ECE 510 OCE
BDDs and Their Applications

Lecture 4. Variable Reordering

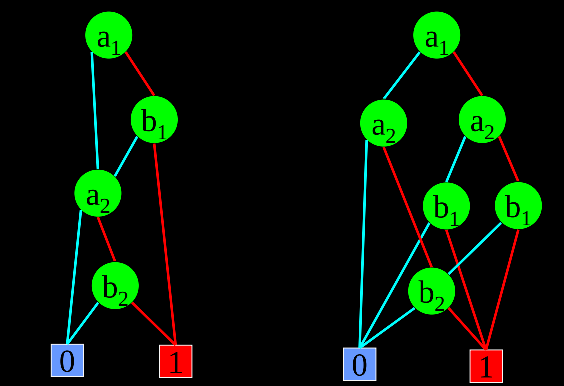
April 6, 2000

Alan Mishchenko

Overview

- Connection between variable ordering and BDD size
- The size of BDDs for symmetric functions
- Exponential lower bounds
- Exact minimization is NP-complete
- The fan-in based heuristic to find good static variable ordering for BDDs representing outputs of logic networks
- Dynamic reordering based on variable swap
- Reordering algorithms: window permutation, sifting, block sifring, symmetric sifting

Variable Ordering for $F = a_1 b_1 + a_2 b_2$



$a_1 < b_1 < a_2 < b_2$ $a_1 < a_2 < b_1 < b_2$

BDDs for Symmetric Functions

- The value depends only on the number of 1's in the input vector
- For n-variable functions, there are n+1 possible numbers of 1's and so 2^{n+1} functions
- Cofactors of symmetric functions themselves are symmetric functions
- Ik the function is cofactored k times, there are no more than n+1-k different cofactors
- The BDD size is limited by $n*(n+1-k) \sim O(n^2)$

Exponential Lower Bounds

- The dependence of the BDD size on variable ordering is very strong. It would be good if for each function at least one ordering gave a small BDD
- However this is not true: BDD representations shares the fatal property of *all* other representations: the representations for nearly all functions take exponential space!
- The reason is that there are 2^{2^n} functions over n variables. This number is so large that it is impossible to have polynomial size representation for all but a very small percent of the functions.

Exact Minimization of BDDs

- Because the size of BDDs depends on variable order, algorithms for constructing good orders are of great practical importance
- It can be shown that the test whether the given BDD has the minimum possible size is an NP-complete problem
- Hence, it is an NP-complete problem to construct the optimal oder for a given BDD, and efficient algorithms cannot be expected

Exact Minimization: Historical Picture

- 1986, in the original paper introducing BDDs, Randy Bryant stated without proof that the construction of an optimal order is NP-complete
- 1993, S.Tani, K.Hamaguchi, S.Yijima proved the weaker case (for shared BDDs)
- 1996, B.Bollig and I.Wegener proved the theorem for BDDs with exactly one root

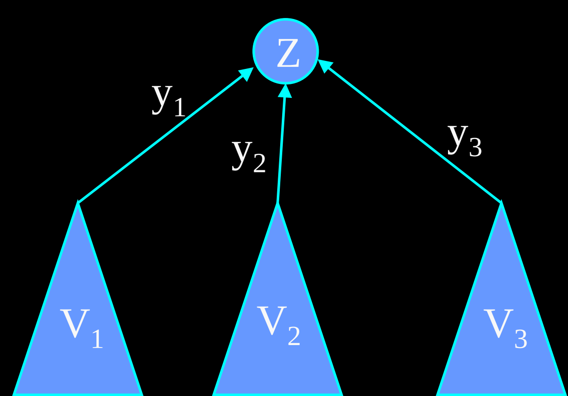
Optimizing Variable Ordering

- Depending on the initial representation (Boolean formula, net list, SOP, etc.), the problem of finding a good static order may have many formulations
- Often semantic information can be used
- The initial functions may be already in the form of BDDs

Fan-in Heuristic for Static Ordering

- Transitive fan-in for the node of the combinational boolean network is
  - 0, if it is an output node
  - or $1+\max_{wi}(\text{ TFI(wi) })$, otherwise
- Heuristic:
  - transform the network to have one output
  - perform a depth-first search of the network starting from the outputs and, when there are two or more predecessors, prefer the one with the largest TFI
  - the earlier the variable appears in a predecessor's variable list, the earlier it is put in the global list

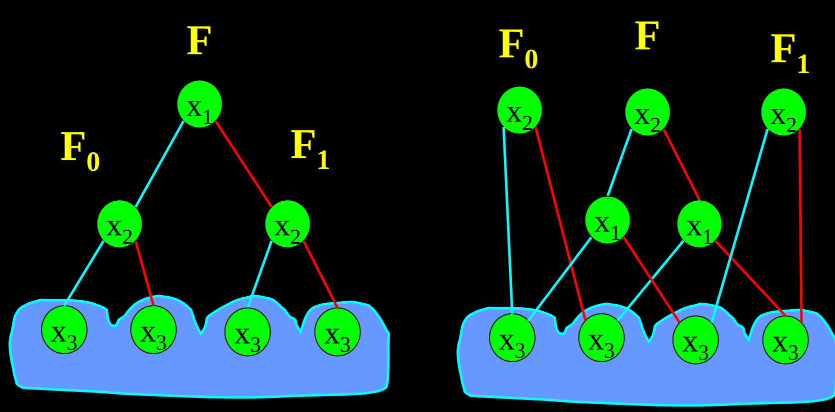
Motivation for the fan-in heuristic



Motivation for Dynamic Reordering

- Static methods are too problem specific (for example, the fan-in heuristic exploit the topology of the circuit)
- The methods work heuristically
- Static ordering even if good is not ideal. In order to succeed in many cases, we need to change the variable ordering as the computation proceeds

Variable Swap is a Local Operation



Algorithms for Dynamic Reordering

- Window permutation
- Sifting algorithm
- Block sifting
- Symmetric sifting
- Block-restricted sifting

Quantitative Comparison