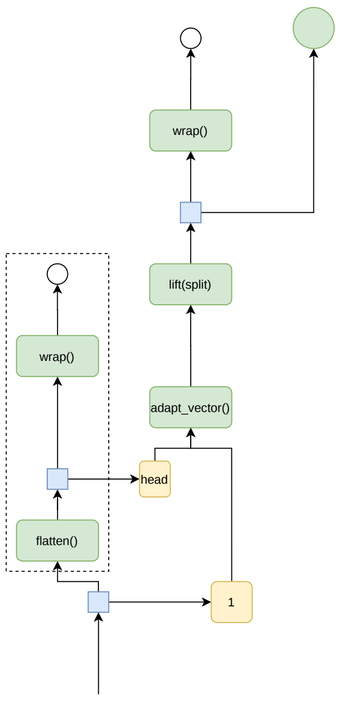
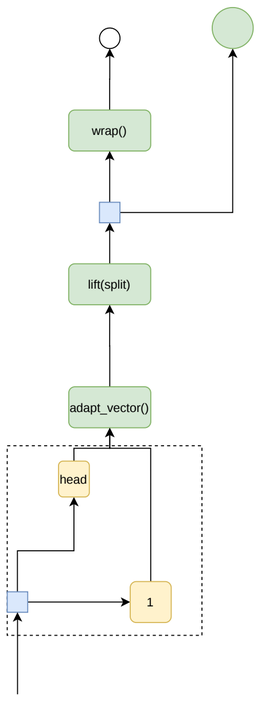
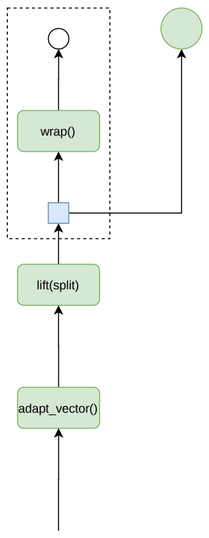
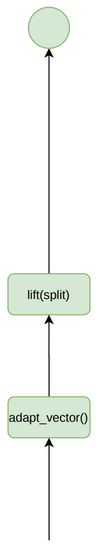
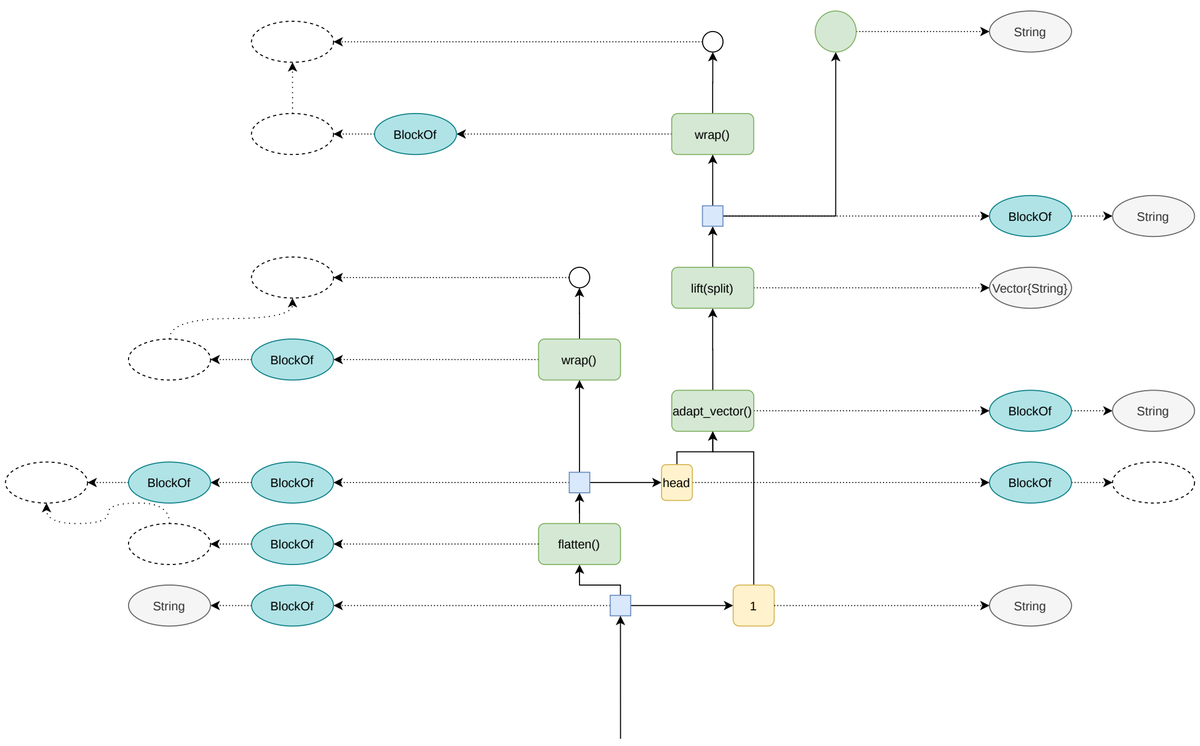

wrap()
lift(split)
wrap()
adapt_vector()
head
flatten()
1


wrap()
lift(split)
adapt_vector()
head
1


wrap()
lift(split)
adapt_vector()


lift(split)
adapt_vector()




```
struct EvalNode
    p::Pipeline
    i::NodeRef
end
```

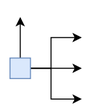
```
Nothing
```



```
struct PartNode
    n::NodeRef
    k::Int
end
```

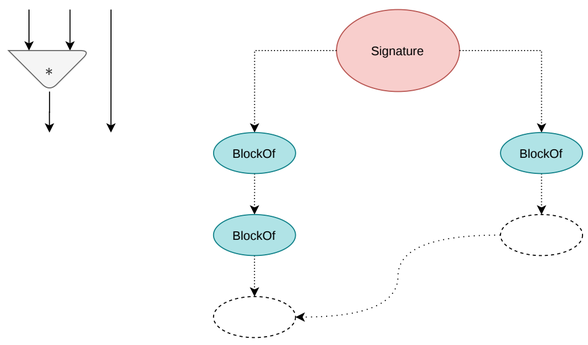
```
struct HeadNode
    n::NodeRef
end
```

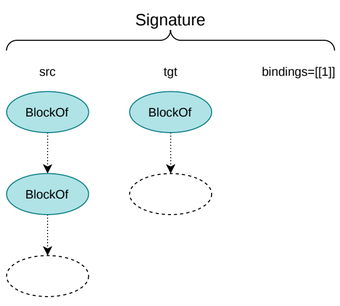
```
struct JoinNode
    head::NodeRef
    parts::Vector{NodeRef}
end
```



```
struct RootNode
end
```

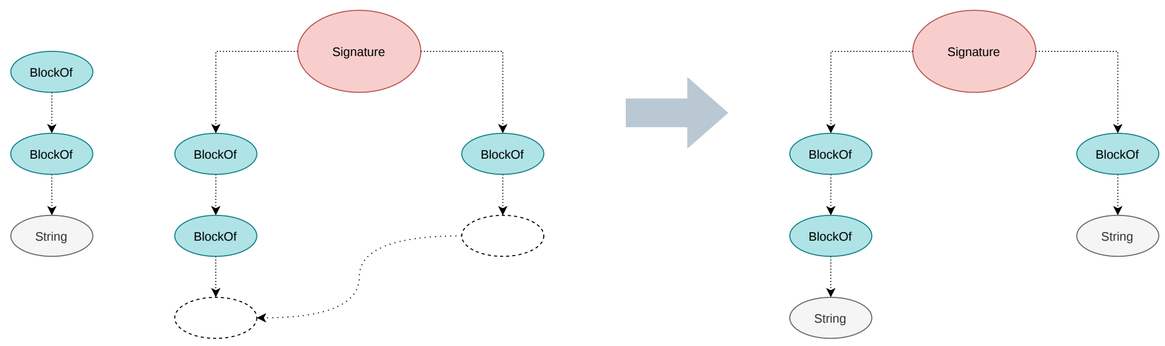
```
mutable struct NodeRef
    node::Union{Nothing,Node}
    shp::AbstractShape
end
```



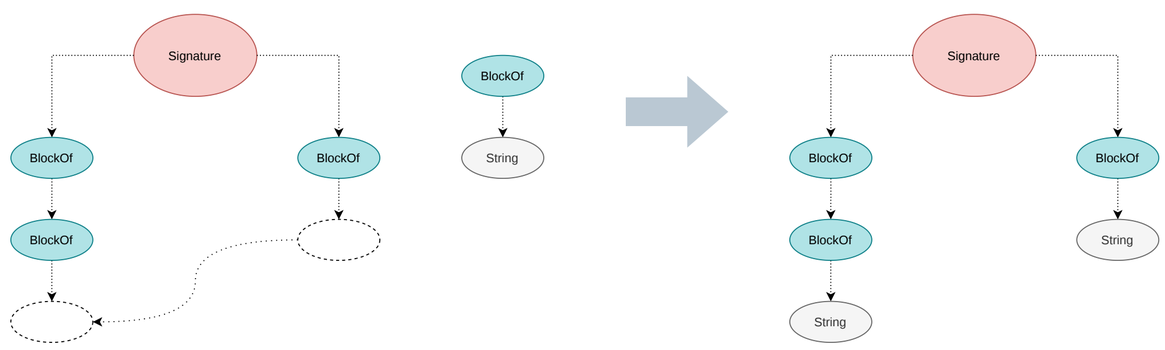


substitution(shp, repl)

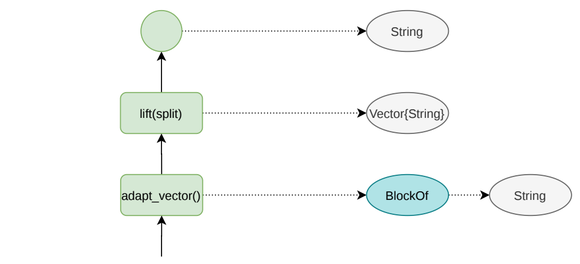
refine_source(shp, sig)



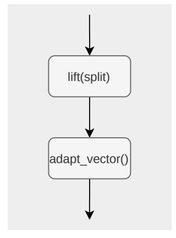
refine_target(sig, shp)



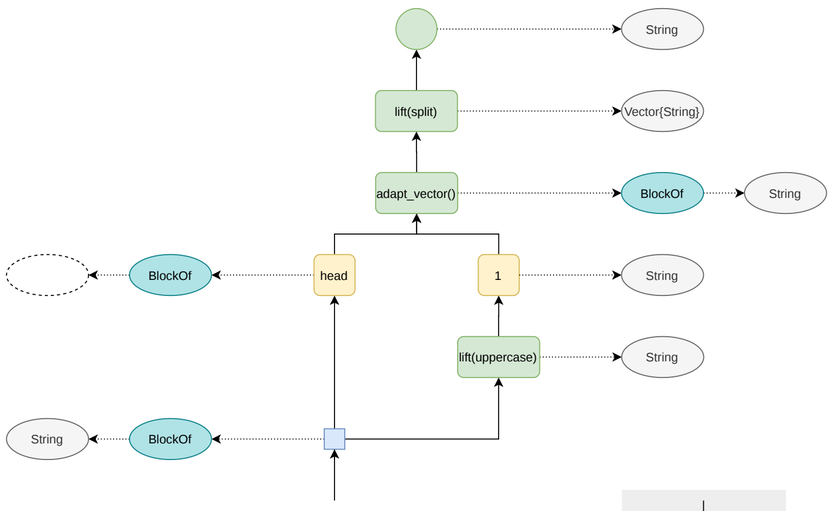
@query "Hello World" split(it) # after rewrite_unwrap

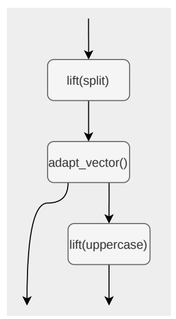


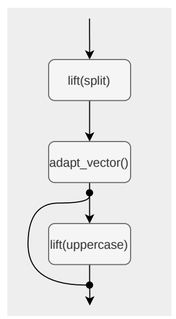
```
chain_of(
    lift(split),
    adapt_vector())
```



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

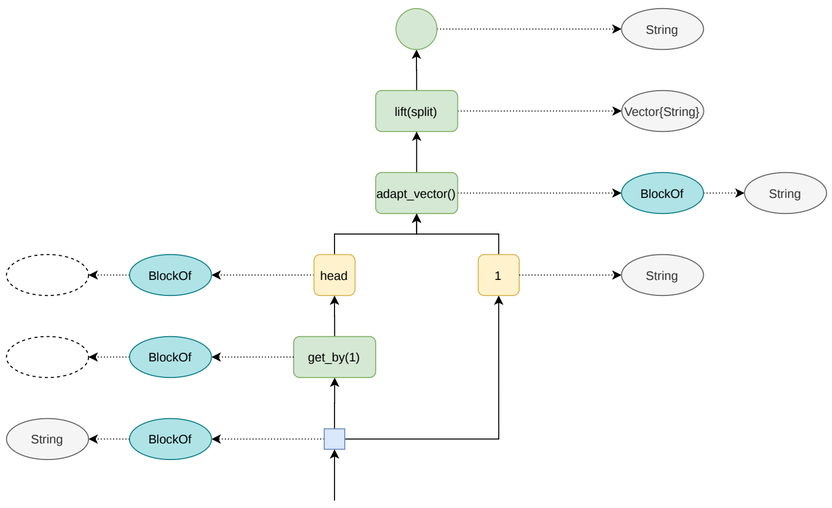


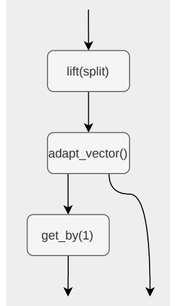


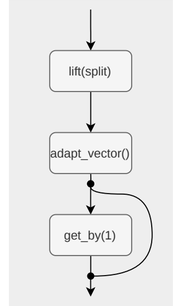



```
chain_of(
    lift(split),
    adapt_vector(),
    with_elements(lift(uppercase)))
```

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

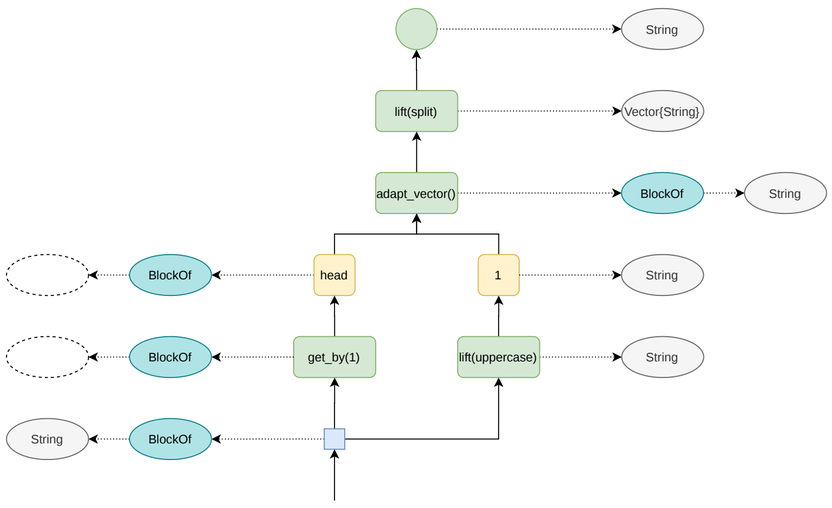


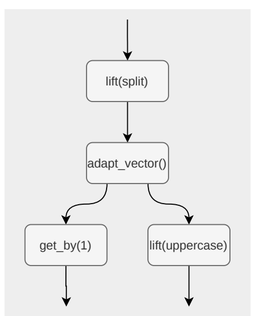


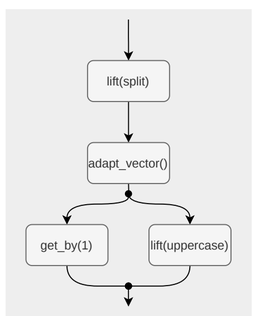


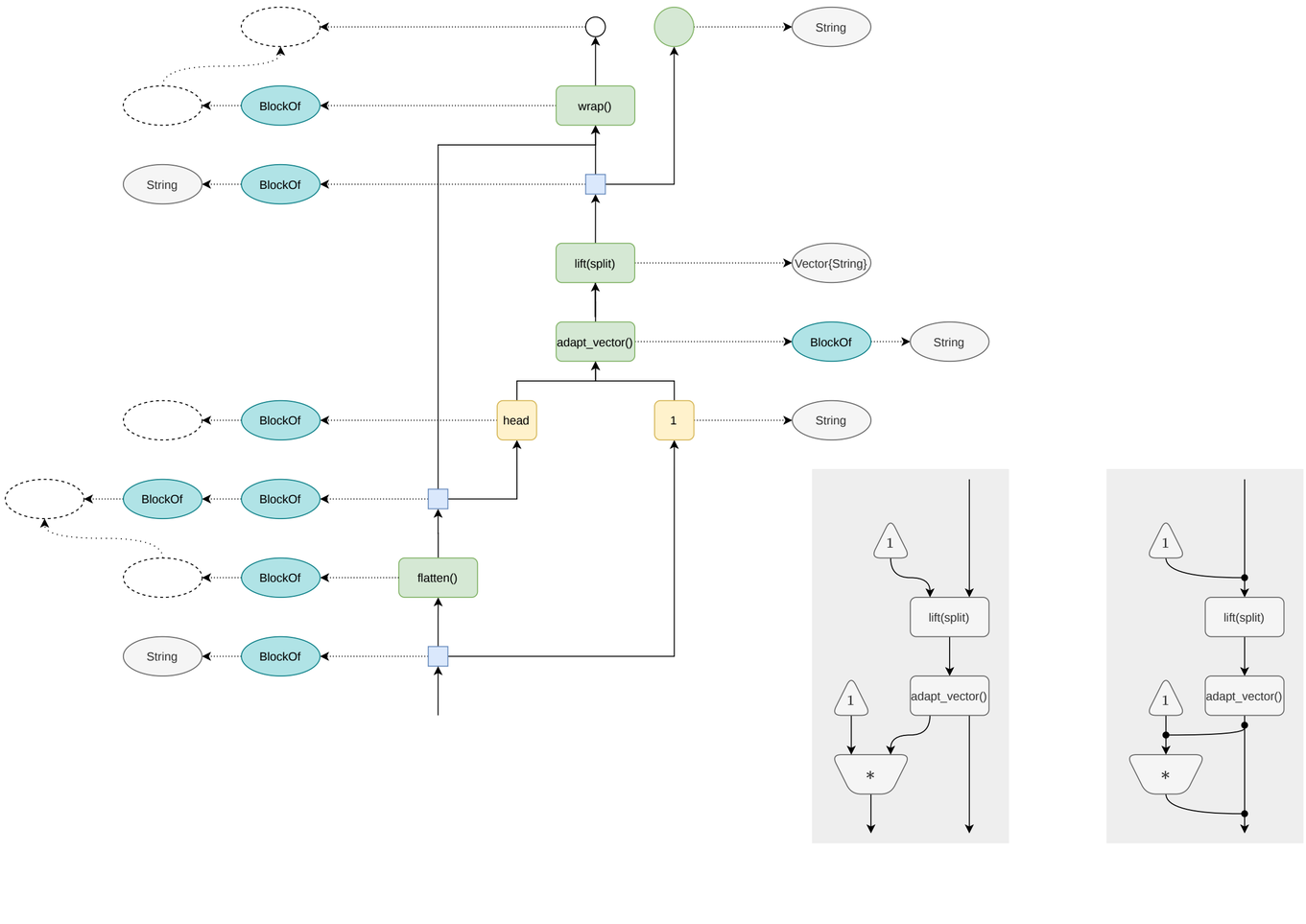

```
chain_of(
    lift(split),
    adapt_vector(),
    get_by(1, x0to1))
```

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








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

@query "Hello World" split(it)