

CS623: CAD FOR VLSI DESIGN
Lecture 36

Unate Function Complementation

Let F be unate.

If F is positive unate then

$\overline{F} = x_j h_0 + h_1$ for any variable x_j , here $h_1 \subseteq h_0$.

If F is negative unate then

$\overline{F} = h_0 + \overline{x_j} h_1$ for any variable x_j , here $h_0 \subseteq h_1$.

The Recursive Procedure

Choose a variable,

recursively complement a cover consisting of cubes not containing the variable, and

complement the original cover with that variable dropped,

tack the variable onto cubes of the first result, and

concatenate the two results.

The Personality Matrix

Given a cover F with k cubes and n variables,

$M_{ij} = 1$ if $F_{ij} \neq 2$, else 0, $1 \leq i \leq k$, $1 \leq j \leq n$.

Procedure UNATE_COMPLEMENT(F)

// F is a unate cover

1. Compute Personality Matrix M of F .
2. Record for every variable, if F is monotone increasing or decreasing and store in an array V .
3. $\overline{M} = \text{PERS_UNATE_COMPLEMENT}(M)$;
4. Get back the cover of $\overline{F}$ from $\overline{M}$ and V .

Procedure

PERS_UNATE_COMPLEMENT(M)

Begin

$$\overline{M} = \phi;$$

Compute $\overline{M}$ on a special case and return.

If not a special case, select the variable x_k which appears in the largest cube and of all such variables find the one that appears frequently. This can be got from M .

Find cofactors of M with respect to x_k and complement them recursively.

Merge the results depending on whether F is monotonically increasing on x_k or otherwise.

End.

Special Cases

When there is a row of all 0's in M , then the function is a tautology and its complement is a contradiction, i.e. empty.

When M is empty, then its complement is a tautology.

M has only one term, then compute its complement using Demorgan's law

These concepts will be used in the context of IRREDUNDANT_COVER computations, which are to be presented in the Seminars.

Simplify(F)

// F is a cover

Begin

If (F is unate) return

$(F' \leftarrow UNATE_SIMPLIFY(F))$

$j \leftarrow BINATE_SELECT(F);$

$F' \leftarrow MWC(x_j SIMPLIFY(F_{x_j})$

$+ \overline{x_j} SIMPLIFY(F_{\overline{x_j}}));$

if $|F| < |F'|$ return(F) else return (F');

end

Case Study:

$$f = \bar{B}\bar{C}\bar{D}\bar{F} + \bar{A}\bar{B}CE + ACF$$

$$+ \bar{A}BD + B\bar{E}F + B\bar{C}\bar{D}F + A\bar{B}C\bar{D}\bar{E} +$$

$$\bar{B}\bar{C}\bar{D}F + A\bar{B}C\bar{D}E + B\bar{C}\bar{D}F + ABD\bar{F} + ABDF.$$

The function in matrix form

$$\begin{vmatrix} 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 & 2 & 1 \\ 0 & 1 & 2 & 1 & 2 & 2 \\ 2 & 1 & 2 & 2 & 0 & 1 \\ 2 & 1 & 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 & 2 & 0 \\ 1 & 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 2 & 1 & 2 & 1 \end{vmatrix}$$

The most binate variable is B

The cofactors got by simplification for B are

$$\begin{vmatrix} 1 & 2 & 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 1 & 2 & 2 \\ 2 & 2 & 2 & 2 & 0 & 1 \\ 2 & 2 & 2 & 2 & 0 & 2 \end{vmatrix}$$

The cofactors got by simplification for $\bar{B}$ are

$$\begin{vmatrix} 1 & 2 & 1 & 2 & 2 & 1 \\ 0 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 0 & 0 & 2 & 2 \\ 1 & 2 & 2 & 0 & 2 & 2 \end{vmatrix}$$

Rows 1 and 4 of cofactor of B contains rows 1 and 3 of cofactor of $\bar{B}$

The merge returns

$$\begin{vmatrix} 1 & 1 & 2 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 & 2 \\ 2 & 1 & 2 & 2 & 0 & 1 \\ 2 & 1 & 2 & 2 & 0 & 2 \\ 0 & 0 & 1 & 2 & 1 & 2 \\ 1 & 0 & 2 & 0 & 2 & 2 \\ 1 & 2 & 1 & 2 & 2 & 1 \\ 2 & 2 & 0 & 0 & 2 & 2 \end{vmatrix}$$

The function is $ABF + BD + B\bar{E}F + B\bar{C} + \bar{A}BCE + A\bar{B}\bar{D} + ACF + \bar{C}\bar{D}$.

A more powerful minimizer

$$BD + B\bar{E}F + \bar{A}\bar{B}CE + A\bar{B}\bar{D} + ACF + \bar{C}\bar{D}$$

Learn this in the seminar and we shall close synthesis/minimizations with this. Thanks!!!