Linux Verification Project Part II

Objective: Use the *matching pair graph* (MPG), *event flow graph* (EFG), and *control flow graph* (CFG) as evidence to verify correct pairing of the given spin lock instances. The instances are specified later.

Guidelines for Reporting Verification Results:

- Articulate how the pairing is verified. Describe the strategy you have used for pairing.
- In case you determine that the pairing is violated on some path, provide the specifics of the path and your reason for claiming that the pairing is violated.
- If you encounter an inconclusive situation where you cannot complete the verification, explain why you consider it to be an inconclusive situation.
- Keep track of time and effort you spend on each instance and report it. In particular, explain how you arrived at the verification strategy. Was it immediately clear? If not, what was the effort to arrive at the strategy? Did the MPG help you to arrive at the strategy? How? What else has helped you to perform the verification?
- The report must be informative clear and concise.

Articulating the Strategy:

As stated in the guidelines above, you need to describe the verification strategy. Here are some suggestions. While articulating the strategy, try to think of important points to cover in order to communicate the strategy. To get you started, here are some points to think of:

- What are the nodes in the MPG where you started the verification? Why?
- Could you do modular verification? If yes, describe the steps you followed. As discussed during lectures, each step in modular verification involves verifying a subset of MPG nodes. The nodes verified in one step are used to verify the next subset of nodes in successive steps.
- There could be a common observation that applies to verify a subset of nodes. Try to come up with such common observations and state them. For example, state the common observation, if you have used one to verify all nodes in particular step of modular verification.
- Were you able to verify all root nodes? What evidence did you generate to verify each root node?
- Are the MPG, EFG, and CFG sufficient for verification?
- Are there cases where the EFG is not sufficient? Why? What is the additional useful evidence CFG provides in those cases?
- Are there cases where you need to check the source code? Why?
- Are there cases where the MPG has extraneous nodes not relevant to verification? Justify with specifics.
- Are there cases where the EFG has extraneous nodes not relevant to verification? Justify with specifics.
- Does the verification require a path feasibility check? Are you able to do it? If not, what makes that path feasibility check difficult?

Write a Concluding Section:

Write a concluding section that summarizes your efforts, what you have learned from the project, and your suggestions for us to improve the project. Here are some specifics you can comment on.

- The experience and knowledge you gained about verification. Is this your first experience of being exposed to large software?
- How do the instances assigned for verification vary in terms of the difficulty and the strategies you used for verification?
- How important is it to have MPG, EFG, and CFG as graphical evidence? Qualitatively, how difficult do you think it would be to verify the given instances using the source code and not the graphical evidence?
- Think of some way to quantify the time the graphical evidence has saved and describe the quantified savings.
- Is the graphical evidence important for the accuracy of verification? How? Give some examples of likely errors that would occur without the support of the graphical evidence.

- Compare this part of the project to whatever other experiences you may have that are similar in terms of understanding and reasoning about software written by others.

The Website and the Spreadsheet:

We have added more information and improved the organization of the website and the spreadsheet to make them more useful. Use the new website at: http://kcsl.ece.iastate.edu/linux/spin/3.19-rc1/. The spreadsheet is given along with this project statement. The key changes to the website and the spreadsheet are:

- Each lock instance is assigned a unique ID that you can use either on the website or the spreadsheet.
- Each instance has a key for browsing the source on the Linux organization website: http://lxr.free-electrons.com/source/. The instructions for browsing are the same as in Part I and they are included at the end of this section.
- The website and the spreadsheet include a unique ID for each big MPG.
- The spreadsheet has two worksheets: (a) the worksheet for lock instances, (b) the worksheet that provides for each lock instance the details of its small MPG, and the EFGs and CFGs.
- The lock instance worksheet has one row per lock instance. Each row includes the lock ID and the web link to the particular lock instance. You can use either the lock ID or the web link to search the lock instance on the website.
- The details worksheet has multiple rows for each lock ID. Each of those rows corresponds to a pair of CFG and EFG for that lock instance.

You can browse the Linux source code using the *source link*: http://lxr.free-electrons.com/source/

You can access the source code as follows. For example, you can view the source code for the lock instance 1459754158328 by following the steps below:

1. The spreadsheet gives the source key – [1850+25+/linux-3.19-rc1/drivers/staging/lustre/lustre/ldlm/l_lock.c]
2. Copy the path after linux-3.19-rc1. In this case, you will copy drivers/staging/lustre/lustre/ldlm/l_lock.c
3. Concatenate with the source link. In this case, you get http://lxr.free-electrons.com/source/drivers/staging/lustre/lustre/ldlm/l_lock.c
4. If you don't specify version you will be directed to the latest kernel version. So add ?v=3.19 at the end. For kernel version 3.19, the URL would be: http://lxr.free-electrons.com/source/drivers/staging/lustre/lustre/ldlm/l_lock.c?v=3.19

   Verify each of the following instances:

Lock ID	Points	Macro Model #Nodes	Macro Model #Edges
1459680926937	10	29	71
1459754158328	5	12	19
1459765699942	5	8	10
1459755403205	5	17	31
1459734022779	5	4	3
1459673583201	5	10	15