**Unified Solver Strategy for Floating-Point**
Jaideep Ramachandran, Northeastern University
jaideep@ccs.neu.edu

**GOAL**
To come up with a strategy for solving Floating-Point Arithmetic formulas that takes into account:
- Nature of input formulas (e.g., complexity)
- Applicability of abstractions
  * Performance of “proxy theories”

**EXAMPLES**
- Linear with arithmetic operations reordered
  $$|(x + (y + z)) - ((x + y) + z)| > \epsilon$$
- Non-linear
  $$10.25 \leq x^2 + y^2 \leq 10.5$$
- Non-linear with arithmetic operations reordered
  $$|(x + y)^2 - ((x^2 + (2 \times x) \times y) + y^2)| > \epsilon$$
  May need different strategies to solve!

**SOLUTION: Unified Strategy**

Require: $f$: FPA formula
1: if Linear($f$) then
2: return Molly$_{MRFPA}(f)$ // mixed real-float reasoning
3: result := Molly$_{RA}(f)$ // pure real abstraction
4: if result $\neq$ failed then
5: return result
6: result := Molly$_{d\text{REAL}}(f)$ // numerical solving
7: if result $\neq$ failed then
8: return result
9: return Molly$_{RPFPA}$ // reduced precision

**Model Lifting Architecture**

![Diagram showing the model lifting architecture]

**Molly Configurations**

<table>
<thead>
<tr>
<th>Spec</th>
<th>Name</th>
<th>MOLLY$^{RA}$</th>
<th>Lazy Realizer</th>
<th>MOLLY$^{MRFPA}$</th>
<th>MOLLY$^{d\text{REAL}}$</th>
<th>MOLLY$^{RPFPA}$</th>
<th>APPROX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy theory</td>
<td>RA</td>
<td>RA</td>
<td>RA</td>
<td>Reals $+$ $\delta$-sat</td>
<td>RPFPA</td>
<td>RPFPA</td>
<td></td>
</tr>
<tr>
<td>Proxy solver</td>
<td>Z3</td>
<td>Realizer++</td>
<td>Realizer++</td>
<td>dReal</td>
<td>Mathsat</td>
<td>Mathsat</td>
<td></td>
</tr>
<tr>
<td>Lifting?</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
</tr>
<tr>
<td>Refinement?</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Experimental evaluation indicates there is no clear winning configuration across all formulas. Hence the need for a unified strategy.