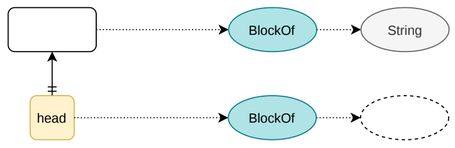
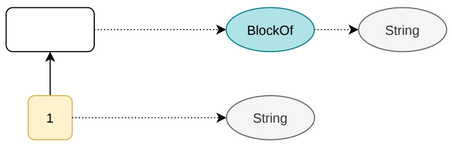
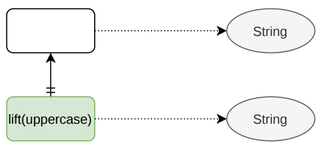
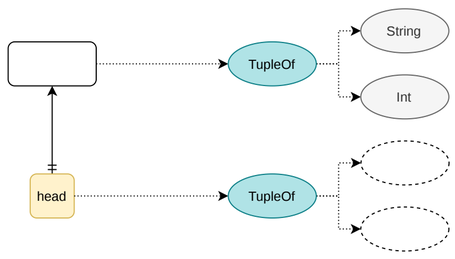
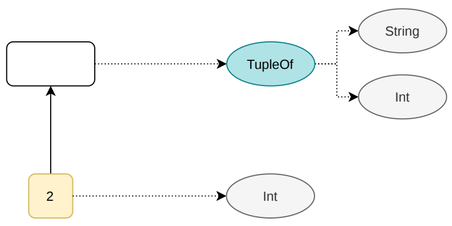

String

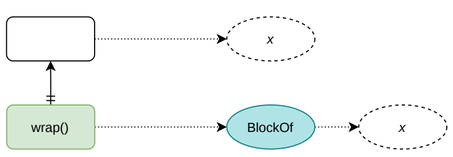

BlockOf
String
head
BlockOf

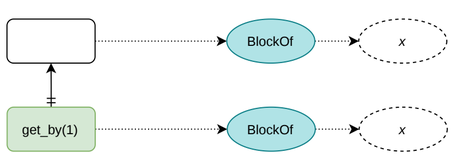

BlockOf
String
1
String

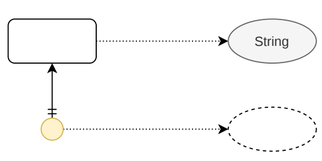

String
lift(uppercase)
String

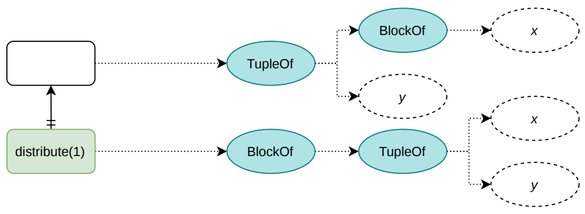

TupleOf
String
Int
head
TupleOf

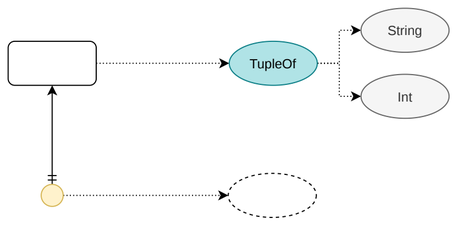

TupleOf
String
Int
2
Int

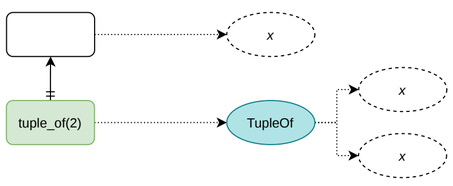

x
wrap()
BlockOf
x

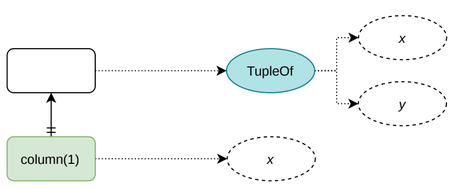

BlockOf
x
get_by(1)
BlockOf
x

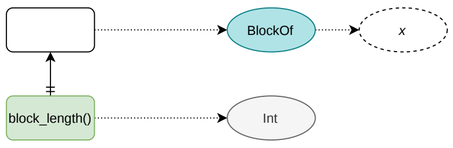

String

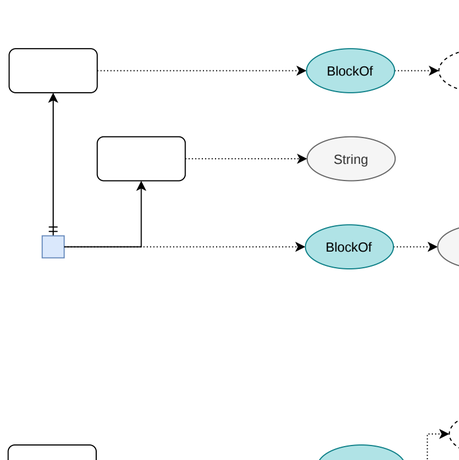

BlockOf
x
TupleOf
y
distribute(1)
BlockOf
TupleOf
x
y

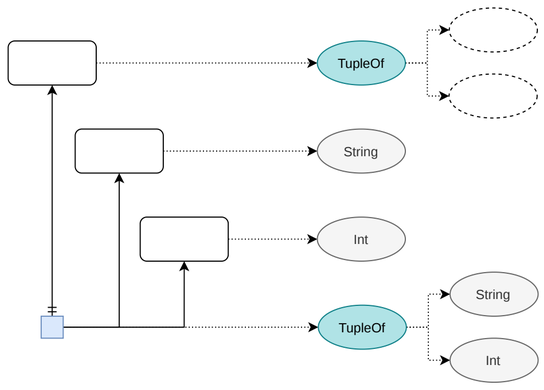

String
TupleOf
Int

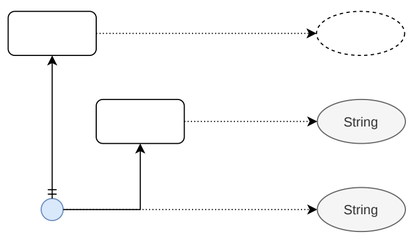

x
tuple_of(2)
TupleOf
x
x


TupleOf
x
y
column(1)
x

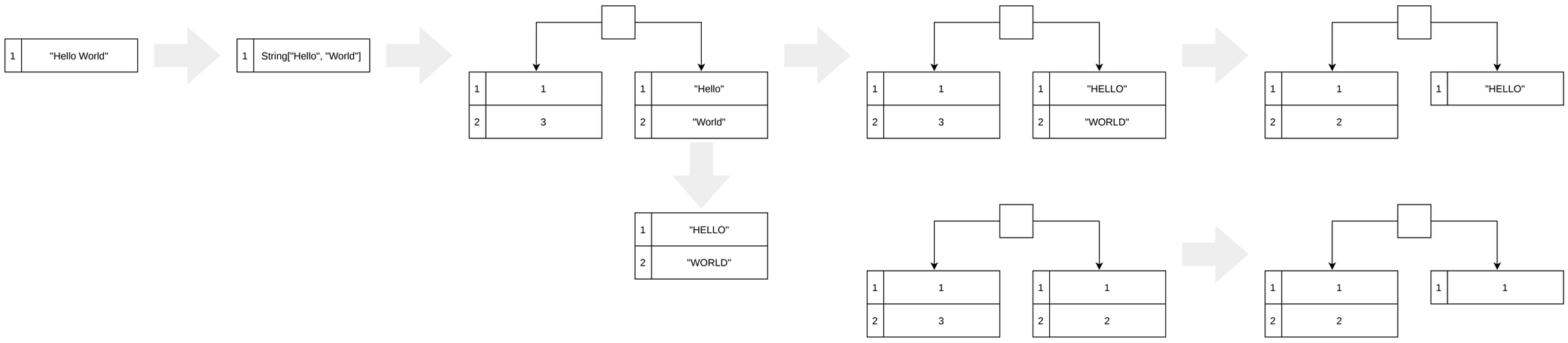

BlockOf
x
block_length()
Int

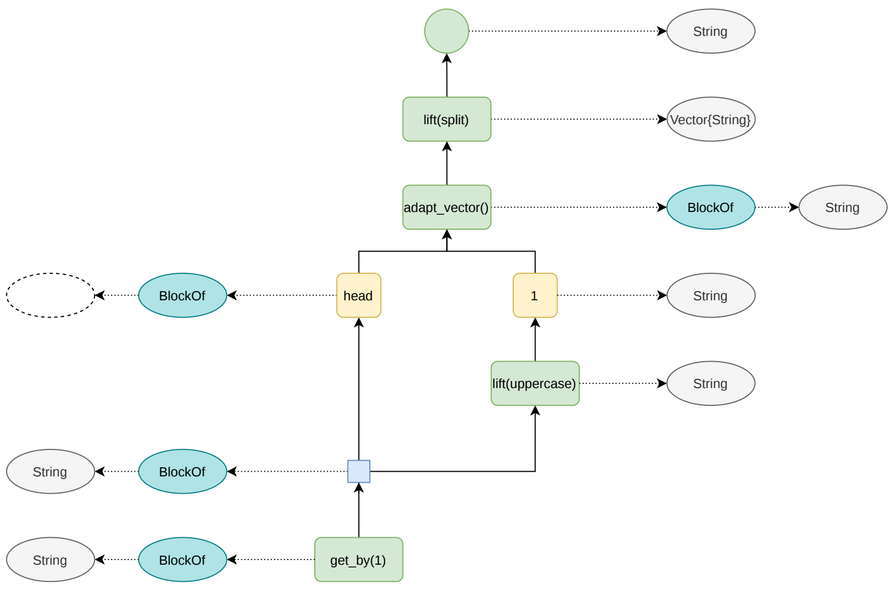

BlockOf
String
BlockOf
String

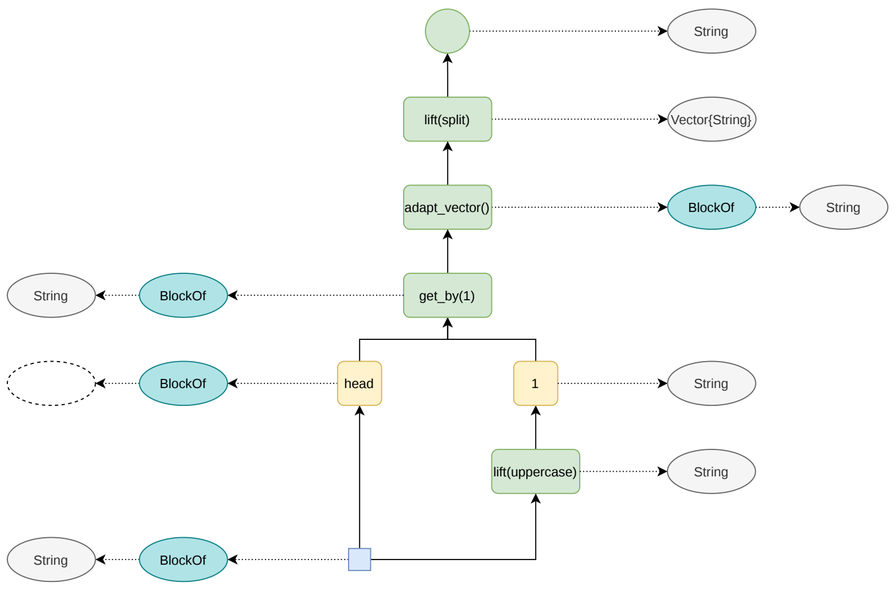

TupleOf
String
Int
String
TupleOf
Int

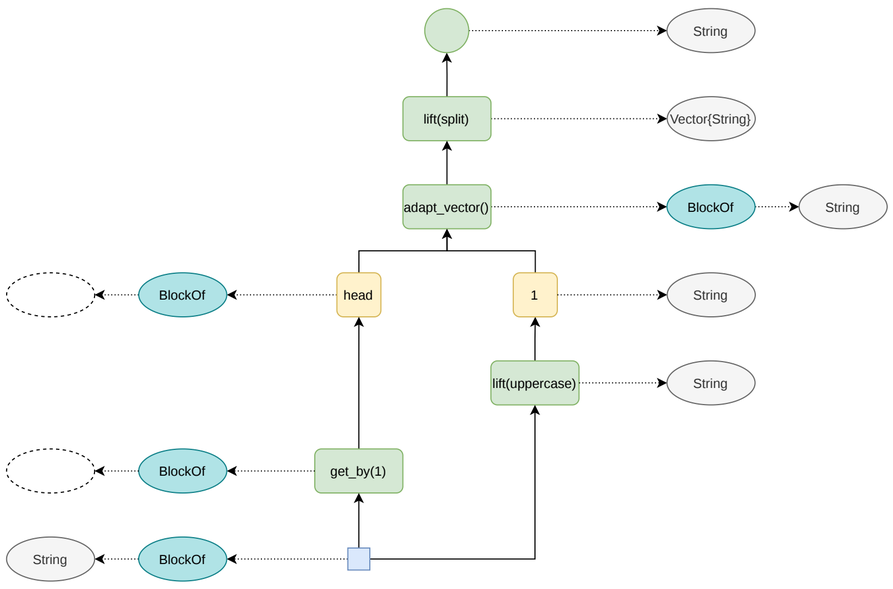

String
String

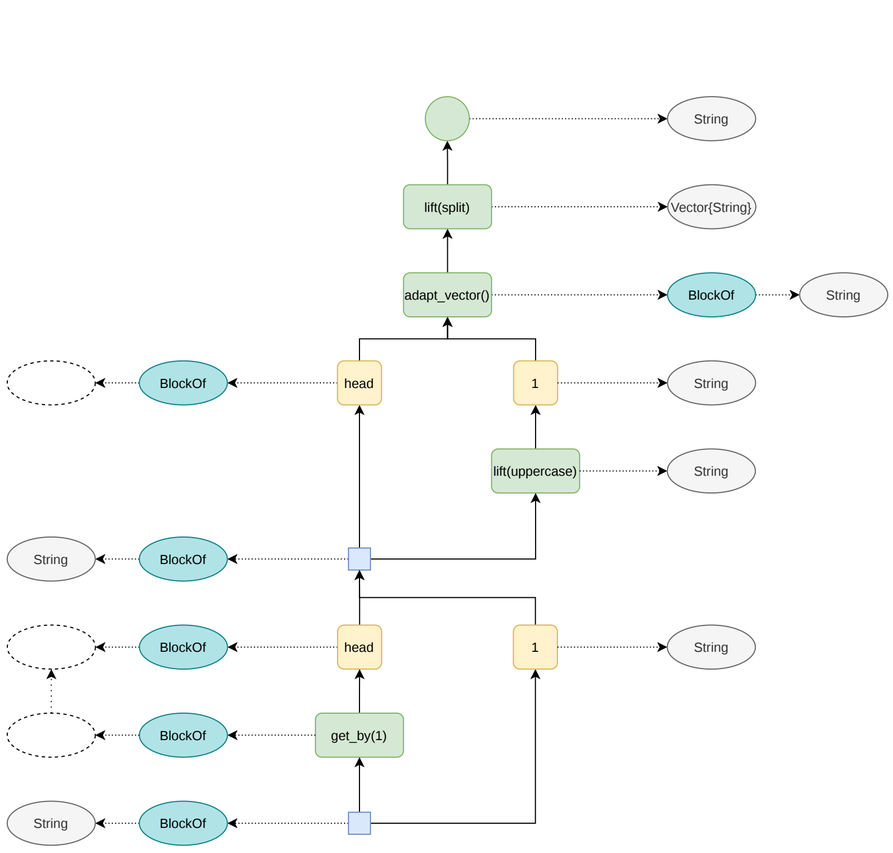
```
chain_of(lift(split),
         adapt_vector(),
         with_elements(lift(uppercase)),
         get_by(1, x0to1))
```

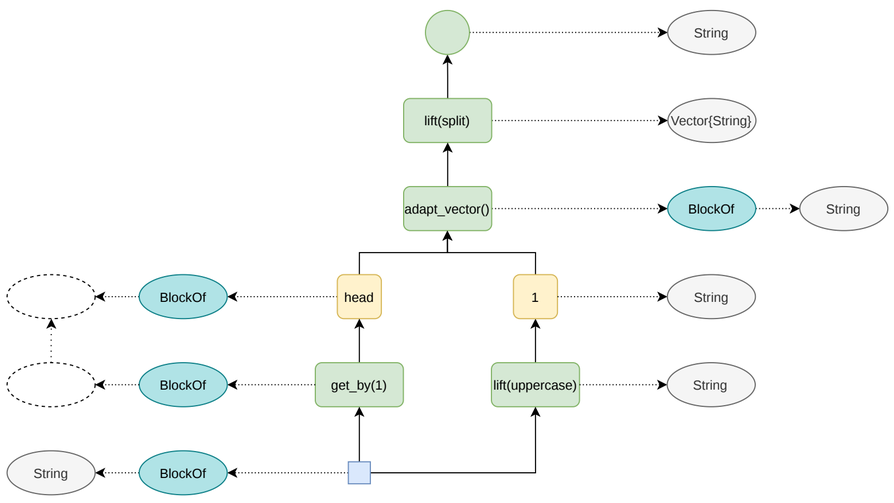




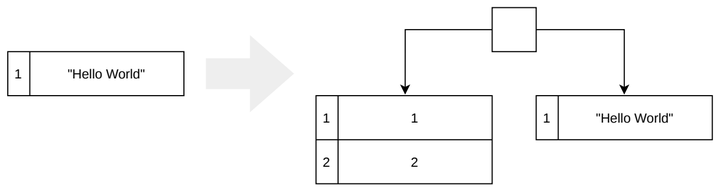


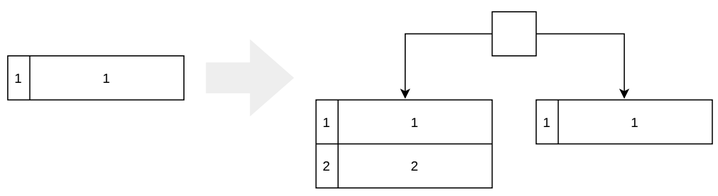


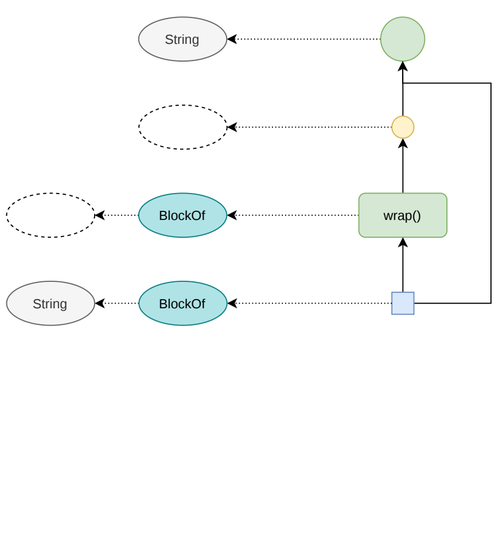




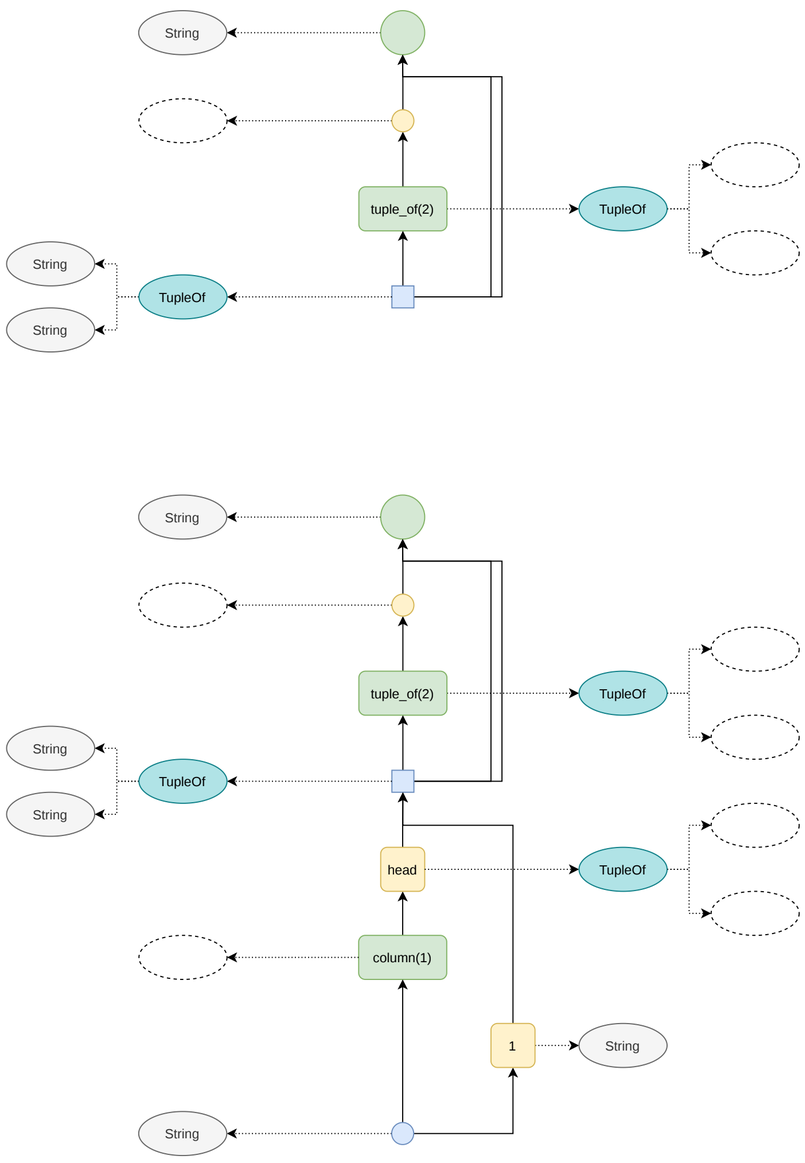
wrap()

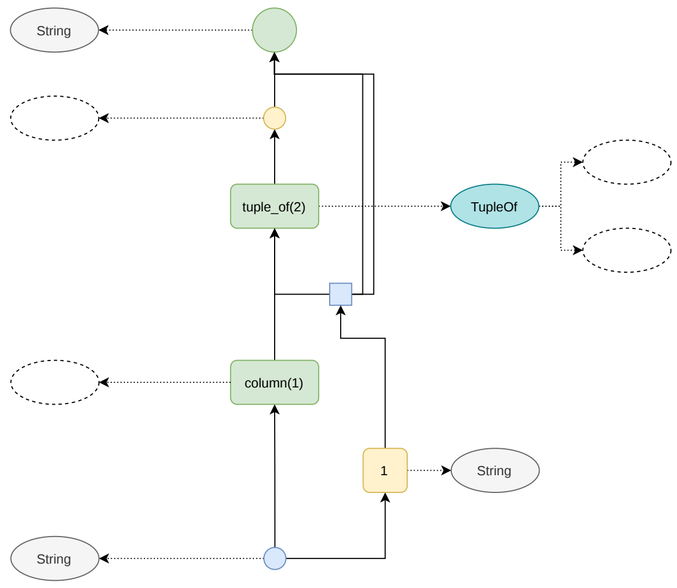

1
"Hello World"
1
1
2
2
1
"Hello World"


1
1
1
1
2
2
1
1

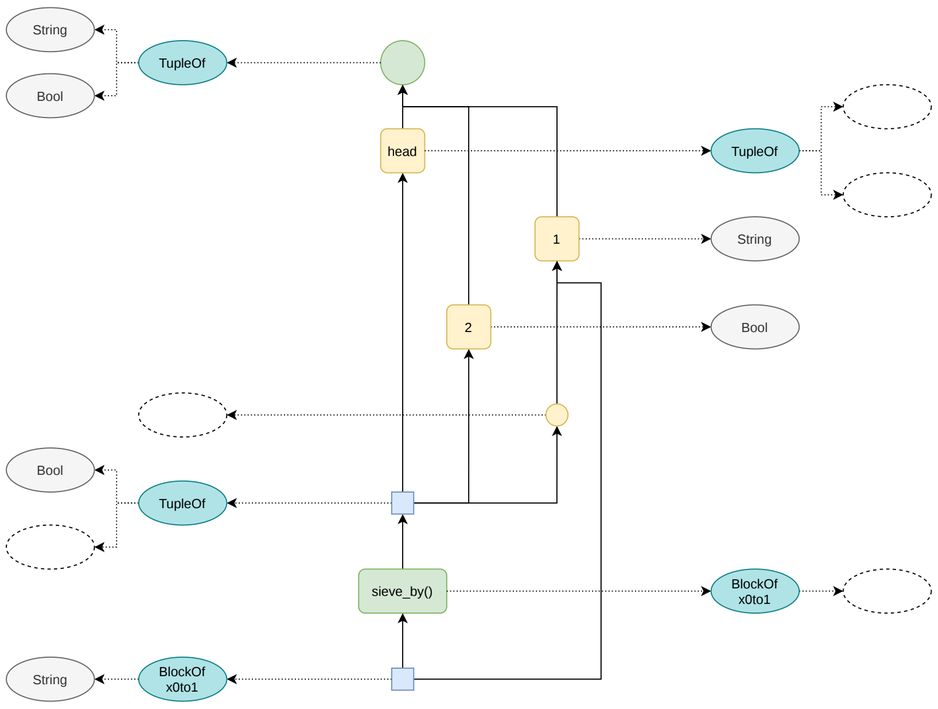

String
BlockOf
wrap()
String
BlockOf

chain_of(tuple_of(2), column(1))

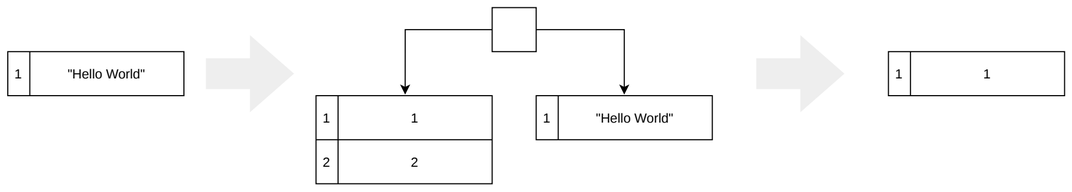


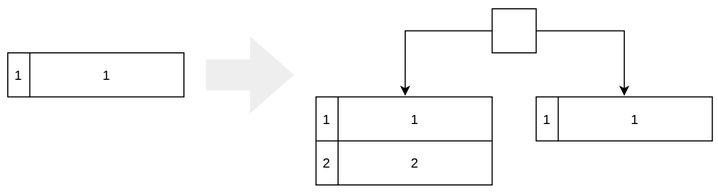


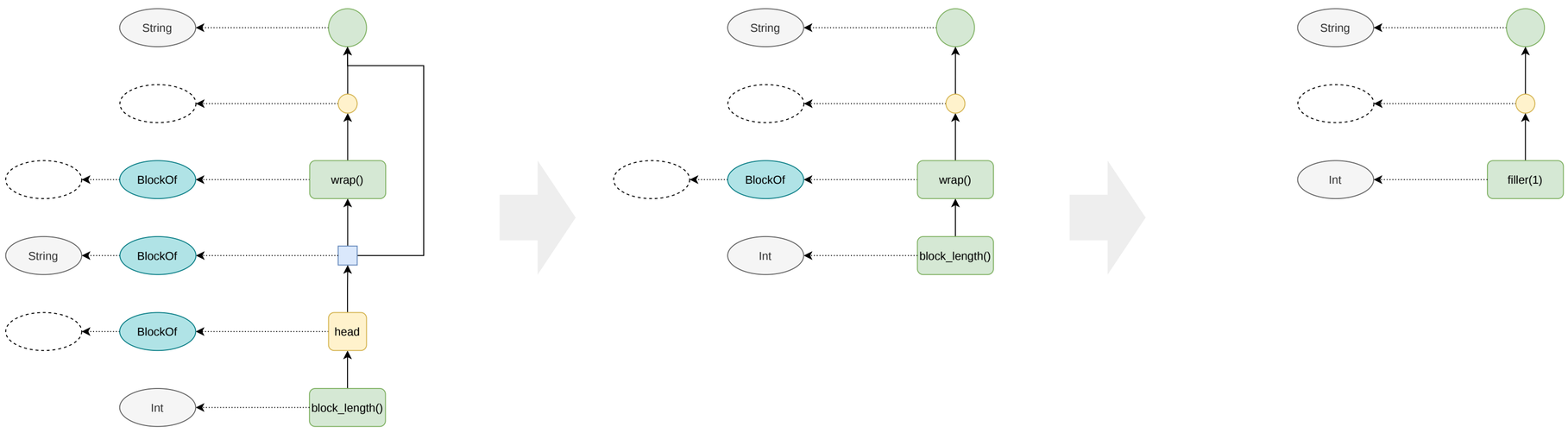
sieve_by()


String
TupleOf
Bool
head
TupleOf
1
String
2
Bool
Bool
TupleOf
sieve_by()
BlockOf
x0to1
String
BlockOf
x0to1

chain_of(wrap(), block_length())

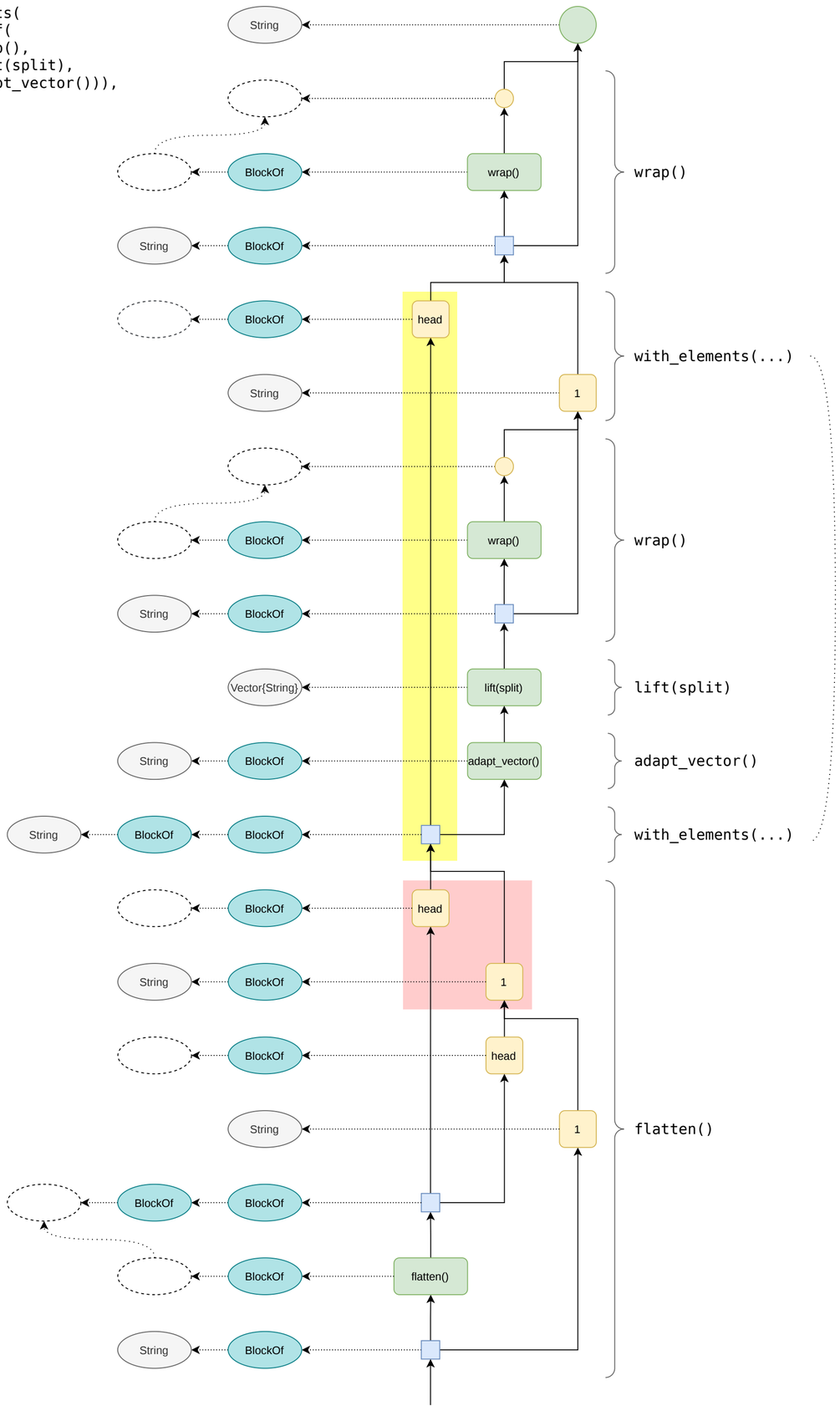


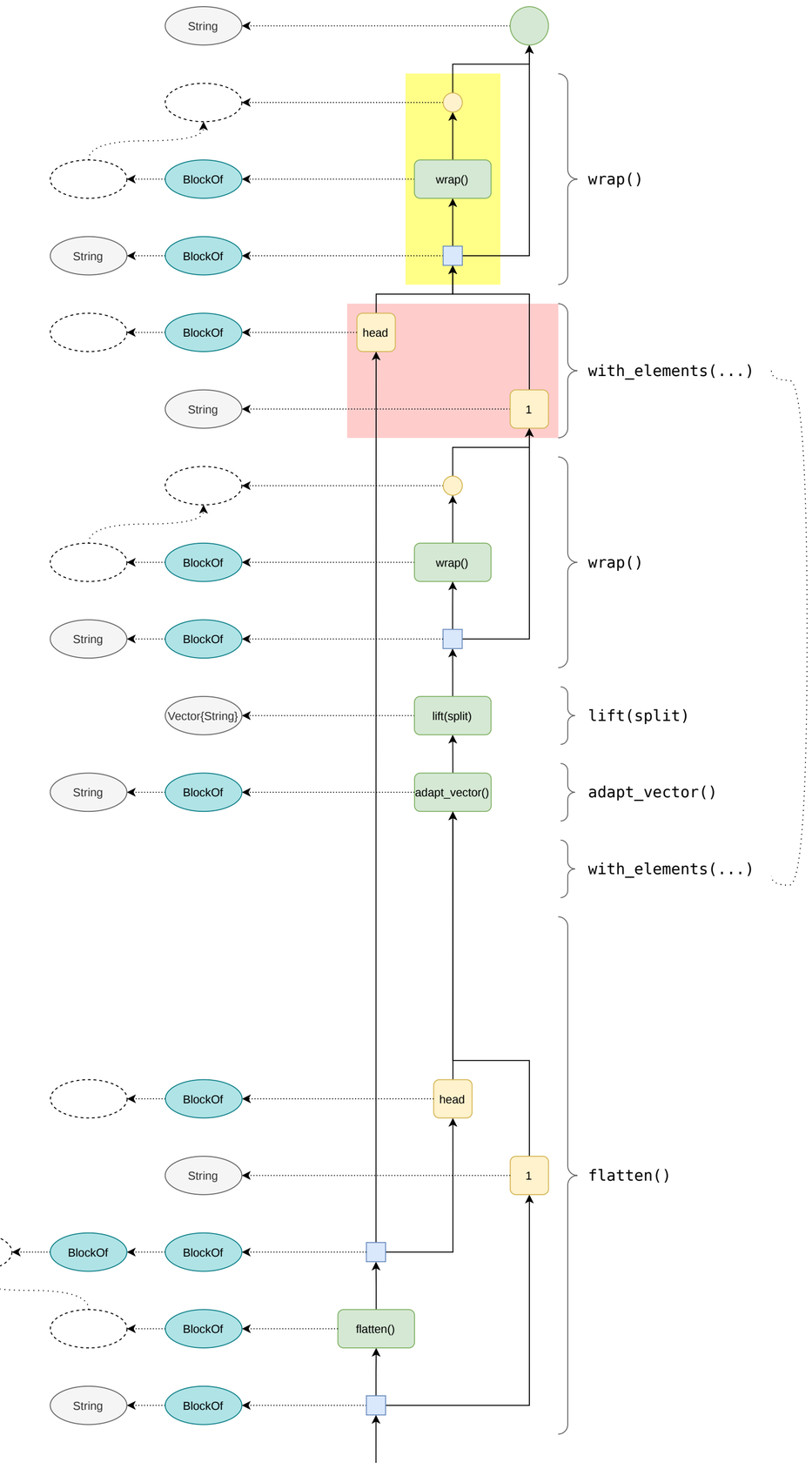




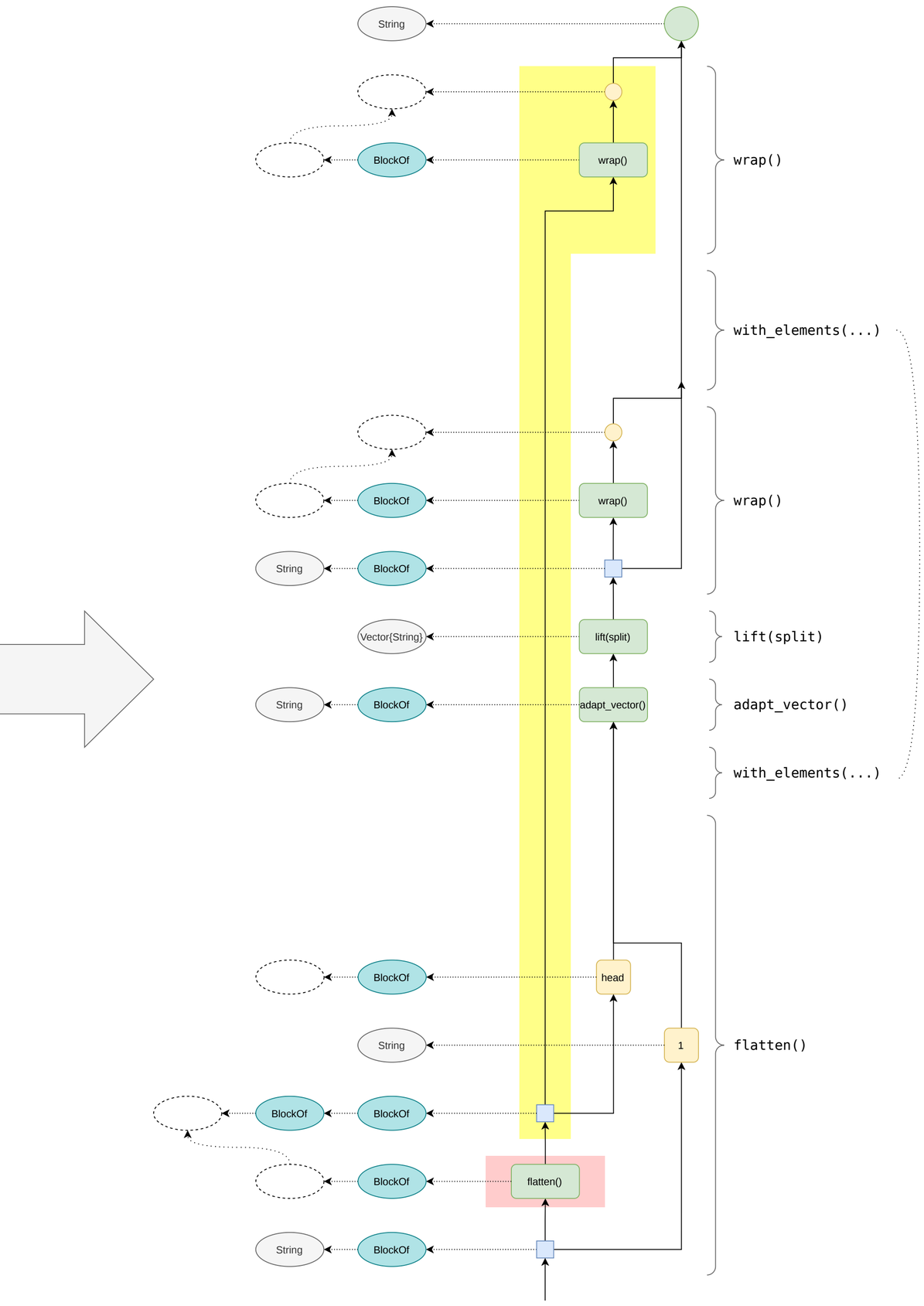
@query "Hello World" split(it)

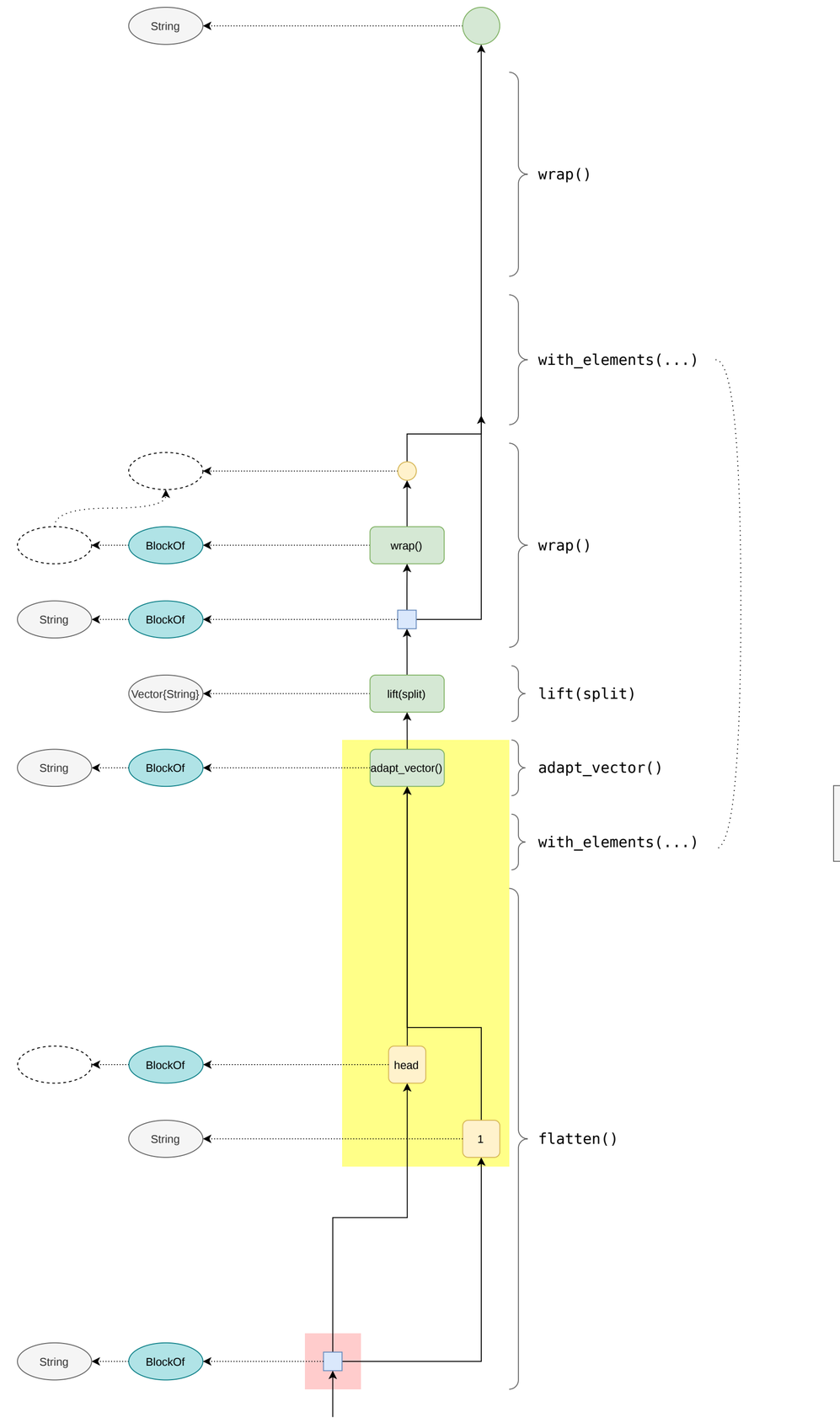
```
chain_of(
    wrap(),
    with_elements(
        chain_of(
            wrap(),
            lift(split),
            adapt_vector()))),
    flatten())
```

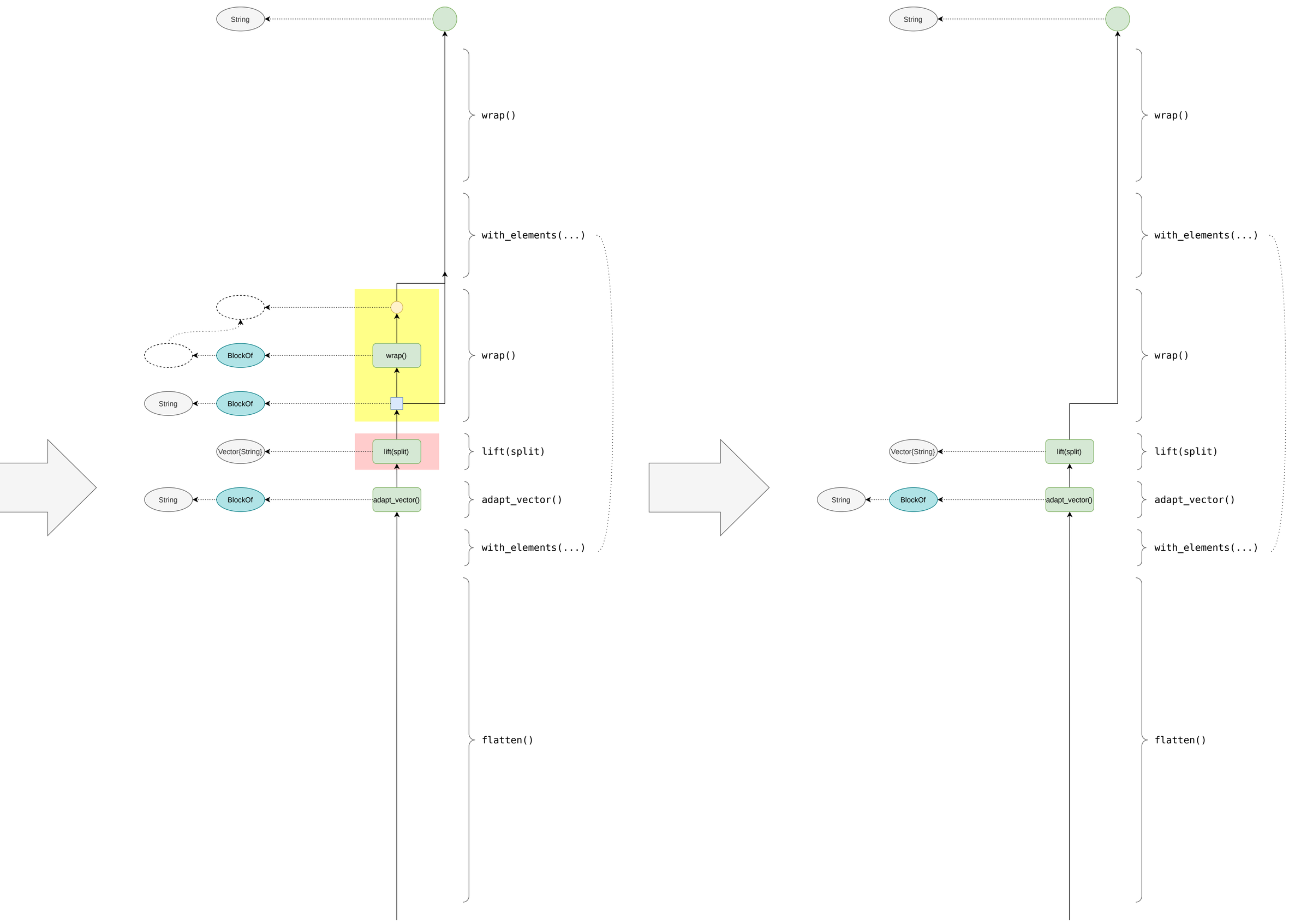




untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}

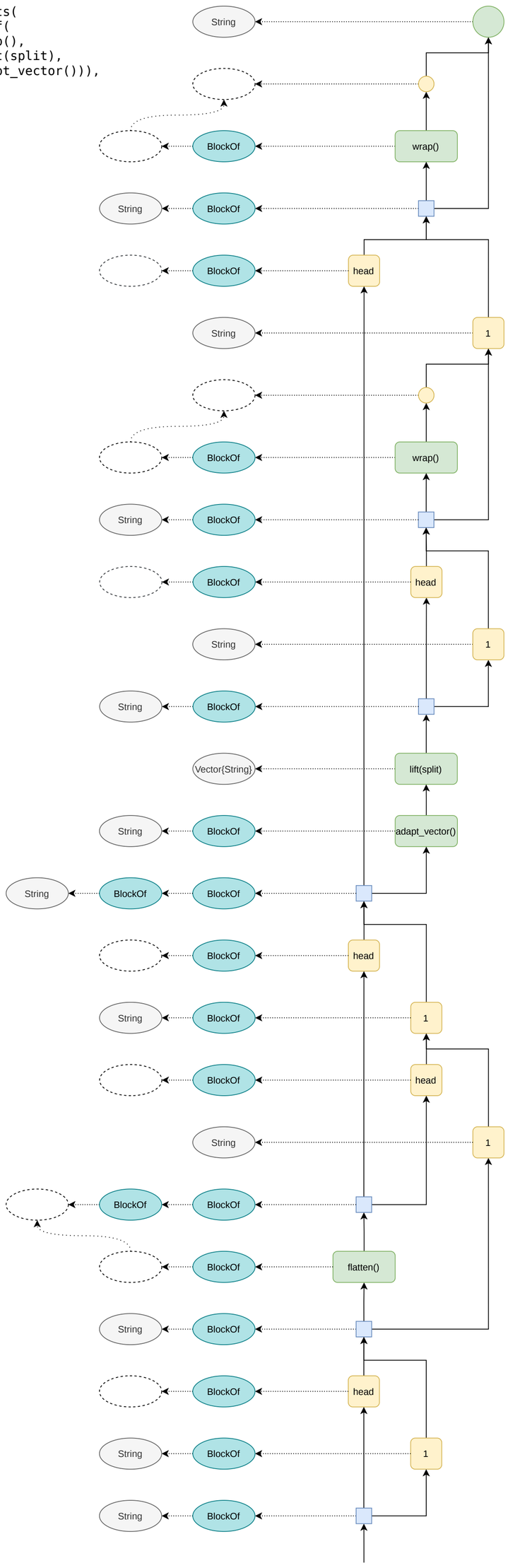

String
BlockOf
wrap()
wrap()
with_elements(...)
BlockOf
wrap()
wrap()
String
BlockOf
Vector{String}
lift(split)
lift(split)
String
BlockOf
adapt_vector()
adapt_vector()
with_elements(...)
BlockOf
head
String
1
flatten()
BlockOf
BlockOf
BlockOf
flatten()
String
BlockOf

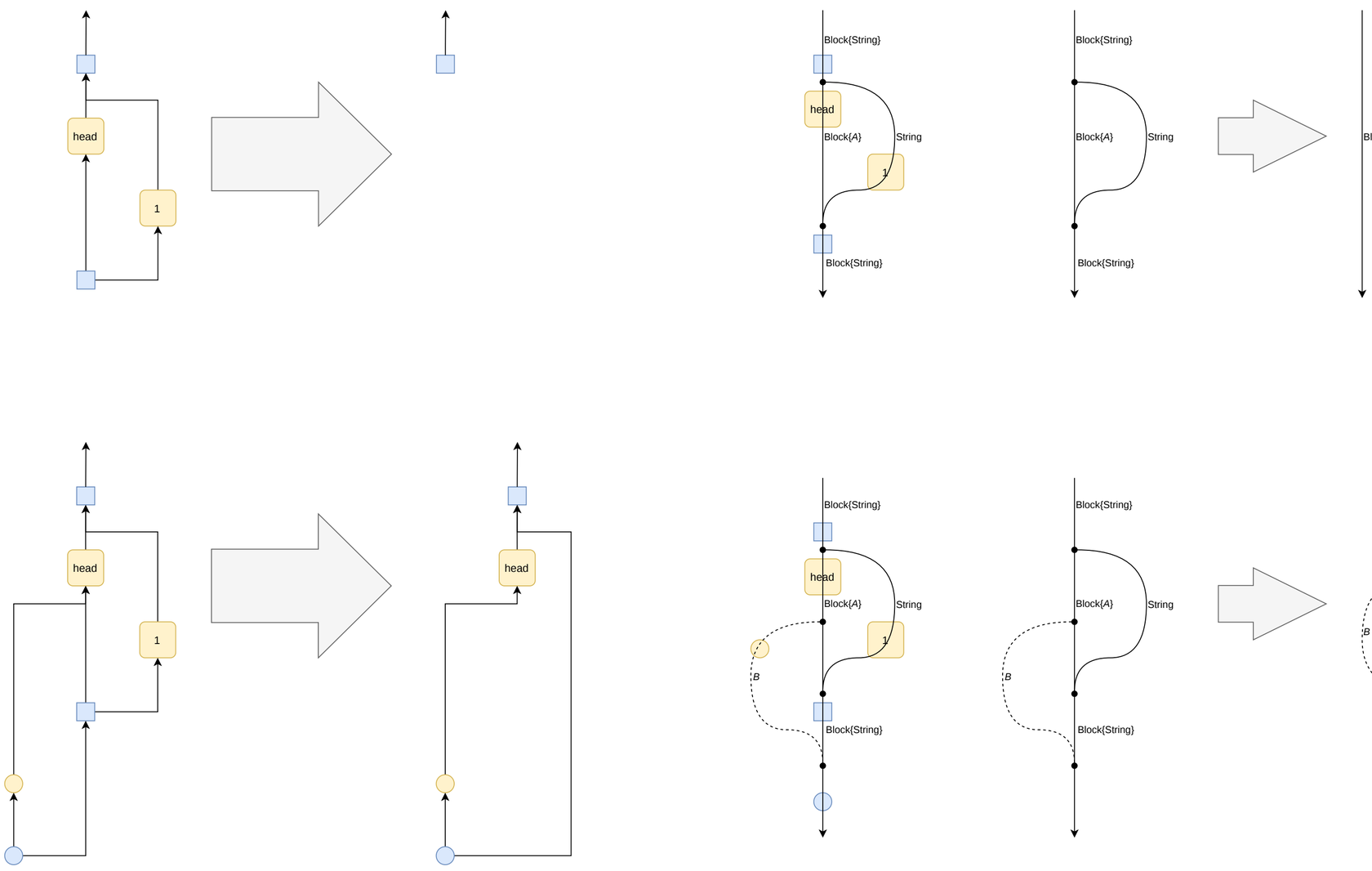

String
wrap()
wrap()
with_elements(...)
BlockOf
wrap()
wrap()
String
BlockOf
Vector{String}
lift(split)
lift(split)
String
BlockOf
adapt_vector()
adapt_vector()
with_elements(...)
BlockOf
head
String
1
flatten()
String
BlockOf

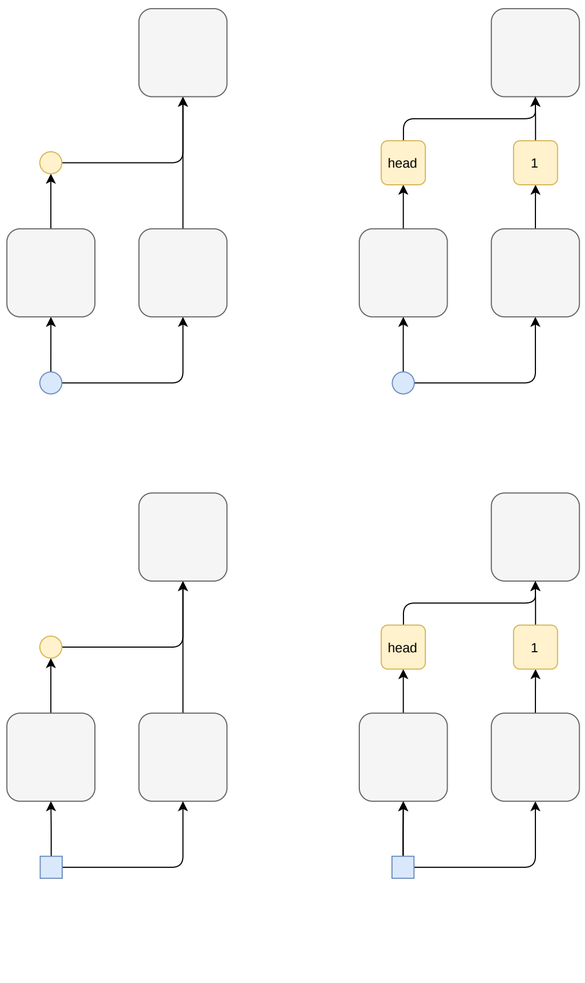

String
wrap()
with_elements(...)
BlockOf
wrap()
String
BlockOf
Vector{String}
lift(split)
lift(split)
String
BlockOf
adapt_vector()
adapt_vector()
with_elements(...)
flatten()
String
wrap()
with_elements(...)
wrap()
Vector{String}
lift(split)
lift(split)
String
BlockOf
adapt_vector()
adapt_vector()
with_elements(...)
flatten()

@query "Hello World" split(it)

```
chain_of(
    wrap(),
    with_elements(
        chain_of(
            wrap(),
            lift(split),
            adapt_vector())),
    flatten())
```







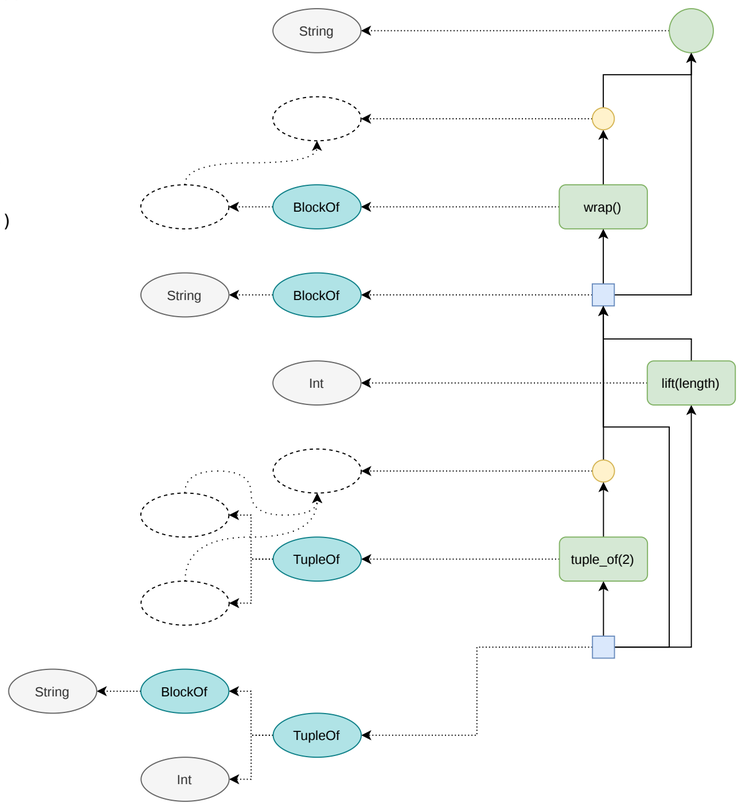
head
1

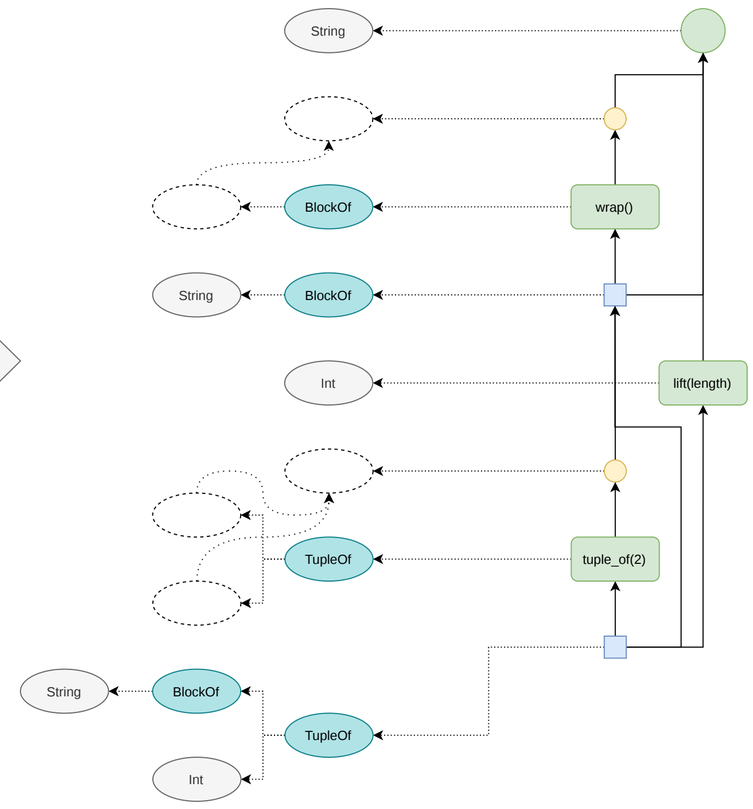
head
1

```
chain_of(
    wrap(),
    tuple_of(
        pass(),
        lift(length)))
```

```
tuple_of(
    wrap(),
    chain_of(wrap(), lift(length))))
```

```
chain_of(
    tuple_of(wrap(),
             pass()),
    tuple_of(
        column(1),
        lift(length)))
```



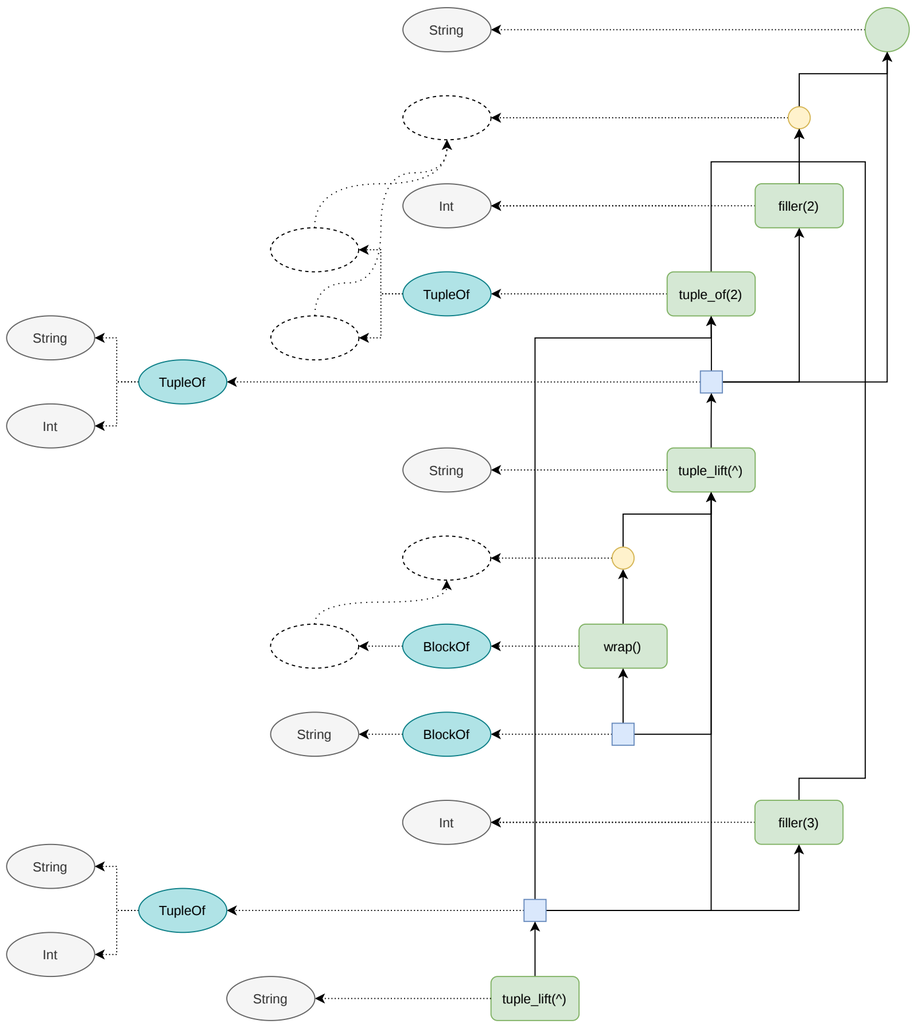



```
tuple_of(
    wrap(),
    lift(length)))
```

untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}

chain_of(f, tuple_of(g, h)) => tuple_of(chain_of(f, g), chain_of(f, h))

```
{it ^ 2, (it ^ 2) ^ 3}
```



untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}

@query "Hello World" split(it)

```
chain_of(
    wrap(),
    with_elements(wrap()),
    with_elements(lift(split)),
    with_elements(adapt_vector()),
    flatten())
```