

# A Haskell Code Generator from I/O Examples

## User Manual

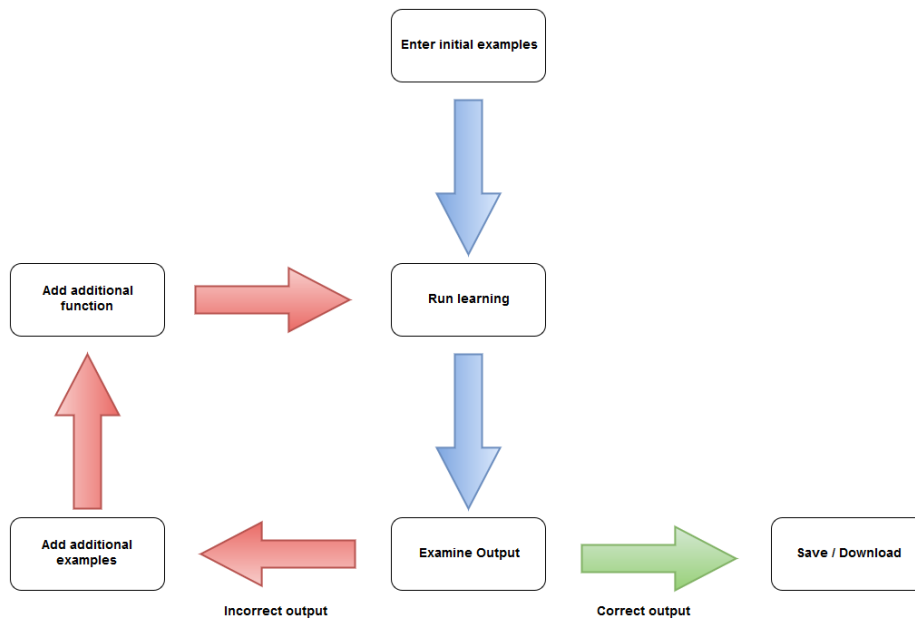


Figure 1: The iterative learning workflow

### Entering examples

In order to use this tool, you first have to enter examples to learn, which specify how the target program behaves on specific inputs.

You can change the number of input arguments by editing the "No. Args" box. While there are no restrictions on the possible arguments, be aware that any increase in number of arguments can greatly affect learning performance.

Currently supported argument types are :

- Integer
- String

## The learning step

After entering your examples, you can perform learning by clicking the "Run ASP" button.

After a short computation step, the learned haskell is displayed, and there are a few options on how to proceed :

- If the learned program is incorrect, then you can freely add more examples which specify the missing behaviour.
- If you are unsure about the correctness of the program, you can simply test some more complicated answers using example autocompletion.
- If you are happy with the learned program, you can download the generated code, or save it to be re-used by other tasks.

## Example autocompletion

If you want to test more complicated values, you can provide examples with valid input, but no output. After learning, these outputs will be completed by the tool, and by analysing the result compared to the expected value, you can discern the correctness of the learned program.

If you are unsure about program correctness, you can add more examples to be autocompleted which will then be ran on the learned program without a new learning task being started.

## Combining functions

*Note: Feature not in build yet :(*

Once you have successfully learned a function, you may reuse it in another. First, add a new learning task by opening a new tab. Then, select it in the list of allowable functions. Any learning performed on this new task will have knowledge of the initial function and make use of it in a preferential manner.

## Handling errors

Sometimes, it may not be possible to learn a program from a set of examples. They may be contradictory, or the tool may not be robust enough. In this case, you have a few options open to you :

- Try to remove certain examples to find out which one might be causing a contradiction.
- Separate your function into multiple smaller ones. For example, a Merge Sort program could be separated into smaller splitting and comparison programs