Exploiting Partitioned Transition Relations for Efficient Symbolic Model Checking in CTL

Aleš Časar, Zmago Brezocnik, Tatjana Kapus
University of Maribor, Slovenia
{casar, brezocnik, kapus}@uni-mb.si

Abstract

We present an efficient tool for symbolic state space traversal of finite state machines. Both algorithms for searching reachable states and for model checking in CTL owe their efficiency primarily to the use of partitioned transition relations. Partitioning of the relations is fully automatic.

Symbolic state space traversal techniques show their superiority over the enumeration based methods when searching reachable states of a FSM. Sets of states and transitions can be represented by characteristic functions, and because they are boolean, we can represent them compactly by BDDs. The transition relation of a large circuit may result in a huge BDD that either does not fit into computer memory or leads to unacceptably long computation times. There are several approaches to overcome this problem. We describe the use of partitioned transition relations for an efficient symbolic state space traversal and model checking in CTL.

If the transition relation $T$ of a FSM is represented by a so-called monolithic transition relation, i.e., using a single BDD, the algorithms for computing reachable state sets might not be efficient enough if the size of $T$ is large. Therefore we do not build a single BDD for the whole transition relation $T$, but rather partition $T$ in some groups of transition relations $T_j$ of individual state variables and represent each group by a (smaller) BDD. $T$ is then called a partitioned transition relation.

Although existential quantification does not distribute over conjunction, conjuncts can be moved out of the scope of an existential quantification if they do not depend on any of the variables being quantified. FSMs that represent circuit behaviour exhibit locality, so it is very common that many of the $T_j$ depend on only small number of the input and state variables. We developed a heuristic algorithm for partitioning and reordering of functions and variables for existential quantification in the formula for computing the next reachable states from a set of present states, represented by its characteristic function $S_i$, that allows us to apply some existential quantifiers before the BDD of the whole transition relation has been built. $S_i$ may depend on all state variables of the FSM, so we leave it within the scope of all existential quantifiers of the state variables. In contrast to the function $S_i$, functions $T_j$ are constant throughout the computation and at the beginning we can determine their place with respect to the existential quantifiers once and for all. The heuristics was also employed for backward searching of reachable states within CTL model checking algorithm.

We wrote an efficient software based on those algorithms, using our own BDD package [1] that is an efficient ITE-based implementation of ROBDDs with complemented edges. Partitioning and reordering of individual transition relations is automatic and needs no user intervention. Introducing partitioned transition relation may decrease the CPU times for large circuits dramatically.

All experiments were done on a Silicon Graphics Indy workstation (84.9 SPECint92) with 64 MB of RAM. Experimental results on the ISCAS'89 benchmark circuits show [2] that our algorithm for searching reachable states is more efficient (e.g., CPU time for s1238 is 1.61 s) than the algorithms that exploit a generalized cofactor and image restrictor and that use range restrictor "constrain". The efficiency of our CTL model checking software was tested on several synchronous sequential circuits. For example, the following properties of a $n$-request bus arbiter were verified: exclusivity, causality, allocation, and starvation (3$n$ + 1 CTL formulas). Verification of the largest arbiter tackled ($n = 10$; 20 latches) took 252.13 s.

References
