MLL proof nets as error-correcting codes

Satoshi Matsuoka
AIST
The structure of the talk

1. Introduction to error-correcting codes
2. Introduction to MLL proof nets
3. How to analyze MLL proof nets using coding theory
4. Our results so far
Hamming \(<7,4>\) code

- A subset of \(\{0,1\}^{7}\) called code words
- Satisfying
  1. \(x_1 + x_2 + x_4 + x_5 = 0\)
  2. \(x_2 + x_3 + x_4 + x_6 = 0\)
  3. \(x_1 + x_3 + x_4 + x_7 = 0\)

where \(x_i \in \{0,1\}\)

  + is exclusive or (or parity check)
Hamming $<7,4>$ code (cont.)

1. $x_1 + x_2 + x_4 + x_5 = 0$
2. $x_2 + x_3 + x_4 + x_6 = 0$
3. $x_1 + x_3 + x_4 + x_7 = 0$
### Hamming <7,4> Code (cont.)

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Hamming <7,4> code (cont.)

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Hamming \( <7,4> \) code (cont.)

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Hamming $<7,4>$ code (cont.)

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Hamming $<7,4>$ code (cont.)

- $2^4 = 16$ words are (legitimate) codewords
- Other words ($2^7 - 2^4 = 112$) are not
Hamming $<7,4>$ code (cont.)

- distance of $w_1, w_2 \in \{0,1\}^{7}$
  $$d(w_1, w_2) = | \{ i | w_1(i) \neq w_2(i) \} |$$
- Example
  $$d(0101000, 00110011) = 4$$
- The distance of code $C$, $d(C)$:
  the minimum distance of different codewords
- Hamming $<7,4>$ code $C$ has $d(C) = 3$
Hamming $<7,4>$ code (cont.)

- So, Hamming $<7,4>$ code is 1-error correcting
Hamming <7,4> code (cont.)

- On the other hand, Hamming <7,4> code is **2-error detecting**

- But, **1-error correcting** and **2-error detecting** are not compatible
The structure of the talk

1. Introduction to error-correcting codes
2. Introduction to MLL proof nets
3. How to analyze MLL proof nets using coding theory
4. Our results so far
Links

- **ID-links**
  \[
  p \downarrow p
  \]

- **par-links**
  \[
  \begin{array}{c}
  A \\
  \circ
  \\
  B
  \end{array}
  \]

- **tensor-links**
  \[
  \begin{array}{c}
  A \\
  \otimes
  \\
  B
  \end{array}
  \]
MLL proof structure (also MLL proof net)

\[ \Theta \Gamma = \begin{array}{c}
p^\perp \\
p \\
p \\
p^\perp \\
\end{array} \begin{array}{c}
p \\
\otimes \\
p^\perp \\
\end{array} \begin{array}{c}
L \\
\otimes \\
( p \otimes p^\perp )
\end{array} \]
Graph-theoretic characterization theorem

- Theorem (Girard, Danos-Regnier)
  \( \Theta \) is MLL proof net
  iff
  for any DR-switching \( S \), the DR-graph \( \Theta_S \) is acyclic and connected
DR-graph 1 for Θ1

\[ S(L) = \text{Right} \]

\[
\begin{align*}
&\quad p^\perp \\
\quad &\quad p \\
\quad &\quad p \\
\quad &\quad p^\perp \\
\quad &\quad p \\
\quad &\quad p^\perp \\
\quad &\quad (p \otimes p^\perp) \\
\quad &\quad L \\
\quad &\quad p^\perp
\end{align*}
\]

acyclic and connected(tree)
DR-graph 2 for $\Theta 1$

$S(L) = \text{Left}$

```
\begin{array}{c}
p \perp \\
p \\
p \otimes p \perp \\
L \\
p \perp \otimes (p \otimes p \perp)
\end{array}
```

acyclic and connected (tree)
MLL proof structure (but not MLL proof net)

\[ \Theta 2 = \]

\[
\begin{array}{c}
\bot \\
\downarrow \\
\bot \otimes (p \otimes p^\bot) \\
\end{array}
\]
DR-graph 1 for $\Theta 2$

$S(L) = \text{Right}$

$\begin{array}{c}
\, \\
\downarrow \scriptstyle p^\perp \\
\downarrow \\
\scriptstyle L \\
\downarrow \\
\scriptstyle p \not\bowtie p^\perp \\
\downarrow \\
\scriptstyle p^\perp \not\bowtie (p \not\bowtie p^\perp) \\
\end{array}$

acyclic and connected (tree)
DR-graph 2 for $\Theta 2$

$S(L) = \text{Left}$

$p^\perp$ \quad p \quad p^\perp \quad p$

$p$ \quad L \quad $p^\perp$

$p^\perp \otimes (p \nabla p^\perp)$

has a cycle
The structure of the talk

1. Introduction to error-correcting codes
2. Introduction to MLL proof nets
3. How to analyze MLL proof nets using coding theory
4. Our results so far
The Basic Idea

• **PS-family**: a set of MLL proof structures such that each member is reachable from the other members by *several* tensor-par exchanges

• Partition MLL proof structures into PS-families

• Regard each PS-family as a code
One of four members of a PS-family

$\theta_1 =$

$\begin{array}{c}
p^\perp \\
\downarrow \\
p \otimes p^\perp \\
\downarrow \\
p^\perp \otimes (p \otimes p^\perp)
\end{array}$
One of four members of a PS-family

\[ \Theta 2 = \]

\[ p^\perp \quad p \quad p^\perp \quad p \]

\[ p \otimes (p \otimes p^\perp) \]
Hamming distance on a PS-family

- distance of $\Theta_1, \Theta_2 \in$ PS-family $F$
  \[ d(\Theta_1, \Theta_2) = \text{the number of “locations” where multiplicative links are different} \]
- For each PS-family $F$, $d(F)$ is the minimum distance of different MLL proof nets in $F$
Example

\[
\begin{align*}
\left\{ \begin{array}{c}
p^\perp \\ L \\ p \otimes (p \otimes p^\perp)
\end{array} \right. \\
\left\{ \begin{array}{c}
p \\ L \\ p \otimes (p \otimes p^\perp)
\end{array} \right. \\
= 2
\end{align*}
\]
The structure of the talk

1. Introduction to error-correcting codes
2. Introduction to MLL proof nets
3. How to analyze MLL proof nets using coding theory
4. Our results so far
First Question

• How do we have properties about $d(F)$?
Proposition

- Let $F$ be a PS-family.
  If $\Theta_1$ and $\Theta_2$ are MLL proof nets and both belong to $F$, then the number of ID-links (tensor-links, and par-links) of $\Theta_1$ is the same as that of $\Theta_2$. 
Theorem
If PS-family $F$ has more than two MLL proof nets, then $d(F)=2$.
So, such a PS-family is just one-error detecting.

Idea of Proof
If $\Theta, \Theta' \in F$, then we can have a sequence
\[ \Theta \Rightarrow \Theta_1 \Rightarrow \cdots \Rightarrow \Theta_n \Rightarrow \Theta' \]
such that $\Theta_1, \ldots, \Theta_n$ are MLL proof nets
where $\Theta_a \Rightarrow \Theta_b$ if $\Theta_b$ is obtained from $\Theta_a$ by
- replacing a tensor-link by a par-link and a par-link
  by a tensor-link exactly two times

Using graph-theoretic characterization theorem,
nontrivial (at least for me)
Using reduction to absurdity
Summary

- We can incorporate the notion of Hamming distance into MLL proof nets naturally
- Got an elementary result
- But it’s ongoing work
- Need to get more results (composition of two PS-families, characterization of PS-families with n MLL proof nets,....)
- The manuscript can be found in http://arxiv.org/abs/cs/0703018