajar en un emprendimiento, me terminé aburriendo ya que los temas que abordábamos me parecían poco motivantes. Mi plan era volver donde hice mi pregrado, al Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile, pero para hacer un magíster. Hablé con el profesor Johan Fabry, el mismo que me guió en mi pregrado, y, debido a diversos problemas que tenía en ese momento, me recomendó hacer un doctorado con un tema que era increíble para mí en ese entonces: robótica. Debo admitir que mi mente se iluminó de inmediato, en el DCC nunca tuve contactos con robots.

Mi paso por el doctorado fue, por decirlo de alguna forma, complejo. Al comienzo todo era emocionante, eso es lo que pasa cuando uno aprende algo totalmente nuevo y fascinante (recordemos que nunca tuve un paso por investigación antes de eso). Luego, el camino se volvió bastante tortuoso, no fue fácil para mí encarar el mundo de la academia e investigación, y mi salud mental se vio afectada bastante (al parecer un tema más que conocido en este mundo y, por alguna razón, tabú). Sin embargo, aprendí a golpes a cómo llevar este proceso y, finalmente, supe llevar mi proyecto de investigación. Además, justo en medio de mi doctorado Johan tuvo que dejar la Universidad y ahí Alexandre Bergel me tomó bajo su tutela. No fue un proceso fácil, pero le agradezco enormemente a Johan y Alex el ayudarme en esta carrera y darme la confianza que necesitaba para llevar el doctorado. Claramente, como estudiantes, nos faltan grupos de ayuda para que podamos llevar esta carrera de forma saludable.

Mi tema de doctorado consistió en desarrollar un lenguaje de programación para comportamientos robóticos con una característica en particular, el robot se mueve al mismo tiempo que se está programando. A esto se le conoce como programación en vivo. El objetivo de este lenguaje es hacer más fácil el desarrollo de comportamientos robóticos. Este tipo de unión entre



una disciplina netamente ligada a computación (programación en vivo) con la robótica fue algo novedoso en su tiempo y que, afortunadamente, he visto como va aumentando en popularidad, con workshops y conferencias dedicadas sólo a unir la computación y la robótica. Me alegra ver que un tema tan importante como éste, que muchas veces es dejado de lado, esté siendo tomado en cuenta y mucha gente le esté dedicando el tiempo que merece.

Aunque mi trabajo con este lenguaje de programación me enseñó mucho sobre el proceso de programar robots, al evaluarlo no pudimos comprobar nuestra hipótesis, no podemos afirmar que el lenguaje facilita, de alguna forma, el desarrollo de comportamientos robóticos. Aun así creo que vamos en la dirección correcta, programar un robot requiere la integración de diferentes disciplinas, todas sumamente complejas. No sólo se van a producir robots más complejos a través de mejorar la inteligencia artificial, algoritmos de control, visión computacional, etc., sino también es importante ayudar a que los programas robóticos sean más fáciles de escribir y de integrar al robot mismo, y con más capacidades. Todo esto para hacer comportamientos robóticos cada vez más complejos y útiles para la sociedad.

El tema de la robótica me lleva hoy en día a investigar y diseñar arquitecturas para drones, haciendo un postdoctorado en la Universidad del Sur de Dinamarca (SDU), en el marco del proyecto HealthDrone. En este proyecto queremos transportar medicinas y otros artículos médicos entre diferentes hospitales y centros médicos que pueden estar ubicados en zonas de difícil acceso, incluyendo islas donde sólo se puede llegar en barco. La idea es usar drones para reducir el costo y el tiempo de traslado de estos artículos médicos.

Aunque estoy en Dinamarca no me he desligado del mundo de la robótica en Chile. Hemos iniciado (con otras personas ligadas a la robótica) una corporación sin fines de lucro llamada Cuac. Con esta corporación trabajamos para potenciar la robótica y su educación en Chile.

Matías Toro

Título tesis: Abstracting Gradual Typing: Metatheory and Applications

Profesor guía: Éric Tanter

Egresé de ingeniero civil en computación de la Universidad de Chile el año 2007, y luego me dediqué a trabajar en la industria aeronáutica por siete años. El trabajo, a pesar de no ser trivial, resultaba monótono y a veces tedioso. No me imaginaba todo el resto de mi vida haciendo lo mismo. Buscando nuevos desafíos, y dado que por temas familiares me complica salir al extranjero, el año 2013 volví al Departamento de Ciencias de la Computación (DCC) para realizar el Magíster en Ciencias, mención Computación. Es ahí donde conocí a mi profesor guía Éric Tanter, el cual me reintrodujo al área de lenguajes de programación. Me gradué del magíster en el 2013, y en el 2014 se hizo natural extender mi trabajo hacia un doctorado.



Mi tesis de doctorado se llama "Abstracting Gradual Typing: Metatheory and Applications", y la investigación se centró en los lenguajes de programación graduales, los cuales buscan integrar sistemas de tipos estáticos (como el de Java) con sistemas de tipos dinámicos (como el de Python). Con sistemas de tipos graduales el programador puede escoger qué expresiones anotar con información de tipos estática, y cuáles dejar sin especificar. El sistema de tipos gradual chequea en tiempo de ejecución lo que no puede verificar durante la etapa de compilación, asegurando así que no se violen las anotaciones estáticas.

El enfoque clásico para diseñar lenguajes graduales es usualmente *ad-hoc*, pero existen metodologías que sistematizan este proceso. Una de ellas es *Abstracting Gradual Typing* (AGT), que ayuda a construir sistemáticamente lenguajes graduales a partir de lenguajes estáticamente tipados usando interpretación abstracta. Mi trabajo de investigación exploró esta (casi nueva en ese entonces) metodología, aplicando AGT a distintas disciplinas de tipo y mecanismos de lenguajes complejos.

La mayor parte de mi investigación fue teórica y se puede resumir en lo siguiente. Se partía de un sistema de tipos complejo existente, que satisfacía cierta propiedad formal, se aplicaba sistemáticamente AGT, y luego se observaba si el lenguaje gradual resultante también cumplía o no con dicha propiedad. Éste no fue el caso de todos los lenguajes estudiados,

por lo que se tenía que ir modificando ciertas abstracciones y reglas de evaluación, para que pudiera cumplir con la propiedad, sin perder otras propiedades intrínsecas a los lenguajes graduales. Estas iteraciones conllevaron muchas demostraciones matemáticas distintas, las que consumieron la mayor parte del tiempo de mi doctorado.

Toda esta experiencia fue una montaña rusa de emociones, donde uno se esperaba de tener una nueva idea o solución que luego se derrumbaba al encontrar algún problema en alguna demostración de algún lema. Trabajar por meses en una demostración matemática para luego ver que había un error (a veces a días antes del *deadline* de una conferencia), sumada a la presión de terminar a tiempo el doctorado fue muy estresante. Llegué a soñar con demostraciones (y hasta encontré algunos errores en demostraciones así).

La otra parte difícil fue la de escribir *papers*, ya que para hacer investigación no sólo sirve ser bueno técnicamente, sino que también se debe saber transmitir las ideas. Aprendí que escribir un *paper* puede ser muy parecido a desarrollar un software: no es recomendable partir ciegamente, sino que hay que darle estructura a las ideas y planificar cómo se van a presentar las cosas de manera de que todo fluya. También aprendí a usar otro tipo de inglés empleado en artículos científicos, que es distinto al que uno podría estar acostumbrado a leer.

Cuando hice el doctorado no había un curso que te enseñara todo esto y lo que aprendí, lo aprendí de Éric. Sigo sintiendo que es mi punto débil y que me falta mucho por aprender aún. Relacionado con esto, también tuve que aprender a presentar artículos científicos. Muchas veces gastaba semanas preparando e iterando una presentación. A pesar de todas las dificultades, es muy gratificante finalmente llegar a publicar un *paper* y presentarlo. ¡Vale la pena el esfuerzo!

Actualmente me encuentro haciendo un postdoctorado en el DCC continuando mis temas de investigación pero en temas relacionados con privacidad diferencial en lenguajes de programación.

Mauricio Quezada

Título tesis: Knowledge Discovery from News Events on Twitter

Profesora guía: Bárbara Poblete

Estudié Ingeniería Civil en Computación en el Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile. Mi gusto por los distintos temas que vi en los cursos que tuve durante la ingeniería, más la buena relación que he tenido con algunos profesores del Departamento me llevaron a continuar con un magíster, y luego, con el doctorado.

Mi tesis de doctorado consistió en una exploración de distintas formas de extraer conocimiento desde la información que comparten los usuarios de Twitter sobre eventos noticiosos. Estas formas de extracción se basan en la suposición de que el contexto en que se publica esta información es muy importante para agrupar contenido similar. Por ejemplo, uno de los trabajos consistió en representar los tuits que expresan algún comentario relacionado a un evento noticioso en particular, como la muerte de Nelson Mandela en 2013, como la diferencia de tiempo en que fueron publicados dos mensajes consecutivos. Esta simple representación nos permitió observar que cierto tipo de noticias generan mayor actividad de los usuarios, y que los mensajes que publican en este tipo de noticias son muy distintos a los mensajes que publican sobre noticias con menores niveles de actividad. Otro aspecto importante es que un mensaje individual no dice mucho sobre la noticia, pero el considerar una gran cantidad de ellos nos permite observar patrones interesantes. Este trabajo lo realicé durante mi tesis de magíster —y luego profundizado durante el doctorado—, en conjunto con Janani Kalyanam y Gert Lanckriet, en ese entonces de la Universidad de California, San Diego.



Una de las cosas más complicadas que enfrentamos durante el desarrollo de mi tesis fue la falta de conjuntos de datos “correctos” sobre los cuales hubiéramos podido evaluar nuestros modelos. Debido a la gran variabilidad de la información (una misma noticia no ocurre dos veces) y la naturaleza de los modelos que propusimos (orientados a resolver nuevas tareas en la minería de datos) nos exigió pensar en formas novedosas y válidas de evaluar la metodología. En pocas palabras, la evaluación consistió en identificar que nuestros modelos hacían resaltar patrones interesantes en otros aspectos de los datos.

Lo más desafiante del doctorado fue poder gestionar mi tiempo y definir bien los objetivos de cada etapa. Siempre aparecían nuevas ideas o cosas interesantes en las que trabajar, por lo que definir bien el plan —y uno no sabe qué va a encontrar al final— fue complicado. Por otro lado, creo que lo más interesante ha sido poder desarrollar distintas habilidades con el tiempo; simplemente el tener la experiencia de trabajar en investigación va generando nuevas capacidades que uno empieza a notar hacia el final del doctorado. También el poder hacer clases en distintas instancias fue muy gratificante, aunque estresante, ya que tuve la oportunidad de transmitir lo que he ido aprendiendo.

Decidí no seguir una carrera académica por varios motivos. Actualmente soy cofundador y CTO de Cero.ai, una empresa que automatiza procesos de comunicación entre empresas y personas.

Daniel Hernández

Título tesis: The Problem of Incomplete Data in SPARQL
Profesor guía: Claudio Gutiérrez

Cuando estaba el colegio, aún sin decidir que estudiaría, mi interés era estudiar algo que me permitiera poder entender el mundo, y por ello pensaba que cualquier carrera que tuviera que ver con ciencias me podría gustar. Escogí entrar a la Escuela de Ingeniería de la Universidad de Chile porque tenía un Plan Común que conducía a muchas carreras, lo que me permitiría más tarde decidir qué estudiar. Al final me decanté por computación, pues me gustaba y a la vez sentía que era una ciencia bastante general. Luego de hacer un magíster y un año de trabajar haciendo clases en la Universidad de Talca, volví al Departamento de Ciencias de la Computación (DCC) donde había estudiado, esta vez para hacer un doctorado. Me decidí a ello porque disfruté el año que trabajé haciendo clases y también por la recomendación de Claudio Gutiérrez (mi profesor guía).

Lo más complejo de mi doctorado fue sin duda el proceso de maduración que se produce cuando uno pasa de intentar resolver un problema a entender cuál es el problema que uno está resolviendo, y el impacto que puede tener lo que uno está haciendo. Este proceso va acompañado con lo difícil que resulta comunicar los resultados de la investigación, escribirlo de manera clara y siguiendo las prácticas de otros investigadores. Como he terminado mi doctorado hace poco tiempo, puedo recordar el camino que he seguido y percibir el cambio que se produce en este proceso de maduración. Lo que hoy me resulta evidente, antes no lo era.

Lo otro que requiere esfuerzo es mantenerse focalizado. Para investigar uno tiene que simplificar al máximo el problema abordado. Quitarle todos sus aspectos no esenciales hasta que el problema sea lo suficientemente claro como para poder enfrentarlo y luego poder comunicarlo. Hacer esto no es fácil. Al simplificar un problema uno termina generando una larga lista de variantes y preguntas sin resolver, para retomar algún día. También resulta un poco desalentador estar enfocado en un problema que se hace cada vez más pequeño al lado de la larga lista que voy dejando al



lado, va un poco contra la motivación inicial de comprender el mundo. Esto produce la sensación de que uno cada vez sabe menos. Por suerte, a lo largo del doctorado me hice consciente de este fenómeno, lo que ahora me ayuda a lidiar con ello.

Mi relación con Claudio fue siempre muy buena y puedo decir que aprender de su mirada general fue una de las cosas más positivas del doctorado. También tengo que agradecer a Aidan Hogan y Renzo Angles, de quienes también aprendí sus diferentes miradas cuando trabajábamos en algún *paper*. El ambiente del DCC es muy bueno para hacer un doctorado, porque tiene una comunidad amable con la cual compartir y reflexionar, y gente muy admirable.

Mi tesis de doctorado “The Problem of Incomplete Data in SPARQL”, estudia cómo las nociones de la información incompleta se manifiestan en el lenguaje de consulta SPARQL. Este lenguaje fue definido por el World Wide Web Consortium (W3C) para los datos de la Web, en particular, para lo que se conoce como Web Semántica. El modelo de datos de SPARQL, llamado RDF, fue diseñado teniendo en consideración que la Web es un espacio en el cual múltiples actores publican de manera independiente, con diferentes creencias y maneras de modelar (o entender) el mundo. Esto nos lleva a que todos los conjuntos de datos en la Web sean considerados incompletos. Por el contrario, SPARQL es un lenguaje que surge (varios años después de RDF) de la necesidad de explorar un conjunto acotado de datos RDF, es decir, de la manera tradicional. Esta diferencia entre RDF y SPARQL produce incompatibilidades entre ambos lenguajes.

Para entender mi trabajo creo que es necesario revisar la noción filosófica que tenemos de la noción de “entender”. Desde el punto de vista científico creo que el concepto de “entender” está relacionado con aquello que ocurre cuando uno analiza un conocimiento bajo una formulación o teoría diferente de la original. En mi tesis, yo tomo las definiciones del lenguaje SPARQL y las analizo bajo la teoría de información

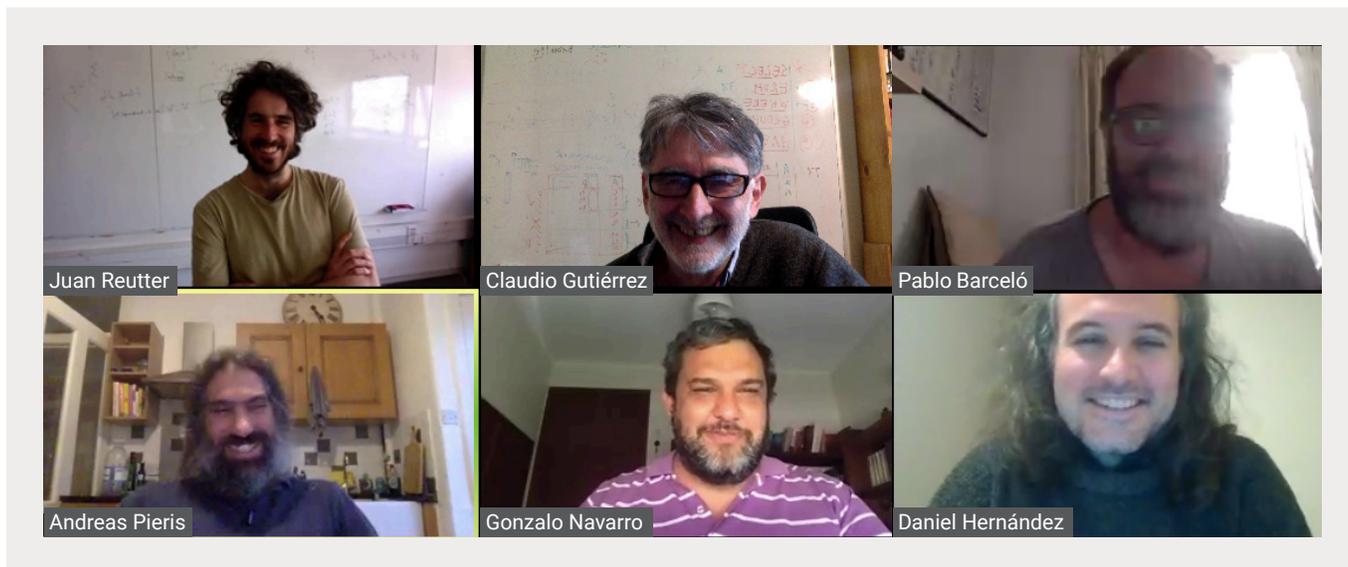
incompleta para bases de datos. La formulación de SPARQL consiste en una serie de reglas definidas de forma semiformal que describen una función que toma una base de datos en lenguaje RDF y una consulta en lenguaje SPARQL y entrega un conjunto de soluciones. Por otra parte, la semántica del lenguaje RDF consiste en asociar cada base de datos con un conjunto de posibles modelos del mundo representado. Bajo la teoría de información incompleta de las bases de datos, la pregunta natural es si la semántica de SPARQL es consistente con la semántica de RDF. Una definición concreta de esto es, por ejemplo, saber si las soluciones que se entregan para una consulta y una base de datos dadas son aún válidas para todos los modelos del mundo que la base de datos representa. A las soluciones que poseen tales características se las conoce como *certain answers* o soluciones seguras.

Para analizar el problema de las soluciones seguras en SPARQL tomé en consideración un fragmento de SPARQL con una semántica bien definida y una simplificación de la semántica de RDF que considera a los datos como sentencias con variables (y por ende incompletas). Por ejemplo, una sentencia como “Juan tomó el bus desde Santiago a x ” es incompleta porque, si bien sabemos que el bus que Juan tomó tenía un lugar de destino, no sabemos cuál era. El lenguaje RDF tiene un elemento que coincide exactamente con

esta noción de variable: los “nodos blancos”. Usando esta simplificación podemos formular la pregunta: ¿Produce SPARQL soluciones que no sean seguras? La respuesta es afirmativa. Una solución que no es segura se produce, por ejemplo, si la base de datos dice que “Juan tomó el bus de Santiago a x ” y la respuesta a la consulta “¿a qué lugar Juan no tomó el bus?” incluye a Curicó como respuesta. Esta respuesta es insegura porque en un mundo posible la variable x puede tomar el valor Curicó.

La pregunta que sigue es cómo podemos modificar la semántica de SPARQL para obtener sólo respuestas seguras. Una semántica de dichas características debe considerar que el problema de si una solución es segura está en la clase de complejidad $coNP$ (muy complejo), mientras que el fragmento SPARQL de nuestra formulación se puede computar de una forma muy eficiente (AC^0). Entonces, una parte de mi tesis consistió en proponer y evaluar experimentalmente la factibilidad práctica de un método aproximado para la evaluación de SPARQL, que entrega sólo respuestas seguras, pero que algunas veces no las entrega todas.

Actualmente, estoy trabajando en la Universidad de Aalborg, en Dinamarca, como postdoc en DAISY - Center of Data Intensive Systems.





dcc
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

NB Nano Break

Podcast del Departamento de
Ciencias de la Computación de
la Universidad de Chile

Disponible en:



DCC UChile



Nano Break

REVISTA DEL DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN DE LA UNIVERSIDAD DE CHILE

Bits

DE CIENCIA