Formalising A Framework for Distributed Algorithms in Isabelle/HOL

Tom Ridge
LFCS,
School of Informatics,
University of Edinburgh.
T.J.Ridge@sms.ed.ac.uk

This extended abstract describes work in progress to formalise a framework for distributed algorithms in Isabelle/HOL, with a view to verifying the existence of simulation relations for certain classes of protocols. The Pragmatic General Multicast (PGM) protocol is a real world protocol to which the results apply.

> [PGM] is a reliable multicast transport protocol for applications that require ordered or unordered, duplicate-free, multicast data delivery from multiple sources to multiple receivers. PGM guarantees that a receiver in the group either receives all data packets from transmissions and repairs, or is able to detect unrecoverable data packet loss

The PGM protocol executes roughly as follows: A sender wishes to multicast a message to a set of receivers, reachable via intermediate nodes in a network having a predetermined (and assumed static) tree topology. The message is tagged and passed down to the receivers. If the receivers were to acknowledge (ACK) the receipt of the message, there would be an explosion of ACK messages at the sender. This restricts the extent to which such a protocol could scale. For this reason, receivers keep track of which message they are waiting for, and only signal (negative acknowledge, or NAK) when they detect they are missing a message in the sequence.

The framework is taken from Maidl (2003) where the author describes the framework, and shows that instances of certain protocols can be simulated by much simpler instances. This result is then applied to the PGM protocol to show that LTL properties (concerning the sender node and the receiver nodes) of full instances of the PGM protocol (sender is connected to receivers via intermediate nodes) are preserved by related restricted instances (where the sender is connected directly to each receiver). This is a formal counterpart to the informal observation that an intermediate network between a sender and a receiver can be viewed as a homogeneous "network cloud": modulo some notion of the external view of an execution, the intervening network is invisible to the sender and receiver.

The primary motivation for this work is to apply theorem proving technology to a real world example, as a test of what can currently be achieved, whilst also contributing to increased confidence in the example itself (and detecting any omissions or inadequate assumptions). The proofs found in Maidl (2003) have small conceptual complexity. Never-the-less they contain a large amount of detail- detail which it is hard for a human being to assert free from error. It is hoped that formalisation will result in proofs that are not significantly longer than those in Maidl (2003), but with the added advantage that it is easier to believe that the steps are correct.

A second motivation was to use the Isar mode of Isabelle, recently introduced in Wenzel (1999), in an attempt to make the formalisation human readable. The hope is that if authors can read formalised proofs that are similar to pen and paper proofs, the barrier to relating a formal proof to a pen and paper proof will have been significantly lowered. Not only will authors be more inclined to affirm that the formalised proof is a fair rendering, but perhaps they will be persuaded to try out these powerful technologies for themselves.

The formalised framework in Isabelle consists of standard encodings of Kripke structures and LTL formulae. An early result was the formalisation of (a variant of) the well-known, but apparently never explicitly proved, result that LTL formulae are stuttering invariant (by induction over the structure of LTL terms).

Maidl (2003) proceeds to show that there is a simulation relation between full and restricted instances, by providing trace equivalent finite executions of the restricted instance for finite executions of the full instance. The proof that this entails a simulation relation (and hence trace inclusion) involves the construction of an infinite execution of the restricted instance for every infinite execution of the full instance. Conceptually there is no problem, in this case, in completing finite related instances to an infinite related instance. However, as alluded to in Muller (1998), this is hard to formalise (in fact, Muller claims that simply typed higher order logic is incapable of verifying the soundness of anything more complicated than weak refinement mappings).

One very natural formalisation of this result, Muller (1998), uses the machinery of Domain Theory. Future work might involve incorporating this into the current formalisation. However, the proofs in Maidl (2003) are instances of the weaker notion of refinement mapping, which can be formalised readily within HOL.

The bulk of the formalisation so far concerns establishing properties of trees (the network topology is assumed to be a tree), of functions over multisets (the contents of communication channels are modelled as multisets), and of verifying the existence of a simulation relation for a given protocol class. The verification of the simulation relation is proving to be the most arduous, because the reasoning in Maidl (2003) itself is very concrete and specific. The automated tactics found in Isabelle do not cope well with the complicated terms involved. Happily, Isar features, such as calculational mode, have ameliorated problems of fine grained term manipulation, so that in the absence of automation, the effort involved in manually constructing the proof is greatly lessened. Further, the calculational proof has the added benefit of being human readable.

A non-standard feature of the framework is that the states of network nodes are implicit, being a function of the execution history so far. Thus, states of the Kripke structure are finite executions. This contrasts with other formalisms, such as Isabelle/IOA, where states in the Kripke structure correspond directly to states of the network at a point in time. An advantage of this approach is that the state of the network never has to be treated directly. A disadvantage is that proving properties of the protocol by induction over the tree structure would seem to be unduly difficult. The formalism, therefore, is suited to global notions of correctness (e.g. a simulation relation on global states of a protocol instance), but not well suited to establishing properties that appeal to local notions, and that might be proved by induction. An example of the latter would be a proof that showed a protocol instance consisting of $n + 1$ nodes was a refinement of an instance consisting of n nodes, and then appealed to transitivity of composition of refinement to show that all instances refined by an instance consisting of a single node.

Future work will extend the current formalisation to include the full machinery of simulation relations, and to address liveness issues. Both notions are found in HOLCF/IOA, and it is hoped that aspects of this work can be incorporated. In the long term, the goal is to tackle the formalisation of distributed algorithms under varied formalisms, to determine what factors are important. For instance, the author previously utilised the HOLCF/IOA formalism, which has a global notion of action synchronisation. This means that the network topology is embodied in the actions that nodes can send (for instance, the fact that nodes A and B are connected might be expressed by the presence of a message M_{AB}, rather than any explicit object which corresponds to the network topology). This, coupled with the IOA notion of hiding and renaming of actions (which makes them trace invisible but does not prevent them interacting) caused problems for modular verification.

References

Monika Maidl. Simple representative instantiations for the pgm protocol. 2003. URL http://www.dcs.ed.ac.uk/home/monika/maidl-pgm-reduction.ps. To be published, TACAS 2003.

Olaf Muller. A verification environment for i/o automata based on formalised metatheory, 1998. PhD thesis, TU-Munchen.

Markus Wenzel. Isar - a generic interpretative approach to readable formal proof documents. In *Theorem Proving in Higher Order Logics, 12th International Conference, LNCS 1690*, 1999.