Software Testing, Quality Assurance & Maintenance—Lecture 31

Patrick Lam

March 22, 2017

Part I

Using Linters

source: `jamie-wong.com/2015/02/02/linters-as-invariants/`

First there was C

Statically-typed languages like C:

```
#include <stdio.h>

int main() {
  printf("%d\n", num);
  return 0;
}
```

Compiler saves you from yourself.

Guaranteed invariant:

> “if code compiles, all symbols resolve”.

Less-nice languages

OK, so you try to run that in JavaScript and it crashes right away.

Invariant:

> “if code runs, all symbols resolve”?

Counterexample

But what about this:

```
function main(x) {
  "use strict";
  if (x) {
    console.log("Yay");
  } else {
    console.log(num);
  }
}

main(true);
```

Nope!

Still no invariants

Invariant:

“if code runs without crashing, all symbols referenced in the code path executed resolve”?

Nope

```
function main() {
  "use strict";
  try {
    console.log(num);
  } catch (err) {
    console.log("nothing to see here");
  }
}

main();
```

JavaScriptWorld Problems

When maintaining old code:

- is this variable defined?
- is this variable always defined?
- do I need to load a script to define that variable?

Why is this the developer's problem?

Solution: Linters

```
plam@banach ~> cat foo.js
function main(x) {
  if (x) {
    console.log("Yay");
  } else {
    console.log(num);
  }
}

main(true);

plam@banach ~> nodejs /usr/local/lib/node_modules/
   jshint/bin/jshint --config jshintrc foo.js
foo.js: line 5, col 17, 'num' is not defined.

1 error
```

Invariant

“If code passes JSHint, all top-level symbols resolve.”

Strengthening the Invariant

Add a pre-commit hook.

"If code is checked-in and commit hook ran,
all top-level symbols resolve."

Better yet. . .

Block deploys on test failures.

"If code is deployed,
all top-level symbols resolve."

Even better yet. . .

Hard to tell if code is deployed or not.
Use git feature branches, merge when deployed.

“If code is in master,
all top-level symbols resolve.”