Using Consensusless Covers for Fast Operating on Boolean Functions

Eugene Goldberg        Ludmila Krasilnikova

Institute of Engineering Cybernetics, the Academy of
Sciences of Belarus, Surganov str.6, Minsk 220012, Republic of
Belarus, fax: ++7 0172 31 84 03, phone: ++ 7 0172 39 51 71,
e-mail: katkov@adonis.iasnet.com

Abstract. The paper presents a method for fast operating on covers
of Boolean functions. The method develops the one based on the unate
paradigm (UP)[1]. The proposed method differs from the UP one in two
aspects (1) the initial cover is decomposed into a set of prime rather than
unate subcovers, (2) prime covers are obtained by applying to each re-
ced not prime subcover either branching by the Shannon expansion or a
procedure of making the subcover consensusless in a variable by the con-
sensus operation. Experiments on MCNC-91 two-level logic benchmarks
and random functions show that operations based on the proposed method
are less laborious than their UP based counterparts.

1 Basic Definitions and Propositions

To present the proposed method we need to introduce some definitions and
propositions.

A subset \( C = S_1 \times \ldots \times S_n \) of the Boolean \( n \)-space \( \{0, 1\}^n \) is said to be a cube. A
subset \( S_i \) we shall call the \( i \)-th component of \( C \). A cube \( C \) is called an implicant
of a completely specified single-output Boolean function \( f \) if \( C \subseteq \text{ON}_\text{SET}(f) \)
where \( \text{ON}_\text{SET}(f) \) is the vertices of the \( n \)-space in which \( f \) evaluates to 1. An
implicant \( C \) is said to be a prime if any cube strictly containing \( C \) is not an
implicant. A set of implicants of \( f \) which contain any vertex of \( \text{ON}_\text{SET}(f) \)
is called a cover of \( f \). We shall call a cover prime if all primes of the Boolean
function specified by the cover are in the cover.

Cubes \( C', C'' \) are said to be orthogonal in the \( k \)-th variable if \( S'_k \cap S''_k = \emptyset \).
Let cubes \( C' \) and \( C'' \) be orthogonal only in the \( j \)-th variable. Then a cube
\( S'_1 \times \ldots \times S'_j \times S_j' \times S^n \times S^n \) is said to be produced by the consensus
operation [2]. We shall call a pair of cubes which are orthogonal only in one variable a
consensus pair in the variable. We shall call a cover consensusless in a variable
if any cube produced by the consensus operation from a pair of cubes from \( F \)
which are consensus pair in the variable is contained in some cube from \( F \).

Denote by \( F_{x_j} \) and \( F_{x_j} \) subcovers (sometimes called cofactors with respect
to \( x_j \) and \( \bar{x}_j \) [2]) of subfunctions \( f(x_1, \ldots, 1, \ldots, x_n) \) and \( f(x_1, \ldots, 0, \ldots, x_n) \) of function
\( f(x_1, \ldots, x_j, \ldots, x_n) \) formed from a cover \( F \) of \( f \).

**Proposition 1** If a cover \( F \) is consensusless in all variables, \( F \) is prime.

**Proposition 2** Let a cover \( F \) be consensusless in the \( j \)-th variable. Then the
covers \( F_{x_k} \) and \( F_{\overline{x_k}} \), \( k \neq j \) obtained by Shannon expansion or any cover \( F' \)
obtained by adding to $F$ a set of cubes produced by the consensus operation are consensusless in the $j$-th variable too.

**Proposition 3** Let $F$ be a cover and $F_j'$ be a set of all cubes produced from consensus pairs of cubes from $F$ orthogonal in the $j$-th variable. Then the equivalent to $F$ cover $F_j^* = F \cup F_j'$ is consensusless in the $j$-th variable.

The proofs of the propositions are omitted for short.

### 2 Formulation of the Method

The UP method is to apply the Shannon expansion to the initial cover to decompose the cover into a set of unate subcovers. This allows one to substitute operating on the cover for doing on the unate subcovers. The main property of unate covers that makes performing many operations trivial is that a unate cover is prime [2]. The key point of the method presented in the paper is to decompose the operated cover into prime subcovers which, generally speaking, may not be unate.

The method consists in recursive performing the following algorithm. (1) If there is a variable in which a subcover $F$ is not unate and an ancestor of $F$ was not made consensusless in the variable then step 2 is performed. Otherwise $F$ is prime. (2) The subcover $F$ is either decomposed by branching in a "not processed" variable or made consensusless in one of such variables.

By making the subcover $F$ consensusless in variable $x_j$ is meant the described in proposition 3 procedure of substituting $F$ for cover $F_j^*$. To choose between the two alternatives in step 2, the values $a = \min(2|F| - (n_1^j + n_0^j))$ and $b = \min(|F| + n_1^j \cdot n_0^j)$ are calculated where $n_1^j$ and $n_0^j$ is the number of cubes from $F$ the $j$-th component of which is equal to $\{0\}$ and $\{1\}$ respectively. The value of $2|F| - (n_1^j + n_0^j)$ is equal to $|F_{x_j}| + |F_{\bar{x}_j}|$ and so $a$ describes the most effective way of branching. The value of $|F| + n_1^j \cdot n_0^j$ is the upper bound of $|F_j^*|$ and so $b$ describes the most effective way of making the cover $F$ consensusless in a variable. If $a \leq b$ the branching in a variable minimizing $2|F| - (n_1^j + n_0^j)$ is chosen. Otherwise from $F$ cover $F_j^*$ is obtained where $j$ is the index of variable for which $|F| + n_1^j \cdot n_0^j$ is minimum.

Justification of the method is based on propositions 1-3.

### 3 Experimental Results

To evaluate the efficiency of using the proposed method (further referred to as the EPC- (expansion plus consensus) method) programs EPC-Reduce and EPC-Decomposition have been written. EPC-Reduce implements Reduce operation used in the two-level logic minimizer Espresso [2]. The program was applied to a number of MCNC-91 two-level examples (table 1). When implementing the operation the extension of the EPC-method to the case of multi-output Boolean functions was made. The program EPC-Decomposition is intended just to decompose the cover into a set of prime covers. The program was applied to single-output covers obtained by the pseudorandom number generator (table 2).