report.md 2/7/2020

## Intra-procedural Static Taint Analysis of JavaScript Programs

## How it works

- 1. I've created class named TaintPass.java extends from CompilerPass as Custom Compiler Pass.
- 2. In the **TaintPass**, I have used **ScopeCreator** class to analyze for every function. For example, if a source code contains multiple functions, TaintPass will create multiple **analizer**.
- 3. TaintAnalysis is the main analyzer extends from DataFlowAnalysis with forward direction.
- 4. I've written **computeTainted** method to calculate **sources** may or must reach to sinks.

The following picture is aim to explain as a whole:

```
Compiler pars > Taint Pass
          function saveUserInfo() { | Enter Scope -> Taint Analysis
                                                               Data Flow Analysis

J
             name = retSource(),
             age = retSource(),
             designation = retSource(),
             stream = getStream();
       6
            let userInfo = {
             'id': id,
      10
             'name': name,
             'age': age
      11
                                Enter Scope _____ Taint Analysis
Ly Flow through
Ly Compute
Taintel
      12
      13
           if (name != '' && age > 10 && designation != '') {
      14
      15
      17 }
          function passUserId() {
  let id = retSource();
      20
           sink(id);
      21
                  > Creates new instance
```

The example follows and how it works in **Control Flow Graph**:

```
1 function saveUserInfo() {
2  let id = retSource(),
3    name = retSource(),
4   age = retSource(),
5   designation = retSource(),
6   stream = getStream();
7
8  let userInfo = {
9    'id': id,
10    'name': name,
11   'age': age
```

report.md 2/7/2020

Line number	Node type	Value	Parent	Sources may reach	Sources must reach
1	FUNCTION	saveUserInfo		{}	{}
2	VAR	id		{}	{}
3	VAR	name		{}	{}
4	VAR	age		{}	{}
5	VAR	designation		{}	{}
6	VAR	stream		{}	{}
7	EMPTY			{}	{}
8	VAR	userInfo	{id, name, age}	0	8
	OBJECTLIT	id, name, age		{}	{}
14	IF	{conditional:true}		{}	{}
15	EXPR_RESULT	sink userInfo	{id, name, age}	{id, name, age}	