Documentation for AutomaDeD GUI

AutomaDeD GUI is used to visualize the report generated by AutomaDeD debugging tool.
It parses this file and visualizes the Progress Dependency Graph (PDG) along with stack traces and annotations on source codes corresponding problematic regions.

- **Providing source code locations:**

Report generated by AutomaDeD contains absolute paths to source file locations. If that path is not accessible, alternate search locations can be provided through an environment variable: **AUTOMADED_SOURCE_PATH**

You can provide ":" separated multiple locations where the tool will search for the expected files. As soon as it finds a match for the base-name, it will use that file in the visualizer.

- **Launching the GUI:**

./automaded-gui <input file>

"automaded-gui" which uses required python installations to launch the GUI

<input file> is the input to the GUI. This is also the output generated by AutomaDeD debugging tool.

By default the python set inside the script points to:
python='/usr/gapps/asde/python/chaos_5_x86_64_ib/bin/python'

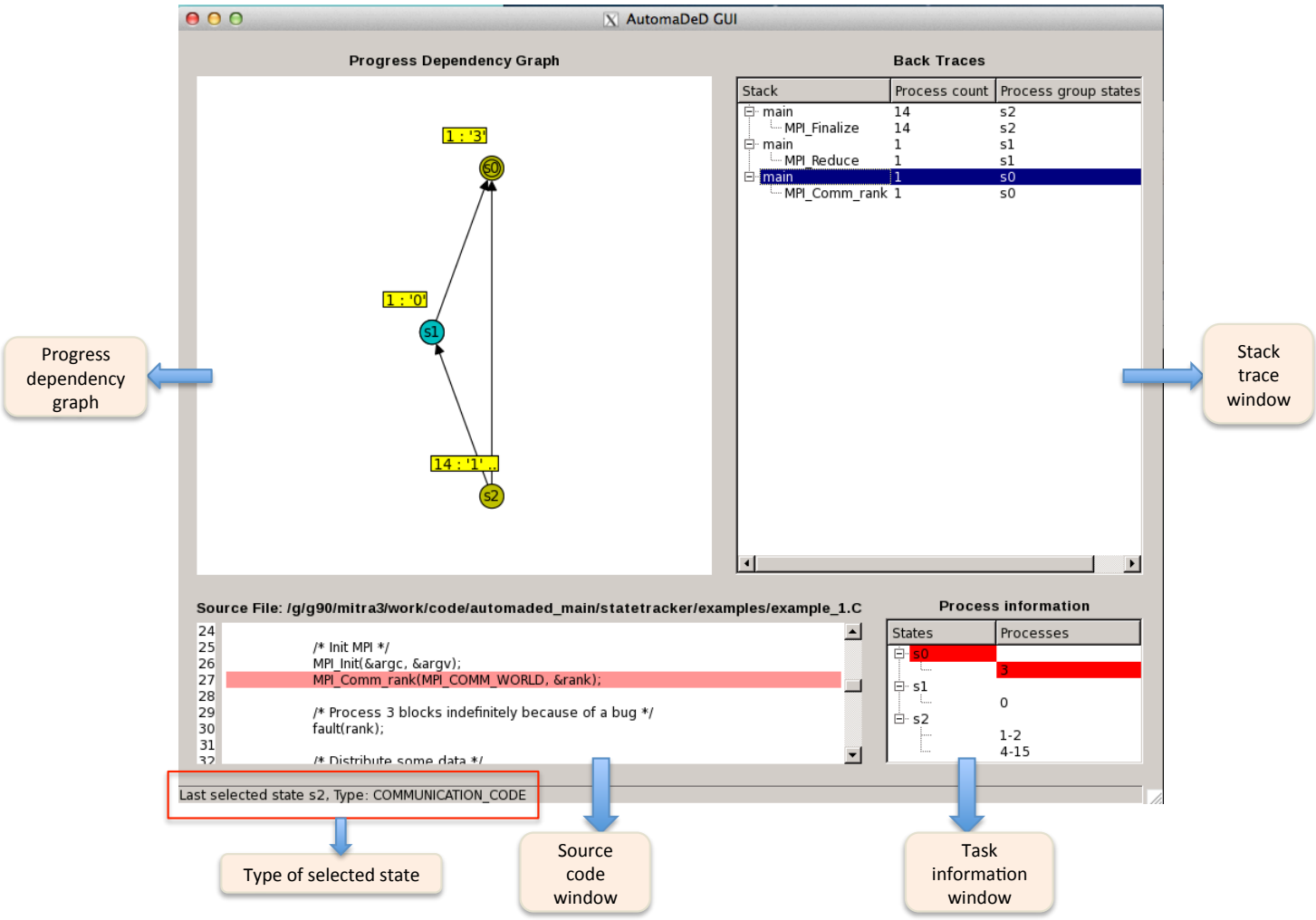
GUI requires following python packages on top of python 2.7:

- ✓ PyQt4, matplotlib, networkx, pylab, pydot, Queue

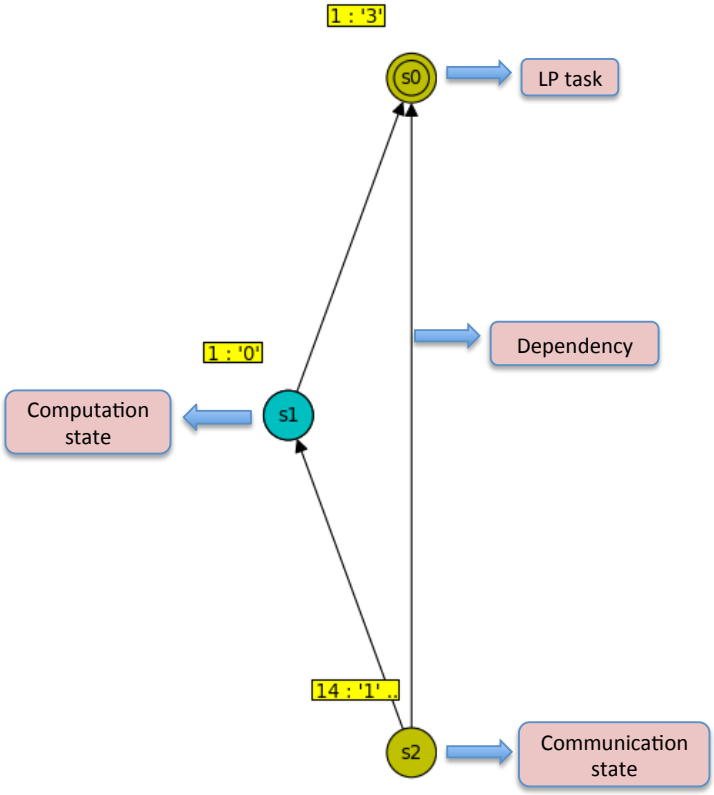
It also requires following packages which are available in LC machines:

- ✓ sys, os, random, getopt, commands, numpy, math

➢ Different components of the GUI:

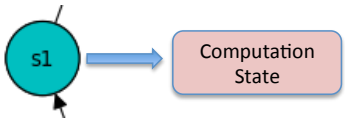


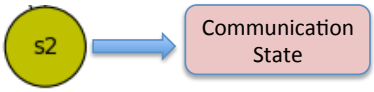
➢ Progress Dependency Graph:

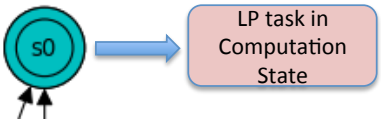


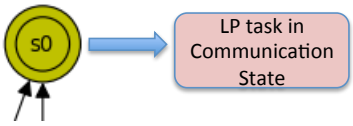
Here processes at state s1 and processes at state s2 are waiting for processes at state s0.

More about state colors and symbols:





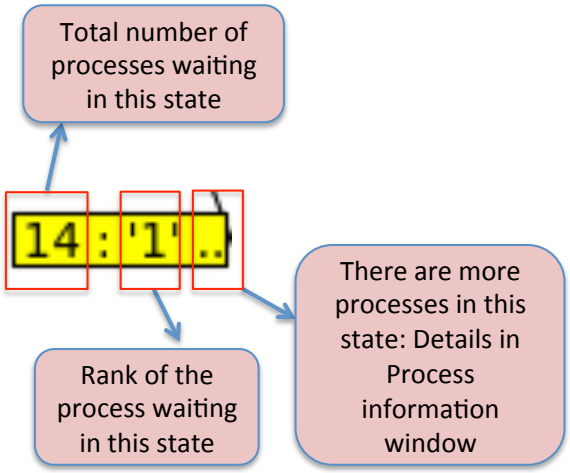




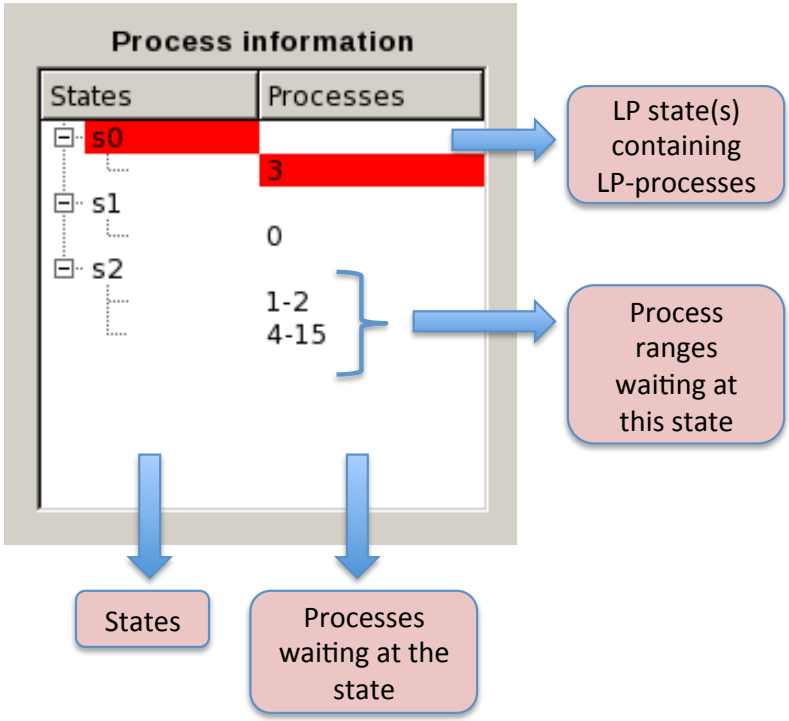
LP task(s)/process(s) has no outgoing edges – which means no other processes are waiting for a Least Progressed (LP) process.

State labels:

Each state has a label, which contains following information:

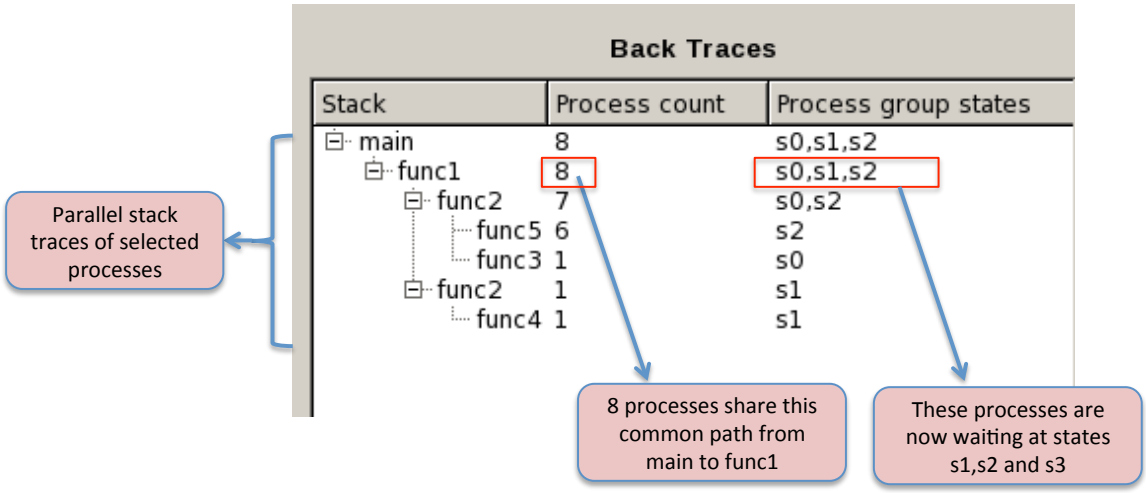


➢ Process information window:



➢ Stack trace window:

Single Click on PDG state: select multiple states in PDG one by one
Double Click on PDG state: select only a particular state in PDG, unselect others



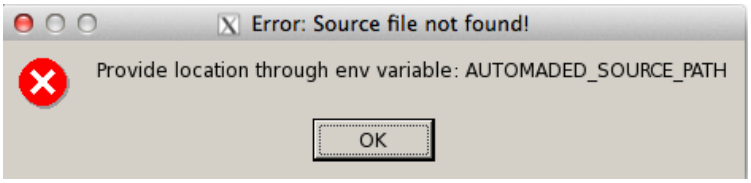
Tooltip: On each row there is a tooltip, which shows the full filename and line number corresponding to that function

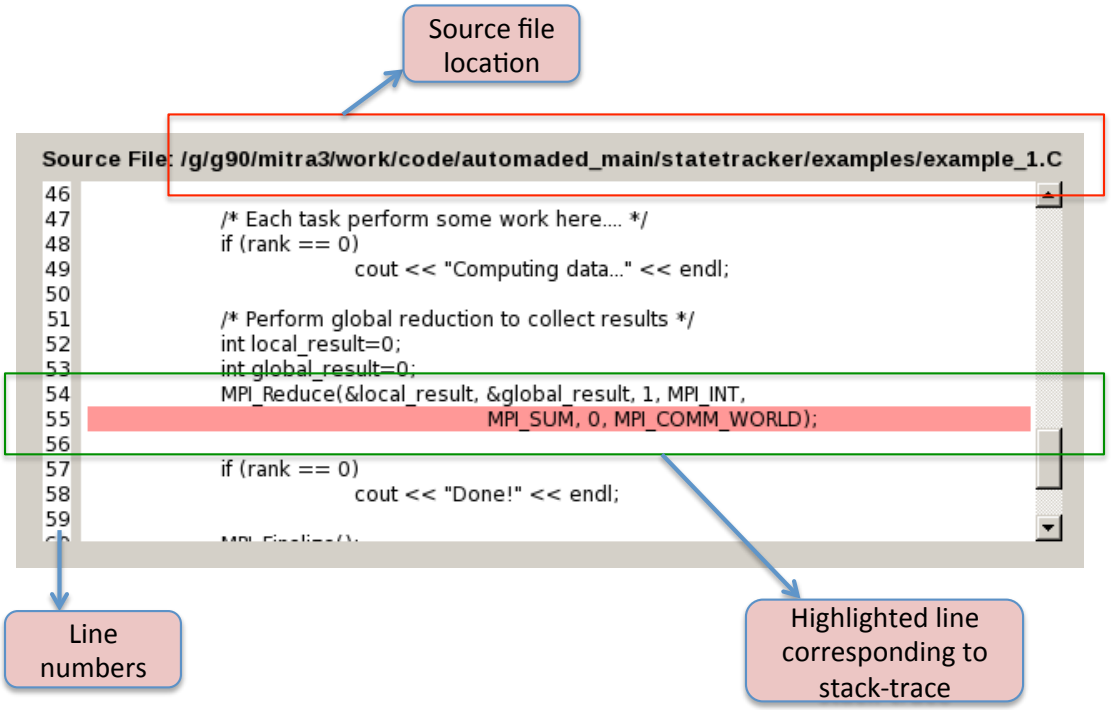
Source code: Single click on any function in this stack trace window will open the source file and will highlight corresponding line.

➢ Source code window:

Source file locations must be readable. To provide custom search path for source files, please provide ":" separated values to the environment variable "AUTOMADED_SOURCE_PATH" before launching the GUI.

If source file is not found following error will be displayed:




Source file location
Source File: /g/g90/mitra3/work/code/automaded_main/statetracker/examples/example_1.C
46
47 /* Each task perform some work here.... */
48 if (rank == 0)
49 cout << "Computing data..." << endl;
50
51 /* Perform global reduction to collect results */
52 int local_result=0;
53 int global_result=0;
54 MPI_Reduce(&local_result, &global_result, 1, MPI_INT,
55 MPI_SUM, 0, MPI_COMM_WORLD);
56
57 if (rank == 0)
58 cout << "Done!" << endl;
59
Line numbers
Highlighted line corresponding to stack-trace