

# Jordan University of Science and Technology

## The Use of Dynamic Temporal Assertions for Debugging

**Authors:** Ziad A. Al-Sharif, Clinton L. Jeffery and Mahmoud H.Said

**Abstract:** Bugs vary in their root causes and their revealed behaviors; some may cause a crash or a core dump, while others may cause an incorrect or missing output or an unexpected behavior. Moreover, most bugs are revealed long after their actual cause. A variable might be assigned early in the execution, and that value may cause a bug far from that last assigned place. This often requires users to manually track heuristic information over different execution states. This information may include a trace of specific variables' values and their assigned locations, functions and their returned values, and detailed execution paths. This chapter introduces Dynamic Temporal Assertions (DTA) into the conventional source-level debugging session. It extends a typical gdb like source level debugger named UDB with on-the-fly temporal assertions. Each assertion is capable of: 1) validating a sequence of execution states, named temporal interval, and 2) referencing out-of-scope variables, which may not be live in the execution state at evaluation time. These new DTA assertions are not bounded by the limitations of ordinary in-code assertions such as locality, temporality, and static hardwiring into the source code. Furthermore, they advance typical interactive debugging sessions and their conditional breakpoints and watchpoints.