Lecture 12

Computer-Aided Heuristic Two-level Logic Minimization
Computer-Aided Heuristic Two-level Logic Minimization

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

Disclaimer: lecture notes based on originals by Giovanni De Micheli
Heuristic minimization

- Provide irredundant 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 implicant enabling the reduction of another

• Irredundant:
  – make cover irredundant
### Example

<table>
<thead>
<tr>
<th>on-set</th>
<th>0000</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>prime implicants</th>
<th>α</th>
<th>0**0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>β</td>
<td><em>0</em>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>γ</td>
<td>01**</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>δ</td>
<td>10**</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ε</td>
<td>1*01</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ζ</td>
<td>*101</td>
<td>1</td>
</tr>
</tbody>
</table>
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 reduction

• Reduce $0**0$ to nothing
• Reduce $\beta = *0*0$ to $\bar{\beta} = 00*0$
• Reduce $\varepsilon = 1*01$ to $\bar{\varepsilon} = 1101$
• Cover is \{\bar{\beta}, \gamma, \delta, \varepsilon\}
Example reshape

- Reshape \{\beta, \delta\} to \{\beta, \delta\}
- where \delta = 10\ast 1
- Cover is \{\beta, \gamma, \delta, \varepsilon\}
Example second expansion

• Expand $\underline{\delta} = 10*1$ to $\delta = 10**$
• Expand $\underline{\varepsilon} = 1101$ to $\zeta = *101$
Summary of Example

• Expansion:
  – Cover: \{\alpha, \beta, \gamma, \delta, \varepsilon\}
  – prime, redundant, minimal w.r.to single cube containment

• Reduction:
  – \alpha eliminated
  – \beta = *0*0 reduced to \beta = 00*0
  – \varepsilon = 1*01 reduced to \varepsilon = 1101
  – Cover: \{\beta, \gamma, \delta, \varepsilon\}

• Reshape:
  – \{\beta, \delta\} reshaped to \{\beta, \delta\} where \delta = 10*1

• Second expansion:
  – Cover: \{\beta, \gamma, \delta, \zeta\}
  – prime, irredudnant (= minimal)
Expand
naive implementation

• For each implicant
  – for each non-* literal (*care* literal)
    ‣ raise it to * (*don’t care*) if possible
  – remove all covered implicants

• Problems:
  – check validity of expansion: 2 ways
    ‣ non intersection of expanded implicant with OFF-set
      • requires complementation of ON-set
    ‣ expanded implicant covered by union of ON-set and DC-set
      • can be reduced to recursive tautology check
  – order of expansions
Heuristics

• First expand cubes which are unlikely to be covered by other cubes
  – Selection: choose implicants with least number of literals in common with other implicants
  – Example: \( f = a'b'c'd' + ab'cd + a'b'c'd \) choose \( ab'cd \)

• Choose expansions to cover the largest number of minterms possible (\( \Rightarrow \) prime implicant)
Reduce Example

• Expanded cover:
  \[ \alpha \quad **1 \]
  \[ \beta \quad 00* \]

• Select \( \alpha \): cannot be reduced and still cover the ON-set

• Select \( \beta \): reduced to
  \[ \beta \quad 001 \]

• Reduced cover:
  \[ \alpha \quad **1 \]
  \[ \beta \quad 001 \]
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 goal and example

• Goal: find a subset of $R^p$ that, together with $E^r$, covers the function

• Example:

\[
\begin{array}{l}
\alpha \quad 00^* \\
\beta \quad *01 \\
\gamma \quad 1*1 \\
\delta \quad 11^* \\
\varepsilon \quad *10 \\
\end{array}
\]

• $E^r = \{ \alpha, \varepsilon \}$

• $R^t = \{ \}$

• $R^p = \{ \beta, \gamma, \delta \}$
Example: continued

• Covering relations:
  – $\beta$ is covered by \{\(\alpha\), \(\gamma\}\}
  – $\gamma$ is covered by \{\(\beta\), \(\delta\)\}
  – $\delta$ is covered by \{\(\gamma\), \(\varepsilon\)\}

• Minimum cover: $\gamma \cup E^r$
Espresso Algorithm

• Compute the complement
• Extract essentials
• Iterate:
  – expand, irredundant, reduce
• Cost functions:
  – cover cardinality $\mathcal{O}_1$
  – weighed sum of cube and literal count $\mathcal{O}_2$
Espresso algorithm

Espresso($F,D$) {
    $R = \text{complement}(F \cup D)$;
    $F = \text{expand}(F, R)$;
    $F = \text{irredundant}(F, D)$;
    $E = \text{essentials}(F, D)$;
    $F = F - E$;
    $D = D \cup E$;
    repeat {
        $\varnothing_2 = \text{cost}(F)$;
        repeat {
            $\varnothing_1 = |F|$;
            $F = \text{reduce}(F, D)$;
            $F = \text{expand}(F, R)$;
            $F = \text{irredundant}(F, D)$;
        } until ($|F| \geq \varnothing_1$);
        $F = \text{last_gasp}(F, D, R)$;
    } until cost($F$) $\geq \varnothing_2$;
    $F = F \cup E$;
    $D = D - E$;
    $F = \text{make_sparse}(F, D, R)$;
}
Summary
heuristic minimization

• Heuristic minimization is iterative
• Few operators applied to covers
• Underlying mechanism
  – cube operation
  – recursive paradigm
• Efficient algorithms
• Preview: next lecture covers efficient boolean representations for computer manipulation