FLOGENT: INFOMATION FLOW SECURITY FOR COGENT

VIVIAN DANG (UNSW)

In a world where the amount of sensitive information being put online is increasing, the risk of this information falling into the hands of malicious people also increases; this calls for more emphasis on information security within systems.

I have been working on FLOGENT, an information flow control system for the programming language COGENT (O'Connor et al., 2016). COGENT is a higher-order, purely functional language with *uniqueness types*. The semantics of COGENT are designed to be easy to reason about, in accordance with its aim of reducing the cost of verification. It is also designed for the implementation of operating systems components such as file-systems, which can compromise the *confidentiality* and *integrity* of the entire system if there are vulnerabilites. As such, ensuring COGENT software is free of vulnerabilites is particularly important.

FLOGENT intends to address the vulnerabilities that are observable on the programming language level, specifically, secret data being directly accessed or modified by unauthorised processes which violates confidentiality and integrity, by extending the Cogent type system with an information flow control feature. Abadi et al. (1999) generalised security type systems to accommodate a functional language with higher order functions. Vassena et al. (2018) then adapted Abadi's work into a HASKELL library called MAC.

My contribution so far is the design and implementation of two primitive operations within MINIGENT, a miniature version of COGENT. The operations are join and unlock, derived from the join and unlabel operations of Abadi et al. (1999). These are sufficient to ensure information flow control in programming language libraries such as MAC (Vassena et al., 2018).

Bibliography

Martín Abadi, Anindya Banerjee, Nevin Heintze, and Jon G. Riecke (1999). "A Core Calculus of Dependency". In: *Principles of Programming Languages*. POPL '99. San Antonio, Texas, USA: ACM, pp. 147–160. ISBN: 1-58113-095-3. DOI: 10.1145/292540.292555.

Liam O'Connor, Zilin Chen, Christine Rizkallah, Sidney Amani, Japheth Lim, Toby Murray, Yutaka Nagashima, Thomas Sewell, and Gerwin Klein (2016). "Refinement Through Restraint: Bringing Down the Cost of Verification". In: *International Conference on Functional Programming*. ICFP 2016. Nara, Japan: ACM, pp. 89–102. ISBN: 978-1-4503-4219-3. DOI: 10.1145/2951913.2951940.

Marco Vassena, Alejandro Russo, Pablo Buiras, and Lucas Waye (2018). "MAC: A verified static information-flow control library". In: *Journal of Logical and Algebraic Methods in Programming* 95, pp. 148–180. ISSN: 2352-2208. DOI: 10.1016/j.jlamp.2017.12.003.