# Synthesizing Programs By Asking People For Help Online

Cassie Jones
Witch of Light
list+sigbovik@witchoflight.com

April 1, 2020

## Abstract

Program Synthesis is a growing area of research, but most approaches are still limited by the difficulty of writing specifications and the lack of awareness of the software's environment such as frameworks, operating systems, and deployment. We propose a distributed social method which satisfies these particular criteria at the cost of some scalability.

**ACH Reference Format:**
Cassie Jones. 2020. Synthesizing Programs By Asking People For Help Online. In *Proceedings of SIGBOVIK 2020*, Pittsburgh, PA, USA, April 1, 2020. (SIGBOVIK '20, ACH).

## Introduction

Tranditional program synthesis is based on enumerating possible programs and checking if they match the specification, via directed search strategies and symbolic execution. For the most part, these take advantage of computers' suitability for doing lots of brute-force work and careful checking.

Working with these synthesis systems involves writing precise specifications in a form that the synthesis tool understands, and then checking that the program output by the synthesizer matches the specification you *thought* you were writing. If there are ambiguities in your specification, synthesis tools will often find a way to solve the problem in the simplest way possible, even if that means it solves a different problem than the one you wanted. Synthesis tools are often quite slow for large programs, and are usually contained to limited domains, for exmaple, writing pure functions over algebraic datatypes [3] or pointer manipulating programs. [4] Program synthesis tools are often oblivious to the function of systems considerations like the behavior of operating systems or external libraries, and adding knowledge of these is time consuming and contribues to synthesis performance problems.

More domain-specific synthesis tools are able to overcome many of these difficulties, since they only need to handle knowledge about their specific domain, but by their nature can't synthesize general programs.

We attempt to solve these limitations by "asking people for help online." We believe that this is a practical, currently deployable approach for producing software. It allows for natural language specifications and systems aware synthesis, and it can provide results all the way through to the deployment stage.

## 1 Our Approach

Our approach, "asking people for help online," overcomes many of the difficulties that other program synthesis approaches suffer from. Since you're asking people for help, they already possess natural language capabilities that we can take advantage of. Furthermore, "people" are able to "ask questions" to refine ambiguous specifications. Depending on your needs and who you're able to find to help you, they can provide assistance ranging from suggesting what code you should write, to contributing changes themselves if they happen to be sufficiently interested.

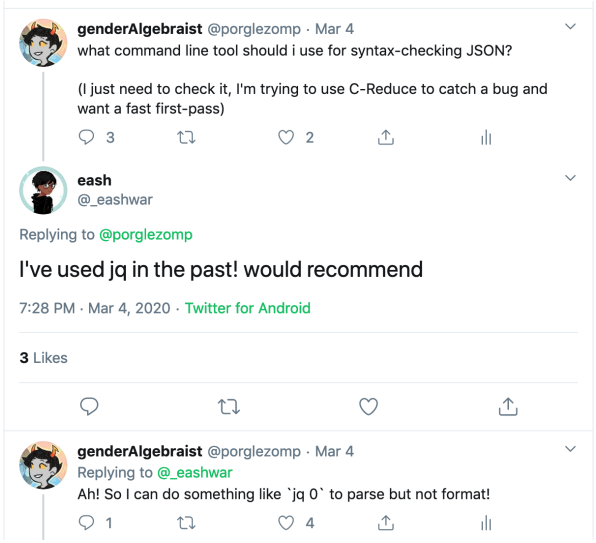


Figure 1: The "asking for help online" approach, applied to a "hole" in a partially implemented C-Reduce interestingness predicate. The query resolved to an executable not yet installed on my machine, but which ended up being a good solution.

### 1.1 Specification Refinement

The first step in synthesis via asking people for help online, is to ask people for help online. To get the best results, you need to find the right place to ask. Some problems can be solved in general programming forums—for instance Stack Overflow can sometimes be helpful if they're not mean to you. Sometimes, your question is too niche to get attention in those places. We have found it very effective to make friends with a variety of programmers in multiple timezones who like answering questions, and then asking on twitter when problems get too difficult or open-ended. If you're integrating with a particular piece of software, library, or language, going to their communities can sometimes be your best bet, but friends are a good first target.

At this point, your synthesis query is a natural language question about the problem you need to solve. When you ask your question, you may get answers, or you may get questions in response about what you're trying to do. If answers don't match your intention, you can also provide extra information to direct the follow-up responses closer to your goal. This is the process of specification refinement, and it is negotiated transparently via the communication process involved in this approach. If you're getting responses, at some point you'll likely have a sufficient specification that someone can help you.

Synthesis queries can also be provided in the forms of partial programs with holes. You aren't limited to sharing machine-readable text, you can even share an image of your partial program by taking a phone picture of your computer screen and sharing that while asking for advice. This is particularly beneficial when your development environment isn't connected to the internet.

Figure 2: This is program synthesis. [1]

### 1.2 Environment Awareness

This synthesis method is aware of the program's environment. It's particular effective at integrating with large frameworks and APIs. When asking people for help online about a React problem or a Yosys problem for instance, there's often help from people who work on those projects. This extends to many other domains and projects as well.

This synthesis approach applies to your whole system. You can use it to synthesize your install scripts, and match existing conventions for different ecosystems. People online are able to access sources of different projects and cite them as justifications, making this approach competitive with other approaches which datamine sources like GitHub. Asking in the right places (like the issue trackers of projects with bugs or missing features) can also synthesizes code inside your dependencies, avoiding the need to fork those dependencies to make your changes.

### 1.3 Cross-Platform

Many synthesis tools are difficult to actually use outside the lab. They may have difficult-to-install dependencies, or only work on certain systems and with certain build approaches, or simply have bad interfaces with lots of set up. Our synthesis approach is fully cross-platform and compatible with existing workflows. Not only is it flexible to *literally any* system that the developer is using, the networked component is also highly flexible. The examples included in this paper involved synthesis work via both IRC and Twitter, with crossover between the two.

```
21:11 <porglezomp> What's the recommended way to distribute extensions? As a
hack I'm installing them into a directory under the yosys data dir so that it
can find its files with +/<extension>/<stuff> paths. Is that fine?
21:20 <cr1901_modern> porglezomp: I found this: https://github.com/YosysHQ
/yosys-plugins/blob/master/vhdl/Makefile#L27-L29
21:20 <cr1901_modern> Also this: https://github.com/tgingold/ghdlsynth-beta/blob
/master/Makefile#L30-L32
21:21 <cr1901_modern> The honest answer in my experience is that "most ppl don't
actually make yosys plugins and just hack directly on yosys". But at least the
above two plugins seems to be consistent in install procedure
21:22 <porglezomp> Ok, nice, just need a slightly deeper subdir.
21:47 <porglezomp> Ah nice, now I can just do yosys -m nangate and things work!
21:52 <cr1901_modern> great :D!
```

Figure 3: The "asking for help online" approach can give results extracted from external data souces like GitHub, and provide synthesis that is tailored to the conventions of a particular software community. Also note that this synthesis query is solving a coupled deployment problem and implementation problem.

## 2 Future Directions

We were only able to evaluate this method by asking for help online. In theory, we believe our approach should naturally extend to asking for help locally, offline. This would have several tradeoffs. Relevant to industry deployment, it can keep the partial programs and specifications more confidental. The offline communication also has latency benefits, which can improve iterations time especially in the specification refinement stage. But, it has a smaller pool of help, and unfortunately, we didn't have access to the local computational resources to evaluate this method, so its analysis must be left to future work.

This approach also shows promise in other domains of software engineering and programming language research, like fault localization and program repair. There are some domains like static type checking for dynamic languages where it could theoretically be deployed, but many of these domains overlap with others, and fall in the areas where this approach breaks down. Some of these domains like fault localization are potentially further automatable via technology like rubber ducks.

## References [2]


[1] Alice Avizandum. 2017. "Liberal: *through sobs* you can't just say everything is nationalized.... Please.... Corbyn: *points at seagull flying past* nationalized". Twitter. https://twitter.com/aliceavizandum/status/875665911952416768

[2] Atticus Maisse. 2019. Hanging indents with a Pandoc bibliography. In *The TEX and LATEX Stack Exchange.* https://tex.stackexchange.com/questions/477219/hanging-indents-with-a-pandoc-bibliography

[3] Nadia Polikarpova, Ivan Kuraj, and Armando Solar-Lezama. 2016. Program Synthesis from Polymorphic Refinement Types. In *Programming Language Design and Implementation 2016*, Santa Barbara, CA, USA, June 13–17, 2016. (PLDI '16).

[4] Nadia Polikarpova and Ilya Sergey. 2019. Structuring the Synthesis of Heap-Manipulating Programs. In *Proceedings of ACM Programming Langanguages 3*, POPL, Article 72. (January 2019). https://doi.org/10.1145/3290385