Minimization of multi-output functions

Tiziano Villa

October 31, 2010

In principle, minimizing a multi-output function requires - with respect to disjoint single-output minimization - computing the primes not only of the single output functions, but also of the products of the pairs, triples etc. of output functions. E.g., consider a multiple-output function including the single-output functions f_1, f_2 and f_3. then one must compute the primes of f_1, f_2, f_3, $f_1 \cdot f_2$, $f_1 \cdot f_3$, $f_2 \cdot f_3$, $f_1 \cdot f_2 \cdot f_3$. E.g., the primes of f_1 and f_2 are the maximal implicants occurring at the same time in f_1 and f_2. In practice, representing multiple-output Boolean functions with the standard cubical representation, it is easy to extend the usual prime computation techniques to them. For instance, consider merging in the multiple-output case: when two adjacent implicants are merged, their output parts are intersected to form the output part of the resulting implicant. If the intersection is null, no new implicant is created. Similarly, when subsuming an implicant by another, one must check that the output parts are the same (see standard textbooks for an illustration of the algorithms).

An interesting way to look to the problem is based on the following observation by Sasao: an n-input m-output Boolean function can be represented by a multiple-valued function of $n+1$ variables where $p_i = 2$, for $i = 1, \ldots, n$, and $p_{n+1} = m$. More precisely, T. Sasao states (in T. Sasao An application of multiple-valued logic to a design of programmable logic arrays. Proc. 8th Int. Symp. on Mult. Val. Logic, 1978) that the Boolean minimization problem for multiple-output functions is equivalent to the minimization of a multiple-valued function of this form.

Theorem 0.1 *Minimizing a two-valued Boolean function with n inputs and m outputs:*

$$f : \{0,1\}^n \mapsto \{0,1\}^m,$$

is equivalent to minimizing a single two-valued output function with n two-valued inputs and one m-valued input:

$$F : \{0,1\}^n \times \{0, \ldots, m-1\} \mapsto \{0,1\}$$

Proof: Given a two-valued Boolean function f with n inputs and m outputs, we construct a single two-valued output function F with n two-valued inputs and one m-valued input, as follows:

$$F = \sum_{i=1}^{m} f_i \cdot Z^i,$$

where f_i is the Boolean function $B^n \mapsto B$ corresponding to the i-th output and Z is an m-valued variable.

Let $X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n} Z^S$ be an implicant of F, where $S_i \subseteq \{0,1\}$, for $i = 1, \cdots, n$ and $S \subseteq \{0, \cdots, m-1\}$.

Denoting with $\leq$ the relation "is covered by " or "is contained by", from the definition of F, if

$$X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n} Z^S \leq F$$

then

$$\forall i \in S:\ X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n} \leq f_i.$$

Let $X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n}$ be an implicant of f_i, $i \in S \subseteq \{0, 1, \cdots, m-1\}$. Again from the definition of F, if

$$\forall i \in S:\ X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n} \leq f_i$$

then

$$X_1^{S_1} X_2^{S_2} \cdots X_n^{S_n} Z^S \leq F.$$

Therefore there is a one-to-one correspondence between implicants of F and implicants of f. Let C be a cover of f and C' be the corresponding cover constructed from C over $S_1 \times S_2 \cdots \times S_n \times S$ with the previous rule. From the above, C' is guaranteed to be a cover of F. Furthermore, the size of the cover C is defined as the number of distinct product terms in C, which is equal to the size of C'.

Hence minimizing F is equivalent to the original problem. ■

Now that we know that multi-output minimization reduces to multi-valued minimization, it is time to ask how is the latter performed in practice. Well, there is an algorithm for it that is the extension of the classical one and, what is more, ESPRESSO implements it (see paper by Rudell on MV espresso). So by using ESPRESSO one relies exactly on that reduction.

There is a trick to reduce multi-output minimization to two-valued one (adding more variables) that we show in the special case of two-output functions. Consider the functions $f_1 = a'bc' + ab'c + a'b'c' + ab'c'$ and $f_2 = a'b' + a'bc$. The primes of f_1 are $a'c'$, ab', $b'c'$ and those of f_2 are $a'b'$, $a'c$. For the two-output function $\{f_1, f_2\}$ introduce a new two-valued variable z such that $z = 1$ when $f_1 = 1$ and $z = 0$ when $f_2 = 1$ (this corresponds to reducing a multiple-output minimization problem to a multi-valued-input single-output problem - in our case, since there are two output functions, the multi-valued variable z is two-valued). Then, we must find the primes of the four-variable function: $z(a'bc' + ab'c + a'b'c' + ab'c') + z'(a'b' + a'bc)$. Its primes are: $za'c'$, zab', $zb'c'$, $z'a'b'$, $za'c$, $a'b'c'$. Note that $a'b'c'$ was not a prime of either f_1 or f_2.

In general multi-output minimization obtains a smaller cover than disjoint minimization of the single-output functions. As an example, consider the following function $\{f_1, f_2\}$

```
.i 3
.o 2
000 10
010 10
100 10
111 10
000 01
010 01
100 01
110 01
111 01
```

f_1 is

```
.i 3
.o 1
000 1
010 1
```

```
100 1
111 1
```

and a minimized cover has 3 cubes

```
.o 1
.p 3
111 1
-00 1
0-0 1
```

f_2 is

```
.i 3
.o 1
000 1
010 1
100 1
110 1
111 1
```

and a minimized cover has 2 cubes

```
.i 3
.o 1
.p 2
11- 1
--0 1
```

Instead a minimum cover of $\{f_1, f_2\}$ has 4 cubes ($< 3 + 2 = 5$ cubes of disjoint minimization)

```
.i 3
.o 2
.p 4
111 11
-00 10
0-0 10
--0 01
```