Effective CNF Encodings for the Towers of Hanoi

Ruben Martins    Inês Lynce

IST/INESC-ID
Technical University of Lisbon, Portugal

LPAR 2008, Qatar
Towers of Hanoi (ToH)

• Only one disk may be moved at a time;
• No disk may be placed on the top of a smaller disk;
• Each move consists in taking the upper disk from one of the towers and sliding it onto the top of another tower.
Towers of Hanoi (ToH)

- Only one disk may be moved at a time;
- No disk may be placed on the top of a smaller disk;
- Each move consists in taking the upper disk from one of the towers and sliding it onto the top of another tower.
Why ToH?

- Existing CNF encodings for ToH are hard to solve for SAT solvers;
- Although SAT technology has never advocated being the best approach for ToH this comes as a surprise;
- We propose a new encoding that incorporates a key number of properties of the ToH which seem to be essential for solving the problem;
- Show that even though SAT solvers are becoming more efficient the modelling still has an importante role for an effective solution.
Given a ToH of size $n$, one may easily find a solution taking into account the solution for a ToH of size $n - 1$;
ToH Properties I

- Given a ToH of size \( n \), one may easily find a solution taking into account the solution for a ToH of size \( n - 1 \);
ToH Properties I

- Given a ToH of size $n$, one may easily find a solution taking into account the solution for a ToH of size $n - 1$;
- The order of the disks to be moved after moving the largest disk is exactly the same as before;
- Notice that there is a relation between the towers involved in the same movement before and after the largest disk is moved that consists in shifting the towers from/to where the disks have to be moved ($T_1 \rightarrow T_2$, $T_2 \rightarrow T_3$, $T_3 \rightarrow T_1$).
ToH Properties I

- Given a ToH of size $n$, one may easily find a solution taking into account the solution for a ToH of size $n - 1$;
- The order of the disks to be moved after moving the largest disk is exactly the same as before;
- Notice that there is a relation between the towers involved in the same movement before and after the largest disk is moved that consists in shifting the towers from/to where the disks have to be moved ($T_1 \rightarrow T_2$, $T_2 \rightarrow T_3$, $T_3 \rightarrow T_1$).
ToH Properties II

- ToH can be solved in $2^n - 1$ steps;
- Considering the relationship between the movement of the disks after/before moving the largest disk we only need to determine the first $2^{n-1} - 1$ steps
ToH Properties II

- ToH can be solved in $2^n - 1$ steps;
- Considering the relationship between the movement of the disks after/before moving the largest disk we only need to determine the first $2^{n-1} - 1$ steps
ToH Properties III

- When moving disks, no two odd/even disks can be moved next to each other.
When moving disks, no two odd/even disks can be moved next to each other.
• When moving disks, no two odd/even disks can be moved next to each other.
ToH Properties III

- When moving disks, no two odd/even disks can be moved next to each other.
ToH Properties III

- When moving disks, no two odd/even disks can be moved next to each other.
• When moving disks, no two odd/even disks can be moved next to each other.
When moving disks, no two odd/even disks can be moved next to each other.
When moving disks, no two odd/even disks can be moved next to each other.
ToH Properties IV

- All disks cycle in a given order between the towers:
  - If \( n \) is even the odd disks will cycle clockwise \((T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1)\) while the even disks will cycle counterclockwise \((T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1)\);
  - If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
All disks cycle in a given order between the towers:

- If $n$ is even the odd disks will cycle clockwise
  \((T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1)\) while the even disks will cycle counterclockwise \((T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1)\);
- If $n$ is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
All disks cycle in a given order between the towers:

- If \( n \) is even the odd disks will cycle clockwise \(( T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1 )\) while the even disks will cycle counterclockwise \(( T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1 )\);
- If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
ToH Properties IV

- All disks cycle in a given order between the towers:
  - If \( n \) is even the odd disks will cycle clockwise \((T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1)\) while the even disks will cycle counterclockwise \((T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1)\);
  - If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
ToH Properties IV

• All disks cycle in a given order between the towers:
  
  • If \( n \) is even the odd disks will cycle clockwise
    \((T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1)\) while the even disks will cycle
    counterclockwise \((T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1)\);
  
  • If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
ToH Properties IV

- All disks cycle in a given order between the towers:
  - If $n$ is even the odd disks will cycle clockwise ($T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$) while the even disks will cycle counterclockwise ($T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1$);
  - If $n$ is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
All disks cycle in a given order between the towers:

- If \( n \) is even the odd disks will cycle clockwise \((T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1)\) while the even disks will cycle counterclockwise \((T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1)\);
- If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
All disks cycle in a given order between the towers:

- If $n$ is even the odd disks will cycle clockwise ($T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$) while the even disks will cycle counterclockwise ($T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1$);
- If $n$ is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
All disks cycle in a given order between the towers:

- If \( n \) is even the odd disks will cycle clockwise \( (T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1) \) while the even disks will cycle counterclockwise \( (T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1) \);
- If \( n \) is odd the odd disks will cycle counterclockwise while the even disks will cycle clockwise.
Existing CNF Encodings for ToH

- The first CNF encoding uses the following set of variables:
  - \{on(d, dt, t), clear(dt, t), move(d, dt, dt', t)\}, where the variable move(d, dt, dt', t) is replaced by conjunction (obj(d, t) \land from(dt, t) \land to(dt', t)).
Existing CNF Encodings for ToH

- The first CNF encoding uses the following set of variables:
  - \{on(d, dt, t), clear(dt, t), move(d, dt, dt', t)\}, where the variable \(move(d, dt, dt', t)\) is replaced by conjunction \(obj(d, t) \land from(dt, t) \land to(dt', t)\).

- A recent encoding proposed by Prestwich produces a more compact formula since it restricts the action \(move(d, tw, tw', t)\) to consider only movements of disks between towers.
  - For example, for 5 towers the previous encoding requires 1,931 variables and 14,468 clauses, whereas this new encoding requires 821 variables and 6,457 clauses.
Effective CNF Encodings for ToH

• The Disk Parity encoding:
  • We improved Prestwich encoding by incorporating properties 2 and 3:
    • Property 2 is incorporated in the encoding by reducing the search space to the first $2^{n-1} - 1$ steps and modifying the goal state so that we have the larger disk on $T_1$ and the remaining disks on $T_2$;
    • Property 3 is encoded through additional clauses.
  • The Disk Cycle encoding:
    • This encoding extends the previous one by additionally incorporating property 4.
Effective CNF Encodings for ToH

• The Disk Parity encoding:
  • We improved Prestwich encoding by incorporating properties 2 and 3:
    • Property 2 is incorporated in the encoding by reducing the search space to the first $2^{n-1} - 1$ steps and modifying the goal state so that we have the larger disk on $T_1$ and the remaining disks on $T_2$;
    • Property 3 is encoded through additional clauses.

• The Disk Cycle encoding:
  • This encoding extends the previous one by additionally incorporating property 4.
Effective CNF Encodings for ToH

- The Disk Sequence encoding:
  - Property 1 recursively determines the disks to be moved at each step;
  - Taking into consideration this we can keep only the variables \( \text{on} \) and drop all the others;
  - Our set of variables is the following: \( \text{on}(d, tw, t) \) with \( 1 \leq d \leq n, 1 \leq tw \leq 3 \) and \( 0 \leq t \leq 2^{n-1} \);
  - By additionally incorporating properties 2 and 3 we are able to produce an encoding that is solved by only using unit propagation.
### Experimental Results I

**Table:** Results for the encodings with a time limit of 10,000 seconds.

<table>
<thead>
<tr>
<th>Size</th>
<th>Prestwich</th>
<th>Disk Parity</th>
<th>Disk Cycle</th>
<th>Disk Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.08</td>
<td>0.01</td>
<td>0.02</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.47</td>
<td>0.03</td>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>3.65</td>
<td>0.70</td>
<td>0.20</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>109.7</td>
<td>5.19</td>
<td>5.18</td>
<td>0.03</td>
</tr>
<tr>
<td>9</td>
<td>7126.57</td>
<td>79.11</td>
<td>7.65</td>
<td>0.09</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>1997.19</td>
<td>973.95</td>
<td>0.23</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>-</td>
<td>1206.37</td>
<td>0.56</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.32</td>
</tr>
</tbody>
</table>
Experimental Results II

**Table:** Results for the new encoding for the ToH.

<table>
<thead>
<tr>
<th>Size</th>
<th>#Vars</th>
<th>#Cls</th>
<th>Mem</th>
<th>GenTime</th>
<th>SolveTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>159,705</td>
<td>827,007</td>
<td>20.9</td>
<td>0.74</td>
<td>3.14</td>
</tr>
<tr>
<td>14</td>
<td>344,022</td>
<td>1,859,150</td>
<td>50.9</td>
<td>1.73</td>
<td>7.33</td>
</tr>
<tr>
<td>15</td>
<td>737,235</td>
<td>4,177,431</td>
<td>121.6</td>
<td>4.07</td>
<td>17.16</td>
</tr>
<tr>
<td>16</td>
<td>1,572,816</td>
<td>9,272,800</td>
<td>294.1</td>
<td>9.44</td>
<td>38.94</td>
</tr>
<tr>
<td>17</td>
<td>3,342,285</td>
<td>20,577,699</td>
<td>708.6</td>
<td>12.31</td>
<td>90.15</td>
</tr>
<tr>
<td>18</td>
<td>7,077,834</td>
<td>45,219,174</td>
<td>1,637.4</td>
<td>50.48</td>
<td>203.05</td>
</tr>
</tbody>
</table>
Conclusions

• Making use of properties of the ToH we were able to improve the previous CNF encodings;
• Taking advantage of those properties we were also able to produce a new encoding that is more compact and can solve ToH by only using unit propagation;
• This is an example that modelling as an important role for an effective solution since a problem that was intractable at first for SAT solvers can now be solved by only using unit propagation.