Bounded Model Checking for Concurrent Behavior with Scheduler

Haitao ZHANG (1010232)

School of Information Science,
Japan Advanced Institute of Science and Technology

August 9, 2012

Keywords: Fixed priority scheduler model, Bounded Model Checking, SMT, OSEK/VDX OS.

With the advancement of the automobile manufacturing technology, demands for the auxiliary functions of vehicles have also increased sharply and tended to be diversified, which have greatly stimulated the application and development of electronic techniques in automobile industry. However, not all of the electronic parts manufacturers use the same production standard, there exist extremely complex correspondence and cooperation between different electronic parts. OSEK/VDX, as a standard for automobile industry, has been proposed by Germany and France automobile manufacturers and applied in many automobile systems to normalize the correspondence and cooperation. Especially, in OSEK/VDX OS, which task to be run is determined by scheduler, in addition, tasks can send service commands to request scheduler for responding to its particular behaviors, such as terminating itself, activating a task and chaining a task. Thus, there may exist a potential risk which is caused by an unreasonable dispatching when tasks run in the system. Therefore, how to check the safety property of a multi-task software based on automobile OSEK/VDX OS has become very difficult and crucial.

Model checking, as a traditional technique, has been applied to checking multi-task software, however, suffers from combinatorial state space explosion when verifying complex multi-task software. Recently, a new

technique called bounded model checking (BMC) has been proposed to overcome the state explosion problem and has been successfully applied to verify the multi-task software. There are many efficient and reliable techniques based on BMC have been proposed to check safety property of multi-task software, e.g., Ganai and Gupta describe a verification framework for BMC to extract high-level design information from an extended finite state machine (EFSM), and several techniques have been employed to simplify the BMC problem, they also describe a lazy method for modeling multi-task concurrent software using shared variables. Grumberg et al. propose a method based on SAT and BMC to check a multi-task system with a series of under-approximated models. However, these techniques focus on the tasks current behaviors, the scheduler's behaviors are not considered in verification process. Therefore, these techniques are not able to check the safety property of a software in which tasks are dispatched by a scheduler to be executed.

In our article, we propose an approach to check the safety property of multi-task software in which tasks are dispatched by fixed priority scheduler (FPS) based on OSEK/VDX OS. In order to accomplish our research purpose, firstly, we analyze the dispatching behaviors of FPS based on OSEK/VDX OS, and then we use an extended finite state machine to establish a model for FPS and describe tasks behaviors. Especially, our FPS model can respond to three types of service commands which are sent by tasks in order to realize tasks particular requests, such as terminating a task, activating a task and chaining a task. Furthermore, as to obtain the execution paths of tasks that are dispatched by FPS, we establish a k-step execution tree to represent all of the possible execution paths. In the execution tree, each possible execution path is from *root-node* to one of *leaf-nodes* and the total amount of execution paths is equal to the total amount of leaf-nodes. In order to ensure each service command which exists in branches of task can be responded by FPS, we insert FPS model into each node of execution tree for obtain all of the dispatching execution paths.

Based on execution tree, we propose two strategies to extract execution paths in which BMC is employed to generate the verification conditions (VCs) based on our execution tree. In addition, Yices which is satisfi-

ability modulo theories (SMT) solver and capable of handling large and propositionally complex formulas in a rich combination of theories is used to check the generated VCs with verification property formula and return the verification results. Finally, we implement two types of tools according to our two strategies of extracting execution paths based on execution tree to evaluate our approach. Using our tools, we can directly get the k-step transition system M which is composed of each execution paths VCs based on FPS dispatching, furthermore, the k-step transition system M can be translated into Yices file with our tools. We also carry out some relevant experiments with our tools, results show that our approach can efficiently check the safety property of multi-task software in which tasks are dispatched by FPS.