FORMAL METHODS FOR SPECIFICATION AND ANALYSIS OF COMMUNICATION PROTOCOLS
Submitted by: Vinit (vv2236)

Formal methods are mathematically based techniques used for specification, system development and verification of software systems. This paper introduces some of the popular formal description techniques (FDTs), which can be used for analysis and verification of communication systems. The syntax, data types, communication and tool support for each of these methods have been described and implementation of the Go back N protocol is given as an example to make this paper more like a tutorial. Each FDT also has a critical evaluation section in which its properties and compatibility with other tools as well as ease of implementation of communication protocols has been discussed. This paper also discusses the advantage of using formal methods along with the traditional testing and simulation techniques to make the system more robust and reliable.

In the first section, the author discusses the basic concepts of formal methods and lists the applications of formal methods in software development, which are system specification, verification, functional testing, prototyping and performance testing. The concept of model checking is also introduced in which the system properties to be checked are mathematically defined and then their validity is verified in all reachable states of the system model. Temporal logic formulas are used to verify the temporal properties of these models. The FDTs described in this paper are based on different operational model. The first FDT given is Specification and Description Language (SDL). SDL is a high-level general-purpose description language and its main area of application is telecommunication systems and protocols. SDL is based on Extended Finite State Machines and uses a graphical syntax. It has a modular approach and a clear distinction between structural and behavioral aspects of a system, which facilitates modeling of layered architectures in communication protocols in this technique. SDL is the most widely used formal specification language in telecommunication companies.

The next protocol discussed is SPIN (Simple Promela Interpreter), which uses a C-like syntax and is freely available. It supports efficient model checking and specification and verification of protocols. Communication within processes here is asynchronous and is done through finite length queues. Estelle is another FDT, which is based on Extended FSM theory and uses Pascal for data manipulation. LOTOS (Language of Temporal Ordering Specifications) is based on process algebra, which is used to model distributed systems and concurrent processes. Its syntax has temporal and stochastic features. The data type is conceptually same as that of SDL’s. Communication between processes here is synchronous. Another FDT used to describe concurrent and distributed systems is Petri Nets. PN also uses graphical modeling in which places (conditions), transitions, edges and tokens are used to specify a system. A condition is verified if the tokens fill the place and this results in a transition, which carry the enabling tokens to new activated places. UPPAAL is a tool used for simulation and verification of real time systems modeled as networks of timed automata.

There are two small sections on Message Sequence Chart and Unified Modeling Language. Both use graphical notation to describe a system.

In the conclusion section, the author has compared the FDTs discussed in this paper and presented it in a tabular form. SPIN is the most successful freeware tool used today. SDL is also widely used mostly due to its graphical notation and support for other notations. LOTOS is not very popular because of its complexity and strict syntax. PN is also very user friendly. Another important factor that has been considered to compare these tools is the learning period and the effort required.