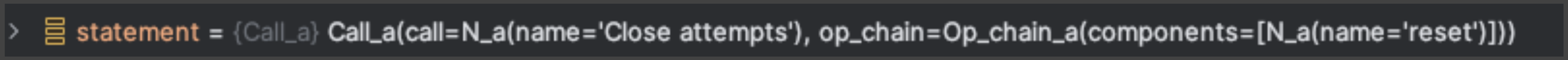
Call parse analysis

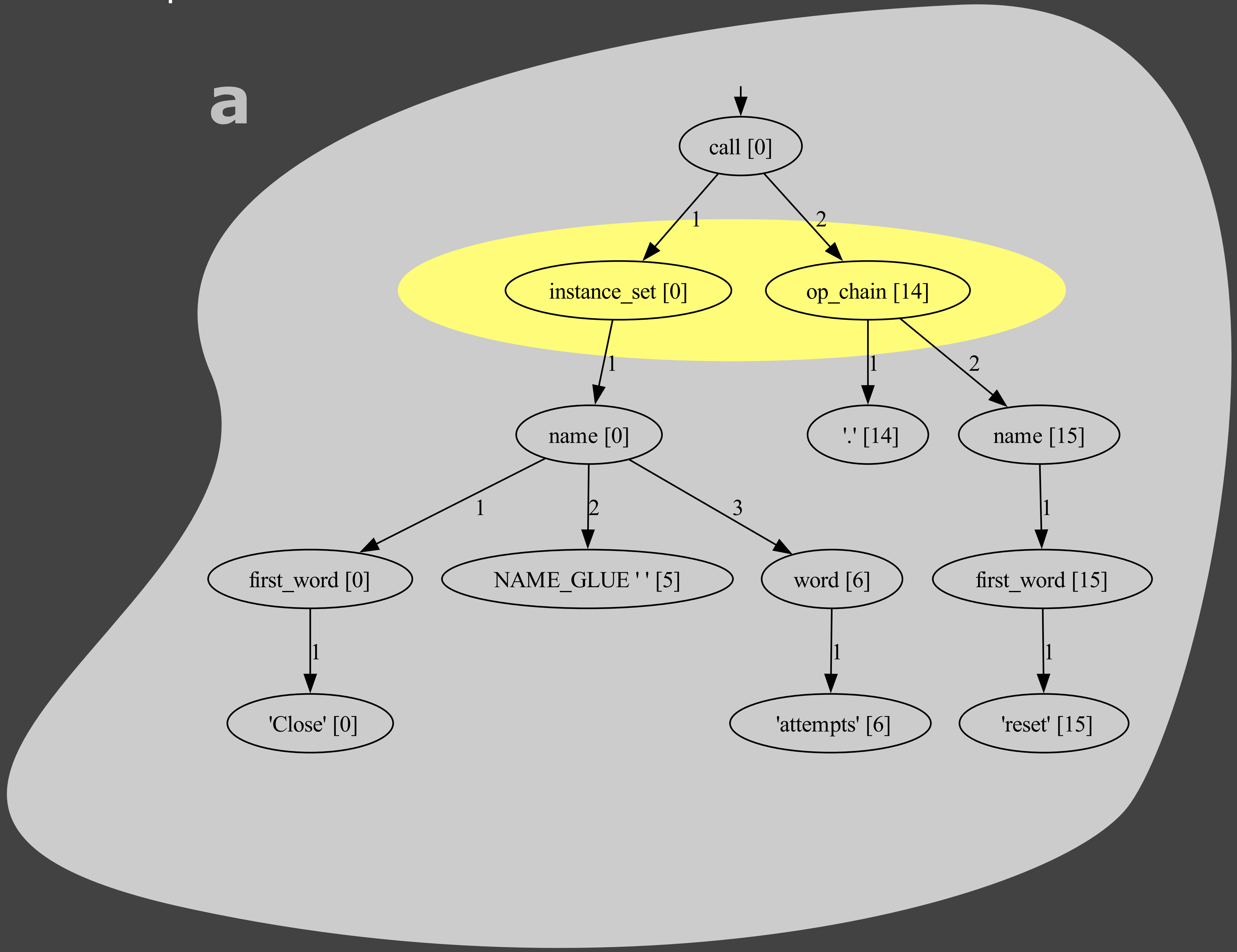
Leon Starr
2025-9-27/ v0.2

Here we study the Scrall parser tree and named tuples resulting from a parse of a variety of operation chains following an attribute write action so that we can figure out how to populate the actions in the *x*UML populate tool

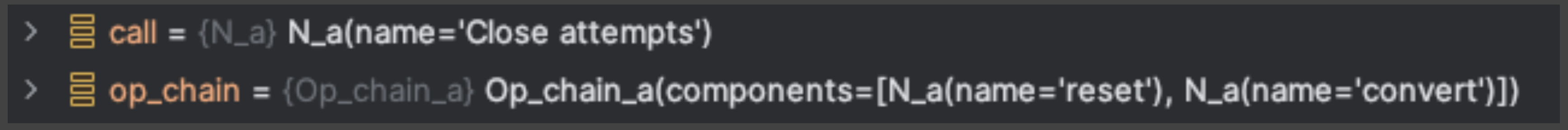


```
> statement = {Call_a} Call_a(call=N_a(name='Close attempts'), op_chain=Op_chain_a(components=[N_a(name='reset')]))
```

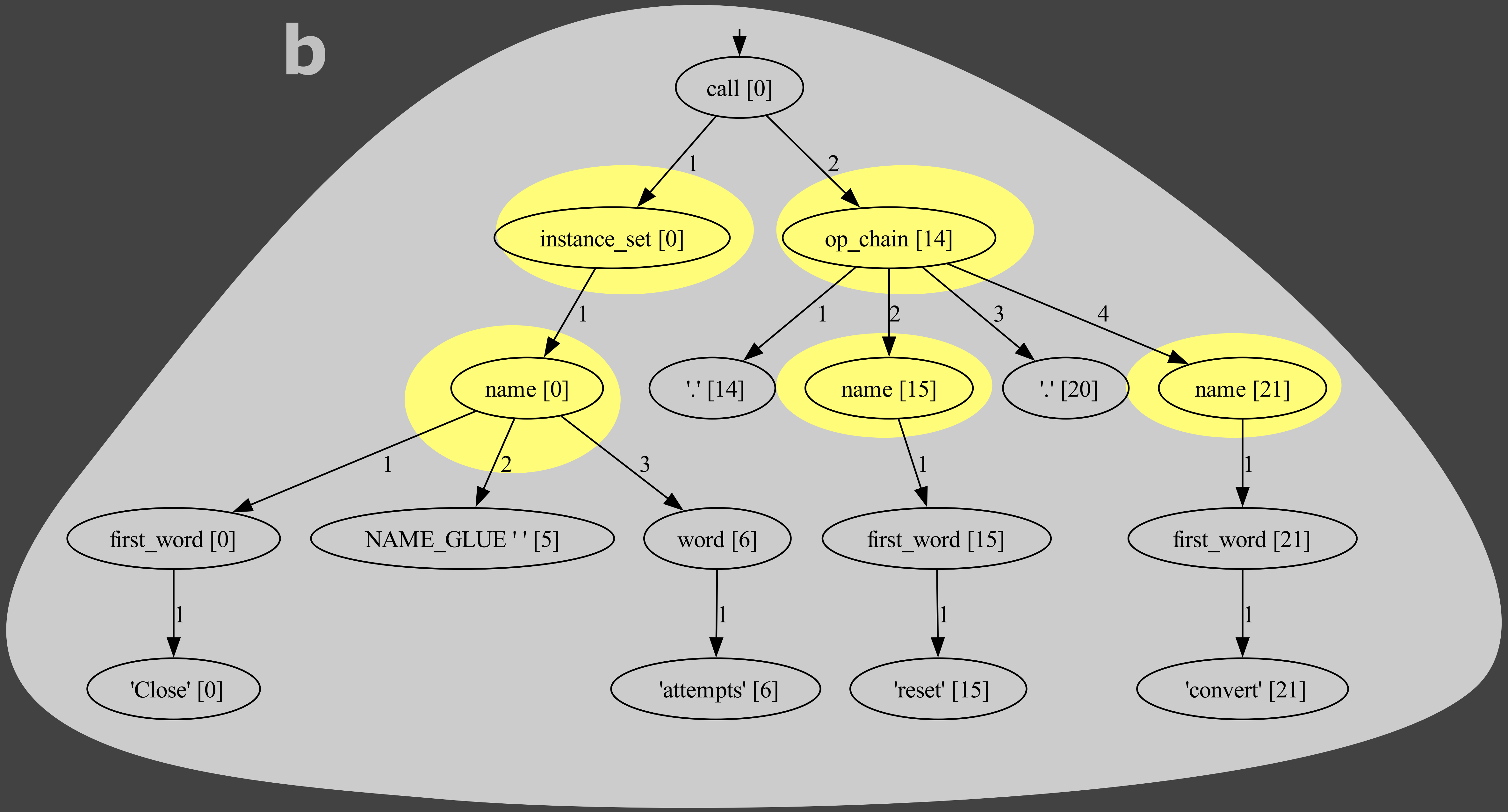
Close attempts.reset



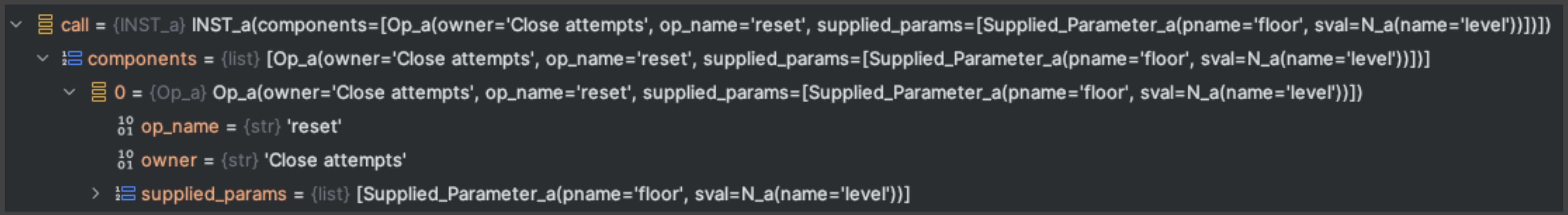
Close attempts.reset.convert



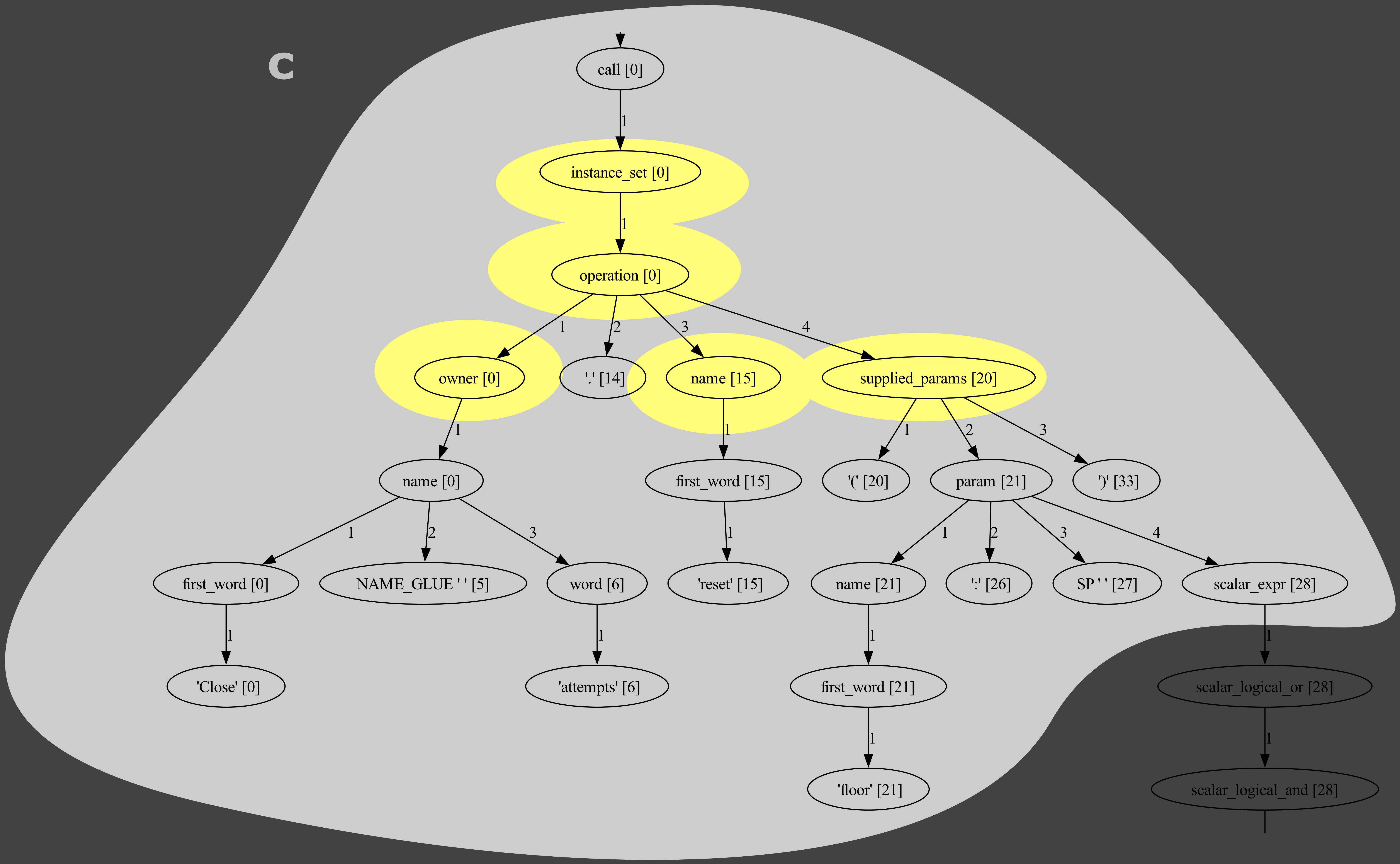
```
> call = {N_a} N_a(name='Close attempts')
> op_chain = {Op_chain_a} Op_chain_a(components=[N_a(name='reset'), N_a(name='convert')])
```



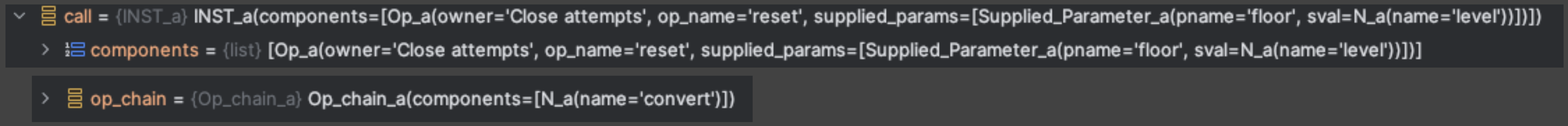
Close attempts.reset(floor: level)



```
⌄ call = {INST_a} INST_a(components=[Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])])
  ⌄ components = {list} [Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])]
    ⌄ 0 = {Op_a} Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])
        op_name = {str} 'reset'
        owner = {str} 'Close attempts'
      › supplied_params = {list} [Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))]
```

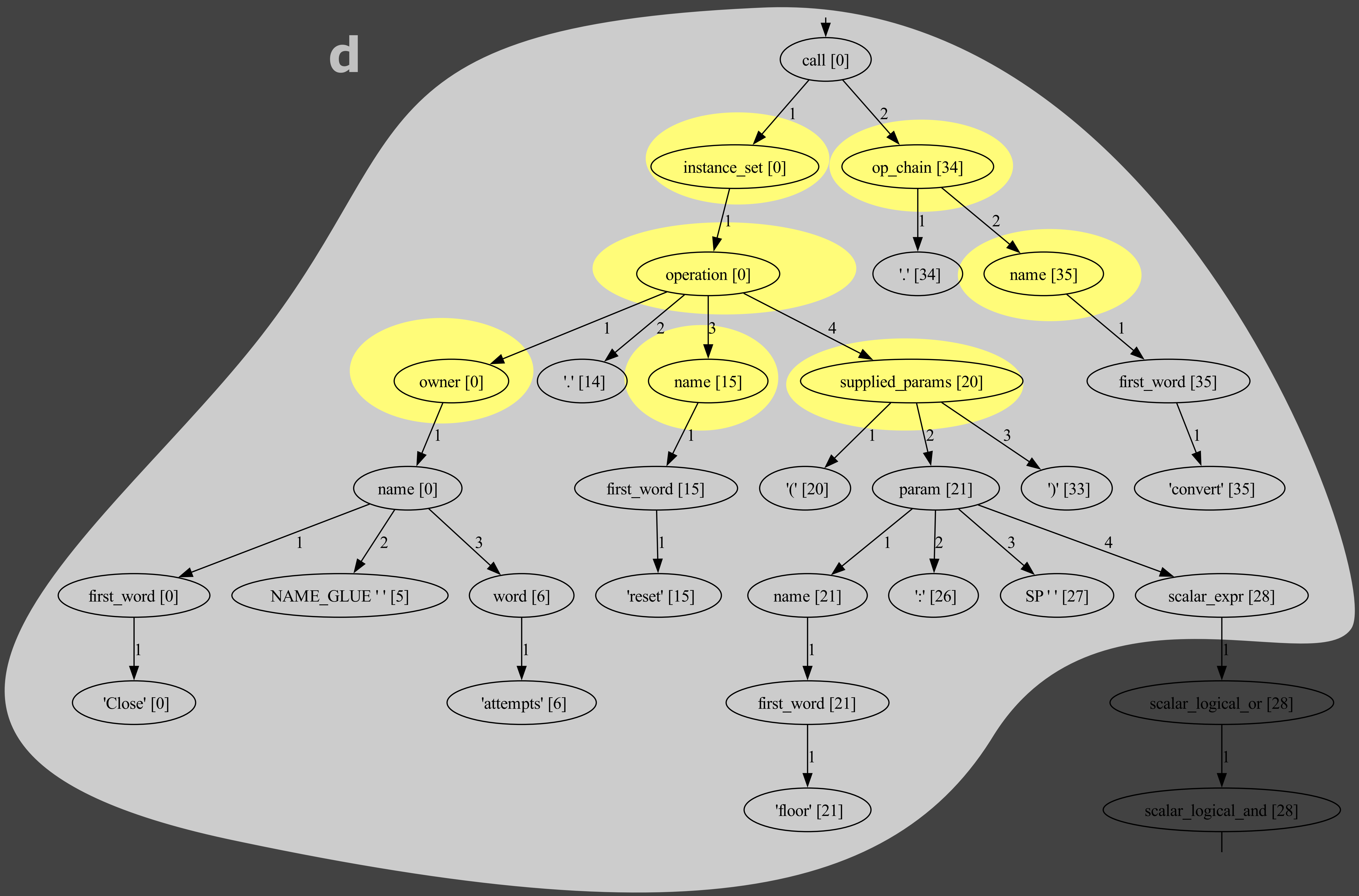


Close attempts.reset(floor: level).convert

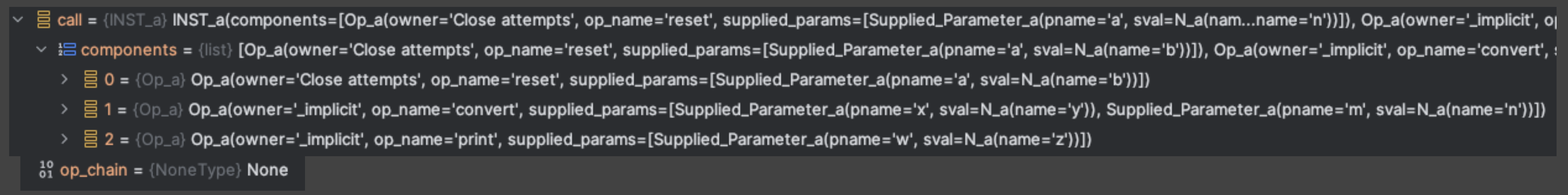


```
⌄ call = {INST_a} INST_a(components=[Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])])
  › components = {list} [Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])]

  › op_chain = {Op_chain_a} Op_chain_a(components=[N_a(name='convert')])
```



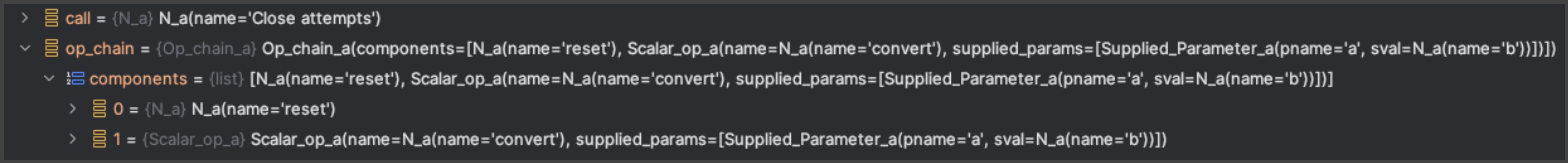
Close attempts.reset(a: b).convert(x: y, m: n).print(w: z)



```
˅ call = {INST_a} INST_a(components=[Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(nam...name='n'))]), Op_a(owner='_implicit', o
  ˅ components = {list} [Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='b'))]), Op_a(owner='_implicit', op_name='convert', 
    > 0 = {Op_a} Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='b'))])
    > 1 = {Op_a} Op_a(owner='_implicit', op_name='convert', supplied_params=[Supplied_Parameter_a(pname='x', sval=N_a(name='y')), Supplied_Parameter_a(pname='m', sval=N_a(name='n'))])
    > 2 = {Op_a} Op_a(owner='_implicit', op_name='print', supplied_params=[Supplied_Parameter_a(pname='w', sval=N_a(name='z'))])
  op_chain = {NoneType} None
```

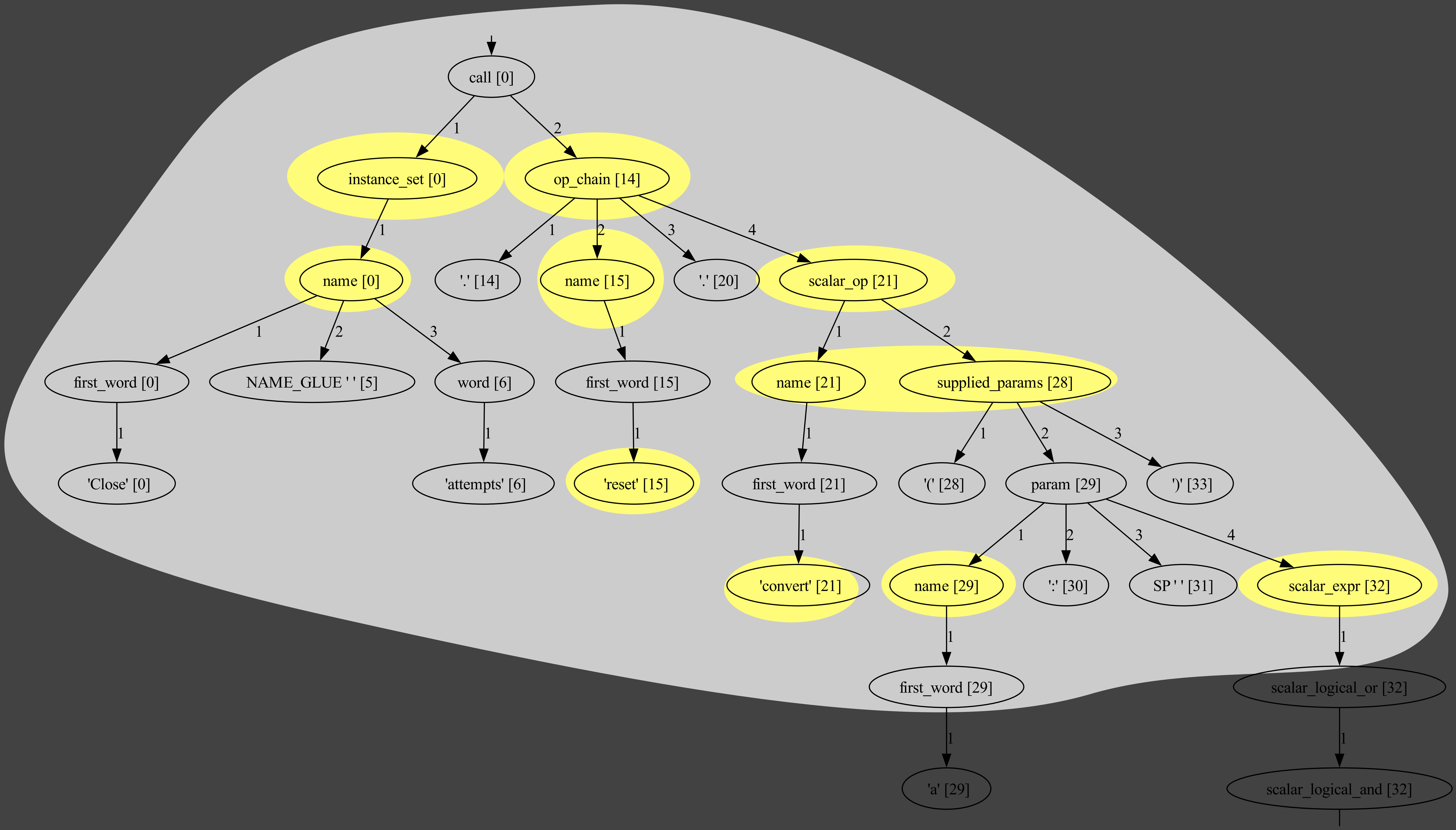
e
call [0]
instance_set [0]
operation [0]
operation [26]
operation [46]
owner [0]
'.' [14]
name [15]
supplied_params [20]
'.' [26]
name [27]
supplied_params [34]
'.' [46]
name [47]
supplied_params [52]
name [0]
first_word [15]
'(' [20]
param [21]
')' [25]
first_word [27]
'(' [34]
param [35]
',' [39]
SP ' ' [40]
param [41]
')' [45]
first_word [47]
'(' [52]
param [53]
')' [57]
first_word [0]
NAME_GLUE ' ' [5]
word [6]
'reset' [15]
name [21]
':' [22]
SP ' ' [23]
scalar_expr [24]
'convert' [27]
name [35]
':' [36]
SP ' ' [37]
scalar_expr [38]
name [41]
':' [42]
SP ' ' [43]
scalar_expr [44]
'print' [47]
name [53]
':' [54]
SP ' ' [55]
scalar_expr [56]
'Close' [0]
'attempts' [6]
first_word [21]
scalar_logical_or [24]
first_word [35]
scalar_logical_or [38]
first_word [41]
scalar_logical_or [44]
first_word [53]
scalar_logical_or [56]
'a' [21]
scalar_logical_and [24]
'x' [35]
scalar_logical_and [38]
'm' [41]
scalar_logical_and [44]
'w' [53]
scalar_logical_and [56]

Close attempts.reset.convert(a: b)



```
>  call = {N_a} N_a(name='Close attempts')
v  op_chain = {Op_chain_a} Op_chain_a(components=[N_a(name='reset'), Scalar_op_a(name=N_a(name='convert'), supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='b'))])])
   v  components = {list} [N_a(name='reset'), Scalar_op_a(name=N_a(name='convert'), supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='b'))])]
      >  0 = {N_a} N_a(name='reset')
      >  1 = {Scalar_op_a} Scalar_op_a(name=N_a(name='convert'), supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='b'))])
```

f


call [0]
instance_set [0]
op_chain [14]
name [0]
'.' [14]
name [15]
'.' [20]
scalar_op [21]
first_word [0]
NAME_GLUE ' ' [5]
word [6]
first_word [15]
name [21]
supplied_params [28]
'Close' [0]
'attempts' [6]
'reset' [15]
first_word [21]
'(' [28]
param [29]
')' [33]
'convert' [21]
name [29]
':' [30]
SP ' ' [31]
scalar_expr [32]
first_word [29]
scalar_logical_or [32]
'a' [29]
scalar_logical_and [32]