HEURISTIC TWO-LEVEL LOGIC OPTIMIZATION

©Giovanni De Micheli

Stanford University
Outline

- Heuristic logic minimization.
- Principles.
- Operators on logic covers.
- Espresso.
Heuristic minimization

- Provide irredudant covers with ’reasonably small’ cardinality.
- Fast and applicable to many functions.
- Avoid bottlenecks of exact minimization:
  - Prime generation and storage.
  - Covering.
Heuristic minimization
Principles

• Local minimum cover:
  – Given initial cover.
  – Make it prime.
  – Make it irredundant.

• Iterative improvement:
  – Improve on cardinality by ’modifying’ the implicants.
Heuristic minimization
Operators

- **Expand:**
  - Make implicants prime.
  - Remove covered implicants.

- **Reduce:**
  - Reduce size of each implicant while preserving cover.

- **Reshape:**
  - Modify implicant pairs: enlarge one and reduce the other.

- **Irredundant:**
  - Make cover irredundant.
Example

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0010$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0100$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0110$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1000$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1010$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0101$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0111$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1001$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1011$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1101$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| $\alpha$ | $0**0$ | $1$  |
| $\beta$  | $*0*0$ | $1$  |
| $\gamma$ | $01**$ | $1$  |
| $\delta$ | $10**$ | $1$  |
| $\epsilon$ | $1*01$ | $1$  |
| $\zeta$  | $*101$  | $1$   |
Example Expansion

- Expand 0000 to $\alpha = 0**0$.
  - Drop 0100, 0010, 0110 from the cover.

- Expand 1000 to $\beta = *0*0$.
  - Drop 1010 from the cover.

- Expand 0101 to $\gamma = 01**$.
  - Drop 0111 from the cover.

- Expand 1001 to $\delta = 10**$.
  - Drop 1011 from the cover.

- Expand 1101 to $\epsilon = 1*01$.

- Cover is: $\{\alpha, \beta, \gamma, \delta, \epsilon\}$. 
Example
Example Reduction

- Reduce $0**0$ to nothing.

- Reduce $\beta = *0*0$ to $\tilde{\beta} = 00*0$

- Reduce $\epsilon = 1*01$ to $\tilde{\epsilon} = 1101$

- Cover is: $\{\tilde{\beta}, \gamma, \delta, \tilde{\epsilon}\}$.
Example
Example
Reshape

• Reshape \{\tilde{\beta},\delta\} to: \{\beta,\tilde{\delta}\}

  — where \tilde{\delta} = 10 \times 1 .

• Cover is: \{\beta,\gamma,\tilde{\delta},\tilde{\epsilon}\}. 
Example
Example
Second expansion

- Expand $\tilde{\delta} = 10\ast1$ to $\delta = 10\ast\ast$.

- Expand $\tilde{\epsilon} = 1101$ to $\epsilon = 1\ast01$. 
Example
Example
(MINI summary)

• Expansion:
  – Cover: \( \{\alpha, \beta, \gamma, \delta, \epsilon\} \).
  – Prime, redundant, minimal w.r. to scc.

• Reduction:
  – \( \alpha \) eliminated.
  – \( \beta = \ast 0 \ast 0 \) reduced to \( \tilde{\beta} = 00 \ast 0 \).
  – \( \epsilon = 1 \ast 01 \) reduced to: \( \tilde{\epsilon} = 1101 \).
  – Cover: \( \{\tilde{\beta}, \gamma, \delta, \tilde{\epsilon}\} \).

• Reshape:
  – \( \{\tilde{\beta}, \delta\} \) reshaped to: \( \{\beta, \tilde{\delta}\} \) where \( \tilde{\delta} = 10 \ast 1 \).

• Second expansion:
  – Cover: \( \{\beta, \gamma, \delta, \epsilon\} \).
  – Prime, irredundant.
Alternative example (ESPRESSO)

- Expansion:
  - Cover: $\{\alpha, \beta, \gamma, \delta, \epsilon\}$.
  - Prime, redundant, minimal w.r. to scc.

- Irredundant:
  - Cover: $\{\beta, \gamma, \delta, \epsilon\}$.
  - Prime, irredundant.
Example
Expand
naive implementation

• For each implicant
  • For each care literal
    * Raise it to don’t care if possible.
  • Remove all covered implicants.

• Problems:
  • Validity check.
  • Order of expansions.
Validity check

• Espresso, MINI:
  – Check *intersection* of expanded implicant with OFF-set.
  – Requires complementation.

• Presto:
  – Check *inclusion* of expanded implicant in the union of the ON-set and DC-set.
  – Can be reduced to recursive tautology check.
Heuristics

• Expand first cubes that are unlikely to be covered by other cubes.

• Selection:
  – Compute vector of column sums.
  – Weight: inner product of cube and vector.
  – Sort implicants in ascending order of weight.

• Rationale:
  – Low weight correlates to having few 1s in densely populated columns.
Example

\[ f = a'b'c' + ab'c' + a'bc' + a'b'c \]

DC-set = \( abc' \)

\[
\begin{array}{ccc}
10 & 10 & 10 \\
01 & 10 & 10 \\
10 & 01 & 10 \\
10 & 10 & 01 \\
\end{array}
\]

• Ordering:

  – Vector: \([313131]^T\)

  – Weights: \((9, 7, 7, 7)\).

• Select second implicant.
Example (2)
Example (3)

- **OFF-set:**

  
  \[
  \begin{array}{ccc}
  01 & 11 & 01 \\
  11 & 01 & 01 \\
  \end{array}
  \]

- **Expand 01 10 10:**

  - 11 10 10 valid.
  
  - 11 11 10 valid.
  
  - 11 11 11 invalid.

- **Update cover to:**

  \[
  \begin{array}{ccc}
  11 & 11 & 10 \\
  10 & 10 & 01 \\
  \end{array}
  \]
Example (4)

11 11 10
10 10 01

- Expand 10 10 01:
  - 11 10 01 invalid.
  - 10 11 01 invalid.
  - 10 10 11 valid.

- Expanded cover:

  11 11 10
  10 10 11
Expand

• Smarter heuristics for choosing literals to be expanded.

• Four step procedure in Espresso.

• Rationale:
  
  – Raise literals so that expanded implicant:

  * Covers a maximal set of cubes.

  * Making it as large as possible.
Definitions

- *free*:
  - Set of entries that can be raised to 1.

- *Overexpanded cube*
  - Cube whose entries in *free* are raised.

- *Feasible cover*
  - Expand a cube to cover another one while keeping it as an implicant of the function.
Expand in ESPRESSO

- **Determine the essential parts.**
  - Determine which entries can never be raised, and remove them from \( free \).
  - Determine which parts can always be raised, raise them, and remove them from \( free \).

- **Detection of feasibly covered cubes.**
  - If there is an implicant \( \beta \) whose supercube with \( \alpha \) is feasible, repeat the following steps.
    * Raise the appropriate entry of \( \alpha \) and remove it from \( free \).
    * Remove from \( free \) entries that can never be raised or that can always be raised and update \( \alpha \).

- **Expansion guided by the overexpanded cube.**
  - While the overexpanded cube of \( \alpha \) covers some other cubes of \( F \), repeat the following steps.
    * Raise a single entry of \( \alpha \) as to overlap a maximum number of those cubes.
    * Remove from \( free \) entries that can never be raised or that can always be raised and update \( \alpha \).

- **Find the largest prime implicant.**
  - Formulate a covering problem and solve it by a heuristic method.
Reduce

- Sort implicants:
  - Heuristic: sort by descending weight.

- For each implicant:

- Lower as many * as possible to 1 or 0.

- **Theorem:**
  - Let $\alpha \in F$ and $Q = F \cup D - \{\alpha\}$.
  - Then, the maximally reduced cube is:
    $\tilde{\alpha} = \alpha \cap \text{supercube}(Q'_\alpha)$. 
Example

• Expanded cover:

  11 11 10
  10 10 11

• Select first implicant:
  – cannot be reduced.

• Select second implicant:
  – Reduced to 10 10 01

• Reduced cover:

  11 11 10
  10 10 01
Irredundant cover
Irredundant cover

- Relatively essential set $E^r$
  - Implicants covering some minterms of the function not covered by other implicants.

- Totally redundant set $R^t$
  - Implicants covered by the relatively essentials.

- Partially redundant set $R^p$
  - Remaining implicants.
Irredundant cover

- Find a subset of $R^p$ that, together with $E^r$, covers the function.

- Modification of the tautology algorithm:
  - Each cube in $R^p$ is covered by other cubes.
  - Find mutual covering relations.

- Reduces to a covering problem:
  - Heuristic algorithm.
Example

\[
\begin{array}{c|ccc}
\alpha & 10 & 10 & 11 \\
\beta & 11 & 10 & 01 \\
\gamma & 01 & 11 & 01 \\
\delta & 01 & 01 & 11 \\
\epsilon & 11 & 01 & 10 \\
\end{array}
\]

- \(E^r = \{\alpha, \epsilon\}\)
- \(R^t = \emptyset\)
- \(R^p = \{\beta, \gamma, \delta\}\).
Example (2)

- Covering relations:
  
  - $\beta$ is covered by $\{\alpha, \gamma\}$.
  
  - $\gamma$ is covered by $\{\beta, \delta\}$.
  
  - $\delta$ is covered by $\{\gamma, \epsilon\}$.

- Minimum cover: $\gamma \cup E^r$
Espresso algorithm

- Compute the complement.

- Extract essentials.

- Iterate:
  - Expand, irredundant, reduce.

- Cost functions:
  - Cover cardinality $\phi_1$.
  - Weighed sum of cube and literal count $\phi_2$. 
Espresso algorithm

\[
\text{espresso}(F, D) \{
    R = \text{complement}(F \cup D);  \\
    F = \text{expand}(F, R);  \\
    F = \text{irredundant}(F, D);  \\
    E = \text{essentials}(F, D);  \\
    F = F \setminus E;  \\
    D = D \cup E;  \\
    \text{repeat} \{  \\
        \phi_2 = \text{cost}(F);  \\
        \text{repeat} \{  \\
            \phi_1 = |F|;  \\
            F = \text{reduce}(F, D);  \\
            F = \text{expand}(F, R);  \\
            F = \text{irredundant}(F, D);  \\
        \} \text{ until } (|F| \geq \phi_1);  \\
        F = \text{last_gasp}(F, D, R);  \\
    \} \text{ until } (\text{cost}(F) \geq \phi_2);  \\
    F = F \cup E;  \\
    D = D \setminus E;  \\
    F = \text{make_sparse}(F, D, R);  \\
\}
\]
Summary
heuristic minimization

- Heuristic minimization is iterative.

- Few operators applied to covers.

- Underlying mechanism:
  - Cube operation.
  - Unate recursive paradigm.

- Efficient algorithms.