

Despite the significant efforts developers invest in creating advanced solvers, ensuring they are completely bug-free is not guaranteed. For work like ours, where the primary goal is to solve a mathematical problem, relying on potentially buggy code poses a significant risk.

While working with Unidom, I accidentally discovered a bug in the model enumeration scenario. This issue occurred in its parallel/distributed mode under certain reasonable settings. All optimal solutions for $7 \leq n \leq 18$ were generated and are available in ``generate_all_v1.zip``, using Unidom under those buggy conditions. Additionally, all optimal solutions for $7 \leq n \leq 19$ were generated ($n=19$ was an open case) and are available in ``generate_all_v2.zip``, using Unidom under settings that did not trigger the bug. However, passing specific test cases and general testing cannot guarantee the absence of bugs—only that no obvious ones are present. Therefore, in other parts of this work, particularly when addressing the next open case, we also incorporate verification.

The test that exposes the bug, along with its visualization, can be found in the `revealed_bug/` folder. This folder also includes a `.sh` script for running Unidom. The `run.sh` script is designed to generate all solutions in parallel and consolidate the results into a single file.

To use the script, move `run.sh` to the main directory of the original Unidom project, adjust the parameters as needed, and then execute it with `./run.sh`.

Please note that when running multiple processes, Unidom may produce identical solutions. If necessary, consider adding a post-processing step to handle these duplicates.

Regarding ``generate_all_v1.zip``, ``generate_all_v2.zip``, and the files they contain, please note the following:

- For some values of n , results are stored in a single file.
- For others, the results are divided into multiple files with the suffix ``_part_X``.
- In each file, solutions are displayed as chessboards where ``Q`` indicates a queen's position, ``x`` marks a square excluded from the candidate set at that search state, and squares marked with ``-`` are empty.
- Each solution in the file is separated by a ``;` from the subsequent one.
- ``generate_all_v1``: Related to the buggy version.
- ``generate_all_v2``: Related to the (hopefully) fixed/corrected version.