RaML 15150 Tests Summary

Yinglan Chen

May 13, 2020

1 HW1

Implement factorial function as below.

```
let rec fact x =
  if x <= 1 then 1
  else let _ = Raml.tick 1.0 in x * fact (x - 1)
let _ = fact 5
```

The rest cannot be implemented since RaML does not support real numbers.

2 HW2

Implement all functions, but none of the function can be analyzed. For example, here is the implementation of `add`

```
let rec add ((n : int), (m: int)) = if n = 0 then m
  else let _ = Raml.tick 1.0 in 1 + add (n-1, m)
```

The error message shows"A bound for {function name} could not be derived. The linear program is infeasible"
The reason is RaML does not support integer recursion (because of negative int issues). A better error message can be: `integer type in recursion detected. Please use rnat instead`
Luckily, this assignment does not use any negative integer. So I change int to rnat type and reimplement HW2
Here is the `add` function after re-implementation

```
let rec add (n, m) =
    Rnat.ifz n
        ((fun () -> m))
        ((fun n' -> let () = Raml.tick 1.0 in succ(add (n', m)) ))

let _ = add (succ(succ(Rnat.zero)),Rnat.zero);;
```

Function `pascal` is still infeasible.

```
let rec pascal ((i: Rnat.t), (j: Rnat.t)) =
    Rnat.ifz j
        ((fun () -> succ(Rnat.zero)))
        ((fun i' ->
            ( let (i_minus_j,_) = (Rnat.minus i j) in
              (Rnat.ifz i_minus_j
              ((fun () -> succ(Rnat.zero)))
              ((fun j' -> let _ = Raml.tick 2.0 in
                          Rnat.add (pascal (i',j')) (pascal(i', j))))))
            )
        ))

let _ = pascal (succ(succ(Rnat.zero)),succ(Rnat.zero));;
```

The reason is pascal has exponential cost. A suggested error message is "The linear program is infeasible. function name has exponential cost". A suggested implementation (during weekly meeting) is when the program sees two recursive calls in one branch, replace one occurrence of the recursive call with a constant and analyze, repeat the same thing on the other occurrence. If both analysis are linear, that implies the function has exponential cost, but I don't know how to achieve that exactly.

There is no natural number comparison in Rnat module. So I use `Rnat.to_int` to implement `n < d` in function `div_mod`

```
exception DivideByZero
let rec divmod (n,d) =
  Rnat.ifz d
  ((fun () -> raise DivideByZero))
  ((fun d' ->
    ( Rnat.ifz n
      ((fun () -> (Rnat.zero, Rnat.zero)))
      ((fun n' ->
        if (Rnat.to_int n) < (Rnat.to_int d) then (Rnat.zero, n)
        else  let (diff, _) = Rnat.minus n' d' in
              let (x,y) = divmod(diff, d) in (Rnat.succ x, y)
      ))
    )
  ))
```

When analyzing `is_prime`, RaML works fine when given degree 1, the tight upper bound, but has Uncaught exception when given any degree larger than 1. The error message is shown below

```
Resource Aware ML, Version 1.4.1, July 2018

Typechecking expression ...
  Typecheck successful.
  Stack-based typecheck successful.

Analyzing expression ...

  Trying degree: 3Uncaught exception:

  Not_found

Raised at file "src/map.ml" (inlined), line 428, characters 6-26
Called from file "src/map.ml", line 1273, characters 23-77
Called from file "raml/annotations.ml", line 703, characters 11-33
Called from file "list.ml", line 111, characters 24-34
Called from file "list.ml", line 111, characters 24-34
Called from file "list.ml", line 111, characters 24-34
Called from file "raml/indices.ml", line 392, characters 17-44
Called from file "list.ml", line 137, characters 24-31
Called from file "src/list0.ml" (inlined), line 27, characters 40-75
Called from file "src/list.ml", line 161, characters 2-19
Called from file "raml/analysis.ml", line 1924, characters 24-70
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1982, characters 25-192
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1672, characters 18-153
```

```
Called from file "raml/analysis.ml", line 1790, characters 25-148
Called from file "raml/analysis.ml", line 1790, characters 25-148
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1982, characters 25-192
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 917, characters 30-42
Called from file "raml/analysis.ml", line 1259, characters 31-64
Called from file "raml/analysis.ml", line 1377, characters 34-55
Called from file "raml/analysis.ml", line 1291, characters 38-62
Called from file "raml/annotations.ml", line 577, characters 19-42
Called from file "raml/analysis.ml", line 1529, characters 21-54
Called from file "raml/analysis.ml", line 1637, characters 33-48
Called from file "raml/analysis.ml", line 1829, characters 16-231
Called from file "raml/analysis.ml", line 1845, characters 32-53
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1982, characters 25-192
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1790, characters 25-148
Called from file "raml/analysis.ml", line 1790, characters 25-148
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1982, characters 25-192
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 917, characters 30-42
Called from file "raml/analysis.ml", line 1259, characters 31-64
Called from file "raml/analysis.ml", line 1377, characters 34-55
Called from file "raml/annotations.ml", line 577, characters 19-42
Called from file "raml/analysis.ml", line 1529, characters 21-54
Called from file "raml/analysis.ml", line 1637, characters 33-48
Called from file "raml/analysis.ml", line 1829, characters 16-231
Called from file "raml/analysis.ml", line 1845, characters 32-53
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1908, characters 25-193
Called from file "raml/analysis.ml", line 1672, characters 18-153
```

```
Called from file "raml/analysis.ml", line 917, characters 30-42
Called from file "raml/annotations.ml", line 577, characters 19-42
Called from file "raml/analysis.ml", line 1529, characters 21-54
Called from file "raml/analysis.ml", line 1637, characters 33-48
Called from file "raml/analysis.ml", line 1829, characters 16-231
Called from file "raml/analysis.ml", line 1845, characters 32-53
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1672, characters 18-153
Called from file "raml/analysis.ml", line 1760, characters 14-133
Called from file "raml/analysis.ml", line 2081, characters 16-28
Called from file "main.ml", line 578, characters 8-21
```

3 HW3

Cannot find a good way to test in RaML. It does not support `assert` in ocaml.
Functions infeasible:

`filterInt, look_and_say, look_say_table, subset_sum, subset_sum_cert`

`subset_sum` and `subset_sum_cert` are exponential function, thus the output is expected. As stated in HW2, a suggested error message is to point out the exponential cost.
The reason why `filterInt` and `look_and_say` are infeasible is that the recursive call does not decrease argument size. So keeping the correctness, I change `tails(x,l)` to `(tails(x,r)` since in this branch, we know a = x needs to be filtered out. So `filterInt` now looks like:

```
let rec filterInt((x:int),(l:int list)) =
  match l with
  | [] -> let _ = Raml.tick 1.0 in []
  | a::r -> if heads(x,l) = 0 then let _ = Raml.tick 1.0 in a :: filterInt(x,r)
        else
            let _ = Raml.tick 1.0 in let tail = tails(x,r) in
            filterInt(x, tail)
```

Similar change is applied to `look_and_say`.
However, `look_say_table` is still infeasible, even after I change nat to Rnat.

```
let rec look_say_table ((l: int list), (n: Rnat.t)) : int list list =
   Rnat.ifz n (fun () -> [l]) (fun n' -> l :: look_say_table(look_and_say(l), n' ))
```

4 HW4

RaML type supports `int * tree * tree` but not `tree * int * tree`, so following code has to change according to that.
Anaylze mode works normally.

5 HW5

cannot support poly type since RaML does not support real.
`'a forest` type, defined as `Node of 'a option * 'a forest list` ,is also not supported because the size of `'a forest list` cannot be determined.

6 HW6

This assignment practice continuation, but polymorphic recursion is not supported.
For example, here is the implementation of `findOne`:

```
let rec findOne (p : 'a -> bool) (t : 'a shrub)
                (s : 'a -> 'b) (k : unit -> 'b) : 'b =
  match t with
  | Leaf n -> if p n then s n else k ()
  | Branch (l, r) ->  findOne p l s (fun () -> findOne p r s k)
```

Here is the output error message:

```
Analyzing function findOne ...
  Trying degree: 2Uncaught exception:

  Analysis.Make(Solver)(Amode).Analysis_error("Polymporphic recursion is
  currently not supported.Check the functional arguments.")
```

After discussion, we think it is too complicated for RaML to analyze, but why the error message occurs (how polymorphic recursion is involved) need further investigation.

```
Exception ("Patterns.Eunsupported_type(_, _, \"expected tuple type\")")
```

7 HW7

Task 1 is not supported since it requires real. Task 2 is implemented in module. Each function is tested separately. The error message is shown as below:

```
Error: analysis failure.

Resource Aware ML, Version 1.4.1, July 2018

Simplify: unsupported type at File "_none_", line 1 (no data constructor (a record?)):

vector
```

After discussion, we think it is because the bounds rely on the function f, thus bringing confusion to RaML. I try to change int to Rnat but that does not change the error message.

8 HW8

HW8 need to use the Sequence Library in SML, but RaML does not currently support that. In order to support, we need to implement the sequence library in RaML. Specific challenge lies in the log n bounds required in both work and span. A possible implementation is to use one tick to represent $\lceil \log n \rceil$ cost.

9 HW9

HW9 is about implementing a tick-tac-toe game. The core algorithm is a min-max estimation which takes exponential cost, so I skip the implementation here.

10 HW10

Task 1: lazylist can be implemented in Ocaml. But due to the absense of String library, and the continuation passing style, RaML cannot analyze the program. The error message is shown as:

```
Error: analysis failure.

Resource Aware ML, Version 1.4.1, July 2018

pattern (/tmp/tmpLEaSjo.raml[9,163+5]../tmp/tmpLEaSjo.raml[9,163+6])
  Ppat_constant Const_int 0
pattern (/tmp/tmpLEaSjo.raml[10,180+5]../tmp/tmpLEaSjo.raml[10,180+6])
  Ppat_var "wild#7/1245"
pattern (/tmp/tmpLEaSjo.raml[11,199+5]../tmp/tmpLEaSjo.raml[11,199+6])
  Ppat_var "wild#8/1246"
;

Exception ("Patterns.Fail(\"no compilation rule found for the given nested pattern\")")
```

Task 2: reference and lazy list. Due to the problem of lazylist, the error message is shown as follows

```
Error: analysis failure.

Resource Aware ML, Version 1.4.1, July 2018

Simplify: unsupported type at File "_none_", line 1 (unsupported recursive type):

unit -> 'a lazylist
```

Task 3: require to implement array using type `int -> 'a ref`. RaML does not support ref type.

```
Error: analysis failure.


Resource Aware ML, Version 1.4.1, July 2018

Simplify: unsupported type at File "_none_", line 1 (no data constructor (a record?)):

'a array
```

After discussion, we change array type to `Rnat.t -> 'a ref`, but that does not change the error message.