504: Machine Learning meets Program Analysis

Assignment 1

Chenglong Wang

Data consistency problems in web applications

Problem When developing web-based user-interactive applications, keeping data-consistencies can be challenging. Data-consistency here means that difference occurrences of a same data should be consistent with each other: for example, given a web application with several occurrences of the current date (e.g. a date in the web page title, a date will appear in texts of a warning window and a date used in the internal computation of the program), all these different occurrences should be consistent – whenever a function modifies one of the data source, all of the occurrences should be modified correspondingly.

The challenge for a user (e.g. me) to keep data consistency in web application development is caused by the following facts.

- Web applications are interactive and the program status mutate when some event is triggered. As different event handlers are implemented in different functions and each handler need to modify different data occurrences in the application. As the code snippets relevant to the same data source are scattered in different locations of the program, it is challenging for programmers to keep these data consistent.
- When the data consistency is not guaranteed, identifying such inconsistency for user is challenging and they are likely to lead to latent bugs. This is caused by the fact that debugging interactive web applications requires generation of events and observing user interface modifications, as the observing process in this case requires the programmer to observe all web pages and it is hard to cover everywhere.

Problem analysis The key point of the problem may not be easily solved due to the following reasons.

- It is not easy to automatically identify which variables are supposed to be consistent with each other.
- Even if we can find out variables that are consistent with each other, analyzing whether they are consistent with each other is challenging. One particular reason is that most web applications are programmed using JavaScript and analyzing JavaScript programs is challenging.

Possible solutions Firstly, to identify variable occurrences that should be consistent each other, we can use natural language informations: 1) variable names can leave a hint, 2) texts in the web page around the data to be displayed can be used to infer the meaning of the data and 3) comments in the program can help identifying the meaning of an object. Once we have a list of potential variables, we can present them to the programmer and ask she to help confirm the list.

Secondly, when we obtain the variables, we can perform program analysis to check whether these variables are consistent with each other or generate test cases to check whether these variables are consistent with each other.

Automatic API migration

Problem When a new incompatible version of an API is released, users face to problem to modify their original code to use the new version of API. Such manual migration requires user energies to study the correspondence between the two version of API and it can be time consuming. In my experience of performing such migration process, I found an example with migration history and imitate it to migrate my program. Thus it is ideal to have a tool that can automatically identify API correspondence between the old and the new versions API based on such examples.

Summary of Comments on wang.pdf

Page: 1

Number: 1 Author: mernst Subject: Highlight Date: 1/11/2016 9:29:25 PM

You didn't mention it as a challenge, but I think it is also interesting to identify different occurrences of a single datum; that is, identifying the data that need to be kept consistent. As written you assume that all the occurrences are already known.

I'm not saying that identifying the consistent data is more interesting what you said, just that it's another thing to consider. (I see that you mention this later, but it came as a surprise in the problem analysis section, which had expected would just expand on problems you had already mentioned.)

Number: 2 Author: mernst Subject: Highlight Date: 1/11/2016 9:28:24 PM

Should they all be using similar abstractions, such as calling certain helper routines?

Number: 3 Author: mernst Subject: Highlight Date: 1/11/2016 9:27:25 PM

Do you mean it's hard for the user to identify, or it is hard for the programmer to identify things that we visible to the user?

Number: 4 Author: mernst Subject: Highlight Date: 1/11/2016 9:27:50 PM

It's not clear to me how identifying inconsistencies is related to debugging. I think I've lost the thread of this paragraph.

Number: 5 Author: mernst Subject: Highlight Date: 1/11/2016 9:30:23 PM

This is confusing: given variables that are consistent we need to analyze whether they are consistent?

Number: 6 Author: mernst Subject: Highlight Date: 1/11/2016 9:32:32 PM

What you believe these are the right solutions? The solutions feel very vacant high level to me. Are they grounded in your own experience, and are they instantiations of what you would do as a programmer or what you would wish to have? Or, are they just an idea that you think might work? It feels a bit more like the latter, perhaps because it feels relatively high-level and ill-defined. And could be helpful to be much more specific about the exact format of the output and exactly how would help the user. Then you could think about how to produce that output, even would be hard to do. I think you'll be more successful at that kind of approach than with just a general idea of the sort of analysis that might possibly be useful.

Number: 7 Author: mernst Subject: Highlight Date: 1/11/2016 9:33:23 PM

Are you assuming that the maintainers did not provide specific guidance in terms of performing the migration?

Number: 8 Author: mernst Subject: Highlight Date: 1/11/2016 9:36:12 PM

Do you mean you found another project that is a client of library L, the same library your program is a client of, and you looked at how that other project was modified? This is just that is difficult to be the first adopter.

Is another problem that, when upgrading uses of the library, the other project might have also introduced irrelevant changes?

Number: 9 Author: mernst Subject: Highlight Date: 1/11/2016 9:35:03 PM

Is there way to do this without these examples, so that it can be useful even to the first people who want to migrate?

Why not solved? Existing method identifies such migration relations based only on this code migration history by evaluation AST subtree similarities between the old sample code and the new sample code[1]. This approach 1) limits the migration relations that can be identified from the sample program and 2) evaluation of the similarity is based on empirical formula thus less convincing[2].

Possible solutions One way we can solve them is to learn source code correspondence with statistical methods and also adopting the API document. We can 1) firstly enumerate combinations of possible mappings based on the document, then 2) rank the combinations based on the development history (sample code) and 3) generate transformation rules to perform API migration.

Page: 2

Number: 1 Author: mernst Subject: Highlight Date: 1/11/2016 9:35:27 PM
Do you have a citation for this?

Number: 2 Author: mernst Subject: Highlight Date: 1/11/2016 9:36:49 PM
Can you explain these limitations? The section is very brief, so it doesn't really convince the reader, even if these facts are true.