Grammar-Based String Refinement Types

Fengmin (Paul) Zhu

Program Strings

```
def get_hostname(url: str) -> str:
  '''Extract the hostname part.'''
  start_host = url.find('://') + 3
  end_host = url.find('/', start_host)
  hostname = url[start_host:end_host]
  return hostname
```

Let's test it:

```
get_hostname("https://www.cispa.de/home")
# Output: 'www.cispa.de'
```

Strings Can Go Wrong

Urls with empty paths trigger bugs:

```
get_hostname("https://www.cispa.de")
# Output: 'www.cispa.d'
```

Urls involving SQL injection may lead to security issues:

```
get_hostname("https://localhost'); DROP TABLE USERS --/")
# Output: "localhost'); DROP TABLE USERS --"
```

This output string instantiates to an unsafe query:

```
INSERT INTO HOSTNAMES VALUES ('localhost'); DROP TABLE
USERS --')
```

Annotate Your Program with String Refinement Types

Refinement type constructors:

```
def lang(G: Grammar) -> type
```
$\{s : \text{str} \mid s \in \mathscr{L}(G)\}$

```
def lang(G: Grammar,
         φ: Condition) -> type
```
$\{s : \text{str} \mid s \in \mathscr{L}(G) \wedge \varphi(s)\}$

Annotate our function with refinement types for the input URL and the output hostname, where both are based on a grammar for URLs:

```
Url: type = lang(URLGrammar['<url>'])
Host: type = lang(URLGrammar['<host>'])

def get_hostname(url: Url) -> Host:
  ...
# tests
```

```
URLGrammar: Grammar = {
  '<url>' :
    '<protocol>://<host><path>',
  '<protocol>' :
    ['http', 'https'],
  '<host>' :
    '<char>*',
  '<path>' :
    ['', '/<char>*'],
  '<char>' : ...
}
```

Dynamic Checking

for detecting bugs & vulnerabilities

Code instrumentation:

```
def get_hostname(url):
  assert Url.accepts(url) # check input
  start_host = url.find('://') + 3
  end_host = url.find('/', start_host)
  hostname = url[start_host:end_host]
  assert Host.accepts(hostname) # check output
  return hostname
```

Manual / random inputs → ISLA Input Specification Language → python™

✓ Test passed

✗ Test failed: assertion violated

Static Checking

for ensuring correctness & safety

extends mypy for Python

Type Checker

$s \in^{?} \mathscr{L}(G)$

Membership test (parsing)

```
get_hostname("https://www.cispa.de/home")
get_hostname(
  "https://localhost'); DROP TABLE USERS --/")
```

$\mathscr{L}(G_1) \subseteq^{?} \mathscr{L}(G_2)$

Language inclusion (subtyping)

```
link: Url = ...
get_hostname(link)
# Urls where protocol='https'
HttpsUrl: type = lang(...)
secure_link: HttpsUrl = ...
get_hostname(secure_link)
```

$s_i \in \mathscr{L}(G_i) \Rightarrow \mathscr{L}(op(s_1, s_2, \ldots)) = ?$

String operation result type inference `reveal_type(url[start_host:end_host])`

✓ Type safe

✗ Type error(s)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Author information: email address fengmin.zhu@cispa.de; home page https://paulz.me.
This project is advised by *Andreas Zeller* (https://andreas-zeller.info/).