

Call parse analysis

Leon Starr
2025-9-26/ v0.1

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

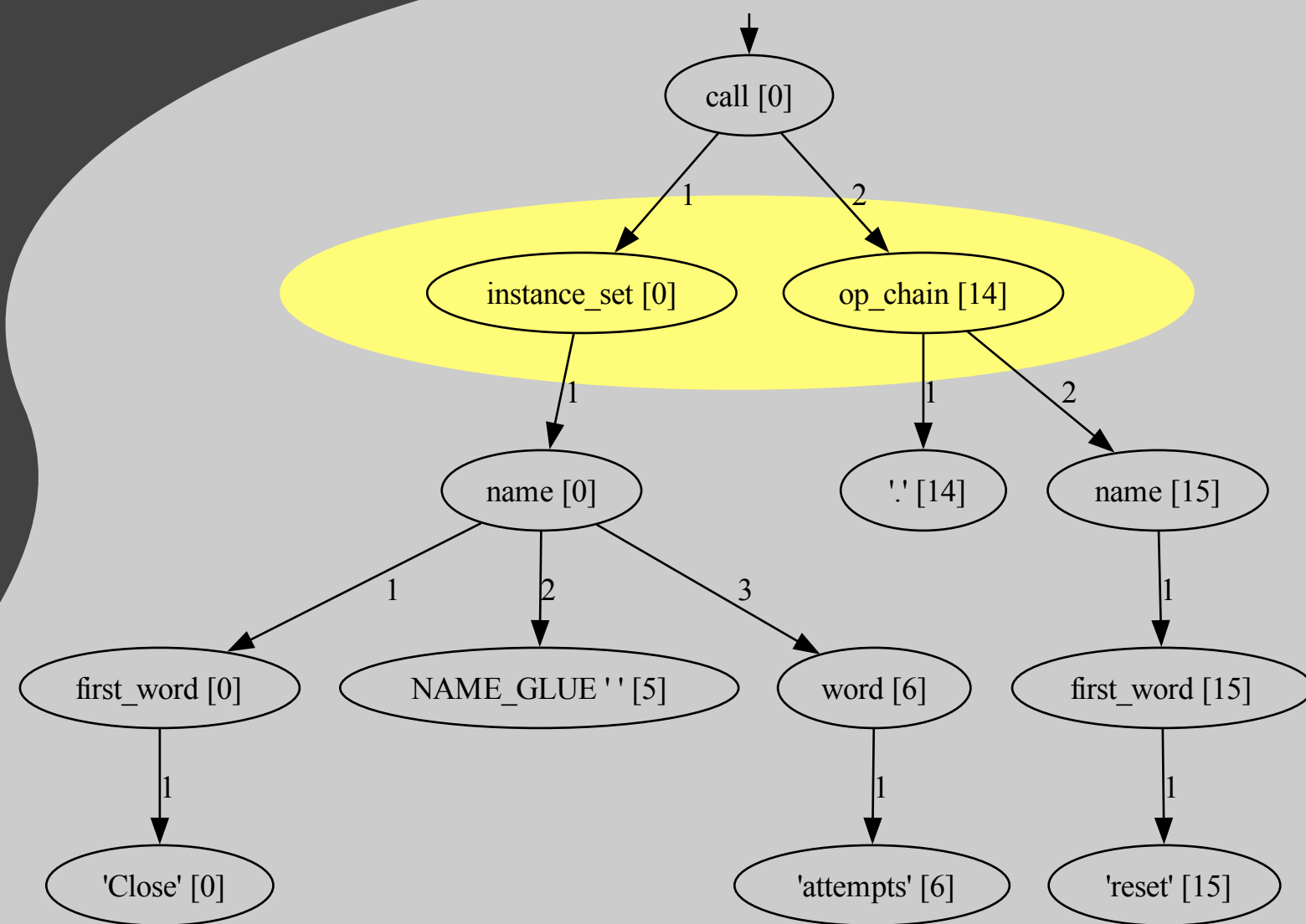


Copyright © 2025, Leon Starr at

MODEL INTEGRATION, LLC

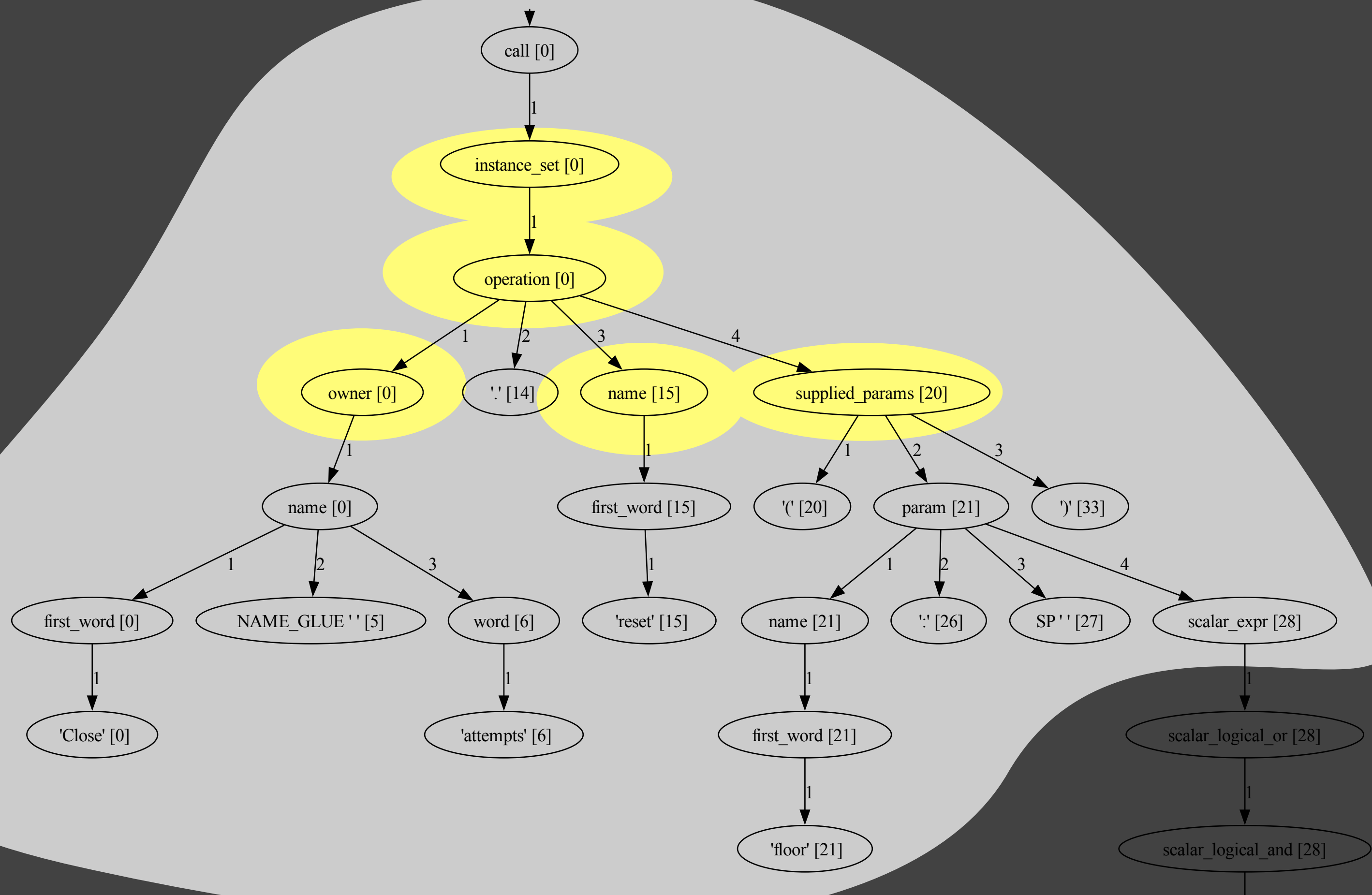
```
> statement = {Call_a} Call_a(call=N_a(name='Close attempts'), op_chain=Op_chain_a(components=[N_a(name='reset')]))
```

Close attempts.reset



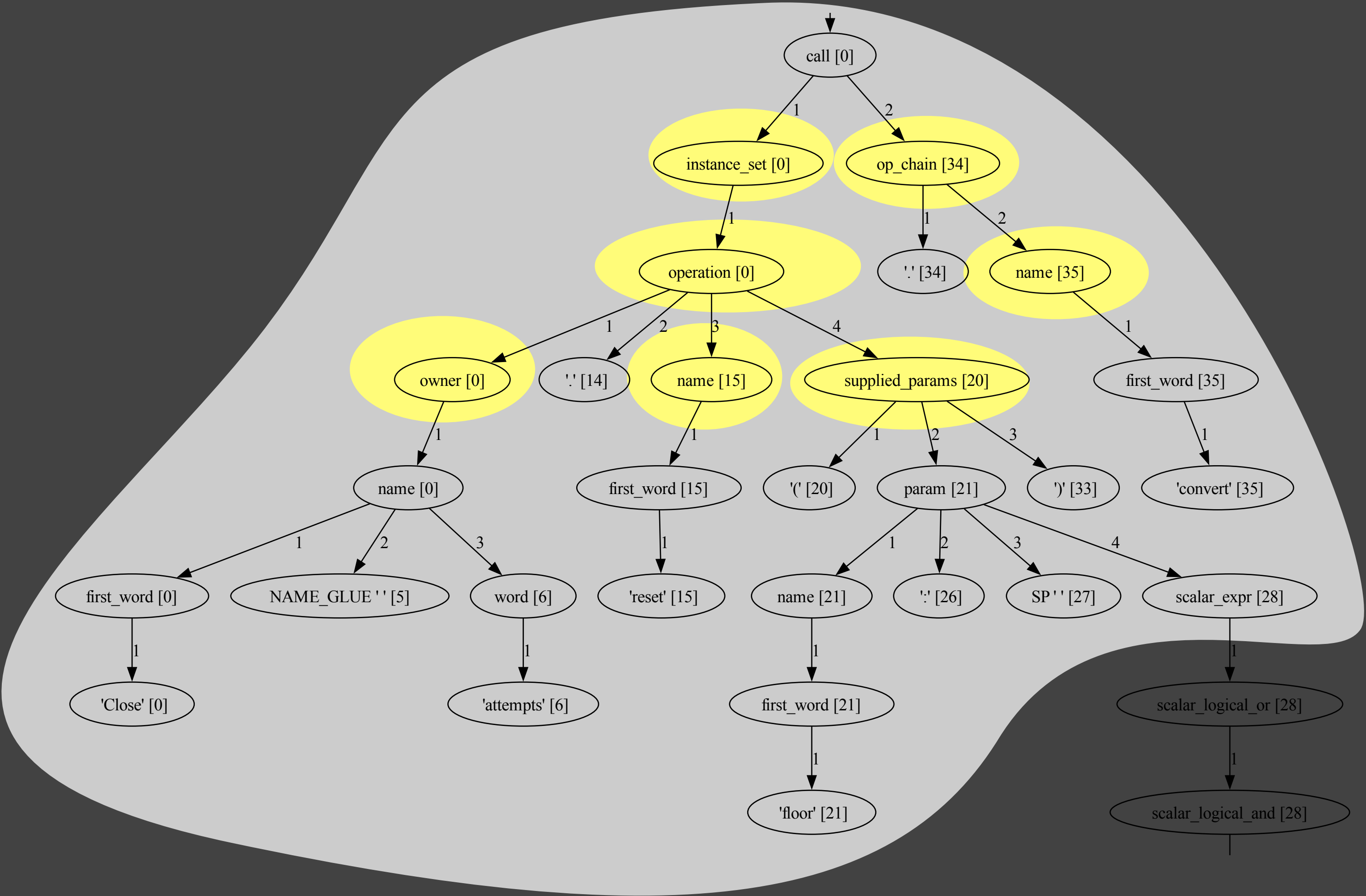
Close attempts.reset(floor: level)

```
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'))]
```



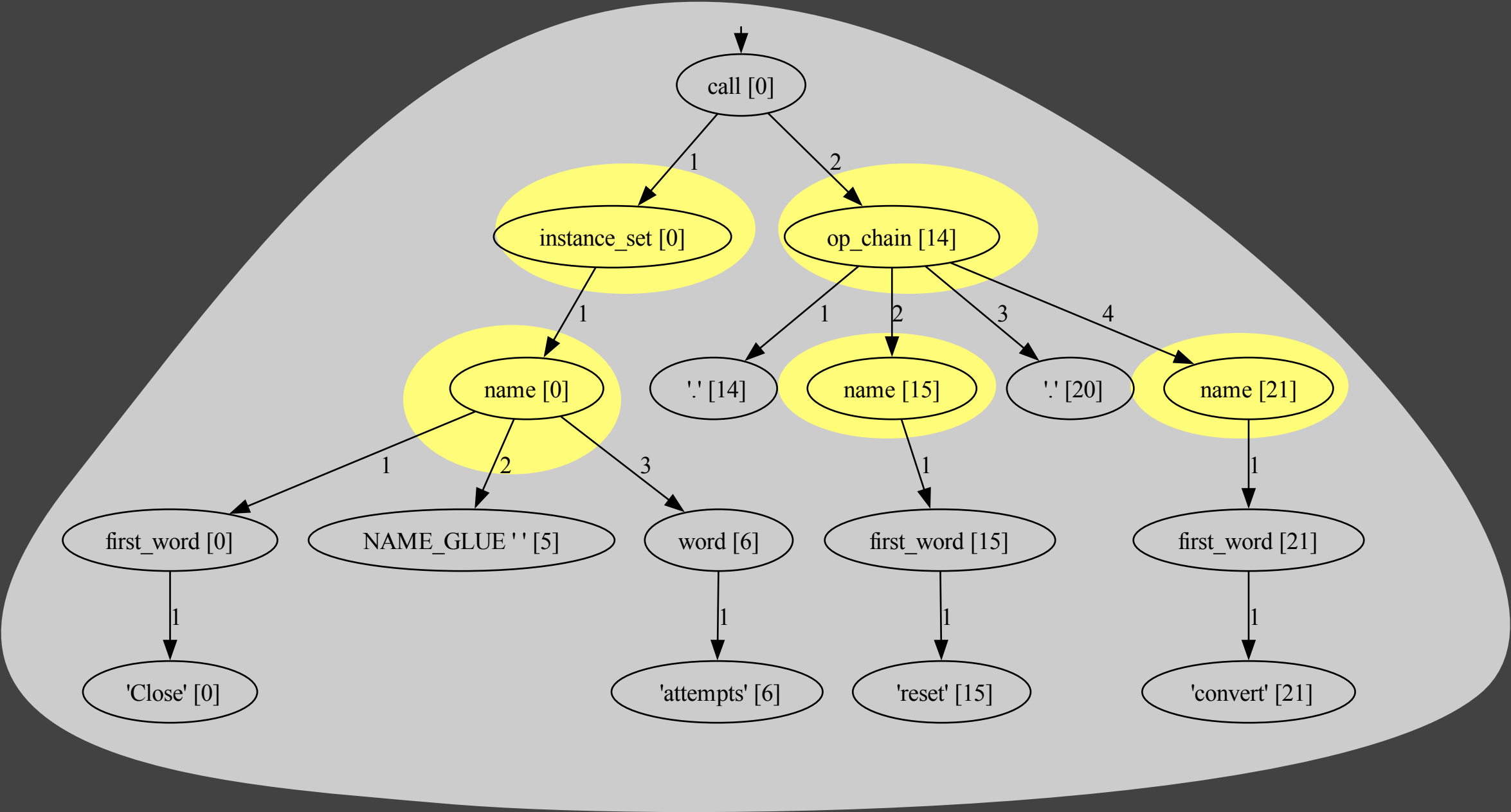
Close attempts.reset(floor: level).convert

```
> 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')])
```



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(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
```

