Exploring Predictability of SAT/SMT solvers

Robert Brummayer\textsuperscript{1} \hspace{1cm} Duckki Oe\textsuperscript{2} \hspace{1cm} Aaron Stump\textsuperscript{2}

\textsuperscript{1}Johannes Kepler University
Linz, Austria

\textsuperscript{2}The University of Iowa
Iowa City, Iowa, USA

EMSQMS’10
Speed is Everything!

- SAT/SMT solvers have made tremendous advances in performance.
- > 100k variables, > 1M clauses.
- Due to:
  - algorithmic advances (clause learning, integration, theory solvers, etc.)
  - good heuristics
  - good engineering
- SAT Competition, SMT-COMP reward speed.
- Applications like program verification need raw power.
Is Speed Everything?

- State-of-the-art on SMT-COMP stagnant.
- But new theories: **expressiveness**.
- For some applications, many easy queries: **embeddability**.
- Even for tough benchmarks, **predictability** an issue.
  - Steve Miller (Rockwell Collins): solver performance is unpredictable.
  - Small change to model $\Rightarrow$ big change to run-time.
  - Problematic for development.
This Talk: Exploring Predictability

**Motivation: incrementally changing formulas.**

- planning/AI applications
  - answer queries based on policies, observations.
  - incrementally changing observations => similar queries.
- software verification
  - call solver to compare code to spec.
  - gradually evolving code/spec => evolving formulas.
- static analysis
  - analyze paths through code.
  - changing code => gradually changing queries.

**Issues with unpredictability.**

- less predictable => harder to embed (e.g., in a compiler).
- end-user frustration.
Measuring Predictability

Population: set of similar instances

- Pick a SAT formula as *seed formula*.
- Generate 50 random variations.
- Run solver to get distribution of solving times.
- Measure of predictability: the standard deviation.

![Graph showing frequency of solving times](image)
Types of changes

Semantics-preserving:
- $l$: literals in each clause are reordered
- $c$: clauses of the formula are reordered
- $n$: variable names are changed
- $lc$: a combination of $l$ and $c$ variations
- $nlc$: a combination of $n$, $l$ and $c$ variations

Semantics-modifying:
- $nlcx$: $nlc$ + one literal of clause is changed (0.01%)
- $nlca$: $nlc$ + one literal is dropped/added to clause (0.01%)

- unary clauses are not modified
- preserves literal/clause ratio
Experiments

- 5 solvers: high ranking in SAT Competition 2009.
- 13 seed formulas: 5 easy, 6 medium, 2 hard.
- Generate 50 instances for each change-type.
- For each seed formula, each solver:
  - Run solver on the 50 instances for the seed.
  - Compute std. dev. of runtimes.
- Graph all std. devs.
Run-Time Std Devs – Semantics Preserving

Brummayer, Oe, Stump

Exploring Predictability of SAT/SMT solvers

EMSQMS’10
Run-Time Std Devs – Semantics Modifying

Compare with:

Brummayer, Oe, Stump
Exploring Predictability of SAT/SMT solvers
EMSQMS'10
Improving Predictability for SMT

Multiple runs on similar formulas:

\[ \Phi \rightarrow \text{SMT Solver} \rightarrow \text{Answer} \]

\[ \Phi' \rightarrow \text{SMT Solver} \rightarrow \text{Answer}' \]

\[ \Phi'' \rightarrow \text{SMT Solver} \rightarrow \text{Answer}'' \]

\[ \cdot \cdot \cdot \]
Multiple runs on similar formulas:

\[ \Phi \overset{\text{SMT Solver}}{\rightarrow} \text{Answer} \]
\[ \Phi' \overset{\text{SMT Solver}}{\rightarrow} \text{Answer}' \]
\[ \Phi'' \overset{\text{SMT Solver}}{\rightarrow} \text{Answer}'' \]
\[ \vdots \]

Idea: pass along some theory lemmas.
Improving Predictability for SMT

Multiple runs on similar formulas:

$\Phi$ → SMT Solver → Answer

$\Phi'$ → SMT Solver → Answer'

$\Phi''$ → SMT Solver → Answer''

\[ \cdots \]

Idea: pass along some theory lemmas.
Dumping theory lemmas

- theory lemmas valid => always safe to add.
- helpful once => may be helpful again.
- lots of lemmas => just dump 10%.
Experiments

- Modified CVC3 and opensmt to dump lemmas.
  - Open-source tools.
  - Very helpful developers (thanks Clark Barrett, Roberto Bruttomesso).
  - Not too hard to modify.
- Selected seed formulas from example divisions.
- Generate 11 mutants for each seed.
  - mutator based on Robert’s SMT fuzzzer/delta-debugger.
  - small number (4) of semantics-modifying changes.
- For each seed formula:
  1. Run solver on seed formula.
  2. Re-run on seed, dumping lemmas.
  3. Re-run on seed + lemmas.
  4. Run mutants.
  5. Re-run mutants + lemmas (from seed).
- Compare times for mutants, mutants + lemmas.
## Results for CVC3: QF_UFIDL

<table>
<thead>
<tr>
<th>name</th>
<th>orig</th>
<th>orig+lem</th>
<th>L</th>
<th>$\tilde{m}$</th>
<th>$\sigma_m$</th>
<th>$\bar{I}$</th>
<th>$\sigma_I$</th>
<th>$m/l$</th>
<th>$\sigma_m/\sigma_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LamportBakery14</td>
<td>3.26</td>
<td>6.1</td>
<td>44</td>
<td>2.49</td>
<td>0.4</td>
<td>2.46</td>
<td>1.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LamportBakery15</td>
<td>2.19</td>
<td>2.22</td>
<td>58</td>
<td>1.97</td>
<td>0.29</td>
<td>1.93</td>
<td>0.27</td>
<td>1.02</td>
<td>1.05</td>
</tr>
<tr>
<td>OOO5</td>
<td>3.16</td>
<td>2.56</td>
<td>62</td>
<td>2.66</td>
<td>0.37</td>
<td>2.55</td>
<td>0.16</td>
<td>1.05</td>
<td>2.23</td>
</tr>
<tr>
<td>sorted_noalloc3</td>
<td>4.86</td>
<td>4.52</td>
<td>61</td>
<td>4.13</td>
<td>0.37</td>
<td>4.34</td>
<td>0.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhard8</td>
<td>2.01</td>
<td>7.75</td>
<td>306</td>
<td>0.07</td>
<td>0.01</td>
<td>0.66</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOO8</td>
<td>3.71</td>
<td>8.57</td>
<td>40</td>
<td>3.51</td>
<td>0.51</td>
<td>3.62</td>
<td>2.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache_unbounded12</td>
<td>4.19</td>
<td>6.07</td>
<td>49</td>
<td>4.2</td>
<td>0.24</td>
<td>5.47</td>
<td>0.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sorted_noalloc5</td>
<td>4.74</td>
<td>4.48</td>
<td>93</td>
<td>3.94</td>
<td>0.44</td>
<td>4.36</td>
<td>0.33</td>
<td>1.32</td>
<td></td>
</tr>
<tr>
<td>OOO6</td>
<td>3.22</td>
<td>6.43</td>
<td>40</td>
<td>2.78</td>
<td>0.41</td>
<td>3.99</td>
<td>1.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sorted_noalloc6</td>
<td>7.03</td>
<td>4.42</td>
<td>239</td>
<td>3.79</td>
<td>1.4</td>
<td>4.26</td>
<td>0.44</td>
<td>1.15</td>
<td>3.15</td>
</tr>
<tr>
<td>vhard5</td>
<td>0.49</td>
<td>1.08</td>
<td>85</td>
<td>0.04</td>
<td>0.01</td>
<td>0.15</td>
<td>0.03</td>
<td>1.2</td>
<td>1.12</td>
</tr>
<tr>
<td>cache_unbounded15</td>
<td>2.85</td>
<td>2.36</td>
<td>81</td>
<td>2.87</td>
<td>0.92</td>
<td>2.24</td>
<td>0.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache_unbounded14</td>
<td>4.04</td>
<td>4.16</td>
<td>35</td>
<td>4.05</td>
<td>0.75</td>
<td>4.13</td>
<td>0.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhard6</td>
<td>0.85</td>
<td>2.16</td>
<td>138</td>
<td>0.04</td>
<td>0.01</td>
<td>0.27</td>
<td>0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache_unbounded17</td>
<td>7.78</td>
<td>19.83</td>
<td>114</td>
<td>4.07</td>
<td>2.04</td>
<td>2.27</td>
<td>5.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache_unbounded16</td>
<td>4.</td>
<td>4.04</td>
<td>35</td>
<td>4.01</td>
<td>0.74</td>
<td>4.04</td>
<td>0.17</td>
<td>1.06</td>
<td>4.3</td>
</tr>
<tr>
<td>vhard16</td>
<td>19.3</td>
<td>120.</td>
<td>2235</td>
<td>0.16</td>
<td>0</td>
<td>7.43</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhard9</td>
<td>2.77</td>
<td>15.39</td>
<td>427</td>
<td>0.08</td>
<td>0.01</td>
<td>0.95</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhard18</td>
<td>29.23</td>
<td>120.</td>
<td>3151</td>
<td>0.19</td>
<td>0</td>
<td>13.08</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vhard11</td>
<td>5.03</td>
<td>31.53</td>
<td>760</td>
<td>0.12</td>
<td>0.85</td>
<td>1.79</td>
<td>33.98(1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

orig = solver on seed
$\tilde{m}$ = mean time, mutants
$\sigma_m$ = std. dev, mutants
L = num dumped lemmas
$\bar{I}$ = mean time, mutants+lemmas
$\sigma_I$ = std. dev, mutants+lemmas
## Results for \texttt{opensmt}: QF\textsubscript{LRA}

<table>
<thead>
<tr>
<th>name</th>
<th>orig</th>
<th>orig+lem</th>
<th>L</th>
<th>(\bar{m})</th>
<th>(\sigma_m)</th>
<th>(\bar{l})</th>
<th>(\sigma_l)</th>
<th>(m/l)</th>
<th>(\sigma_m/\sigma_l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sc-10.ind</td>
<td>6.07</td>
<td>7.55</td>
<td>15</td>
<td>0.2</td>
<td>2.89</td>
<td>0.2</td>
<td>3.35</td>
<td></td>
<td>1.1</td>
</tr>
<tr>
<td>safety-10</td>
<td>1.27</td>
<td>1.36</td>
<td>18</td>
<td>0.54</td>
<td>1.33</td>
<td>0.59</td>
<td>1.06</td>
<td>1.1</td>
<td>1.25</td>
</tr>
<tr>
<td>p5-zenonum_s5</td>
<td>3.29</td>
<td>3.66</td>
<td>37</td>
<td>3.26</td>
<td>0.06</td>
<td>3.66</td>
<td>0.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety-11</td>
<td>1.42</td>
<td>1.65</td>
<td>17</td>
<td>0.66</td>
<td>2.87</td>
<td>0.74</td>
<td>1.72</td>
<td>1.35</td>
<td>1.66</td>
</tr>
<tr>
<td>uart-8.b</td>
<td>2.93</td>
<td>4.12</td>
<td>16</td>
<td>3.32</td>
<td>1.98</td>
<td>2.83</td>
<td>1.88</td>
<td>1.04</td>
<td>1.05</td>
</tr>
<tr>
<td>sc-11.ind</td>
<td>12.33</td>
<td>8.77</td>
<td>17</td>
<td>0.22</td>
<td>4.9</td>
<td>0.22</td>
<td>5.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-0_s10</td>
<td>3.74</td>
<td>4.42</td>
<td>6</td>
<td>6.14</td>
<td>2.06</td>
<td>5.33</td>
<td>0.56</td>
<td>1.21</td>
<td>3.66</td>
</tr>
<tr>
<td>Dep_s8.ms</td>
<td>3.54</td>
<td>2.57</td>
<td>32</td>
<td>1.86</td>
<td>0.45</td>
<td>1.69</td>
<td>0.45</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>uart-7.ind</td>
<td>3.38</td>
<td>3.06</td>
<td>21</td>
<td>0.15</td>
<td>1.49</td>
<td>0.16</td>
<td>1.17</td>
<td>1.2</td>
<td>1.26</td>
</tr>
<tr>
<td>sc-12.ind</td>
<td>19.71</td>
<td>11.25</td>
<td>18</td>
<td>0.24</td>
<td>8.28</td>
<td>0.25</td>
<td>8.23</td>
<td>1.01</td>
<td></td>
</tr>
<tr>
<td>uart-9.b</td>
<td>6.75</td>
<td>8.34</td>
<td>21</td>
<td>5.01</td>
<td>3.05</td>
<td>4.24</td>
<td>2.84</td>
<td>1.1</td>
<td>1.07</td>
</tr>
<tr>
<td>safety-13</td>
<td>2.75</td>
<td>3.01</td>
<td>24</td>
<td>1.02</td>
<td>5.2</td>
<td>0.74</td>
<td>4.81</td>
<td>1.11</td>
<td>1.08</td>
</tr>
<tr>
<td>io-safe-18</td>
<td>5.2</td>
<td>3.13</td>
<td>36</td>
<td>3.06</td>
<td>1.34</td>
<td>3.</td>
<td>1.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9clks.inv.b</td>
<td>5.07</td>
<td>6.4</td>
<td>6</td>
<td>0.41</td>
<td>3.15[1]</td>
<td>0.53</td>
<td>2.79[1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safety-16</td>
<td>1.06</td>
<td>2.5</td>
<td>16</td>
<td>1.62</td>
<td>12.63</td>
<td>1.75</td>
<td>20.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>io-safe-20</td>
<td>5.7</td>
<td>7.15</td>
<td>39</td>
<td>4.88</td>
<td>1.88</td>
<td>4.36</td>
<td>2.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p-0_s13</td>
<td>8.</td>
<td>6.24</td>
<td>6</td>
<td>12.61</td>
<td>3.13</td>
<td>10.93</td>
<td>5.03</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>p7-driv_s7</td>
<td>5.23</td>
<td>12.96</td>
<td>45</td>
<td>2.19</td>
<td>1.51</td>
<td>2.39</td>
<td>2.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sc-15.ind</td>
<td>15.12</td>
<td>42.9</td>
<td>18</td>
<td>0.3</td>
<td>17.62</td>
<td>0.31</td>
<td>18.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>uart-9.ind</td>
<td>7.72</td>
<td>9.21</td>
<td>26</td>
<td>0.22</td>
<td>3.3</td>
<td>0.21</td>
<td>2.91</td>
<td>1.13</td>
<td>1.13</td>
</tr>
<tr>
<td>3nodes.ind</td>
<td>8.81</td>
<td>9.02</td>
<td>24</td>
<td>0.21</td>
<td>0.6[1]</td>
<td>0.19</td>
<td>0.52[1]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(\text{orig} = \text{solver on seed}\) \hspace{1cm} \(\text{L} = \text{num dumped lemmas}\)

\(\bar{m} = \text{mean time, mutants}\) \hspace{1cm} \(\bar{l} = \text{mean time, mutants+lemmas}\)

\(\sigma_m = \text{std. dev, mutants}\) \hspace{1cm} \(\sigma_l = \text{std. dev, mutants+lemmas}\)
Conclusion

- Speed is not everything.
- Attributes like predictability also important.
- Experiments: SAT solvers differ in predictability.
- Passing theory lemmas can help SMT:
  - can improve performance a little (15-35%).
  - can improve predictability (3x, 3.5x).
  - but not predictably(!).
- Future work: try to improve predictability.
  - trade some performance for predictability.
  - canonical forms for SAT formulas?
  - run seed, mutant formula together?
  - use formula diffs? proofs?
Conclusion

- Speed is not everything.
- Attributes like predictability also important.
- Experiments: SAT solvers differ in predictability.
- Passing theory lemmas can help SMT:
  - can improve performance a little (15-35%).
  - can improve predictability (3x, 3.5x).
  - but not predictably(!).
- Future work: try to improve predictability.
  - trade some performance for predictability.
  - canonical forms for SAT formulas?
  - run seed, mutant formula together?
  - use formula diffs? proofs?

Maybe you want to try to improve predictability!