Symbolic scheduling

Abstract: One of the effective techniques for controlling the complexity of task execution in heterogeneous hardware/software systems is symbolic scheduling. It makes it possible to analyze and synthesize execution plans without a comprehensive list by symbolically describing scheduling constraints and system behaviours. With an emphasis on their use in the software and hardware domains, this seminar paper attempts to investigate and assess a variety of symbolic scheduling approaches. The first part of the paper will look at the FunState model, which was presented by Strehl et al. and explicitly represents scheduling and nondeterminism by combining state machines and functional programming. The paper will then examine the ESST (Explicit Scheduler, Symbolic Threads) method put out by Cimatti et al., which improves software model testing by keeping an explicit scheduler in place while symbolically exploring threads. The difficulties of confirming cooperative scheduling policies in multithreaded software systems are addressed by this method. The study will conclude by examining Soviani's high-level synthesis methodology for packet processing pipelines, which uses symbolic scheduling to maximize hardware designs' resource allocation and pipelining. By contrasting these methods, the study aims to determine which symbolic scheduling strategy works best, taking into account aspects like scalability, efficiency, and suitability for real-world systems.