ENGF0002 (Design and Professional Skills)

Scenarios

This document focuses on the differences in function calls between the three prototypes of the programming language. It follows an observation-explanation-conclusion structure in that an observation is laid down, explained, and a theory is formulated it out of it.

Classifier-1 (1.rkt):

```
#lang FunctionCalls

fun times(a, b):
  a * b
end

print(times(2 , times(2 , 4)))
print("Fin.")
#translates to (2 * (2 * 4))
```

```
Welcome to DrRacket, version 7.0 [3m].
Language: FunctionCalls, with debugging; memory limit: 128 MB.
version: 2018-09-04T22:54:09-04:00

------Core 1------

16
"Fin."

------Core 2------

"Fin."

------Core 3------

16
"Fin."
```

Partition after test: {Core 1, Core 3}, {Core 2}

Observation-1: When a function is called from within another, Core-1 and Core-3 return a valid output, but Core-2 returns nothing.

Theory-1: When a function *f*- that takes *n* arguments and returns a single value (of the same type as the arguments)- is used in another *f*, then Core-1 and Core-3 return valid results, but Core-2 fails to return anything.

Example:
fun $f(a_1, a_2, .. , a_n)$*:*
  //function body
end
$f(i_1, i_2, .. , i_{n-1}, f(a_1, a_2, .. , a_n))$*:*

Core-2, however, does not fail to compile as it still prints something afterwards, as seen in the classifier.

Classifier-2 (2.rkt):

```
#lang FunctionCalls

fun greaterThanSeven(n):
  if n < 7:
    greaterThanSeven(n + 1)
  else:
    n
  end
end

print(greaterThanSeven(3))
```

```
Welcome to DrRacket, version 7.0 [3m].
Language: FunctionCalls, with debugging; memory limit: 128 MB.
version: 2018-09-04T22:54:09-04:00

------Core 1------

7

------Core 2------


------Core 3------

ERROR: Program timed out
```

Observation-2: When a recursive function is implemented, then Core-1 returns a valid output. However, Core-2 returns nothing and Core-3 times out.

Theory-2: When a recursive function- a function that calls itself- is implemented in the language, then Core-1 returns a correct output, i.e implements the recursive function correctly. Core-3, on the other hand, times out and fails to compile indicating that its recursion limit may have been reached. However, Core-2, again, returns nothing as a function is being called from within another.

Partition after test: {Core 1}, {Core 2}, {Core 3}

Note, however, that when a factorial function was implemented by me in the languages, then all three of the cores failed. For Core-1, the behaviour was similar to Core-2's in that it returned nothing. Core-3, on the other hand, timed out and I am yet to find a recursive function that does not time out in Core-3 even if one single recursive call is made.