CS6180 Lecture 25 – Event Logic, Computer Security, and Unguessable Atoms

Robert L. Constable

Abstract

The constructive/intuitionistic type theory we have been studying in this course and which is implemented by the Nuprl proof assistant has many novel features introduced by the Cornell researchers who designed and implemented the theory, starting in 1980 [3] presented in the 1986 book *Implementing Mathematics* [6] and continuing for thirty seven years with lively innovative work in Ithaca and abroad [5]. One of the special features of the implementation is that it supports *distributed computation* on its library of definitions, theorems, and tactics.

In this lecture we will continue our discussion of *distributed protocols* with a brief comparison of the easy to understand 2/3 consensus protocol [12] and the widely used *Paxos* protocol invented by Leslie Lamport [9, 10]. It is not easy to understand and even harder to verify, but it has been verified in Nuprl using the logic of events in the hands of dedicated experts such as Mark Bickford, Vincent Rahli, and Robbert van Renesse with help from Dan Arnon [2] when he was working at the EMC Corporation, now part of Dell. We will only discuss the highlights of this major effort to illustrate the expressive power of constructive type theory. This research also provides further evidence that the *Church/Turing thesis* about computability is applicable only to limited to a subset of what we now understood to be computable processes.

We will also look at the type of *Atoms* and how it is used in computer security and distributed systems. This is a fascinating type not found in other type theories. The type has the characteristic that it is not enumerable, yet it is not uncountable. There are two fundamental articles on this topic. One is by Stuart Allen who invented the type and gave a *supervaluation semantics* in the article "An Abstract Semantics for Atoms in Nuprl" [1]. Another key article is by Mark Bickford entitled "Unguessable Atoms: A Logical Foundation for Security" [4] in which Mark shows how to use atoms in security protocols. We will not examine the applications, but the type is very interesting in its own right.

1 Distributed Protocols

We have briefly discussed the 2/3 consensus protocol. It is easy to understand and reason about. We have proven it correct [12]. Industry uses the more complex but more efficient Paxos protocol [9, 10] for which Lamport won the Turing Award. We have also proved this protocol correct based on ideas from Van Renesse [13]. We will not have time to go into details of this protocol and its subtle correctness proof. However, we have a number of interesting and revealing stories about our efforts and our work with industry on this protocol.

In addition to examining these two protocols, we will look at the famous Fischer/Lynch/Paterson [8] result on distributed consensus. This is also called the FLP result, referring to Michael Fischer, Nancy Lynch, and Michael Paterson. They are all friends of mine. We all started out doing computational complexity research. That was a way to "establish credibility." Then we branched out into different aspects of computing theory as can be seen from these diverse references [7, 14, 11].

2 The Atom Type

The type of Atoms is explained by Allen using *supervaluation semantics* [1]. If you look up this semantics in Google you will see a lively discussion by philosophers and by Jon Sterling who is working at CMU with Professor Harper to design and implement a Nuprl-like proof assistant for a variant of our *fully intuitionistic type theory* [5].[1]

This type has the property that it is not finite, yet it is not enumerable. The article by Mark Bickford is included in supplementary material [1].

References

[1] Stuart F. Allen. An Abstract Semantics for Atoms in Nuprl. Technical report, 2006.

[2] Dan Arnon and Navindra Sharma. An Analysis of a Virtually Synchronous Protocol. Submitted to an ACM Transactions on Computer Systems, 2012.

[3] J. L. Bates and Robert L. Constable. Definition of micro-PRL. Technical Report 82–492, Cornell University, Computer Science Department, Ithaca, NY, 1981.

[4] Mark Bickford. Unguessable atoms: A logical foundation for security. In *Verified Software: Theories, Tools, Experiments, Second International Conference*, pages 30–53, Toronto, Canada, 2008. VSTTE 2008.

[5] Mark Bickford, Vincent Rahli, and Robert Constable. The good, the bad and the ugly. In *Thirty-Second Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS.

[6] Robert L. Constable, Stuart F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, NJ, 1986.

[7] M. J. Fischer and M. O. Rabin. Super-exponential complexity of Presburger arithmetic. In R. M. Karp, editor, *Complexity of Computation*, pages 27–41, Amer. Math. Soc., Providence, RI, 1974.

[8] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faculty process. *JACM*, 32:374–382, 1985.

[1] An attempt to stress the unique features of the current Nuprl type theory and it's implementation could also mention the asynchronous distributed computation system and the *event logic* used to reason about distributed computation.

[9] Leslie Lamport. The part-time parliament. *ACM Trans. Computer Systems*, 16(2):133–169, 1998.

[10] Leslie Lamport. Paxos made simple. *ACM SIGACT News*, 32(4):51–58, 2001.

[11] D. C. Luckham, M. R. Park, and M. S. Paterson. On formalized computer programs. *JCSS*, 4:220–249, 1970.

[12] Vincent Rahli, David Guaspari, Mark Bickford, and Robert L. Constable. Eventml: Specification, verification, and implementation of crash-tolerant state machine replication systems. *Science of Computer Programming*, 148(Supplement C):26 – 48, 2017. Special issue on Automated Verification of Critical Systems (AVoCS 2015).

[13] Robbert van Renesse. Paxos made moderately complex. Technical report, Cornell University, 2011.

[14] J.L. Welch, L. Lamport, and N. Lynch. A lattice-structured proof technique applied to a minimum spanning tree algorithm. Laboratory for Computer Science MIT/LCS/TM-361, Massachusetts Institute of Technology, Cambridge, MA, June 1988.