

Anubis - Feature #56

More dot operators

12/07/2008 02:41 PM - SpiceGuid -

Status:New Priority:Low Assignee: Category:Libraries Target version: Platform: Resolution:	Start date: Due date: % Done:0% Estimated time:0.00 hour Triage Stage:
Description I am enamoured with the Anubis overloaded dot operator. I believe the Anubis 'Kleisli applicator' or 'Maybe monad' syntax is the less intrusive one i have ever seen. It encourages a really elegant exceptionless style, so i just beg more in the same vein. public define \$A -> \$C \$B -> \$C g.\$A -> \$B f = (\$A x) -> g(f(x)). public define Maybe(\$U) (Maybe(\$T),Maybe(\$T)) ab.(\$T,\$T) -> Maybe(\$U) f = if ab is (a,b) then if a is { failure then failure, success(a1) then if b is { failure then failure, success(b1) then f(a1,b1) } }. public define Result(\$E,\$U) (Result(\$E,\$T),Result(\$E,\$T)) ab.(\$T,\$T) -> Result(\$E,\$U) f = if ab is (a,b) then if a is { error(e) then error(e), ok(a1) then if b is { error(e) then error(e), ok(b1) then f(a1,b1) } }. 	

History

#1 - 12/10/2008 03:49 PM - SpiceGuid -

I am enamoured with the Anubis overloaded dot operator.

I believe the Anubis 'Kleisli applicator' or 'Maybe monad' syntax is the less intrusive one i have ever seen. It encourages a really elegant exceptionless style, so i just beg more in the same vein.

```
public define $A -> $C
  $B -> $C g.$A -> $B f
  =
  ($A x) |-> g(f(x)).

public define Maybe($U)
  (Maybe($T),Maybe($T)) ab.($T,$T) -> Maybe($U) f
  =
  if ab is (a,b) then
  if a is
  {
    failure then failure,
    success(a1) then
    if b is
    {
      failure then failure,
      success(b1) then f(a1,b1)
    }
  }
  }.

public define Result($E,$U)
  (Result($E,$T),Result($E,$T)) ab.($T,$T) -> Result($E,$U) f
  =
  if ab is (a,b) then
  if a is
  {
    error(e) then error(e),
    ok(a1) then
    if b is
    {
      error(e) then error(e),
      ok(b1) then f(a1,b1)
    }
  }
  }.
```

Moreover i propose an 'exception catch' fonctionality :

```
public define ($E -> Result($E,$T)) -> Result($E,$T)
  try_with (Result($E,$T) x)
  =
```

```
($E -> Result($E,$T) error_handler) |->  
if x is  
{  
  error(e) then error_handler(e),  
  ok(v) then x  
}.
```

#2 - 01/22/2009 10:12 PM - Alain Prouté

- Assignee deleted (*Alain Prouté*)
- Platform deleted (*All*)