APPROACHES TO FUZZY LOGIC DECOMPOSITION

- Graphical Representations
- Kandel's and Francioni's Approach
- Fuzzy to Multiple-valued Function Conversion Approach
- Fuzzy Logic Decision Diagrams Approach
- Fuzzy Logic Multiplexer
Fuzzy Logic Decision Diagrams Approach
Result of Example using (FLDD)
APPROACHES TO FUZZY LOGIC DECOMPOSITION

- Graphical Representations
- Kandel's and Francioni's Approach
- Fuzzy to Multiple-valued Function Conversion Approach
- Fuzzy Logic Decision Diagrams Approach
  ➔ Fuzzy Logic Multiplexer
Fuzzy Logic Multiplexer

\[ d_0, d_1, d_2, d_3 \]

\[ x, \overline{x}, x\overline{x} \]

Output

\[ d_0, d_1, d_2, d_3 \]

\[ \text{Min}, \text{Min}, \text{Max} \]

Output

\[ d_3 xx' \]
Fuzzy Logic Circuit Implemented using Multiplexers
Contents

- Fuzzy logic
- Fuzzy logic systems applications
- Approaches to fuzzy logic decomposition
  ➤ Decomposition program
- Conclusion
DECOMPOSITION PROGRAM

- Need to Decompose Multiple-valued Functions and Relations
- Decomposition Structure
- Multiple-Valued Cube Diagram Bundles
- Upgrading Generalized Universal Decomposer (GUD) to Multiple-Valued Generalized Universal Decomposer (MVGUD)
- Upgrading MVGUD to Relation Multiple-Valued Generalized Universal Decomposer (RMVGUD)
- Results of Using RMVGUD
Need to Decompose Multiple-valued Functions and Relations

- Multiple-valued and Inconsistent Data
- Ways to Create Relations
  - Decomposition Process to Create Relations
  - Program to Change Inconsistency data into Relations
Multiple-Valued Cube Diagram

Bundles
Upgrading GUD to MVGUD
Upgrading MVGUD to RMVGUD

- Modify MVGUD to Read Relations
- Compatibility Checking and Correction for Relations Example
- New Data Structure for Writing Decomposed Relations to Files
Decomposition Structure

- General flow chart of GUD, MVGUD, and RMVGUD Program.
Multiple-Valued Cube Diagram Bundles

- Multiple-Valued Cube Diagram Bundles (MVCDB) internal data structure to hold binary, multiple-valued, and relations.
Upgrading GUD to MVGUD

- Change the reader to read in multiple-valued functions from file.
- Change encoding from binary to multiple-valued.
- Change writer to write out multiple-valued functions to files.
- Need new way of verifying results.
Upgrading MVGUD to RMVGUD

- Modify MVGUD to Read Relations
- Compatibility Checking and Correction for Relations
- New Data Structure for Writing Decomposed Relations to Files
Compatibility Checking and Correction for Relations Example

- Function that needs checked and corrected shown in a decomposition-map.

<table>
<thead>
<tr>
<th></th>
<th>cd</th>
<th>CO</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a b</td>
<td>00</td>
<td>0.1</td>
<td>0.3</td>
<td>2.3</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>01</td>
<td>1.2</td>
<td>—</td>
<td>2.3</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.3</td>
<td>0.4</td>
<td>1.4</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0</td>
<td>0.3</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Compatibility Graph Show Clique

- Clique before checking and correction:
  - clique 0 = 0 1 2
  - clique 1 = 0 3
- Clique after:
  - clique 0 = 0
  - clique 1 = 0 3
  - clique 2 = 1 2
- Compatibility graph and corrected cliques shown left
New Data Structure for Writing Decomposed Relations to Files

- Data structure to store relations before printing relation to output file
- Matrix of lists
- Efficient when few relations
## Results of Relation Decomposition

<table>
<thead>
<tr>
<th>File</th>
<th>Input File</th>
<th></th>
<th>Output File</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of Inputs</td>
<td>No. of Cubes</td>
<td>No. of cubes</td>
<td>No. of Blocks</td>
</tr>
<tr>
<td>hayes</td>
<td>4</td>
<td>132</td>
<td>68</td>
<td>3</td>
</tr>
<tr>
<td>flare 1</td>
<td>10</td>
<td>323</td>
<td>157</td>
<td>21</td>
</tr>
<tr>
<td>flare 2</td>
<td>10</td>
<td>1066</td>
<td>244</td>
<td>18</td>
</tr>
</tbody>
</table>
CONCLUSION

• Advantages of two new approaches to fuzzy function decomposition
  – Eliminates the need for time-consuming conversion to canonical form
  – Eliminates the use of S-maps
  – Enables decomposition of larger size

• Decomposes relations
Backup

• Uses of Fuzzy Logic Systems
• Fuzzy Logic Systems are Best Used in These Areas
• Where Fuzzy Logic Systems are Not the Best Solution
• Advantage of Fuzzy Logic Control Systems over Traditional or Conventional Control Systems
• Implementation and Future Trend of Fuzzy Logic Systems
Sources

Paul Burkey
Compatibility Checking and Correction for Relations Example

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.1</td>
<td>0.3</td>
<td>2.3</td>
<td>1.3</td>
</tr>
<tr>
<td>01</td>
<td>1.2</td>
<td></td>
<td>2.3</td>
<td>0.1</td>
</tr>
<tr>
<td>10</td>
<td>0.3</td>
<td>0.4</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Compatibility Graph Show
Clique
New Data Structure for Writing Decomposed Relations to Files
## Results of Relation Decomposition

<table>
<thead>
<tr>
<th>File</th>
<th>Input File</th>
<th>Output File</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of Inputs</td>
<td>No. of Cubes</td>
</tr>
<tr>
<td>hayes</td>
<td>4</td>
<td>132</td>
</tr>
<tr>
<td>flare 1</td>
<td>10</td>
<td>323</td>
</tr>
<tr>
<td>flare 2</td>
<td>10</td>
<td>1066</td>
</tr>
</tbody>
</table>
Hayes Result Block Diagram

age +

hobby 3

education +

marital +

3
category
Additional Topics

- Uses of Fuzzy Logic Systems
- Fuzzy Logic Systems are Best Used in These Areas
- Where Fuzzy Logic Systems are Not the Best Solution
- Advantage of Fuzzy Logic Control Systems over Traditional or Conventional Control Systems
- Implementation and Future Trend of Fuzzy Logic Systems
  - Fuzzy Logic System Software Tools
  - Fuzzy Logic System Hardware
  - Future Trends of Fuzzy Logic Systems
Sources

- Paul Burkey’s M.S. at PSU, 1999