

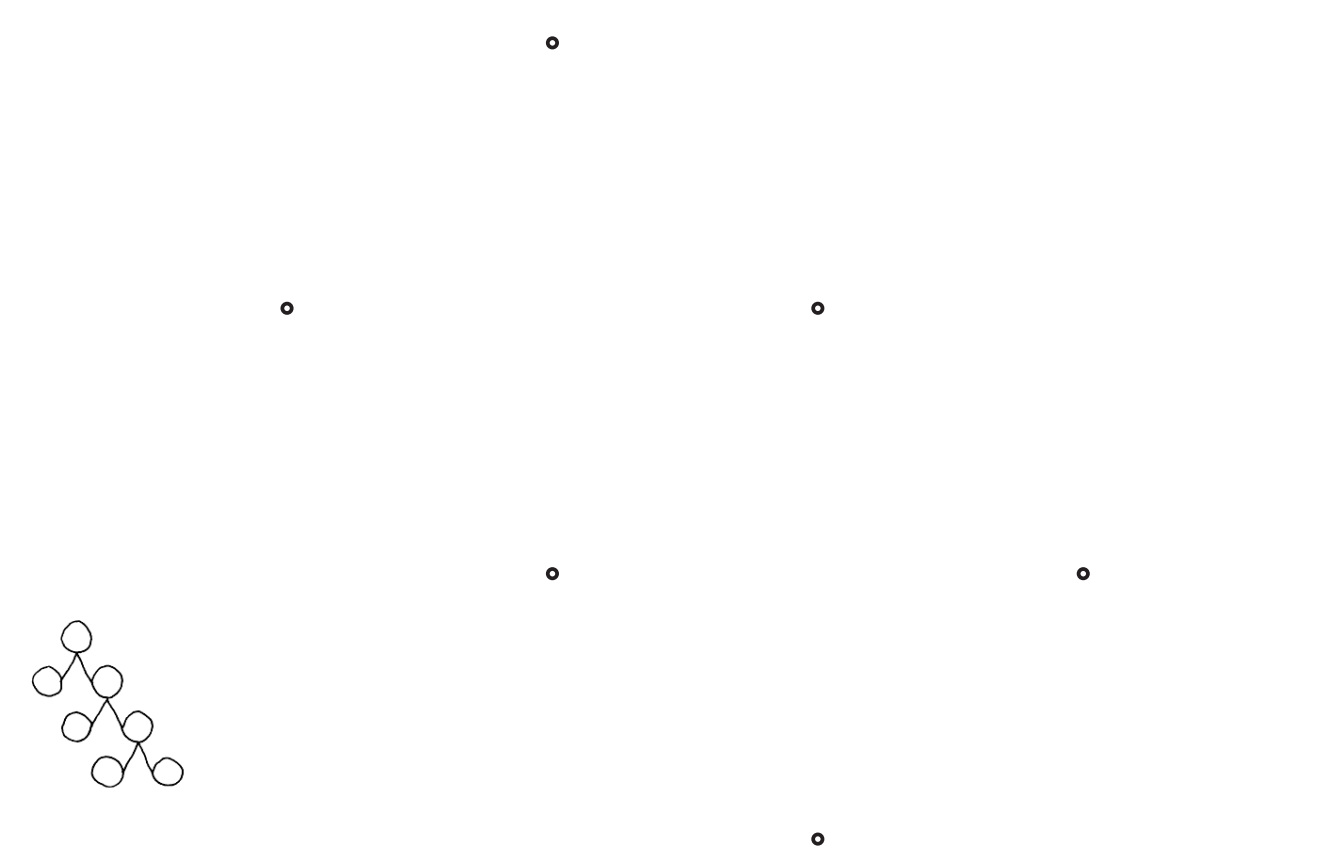


GOAL

- The \LaTeX package TikZ is a powerful, multi-purpose package that can be used to create a host of different graphical objects in a typeset environment
- However, it can be cumbersome and unintuitive for the end user at times
- Want to automatically generate correct and reasonable TikZ code for a graph based on a user drawing of the graph

EXAMPLE

- Here is a successful example of the script's run:



CHALLENGES

- The TikZ interface is complicated and sometimes unintuitive; the full manual is 405 pages long!
- Many end users do not need the full scope of what the TikZ interface offers, and instead just need to draw simple graphs on short notice
- Generation of code needs to avoid creating unreasonably large graphs

SOLUTIONS

- We use canny circle detection to find where the nodes are relative to each other on the original image.
- With this relative data, we plug the relations into a Z3 SAT solver and it outputs a set of coordinates for TikZ to read.
- This set is then translated into TikZ code and compiled into a PDF using \PDFLaTeX .
- This is repeated again separately with two different objective functions: one which minimizes pairwise distances between vertices, and one which minimizes the total width and height of the graph.

RESULTS

- None of the optimization results yielded a significantly different result. The only exception is that one benchmark (#11) failed at runtime under the rectangle objective function
- Out of 21 benchmarks, 13 benchmarks successfully generated the correct nodes
- 3 benchmarks were missing some nodes
- 4 benchmarks had extra nodes
- 1 benchmark did not generate any TikZ code

CONCLUSION

- Conclusions