C2Flowchart Tool

August 2021

1 Introduction

The C2Flowchart Tool is used to translate stateless decision logic from C code to Stateflow flow charts.

2 How to Use the Tool

This section describes what must be done to setup the tool, as well as how to use the tool.

2.1 Prerequisites and Installation

1. Use MATLAB/Simulink/Stateflow R2019a or newer.
2. To install the tool,
   (a) from a `.zip` file — unzip the contents to your desired location.
   (b) Add the contents of the `src` folder to the MATLAB search path.
3. From within the `src` folder, run the command `c2flowchart("../example/example.c")`. If a new model appears containing a Stateflow function, then the tool has been successfully installed.

2.2 Function Arguments

```
function c2flowchart(c_file, export_c_types)
```

Parameters:

1. `c_file`: The path to the C file to be used as an input.
2. `export_c_types`: When this arguments value is 1, the function will export external C types to base workspace using the function `Simulink.exportExternalCTypes`

The function can be called with either one or two arguments. When only one argument is used, the parameter `export_c_types` is set to 0.

2.3 Example

From within MATLAB, navigate to the `src` folder within the directory that the tool was installed in. Take note of the path to the C file that will be used as the input. In this example, we will use the example file located within the `example` folder, using the relative path `"../example/example.c"`.

Now, we will simply call the function `c2flowchart`.

```
>> c2flowchart("../example/example.c")
```

A MATLAB window should open containing the Stateflow function shown in Figure 1. After some minor manual tweaks to clean up the layout of the generated flow chart, it should resemble the function shown in Figure 2.

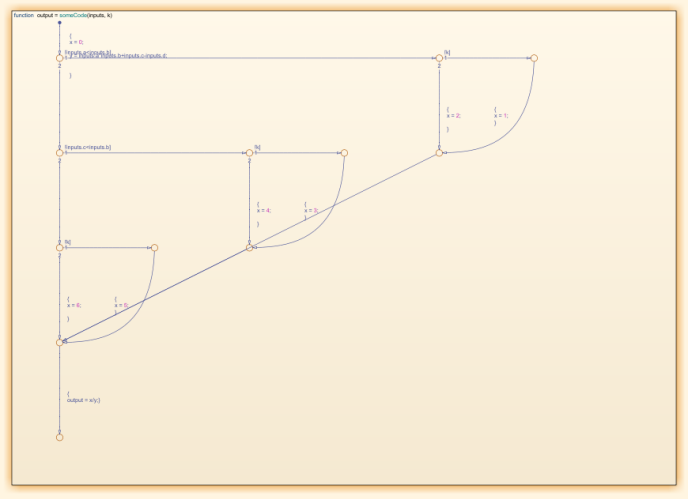

Figure 1: Output of the function `c2flowchart` before formatting

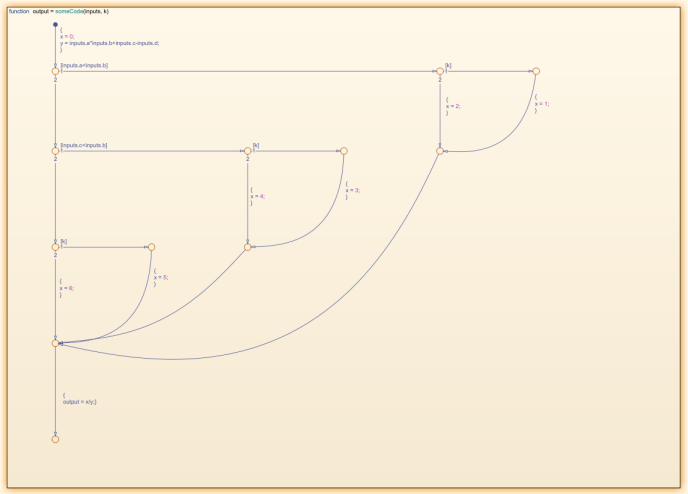

Figure 2: Output of the function `c2flowchart` after formatting

The file `example.c` contains a custom type definition used for its input. To add this custom type definition to the base workspace, use the command:

```
>> c2flowchart("../example/example.c", 1)
```

Your MATLAB workspace should now show the custom type definitions, as in Figure 3.

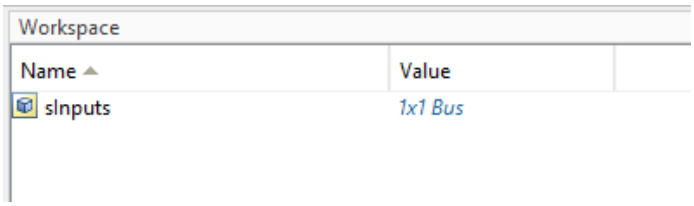

Figure 3: Base workspace after running the command `c2flowchart` using the `export_c_types` parameter

2.4 Folders and Files

The `src` folder contains three files that are necessary to run the tool.

1. `c_json.py` contains code for running the Python library `pycparser`. This is the library that generates the AST for the c file passed to the tool.
2. `c2flowchart.m` is the main file invoked from the Matlab command line.
3. `createFunctionFlowChart.m` is the file responsible for parsing the AST and generating the corresponding Stateflow flow chart.

The AST file generated by `c_json.py` generates into the folder `tmp`. This folder is created by the tool and can be deleted if the tool is not running.

2.5 Errors and Warnings

Any errors or warnings during tool use will be visible in the Matlab Command window. The most likely error is shown is Figure 4. This error occurs when a chart is open that has the same name as the C file that was passed as the first argument. Simply close the existing chart and run the tool again.

```
Error using sfnew (line 65)
A system named 'example' already exists

Error in c2flowchart>buildFlowChart (line 67)
    bdhandle = sfnew(chart_name);

Error in c2flowchart (line 44)
    buildFlowChart(name, objAST); - Show complete stack trace
```

Figure 4: Example error if a chart of the same name is already open