

Call parse analysis

Leon Starr
2025-10-1/ v0.5

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 xUML populate tool



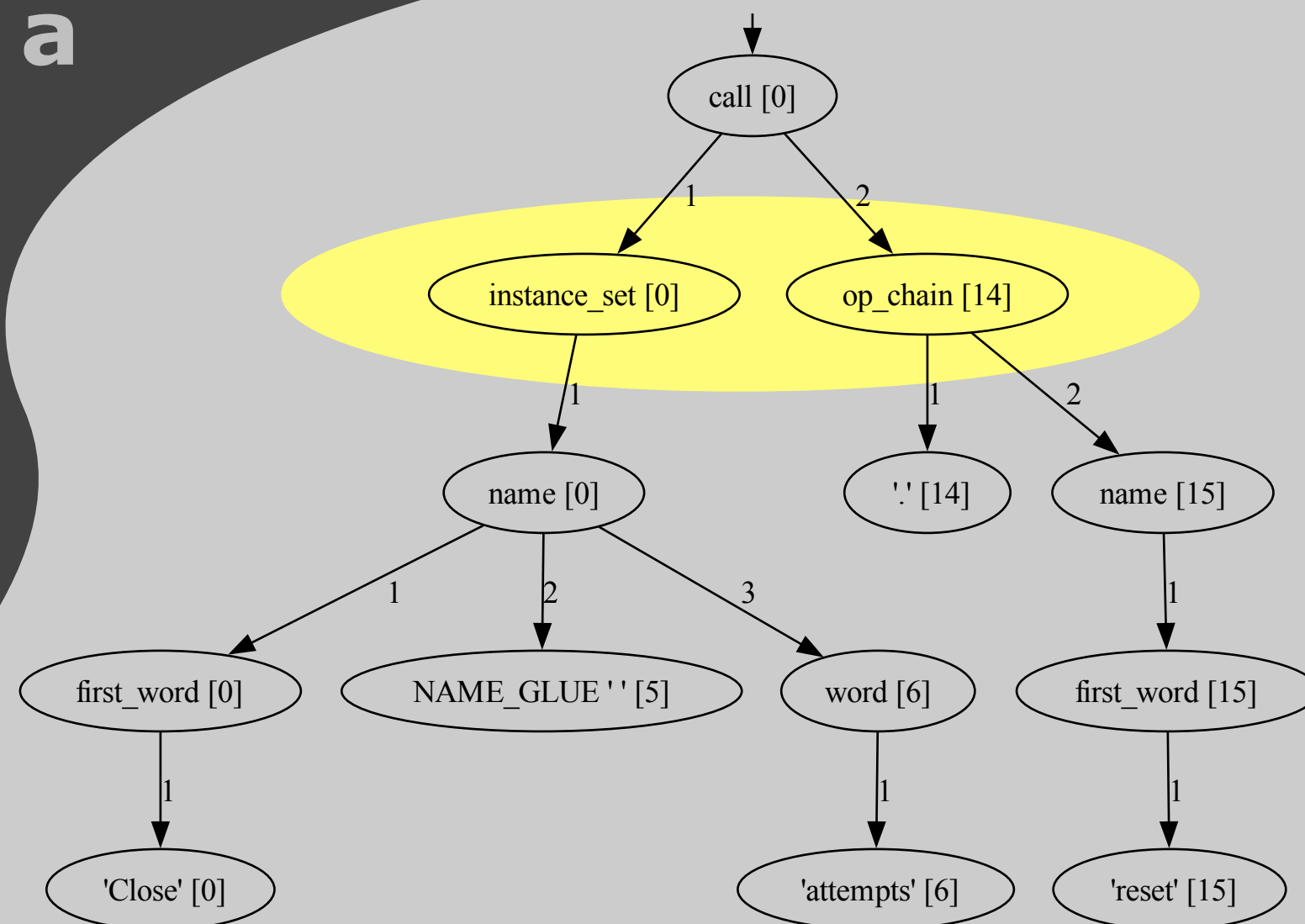
Copyright © 2025, Leon Starr at

MODEL INTEGRATION, LLC

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

Close attempts.reset

a

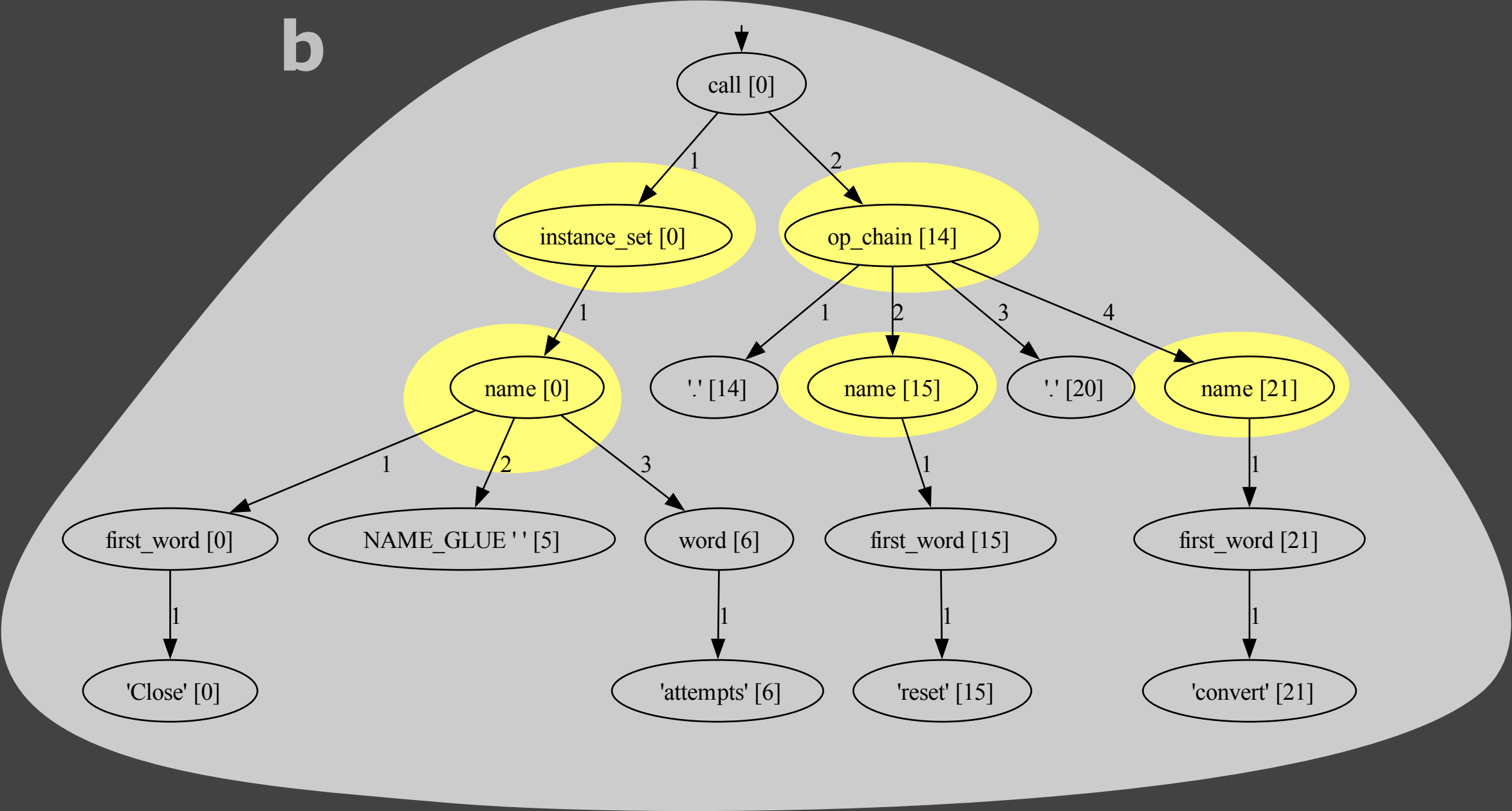


This pattern could be an attribute.type op OR
iset.attribute OR iset.method() OR iset.service()

Not the () indicating that no parameters are specified.
The parse pattern changes to a different form if one
or more parameters are specified.

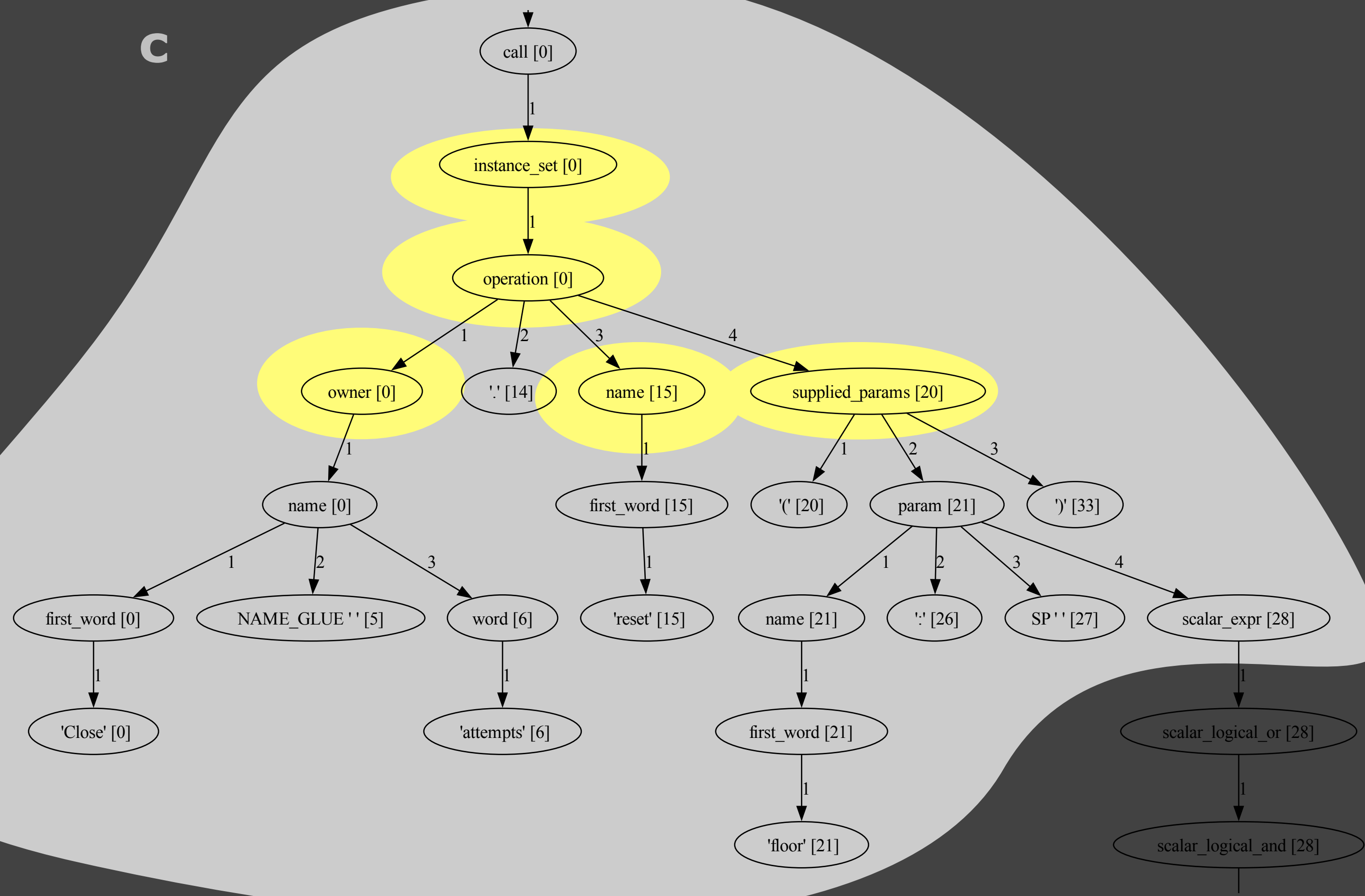
Close attempts.reset.convert

```
> 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)

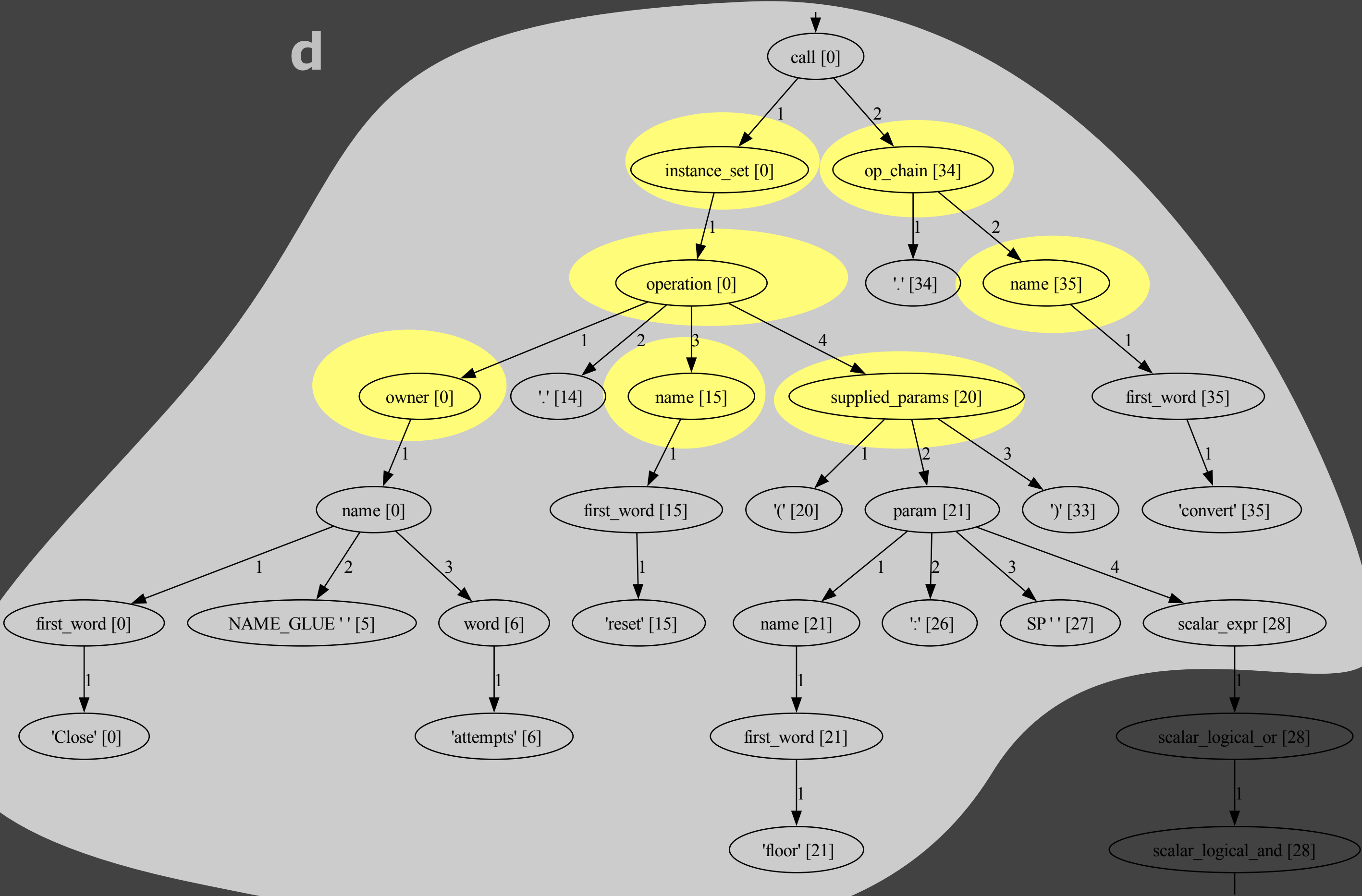
```
call = {INST_a} INST_a(components=[Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])])
components = {list} [Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])]
0 = {Op_a} Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='floor', sval=N_a(name='level'))])
  10 op_name = {str} 'reset'
  01 owner = {str} 'Close attempts'
  > 10 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')])
```

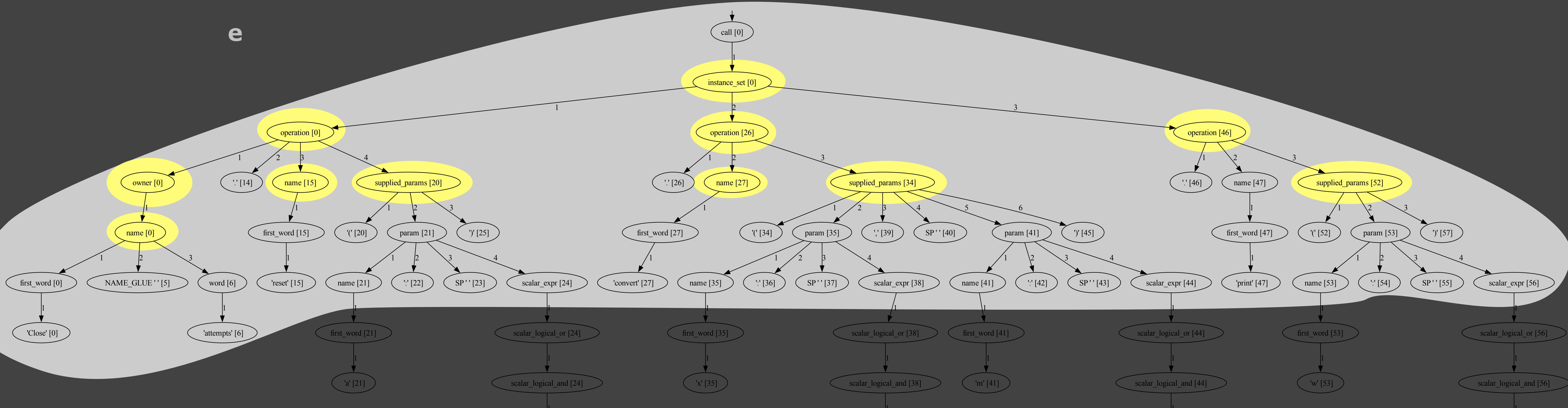
d



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(name='n'))]), 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'))]), Op_a(owner='_implicit', op_name='print', supplied_params=[Supplied_Parameter_a(pname='w', sval=N_a(name='z'))])])
components = (list) [Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='n'))]), 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'))]), Op_a(owner='_implicit', op_name='print', supplied_params=[Supplied_Parameter_a(pname='w', sval=N_a(name='z'))])])
0 = {Op_a} Op_a(owner='Close attempts', op_name='reset', supplied_params=[Supplied_Parameter_a(pname='a', sval=N_a(name='n'))])
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
```

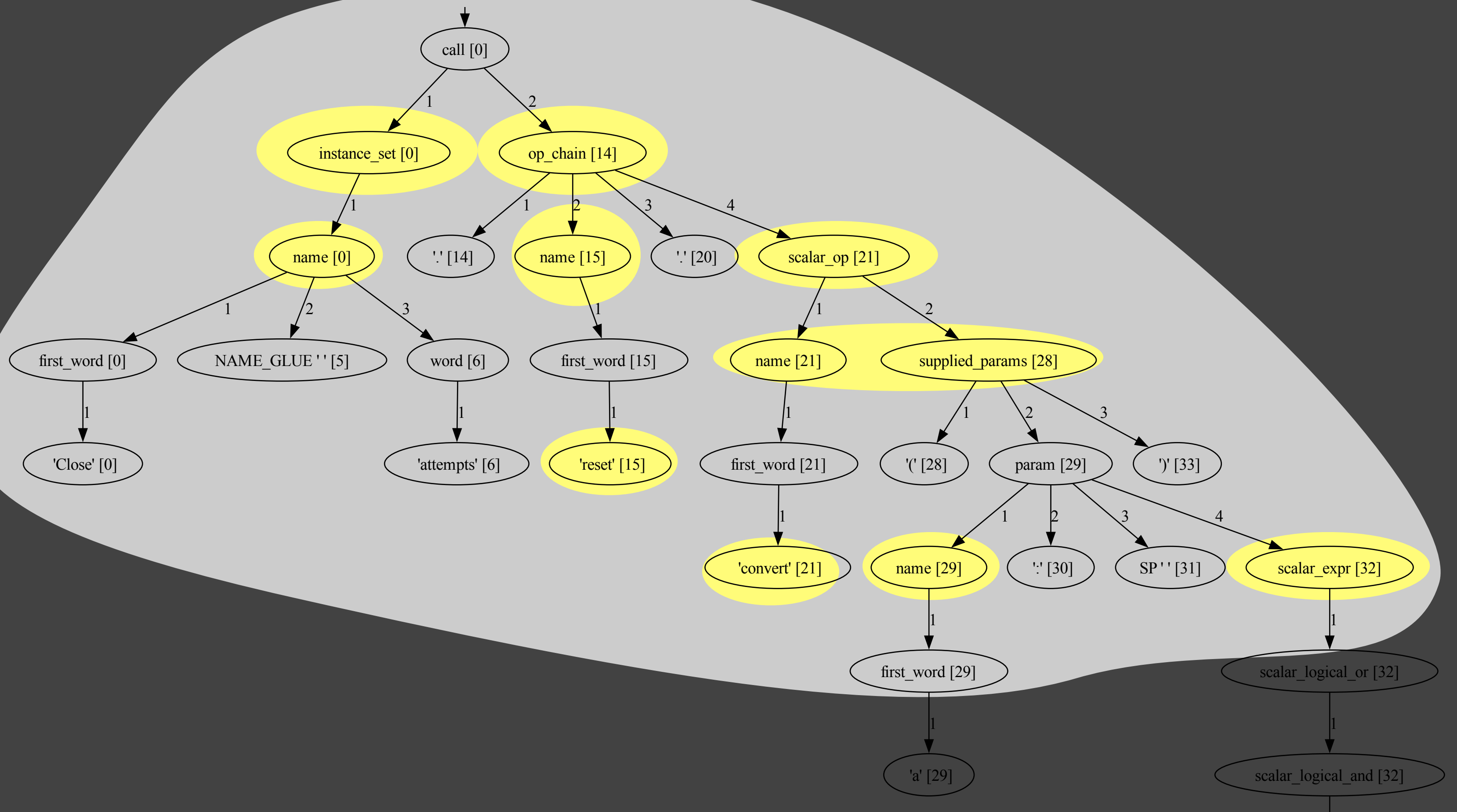
e



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



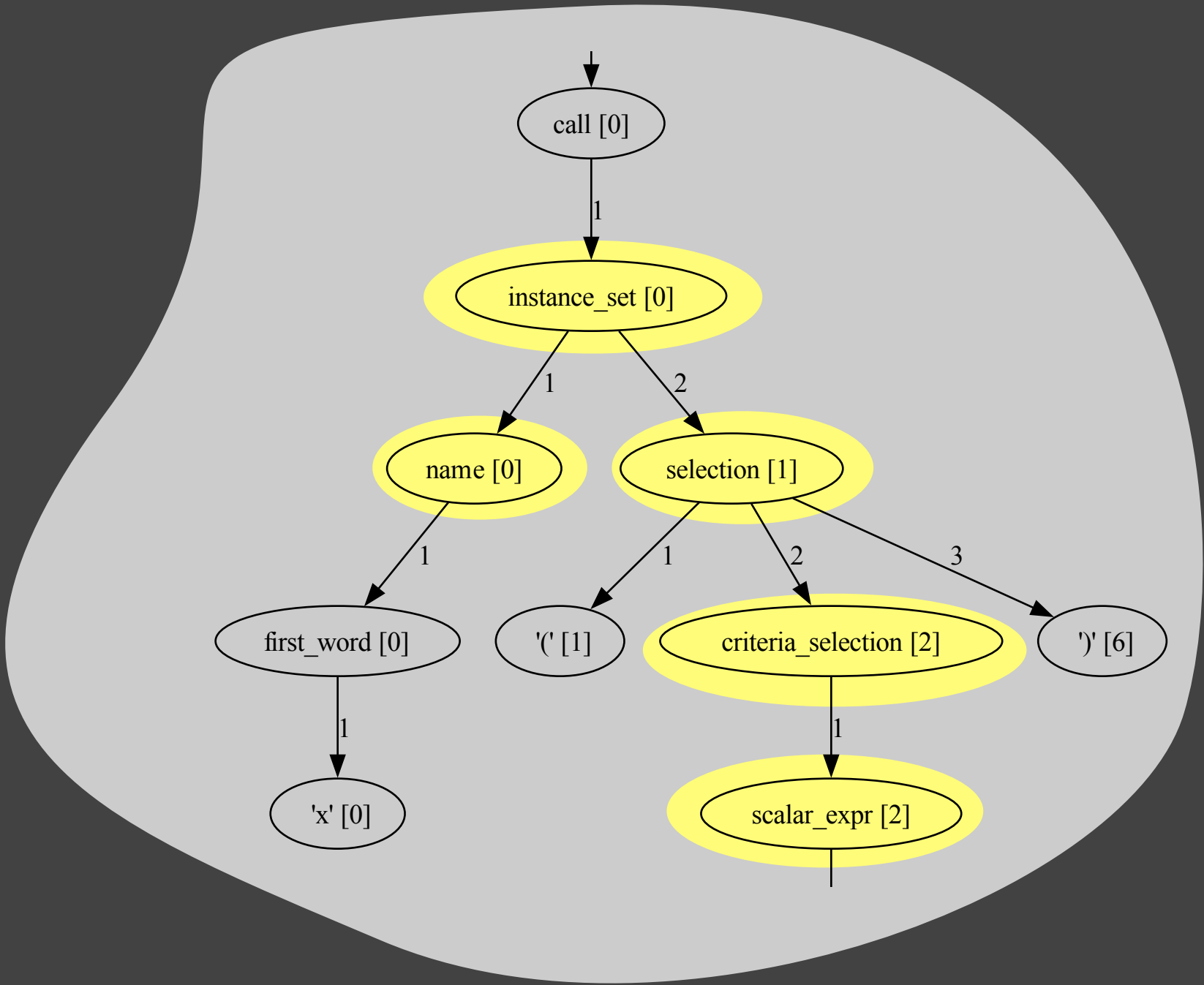
x(a: b)

```

  call = {INST_a} INST_a(components=[N_a(name='x'), Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')]))])
  components = {list} [N_a(name='x'), Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')]))]
  0 = {list} [N_a(name='x'), Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')]))]
  __len__ = {int} 1
  Protected Attributes
  op_chain = {NoneType} None

```

g

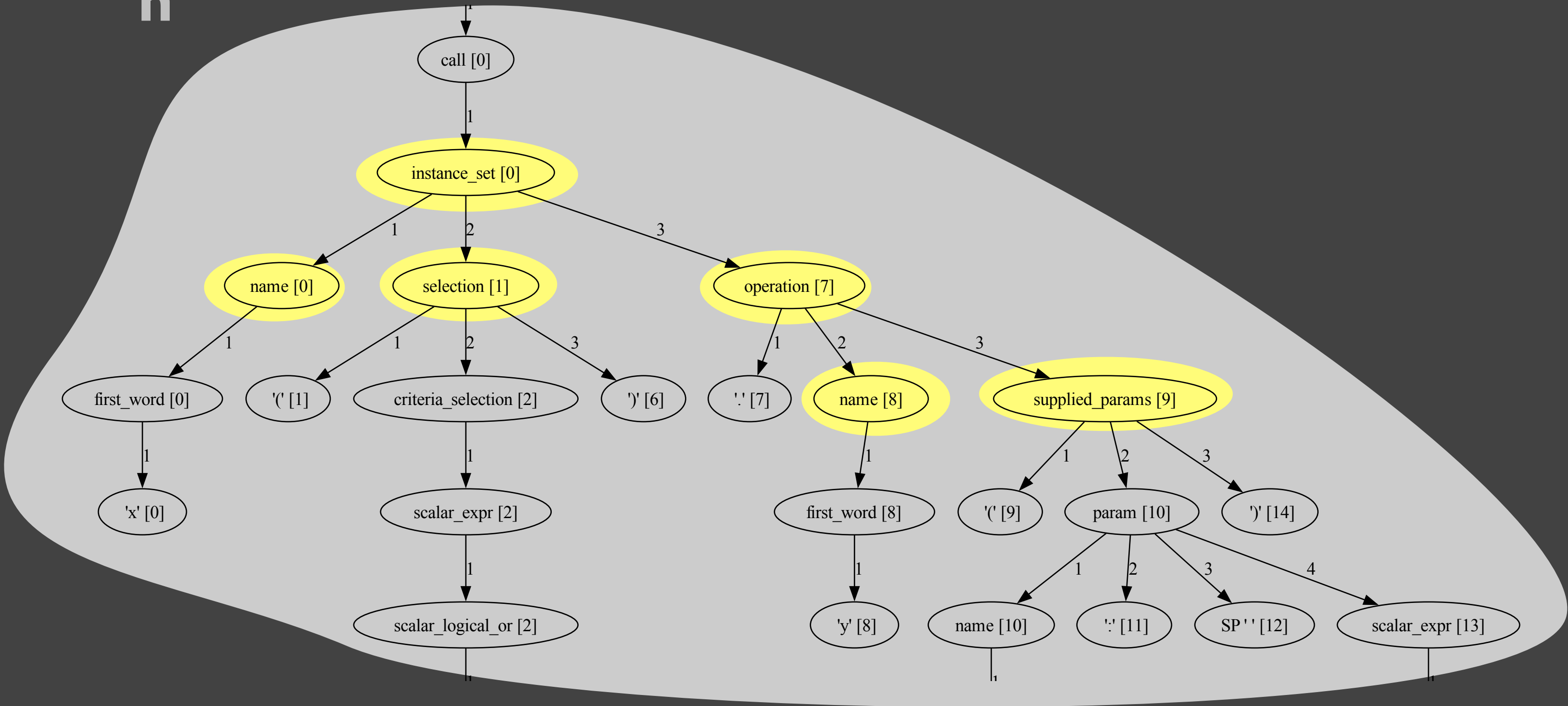


x(a: b).y(c: d)



```
call = {INST_a} INST_a(components=[N_a(name='x'), Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')])]), op_name='y', supplied_params=[Supplied_Parameter_a(pname='c', sval=N_a(name='d'))])
components = (list) [N_a(name='x'), Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')])]), Op_a(owner='_implicit', op_name='y', supplied_params=[Supplied_Parameter_a(pname='c', sval=N_a(name='d'))])
0 = {N_a} N_a(name='x')
1 = {Criteria_Selection_a} Criteria_Selection_a(card='ALL', criteria=BOOL_a(op='==', operands=[N_a(name='a'), N_a(name='b')])])
2 = {Op_a} Op_a(owner='_implicit', op_name='y', supplied_params=[Supplied_Parameter_a(pname='c', sval=N_a(name='d'))])
```

```
op_chain = {NoneType} None
```

h



/R4/Cabin.Ping

```
>  call = {INST_a} INST_a(components=[PATH_a(hops=[R_a(rnum='R4'), N_a(name='Cabin')])])  
>  op_chain = {Op_chain_a} Op_chain_a(components=[N_a(name='Ping')])
```

i

