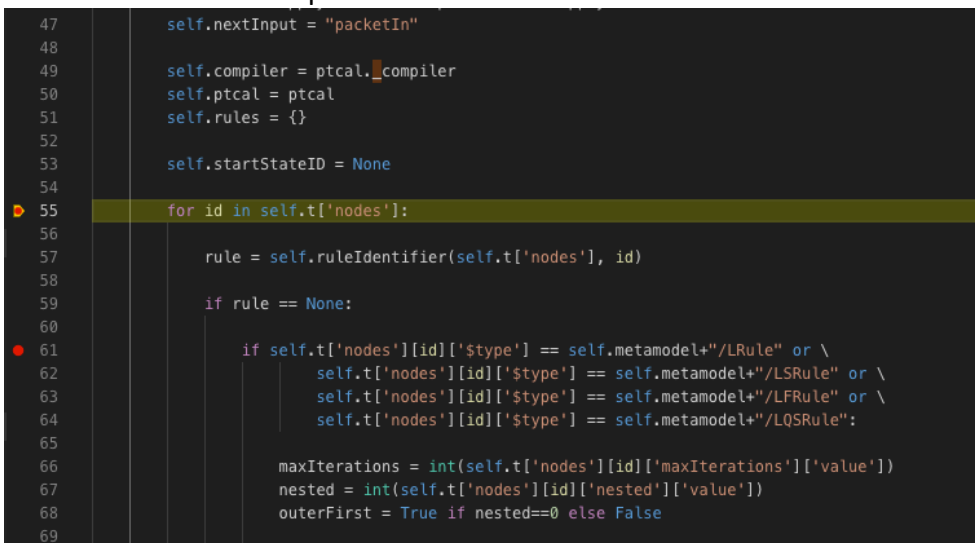
How to debug Atompm ?
Here we are using Visual Studio code to debug Atompm code.
Step1: Start the “httpwsd.js” file from terminal.

```
Last login: Mon May  4 10:11:22 on ttys004
(base) wenjuns-MacBook-Pro:~ wenjunshi$
(base) wenjuns-MacBook-Pro:~ wenjunshi$ cd atompm/
(base) wenjuns-MacBook-Pro:atompm wenjunshi$ node httpwsd.js
   info  - socket.io started
   info  - handshake authorized qcok3PbzCM5dUwlaCxqb
>> {"workerType":"/csworker","workerId":0}
   info  - handshake authorized ggA9a35pf5Np9qnbCxqc
>> {"workerType":"/asworker","workerId":1}
   info  - handshake authorized Jdbs9mV9zJYYLRWQCxqd
   info  - transport end (heartbeat timeout)
```

Step2: Open Atompm from browser.
Step3: Set breakpoint in the source code.
Here is an example:



```
47        self.nextInput = "packetIn"
48
49        self.compiler = ptcal._compiler
50        self.ptcal = ptcal
51        self.rules = {}
52
53        self.startStateID = None
54
55        for id in self.t['nodes']:
56
57            rule = self.ruleIdentifier(self.t['nodes'], id)
58
59            if rule == None:
60
61                if self.t['nodes'][id]['$type'] == self.metamodel+"/LRule" or \
62                        self.t['nodes'][id]['$type'] == self.metamodel+"/LSRule" or \
63                        self.t['nodes'][id]['$type'] == self.metamodel+"/LFRule" or \
64                        self.t['nodes'][id]['$type'] == self.metamodel+"/LQSRule":
65
66                    maxIterations = int(self.t['nodes'][id]['maxIterations']['value'])
67                    nested = int(self.t['nodes'][id]['nested']['value'])
68                    outerFirst = True if nested==0 else False
69
```

In example, I set two breakpoints in the motifcontext.py file.

Step4: Start debugging from the main.py file.
-go to main.py file
-start debugging under that file.
Step5: Go to Atompm browser, hit the breakpoint from there.
In the example, it will hit the breakpoint when loading a transformation file.
So, In the browser,
-open a sample bird model
- load transformation T_run.model.
Then it will go to visual studio code and stops at the first breakpoint we set.
Step6: Check attributes of the transformation.

When your hang over t[“nodes”], you will see more info about the transformation.