Summary of Randal E. Bryant's
"Graph-Based Algorithms for Boolean Function Manipulation"

Several areas of computer science, including digital system verification, cast problems in terms of boolean functions. Common representations of boolean functions, such as truth tables and Karnaugh maps, have size exponential in the number of variables appearing in the formula, regardless of the formula. This paper presents a new data structure for representing boolean functions and a set of algorithms for manipulating the data structure.

The proposed representation produces significantly smaller structures for many boolean functions, with small polynomial bounds for certain definable classes of functions, such as the symmetric functions. The representation is also canonical, meaning that any two boolean formulas that yield the same outputs on the same inputs have the same representation. Canonicity implies that the equivalence of two formulas can be checked quickly; in this case, equivalence testing is a simple isomorphism test of time linear in the size of the representations. Although this representation is similar to ones proposed by other researchers, Bryant is the first to contribute algorithms for combining representations via boolean operations.

Bryant's representation is a form of directed, acyclic graph with a single source node and one or two sink nodes. Given a formula and an ordering on its variables, the graph can be produced by a recursive procedure that either creates a node labeled with a boolean constant if the formula is constant or creates a node for the first variable in the ordering and adds two edges from that node: one (labeled with 0) pointing to the representation for the formula with 0 substituted for the variable and one (labeled with 1) pointing to the representation for the formula with 1 substituted for the variable. Both of the sub-representations are generated relative to the variable ordering minus the first element.

Applying a provided reduction algorithm that eliminates isomorphic subgraphs and bypasses any nodes for which both of its outgoing edges point to isomorphic subgraphs yields the final, canonical representation for F. The output of F under a particular assignment of variables can be determined by traversing F from the source to either the node for 0 or the node for 1, at each node following the edge labeled 0 if the variable is assigned false and following the edge labeled 1 otherwise. Satisfiability of F is determined simply by testing whether the representation of F consists only of the node for constant 0. An algorithm provided for combining two graphs G_1 and G_2 with respect to any boolean operation also yields graphs in this reduced form; this algorithm is efficient, requiring time $O(|G_1| \times |G_2|)$.

Experimental results indicate that this representation is viable in practice for several common examples. One notable exception is integer multiplication, for which this representation does not improve on the exponential worst-case size of other representations. It should also be noted that the succinctness of the representation for a particular function may be heavily dependent on the variable ordering chosen; although the problem of choosing an optimal variable ordering is co NP-complete, it is suggested that good heuristics exist for finding reasonable variable orderings in practice.