Consider the following program:

```c
1 unsigned int func(unsigned int a) {
2    unsigned int *z = 0;
3    if ((a & 1) && ((a & 1) ^ 1))
4        return *z;
5    return 0;
6 }
```

This program is safe, i.e., the unsafe pointer dereference in line 4 is unreachable because the guard in line 3 is never holds; a & 1 holds if the last bit in a is one, and (a & 1) ^ 1 inverts the last bit in a. The clang static analyzer, however, produces the following (spurious) bug report when analyzing the program:

```
main.c:4:12: warning: Dereference of null pointer (loaded from variable 'z')
    return *z;
    ^~
1 warning generated.
```

Experimental evaluation

<table>
<thead>
<tr>
<th>Projects</th>
<th>time (s) (no ref)</th>
<th>time (s) (with ref)*</th>
<th>reported bugs (no ref)</th>
<th>refuted bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmux</td>
<td>86.5</td>
<td>89.9</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>redis</td>
<td>347.8</td>
<td>338.3</td>
<td>93</td>
<td>1</td>
</tr>
<tr>
<td>OpenSSL</td>
<td>138</td>
<td>128</td>
<td>38</td>
<td>2</td>
</tr>
<tr>
<td>twin</td>
<td>225.6</td>
<td>216.7</td>
<td>63</td>
<td>1</td>
</tr>
<tr>
<td>git</td>
<td>488.7</td>
<td>405.9</td>
<td>70</td>
<td>11</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>1167.2</td>
<td>1112.4</td>
<td>196</td>
<td>6</td>
</tr>
<tr>
<td>SQLite3</td>
<td>1078.6</td>
<td>1058.4</td>
<td>83</td>
<td>15</td>
</tr>
<tr>
<td>curl</td>
<td>79.8</td>
<td>79.9</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Memcached</td>
<td>43.9</td>
<td>44.2</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>libWebM</td>
<td>96</td>
<td>96.2</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>xerces-c++</td>
<td>489.8</td>
<td>433.2</td>
<td>81</td>
<td>2</td>
</tr>
<tr>
<td>XNU</td>
<td>3447.1</td>
<td>3405.1</td>
<td>557</td>
<td>51</td>
</tr>
<tr>
<td>Total</td>
<td>7683.7</td>
<td>7408.5</td>
<td>1270</td>
<td>89</td>
</tr>
</tbody>
</table>

* The average time of Z3, Booleter, MathSAT, Yices and CVC4.

In total, 89 bugs were refuted and an in-depth analysis of them show that all of them were false positives. The average time to analyze the projects with refuted bugs was 35.0 seconds faster, a 6.25% speed up. Out of the four projects where no bug was refuted the analysis was 1.0 second slower on average: a 1.24% slowdown.

Acknowledgements

Thanks to George Karpenkov, Artem Degachev, Devin Coughlin, Anna Zaks, Réka Kovács, Dominic Chen and Gábor Horváth for the help during the development of this project.

This project was developed as part of the Google Summer of Code 2018.