

```

struct EvalNode
  p::PipeLine
  l::NodeRef
end

Nothing

struct PartNode
  n::NodeRef
  k::Int
end

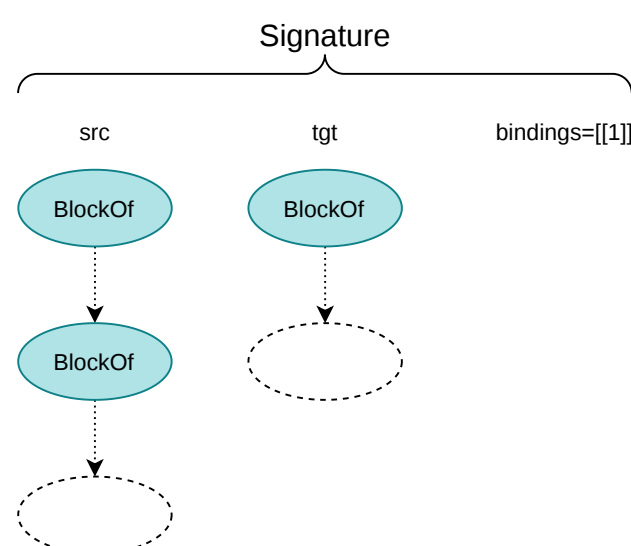
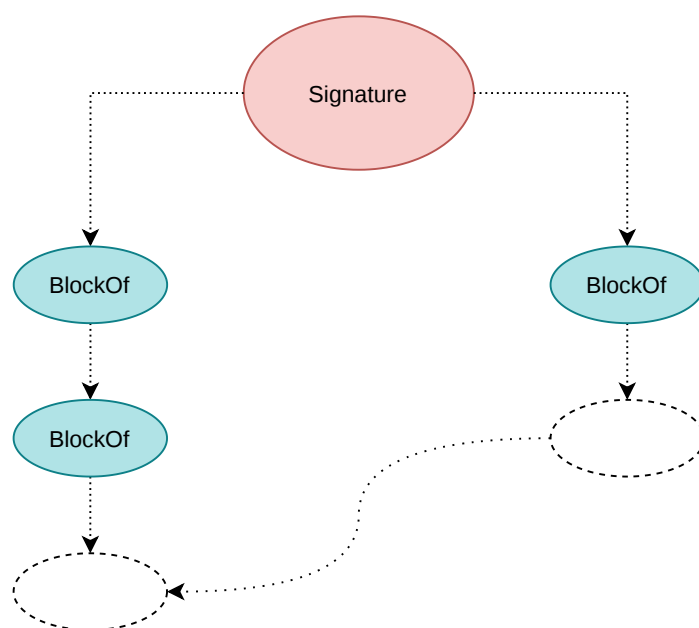
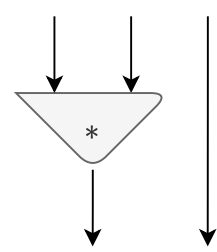
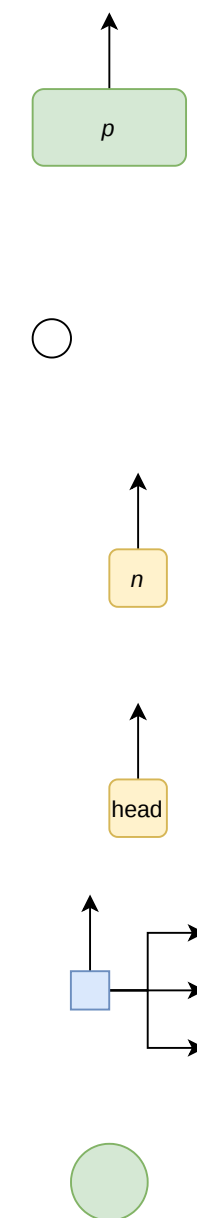
struct HeadNode
  n::NodeRef
end

struct JoinNode
  head::NodeRef
  parts::Vector{NodeRef}
end

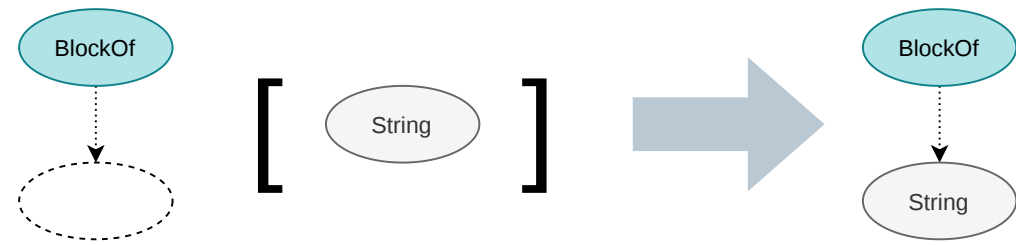
struct RootNode
end

mutable struct NodeRef
  node::Union{Nothing,Node}
  shp::AbstractShape
end

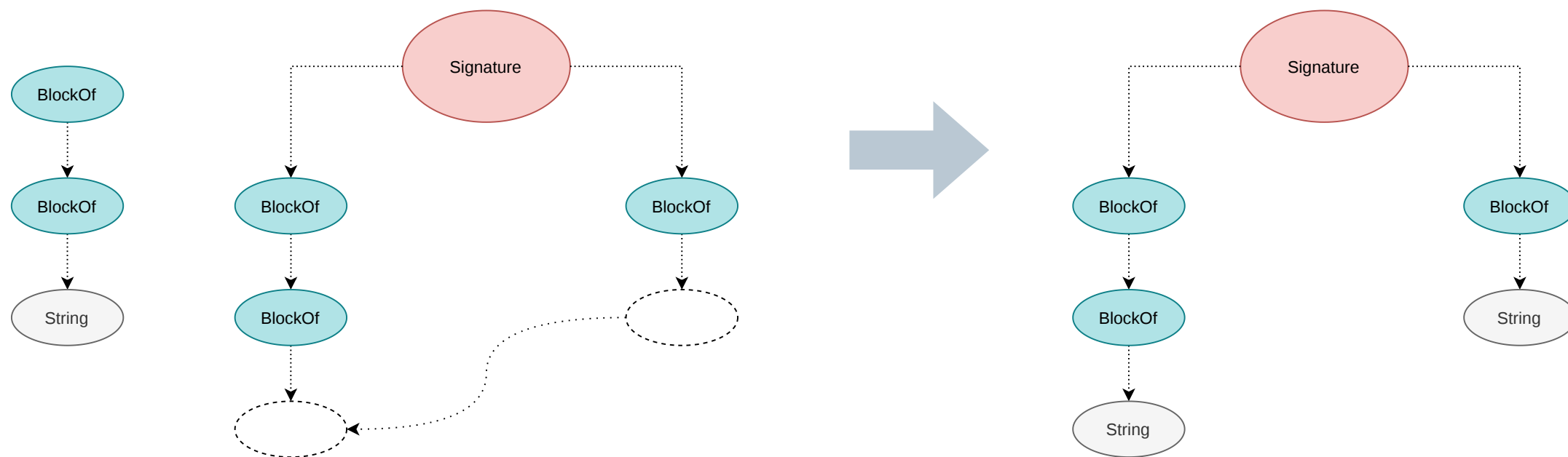
```



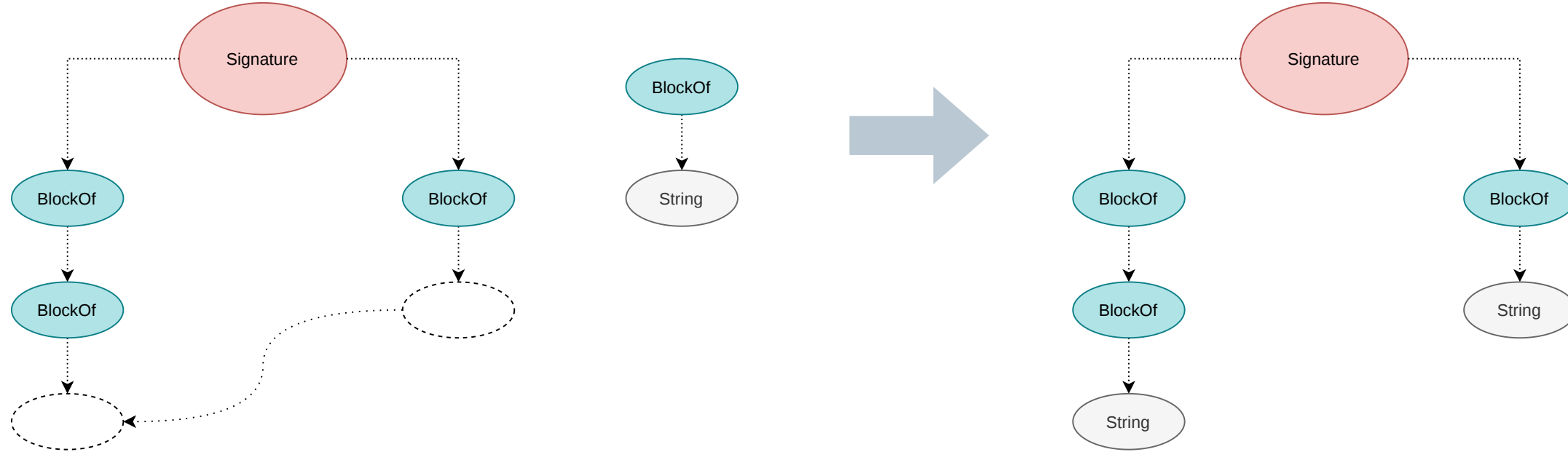
substitution(shp, repl)



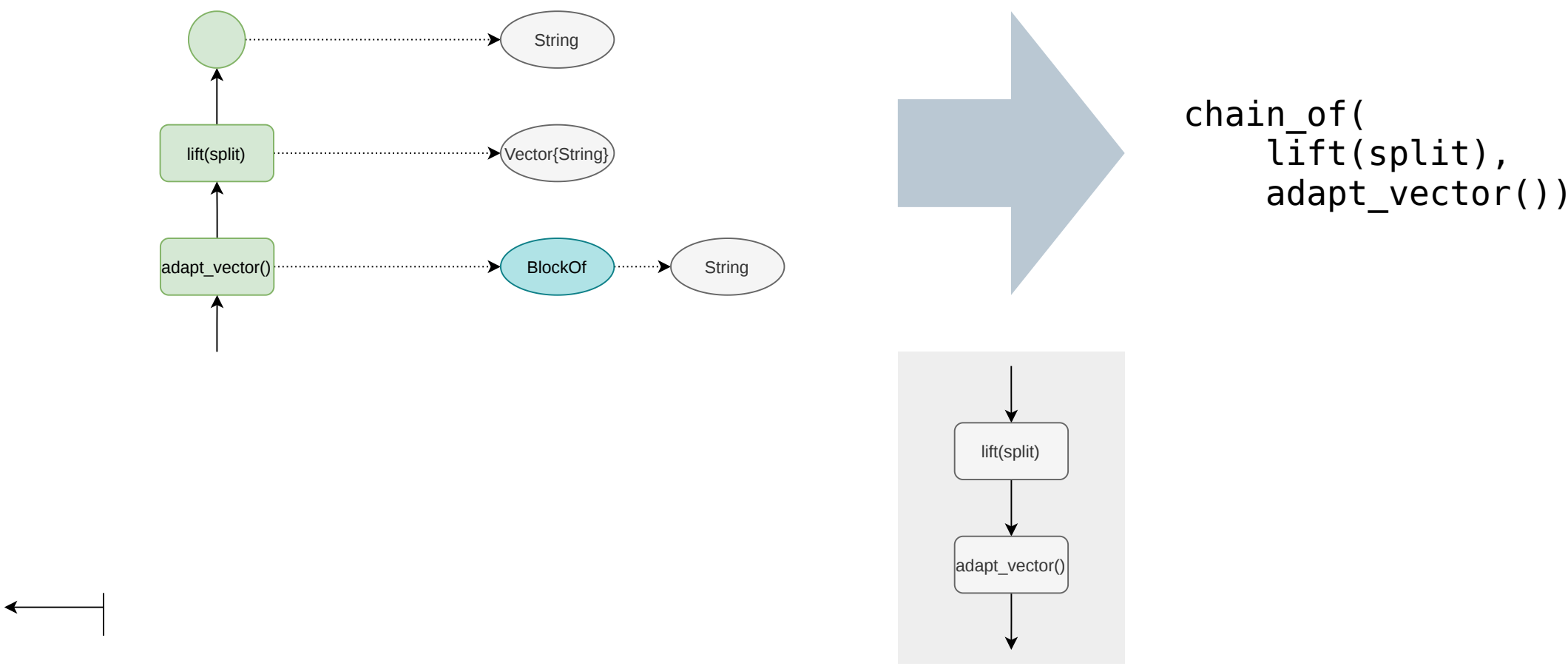
refine_source(shp, sig)



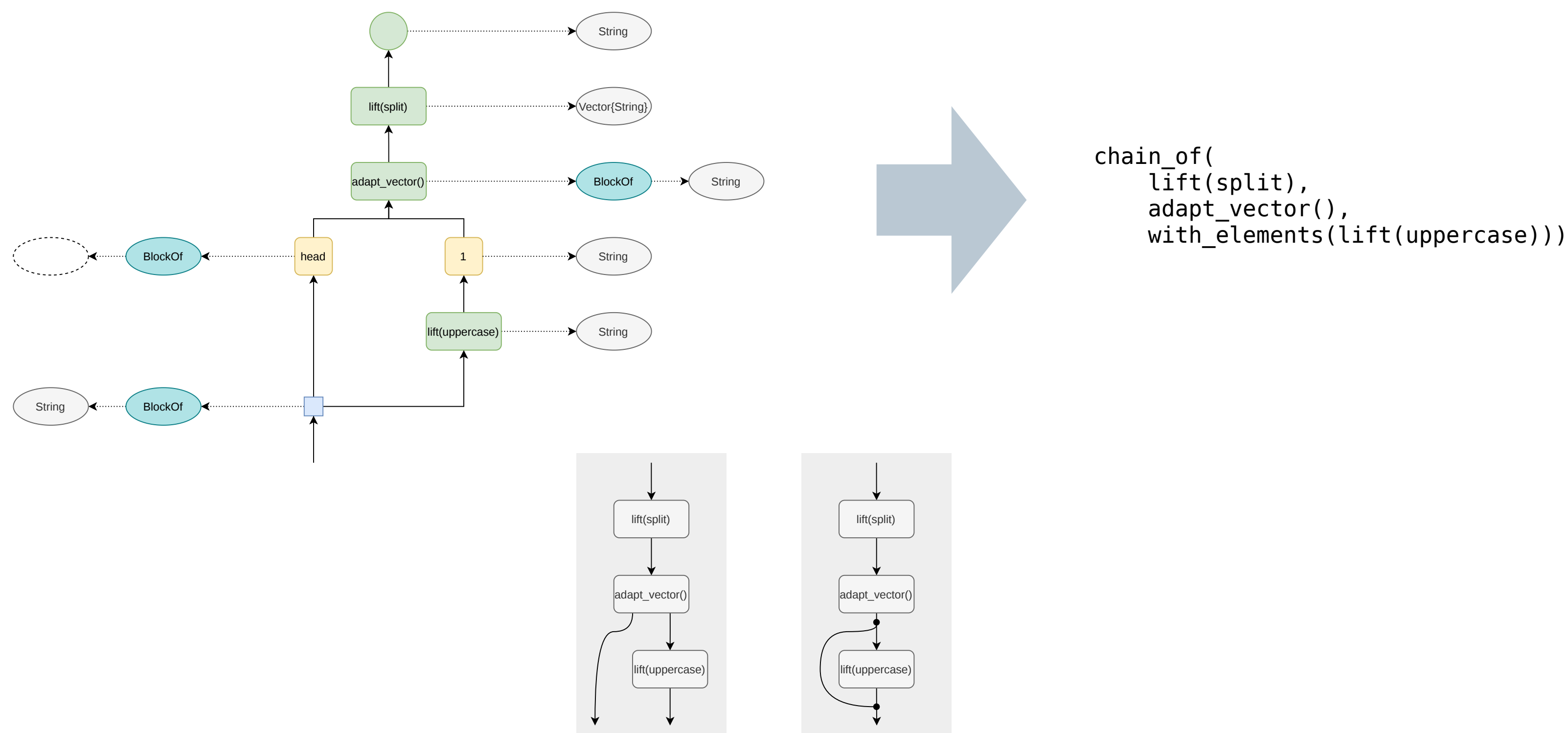
refine_target(sig, shp)



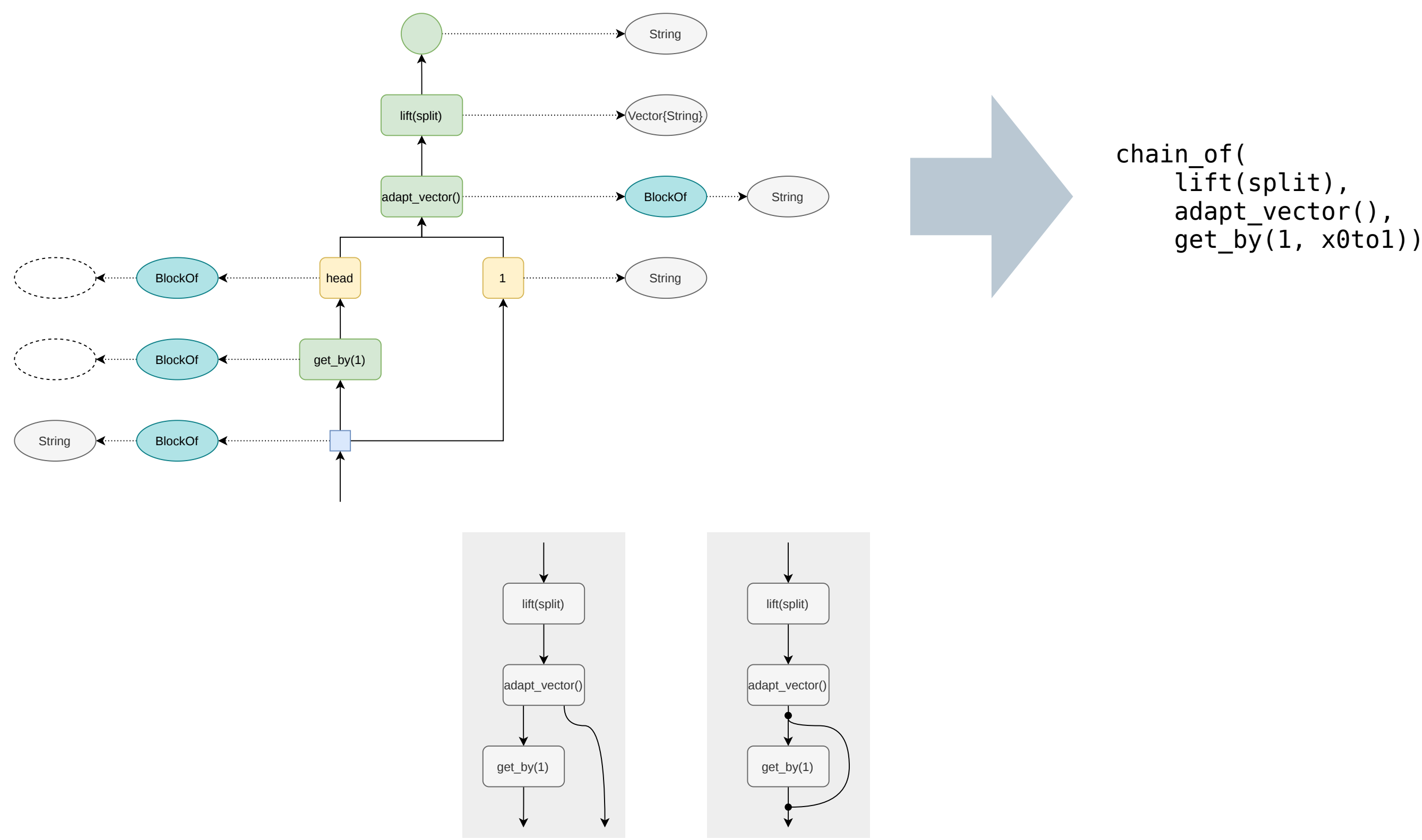
@query "Hello World" split(it) # after rewrite_unwrap



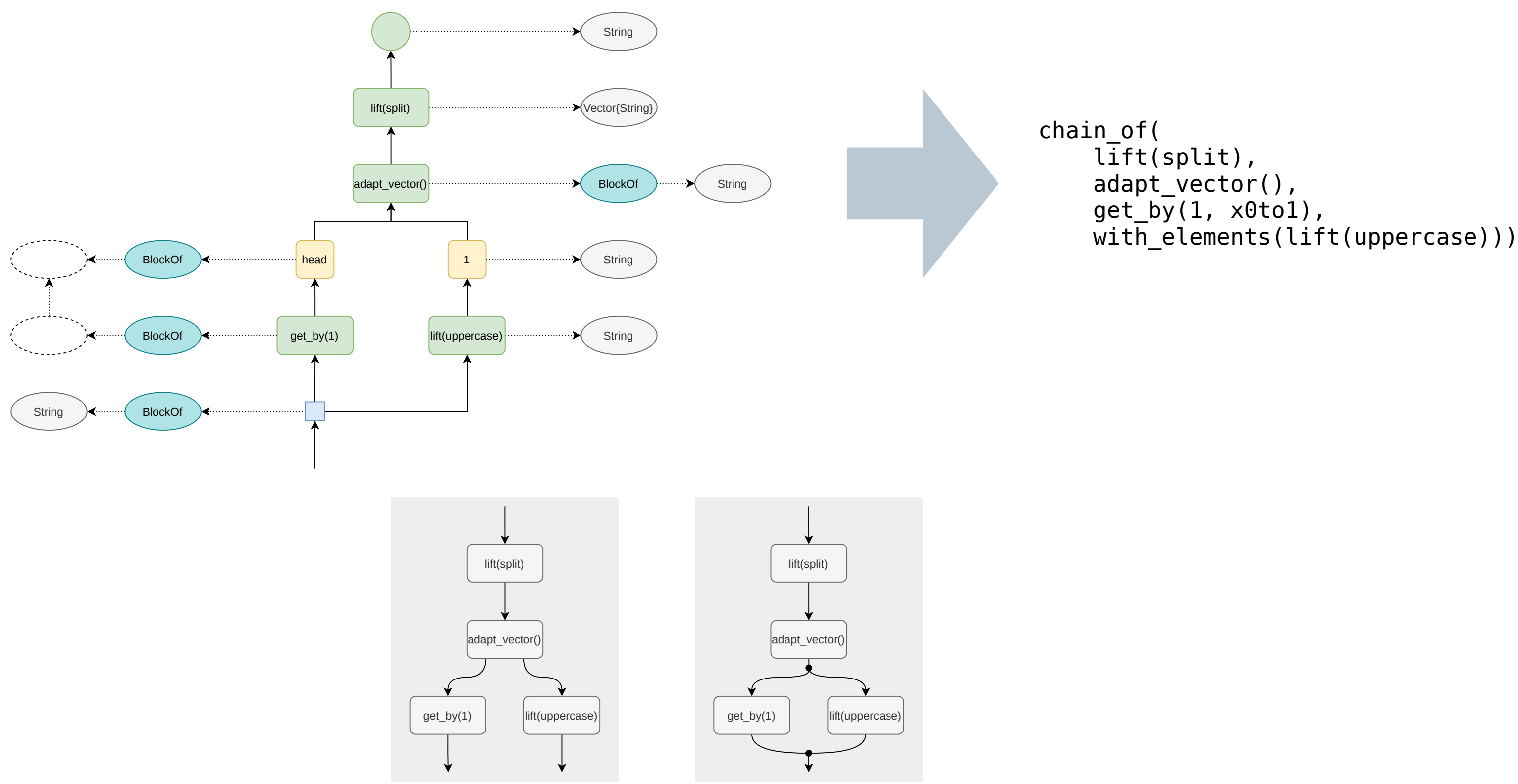
@query "Hello World" split(it).uppercase(it) # after rewrite_unwrap



@query "Hello World" split(it).first() # after rewrite_unwrap



```
@query "Hello World" split(it).uppercase(it).first() # after rewrite_unwrap
```



@query "Hello World" split(it)

