REG with MRS Composition

Elizabeth (Liz) Conrad — DELPH-IN 2024

Overview

- REG as a task
- POGG: Precision-Oriented Graphical Generator
    - Architecture
    - MRS Algebra Implementation
    - Evaluation Metrics
    - Results / Error Analysis
- Discussion
    - Questions (from me to you)
    - Questions (from you to me)

Referring Expression Generation

Referring Expression Generation

- Task: Given some description of an entity or entities, generate an (English) referring expression for that entity
- Can often be divided into two subtasks:
    - Content selection
    - Surface string generation
        - This is what my project addresses

Content Selection via Graphical Comparison (Krahmer et al. 2003)

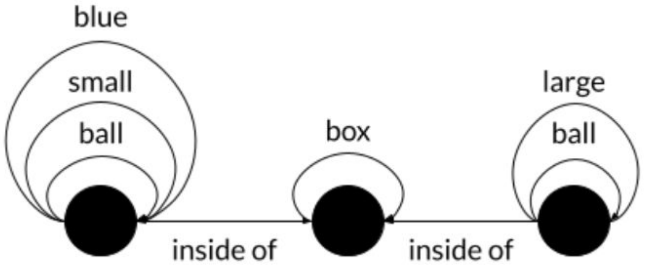


Figure 2.1: Entity graph representation in the style of Krahmer et al. (2003)

- Nodes represent entities
- Self-pointing edges represent properties of the entity
- Edges between nodes represent relationships between entities
- Suitable for ensuring distinguishing properties are selected in content selection

Content Selection via Graphical Comparison (Krahmer et al. 2003)

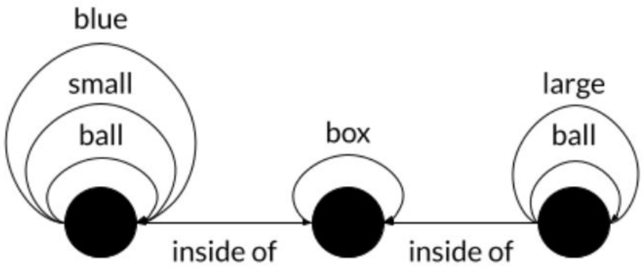
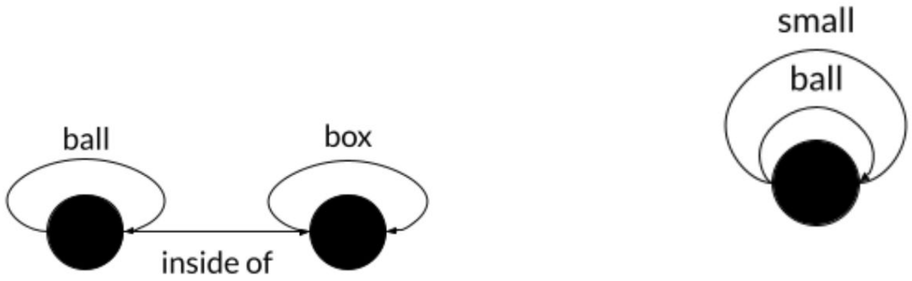


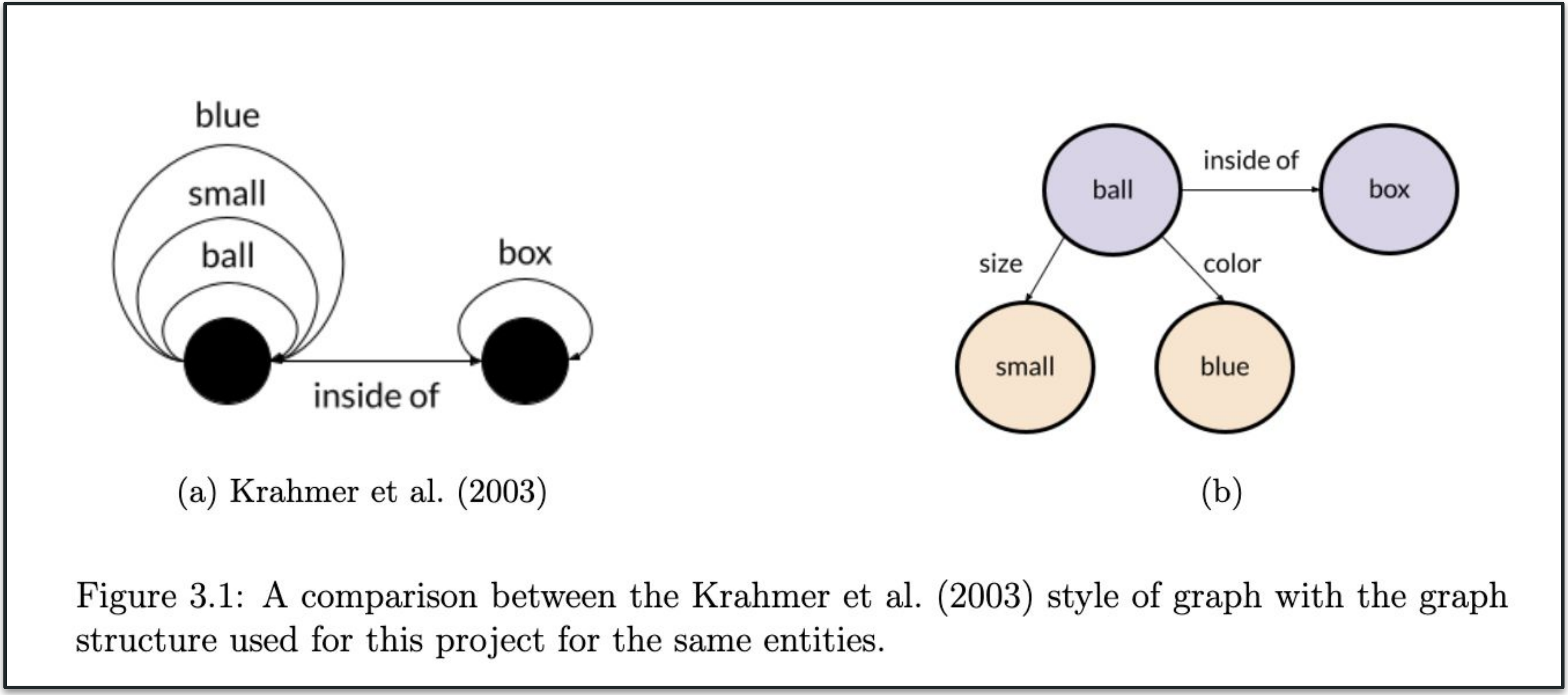
Figure 2.1: Entity graph representation in the style of Krahmer et al. (2003)



(a) A non-distinguishing subgraph for the full graph in Figure 2.1

(b) A distinguishing subgraph for the full graph in Figure 2.1

Modified graphs for surface string generation



(a) Krahmer et al. (2003)

(b)

Figure 3.1: A comparison between the Krahmer et al. (2003) style of graph with the graph structure used for this project for the same entities.

POGG

POGG: Precision-Oriented Graphical Generator

- POGG-internal
  - Graph-to-MRS Algorithm
  - Composition Library
  - MRS Algebra Implementation
- POGG-external
  - Lexicon

All entity data I used for my project came from Eric Zinda :)

Graph-to-MRS Recursive Algorithm

- Recurse from the root to leaf nodes and produce basic MRS fragments along the way for each entity/property node
- By “basic” I mean that, in most cases, the MRS that represents a node contains only one EP
- When returning back up the call stack, combine the fragments to create larger fragments based on what type of composition the edge calls for

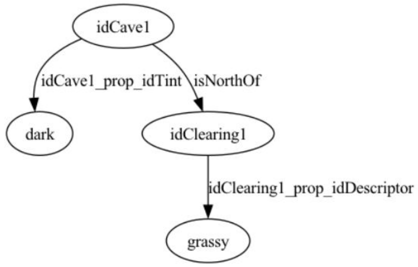
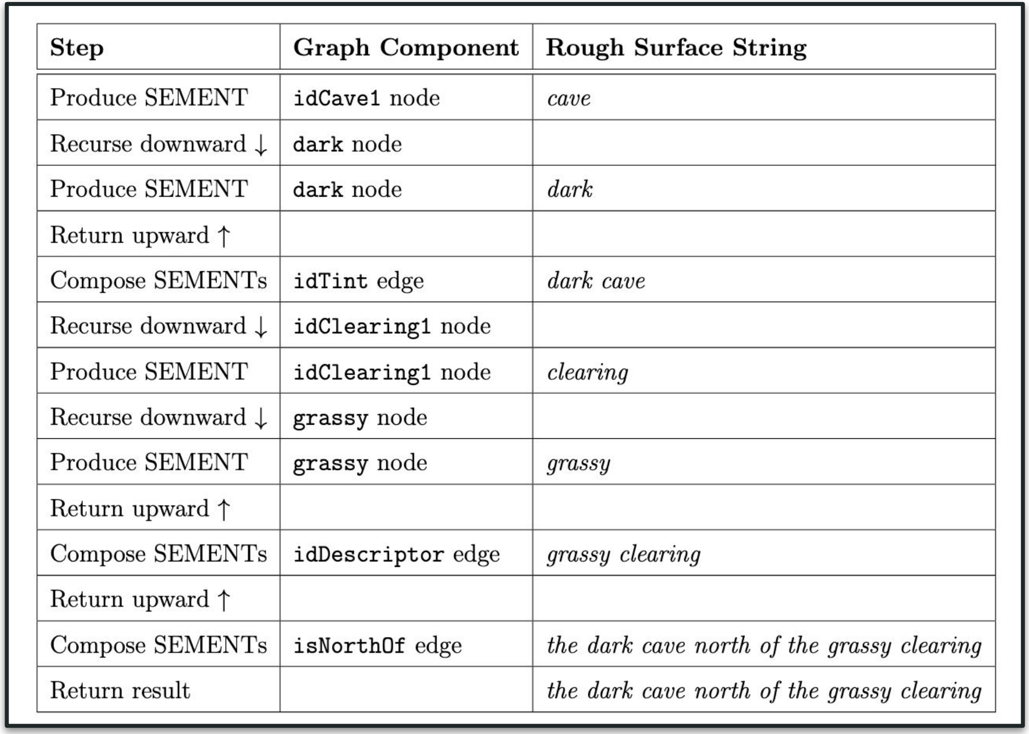
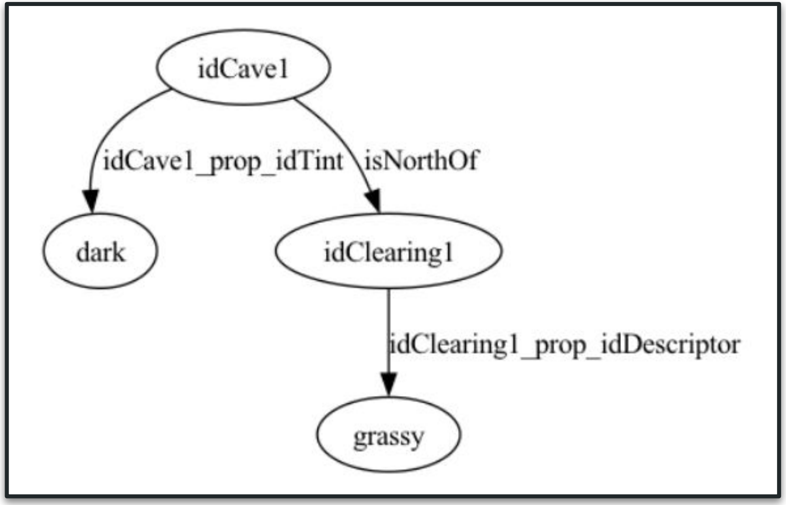


Figure 4.3: Example of an entity graph, should generate strings roughly like *the dark cave north of the grassy clearing*

Graph-to-MRS Recursive Algorithm



Step	**Graph Component**	**Rough Surface String**
Produce SEMENT	`idCave1` node	*cave*
Recurse downward ↓	`dark` node	
Produce SEMENT	`dark` node	*dark*
Return upward ↑		
Compose SEMENTs	`idTint` edge	*dark cave*
Recurse downward ↓	`idClearing1` node	
Produce SEMENT	`idClearing1` node	*clearing*
Recurse downward ↓	`grassy` node	
Produce SEMENT	`grassy` node	*grassy*
Return upward ↑		
Compose SEMENTs	`idDescriptor` edge	*grassy clearing*
Return upward ↑		
Compose SEMENTs	`isNorthOf` edge	*the dark cave north of the grassy clearing*
Return result		*the dark cave north of the grassy clearing*



POGG: Precision-Oriented Graphical Generator

- POGG-internal
  - Graph-to-MRS Algorithm ✔
  - Composition Library
  - MRS Algebra Implementation
- POGG-external
  - Lexicon

All entity data I used for my project came from Eric Zinda :)

Composition Library

Composition Library

- The library of composition functions that a user of POGG would use when filling out their lexicon
- Functions are largely named after syntactic constructions as that feels more intuitive, but technically under the hood they are only performing semantic composition
- Some functions take as input some predicate label and return an MRS fragment for that predicate, such as `adjective_sement`, which could produce an MRS like *red*
- Some functions are designed to take two MRS fragments and compose a larger fragment, such as adjective, which could produce an MRS like *red ball*

Current functions in the composition library

Functions that (for the most part) take a predicate label as input and return an MRS fragment

1. basic
2. noun_sement
3. adjective_sement
4. verb_sement
5. preposition_sement
6. pronoun_sement
7. quant_sement
8. boolean_adjective_sement
9. boolean_pass_part_sement

Functions that take two MRS fragments as input and return a larger MRS fragment

1. adjective (*red ball*)
2. compound (*cake box)*
3. passive_participle (*broken bottle*)
4. possessive (*student's homework*)
5. prefix (*un-locked*)
6. quantify (*the ball*)
7. relative_direction (*north of here*)
8. descriptor
9. boolean

Boolean???

- Some properties are boolean, such as idImmovable in this example
- Unlike *green*, which can be specified to one specific ERG label, when idImmovable appears in a graph it has two possibilities for surface realization depending on the child node
- So the boolean functions are meant to handle these cases

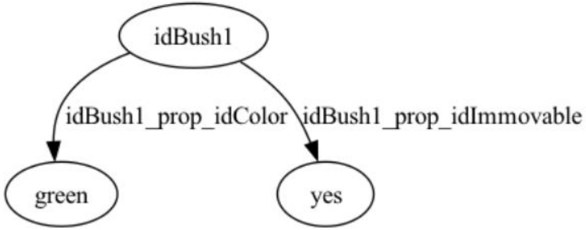


Figure 4.2: Example of a graph with a boolean property

POGG: Precision-Oriented Graphical Generator

- POGG-internal
  - Graph-to-MRS Algorithm ✔
  - Composition Library ✔
  - MRS Algebra Implementation
- POGG-external
  - Lexicon

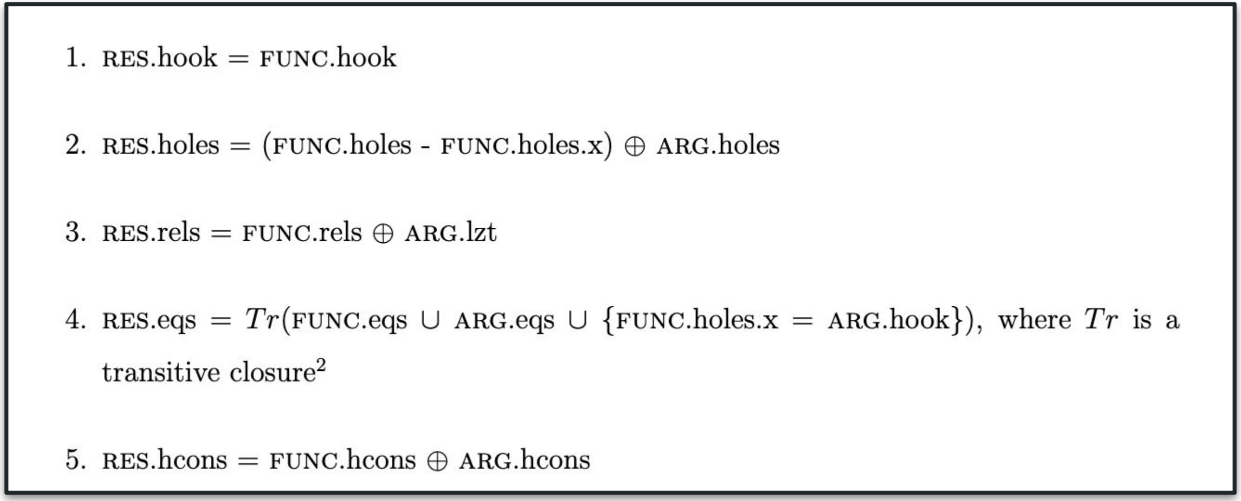
All entity data I used for my project came from Eric Zinda :)

MRS Algebra Implementation

MRS Algebra Implementation

- SEMENT class which is an extension of the MRS class from PyDelphin
  - formally, an MRS does not include a list of holes or list of eqs which I needed to implement the algebra, so I made a subclass of MRS to add these elements
- Five primary functions:
  - create_base_sement
  - op_nonscopal_label_shared
  - op_nonscopal_label_unshared
  - op_scopal
  - op_final

op_nonscopal_

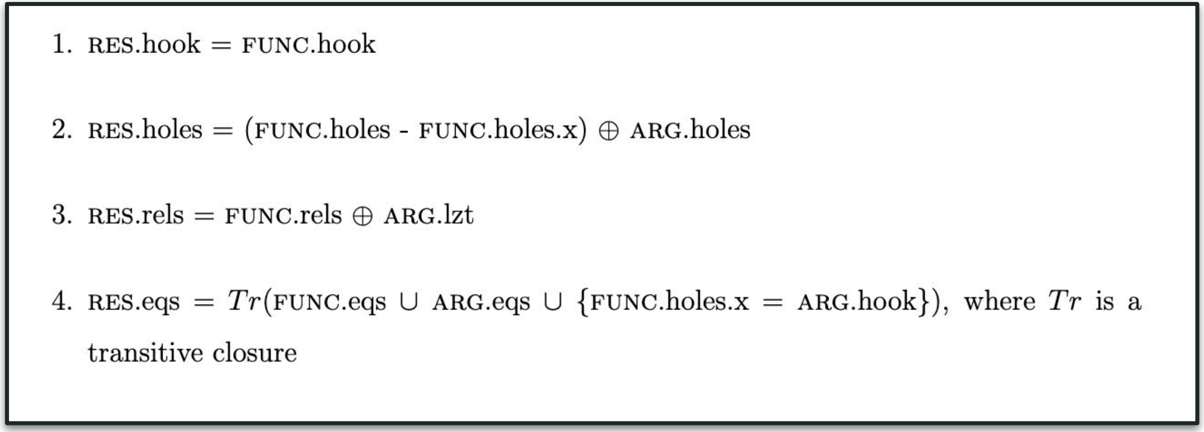


1. RES.hook = FUNC.hook
2. RES.holes = (FUNC.holes - FUNC.holes.x) $\oplus$ ARG.holes
3. RES.rels = FUNC.rels $\oplus$ ARG.lzt
4. RES.eqs = Tr(FUNC.eqs $\cup$ ARG.eqs $\cup$ {FUNC.holes.x = ARG.hook}), where Tr is a transitive closure[2]
5. RES.hcons = FUNC.hcons $\oplus$ ARG.hcons

- Both nonscopal functions do this, but the label_shared version ensures that the LTOP of each fragment is shared
- Question: does the algebra paper discuss the difference in these two cases?

Definition from Copestake et al. 2001

op_scopal

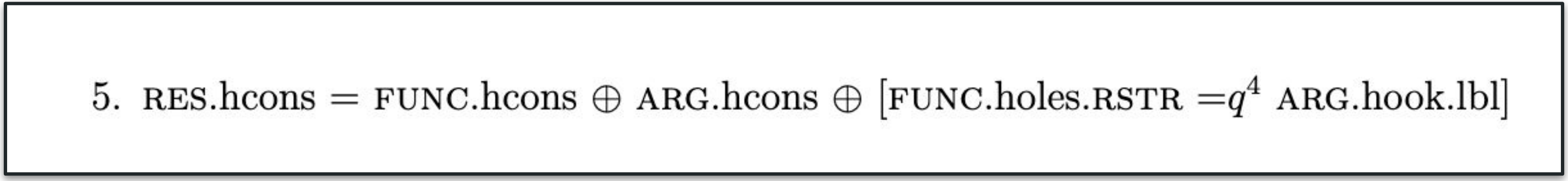


1. RES.hook $=$ FUNC.hook

2. RES.holes $=$ (FUNC.holes - FUNC.holes.x) $\oplus$ ARG.holes

3. RES.rels $=$ FUNC.rels $\oplus$ ARG.lzt

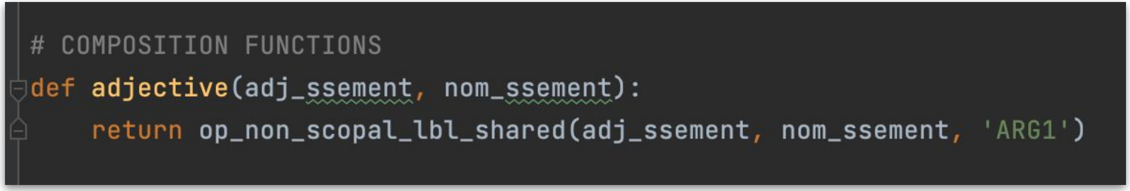
4. RES.eqs $=$ Tr(FUNC.eqs $\cup$ ARG.eqs $\cup$ $\{$FUNC.holes.x $=$ ARG.hook$\}$), where Tr is a transitive closure



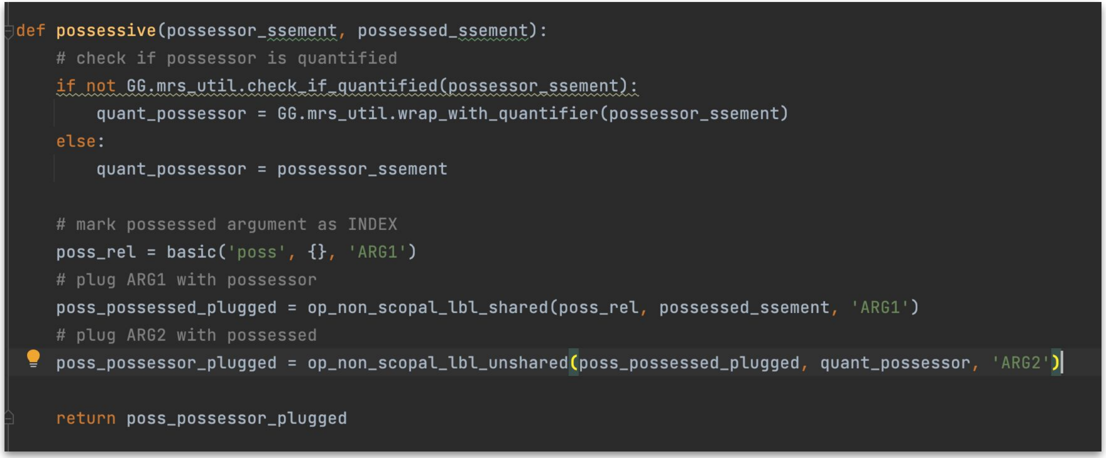
5. RES.hcons $=$ FUNC.hcons $\oplus$ ARG.hcons $\oplus$ [FUNC.holes.RSTR $=q$[4] ARG.hook.lbl]

Definition from Copestake et al. 2001

Revisiting the Composition Library



```python
# COMPOSITION FUNCTIONS
def adjective(adj_ssement, nom_ssement):
    return op_non_scopal_lbl_shared(adj_ssement, nom_ssement, 'ARG1')
```



```python
def possessive(possessor_ssement, possessed_ssement):
    # check if possessor is quantified
    if not GG.mrs_util.check_if_quantified(possessor_ssement):
        quant_possessor = GG.mrs_util.wrap_with_quantifier(possessor_ssement)
    else:
        quant_possessor = possessor_ssement

    # mark possessed argument as INDEX
    poss_rel = basic('poss', {}, 'ARG1')
    # plug ARG1 with possessor
    poss_possessed_plugged = op_non_scopal_lbl_shared(poss_rel, possessed_ssement, 'ARG1')
    # plug ARG2 with possessed
    poss_possessor_plugged = op_non_scopal_lbl_unshared(poss_possessed_plugged, quant_possessor, 'ARG2')

    return poss_possessor_plugged
```

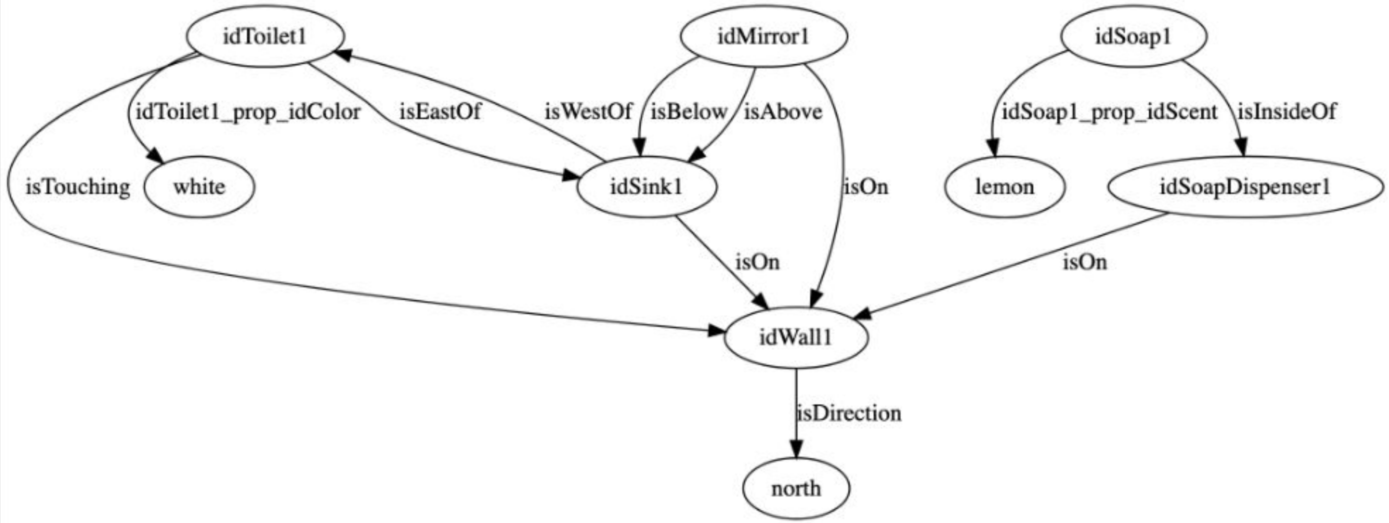
- Specifics of MRS composition are contained in the algebra implementation
- Composition library functions make use of the basic composition functions from the MRS algebra

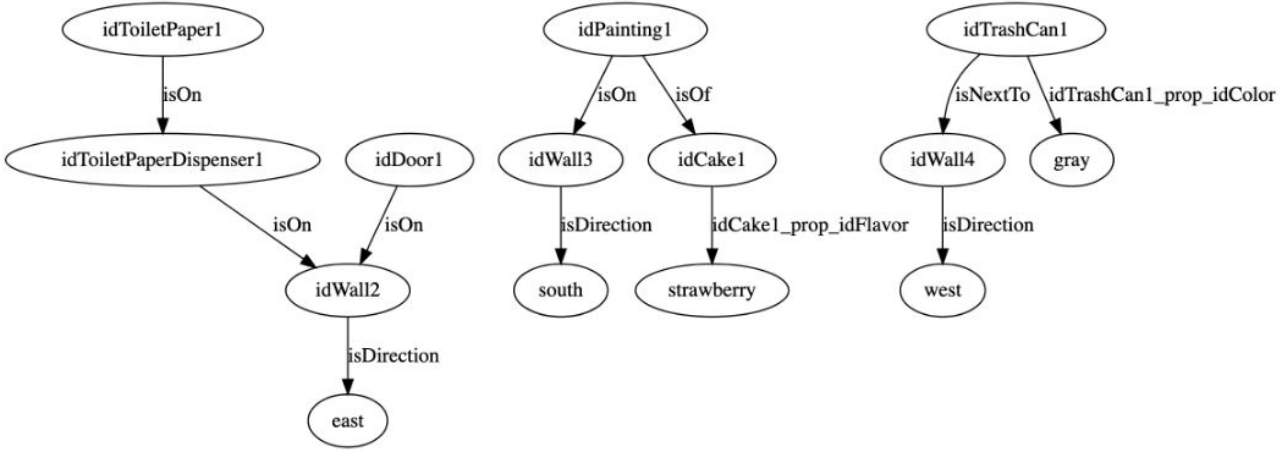
POGG: Precision-Oriented Graphical Generator

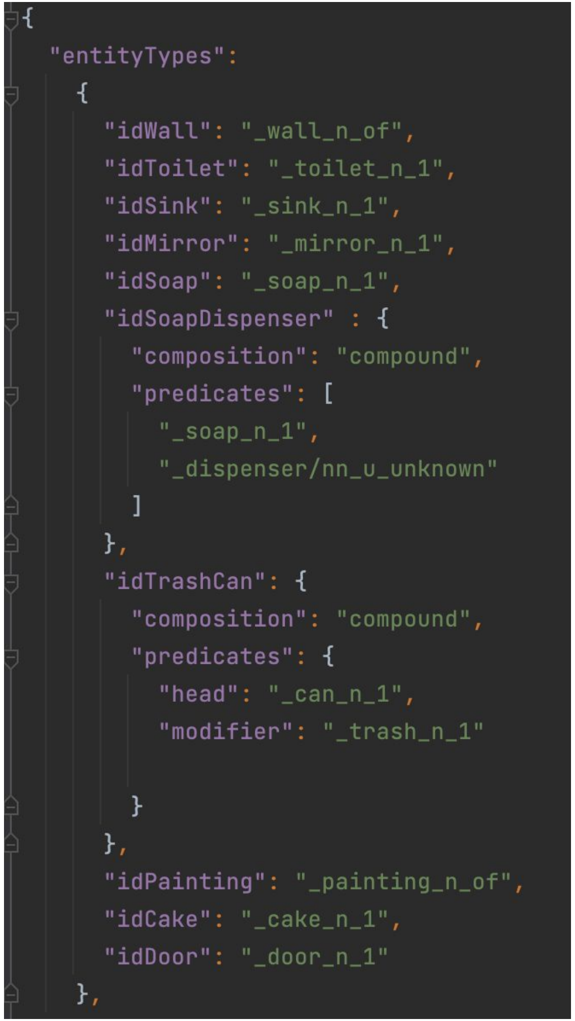
- POGG-internal
  - Graph-to-MRS Algorithm ✔
  - Composition Library ✔
  - MRS Algebra Implementation ✔
- POGG-external
  - Lexicon

All entity data I used for my project came from Eric Zinda :)

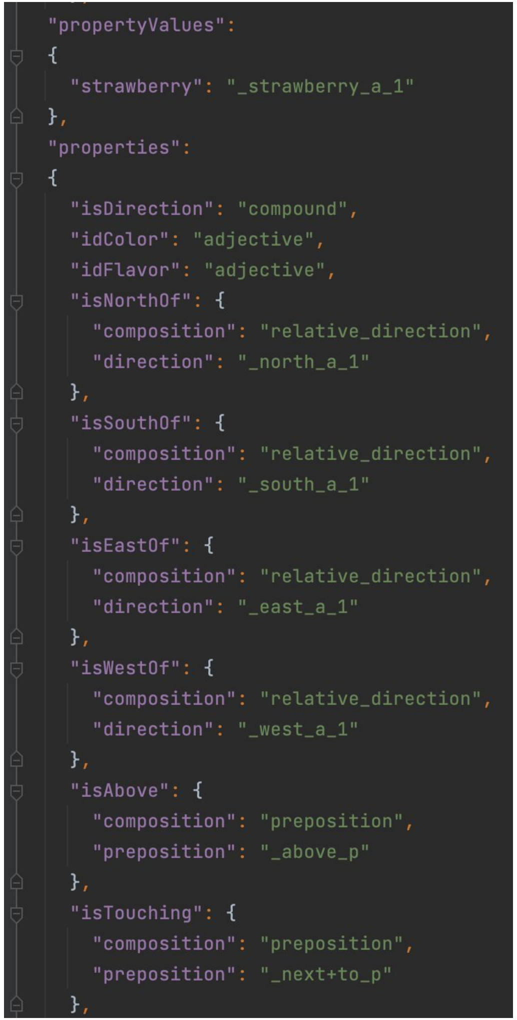
Lexicon


idToilet1
idMirror1
idSoap1
idToilet1_prop_idColor
isEastOf
isWestOf
isBelow
isAbove
idSoap1_prop_idScent
isInsideOf
isTouching
white
idSink1
isOn
lemon
idSoapDispenser1
isOn
isOn
idWall1
isDirection
north


idToiletPaper1
idPainting1
idTrashCan1
isOn
isOn
isOf
isNextTo
idTrashCan1_prop_idColor
idToiletPaperDispenser1
idDoor1
idWall3
idCake1
idWall4
gray
isOn
isOn
isDirection
idCake1_prop_idFlavor
isDirection
idWall2
south
strawberry
west
isDirection
east



```
{
  "entityTypes":
    {
      "idWall": "_wall_n_of",
      "idToilet": "_toilet_n_1",
      "idSink": "_sink_n_1",
      "idMirror": "_mirror_n_1",
      "idSoap": "_soap_n_1",
      "idSoapDispenser" : {
        "composition": "compound",
        "predicates": [
          "_soap_n_1",
          "_dispenser/nn_u_unknown"
        ]
      },
      "idTrashCan": {
        "composition": "compound",
        "predicates": {
          "head": "_can_n_1",
          "modifier": "_trash_n_1"

        }
      },
      "idPainting": "_painting_n_of",
      "idCake": "_cake_n_1",
      "idDoor": "_door_n_1"
    },
```



```
    "propertyValues":
    {
      "strawberry": "_strawberry_a_1"
    },
    "properties":
    {
      "isDirection": "compound",
      "idColor": "adjective",
      "idFlavor": "adjective",
      "isNorthOf": {
        "composition": "relative_direction",
        "direction": "_north_a_1"
      },
      "isSouthOf": {
        "composition": "relative_direction",
        "direction": "_south_a_1"
      },
      "isEastOf": {
        "composition": "relative_direction",
        "direction": "_east_a_1"
      },
      "isWestOf": {
        "composition": "relative_direction",
        "direction": "_west_a_1"
      },
      "isAbove": {
        "composition": "preposition",
        "preposition": "_above_p"
      },
      "isTouching": {
        "composition": "preposition",
        "preposition": "_next+to_p"
      },
```

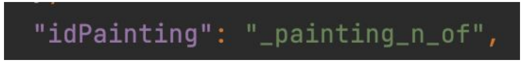
POGG: Precision-Oriented Graphical Generator

- POGG-internal
  - Graph-to-MRS Algorithm ✔
  - Composition Library ✔
  - MRS Algebra Implementation ✔
- POGG-external
  - Lexicon ✔

All entity data I used for my project came from Eric Zinda :)

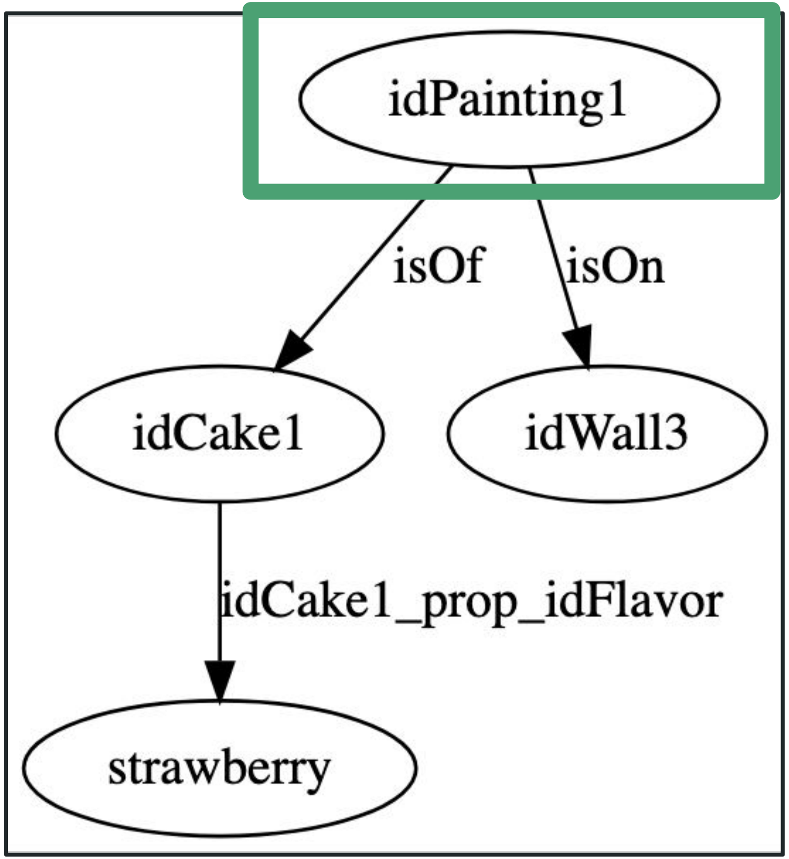
Example

A painting of a strawberry cake on the wall

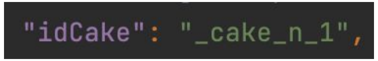
1. Graph-to-MRS algorithm starts recursion at root, **idPainting1**
    a. Produce MRS fragment for *painting*
        i. Consult lexicon
        ii. 

```
"idPainting": "_painting_n_of",
```

        iii. Produce MRS fragment with EP for _painting_n_of

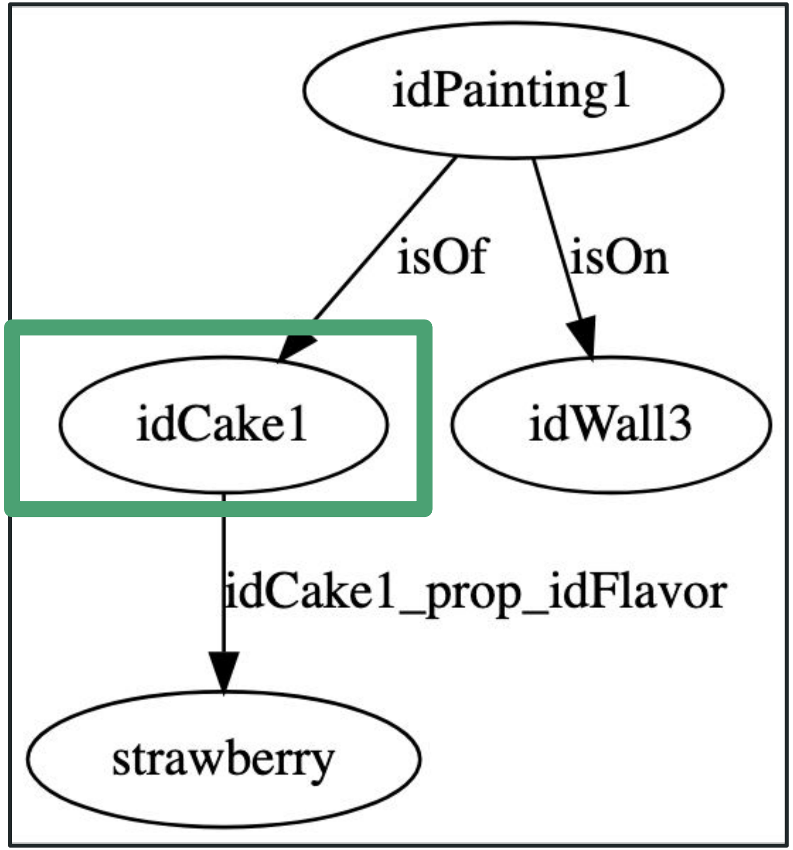


A painting of a strawberry cake on the wall

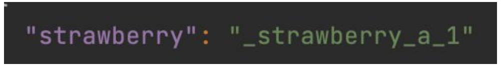
1. Graph-to-MRS algorithm starts recursion at root, **idPainting1**
2. Recurse to **idCake1**
   a. Produce MRS fragment for *cake*
      i. Consult lexicon
      ii. 

```
"idCake": "_cake_n_1",
```

      iii. Produce MRS fragment with EP for _cake_n_1

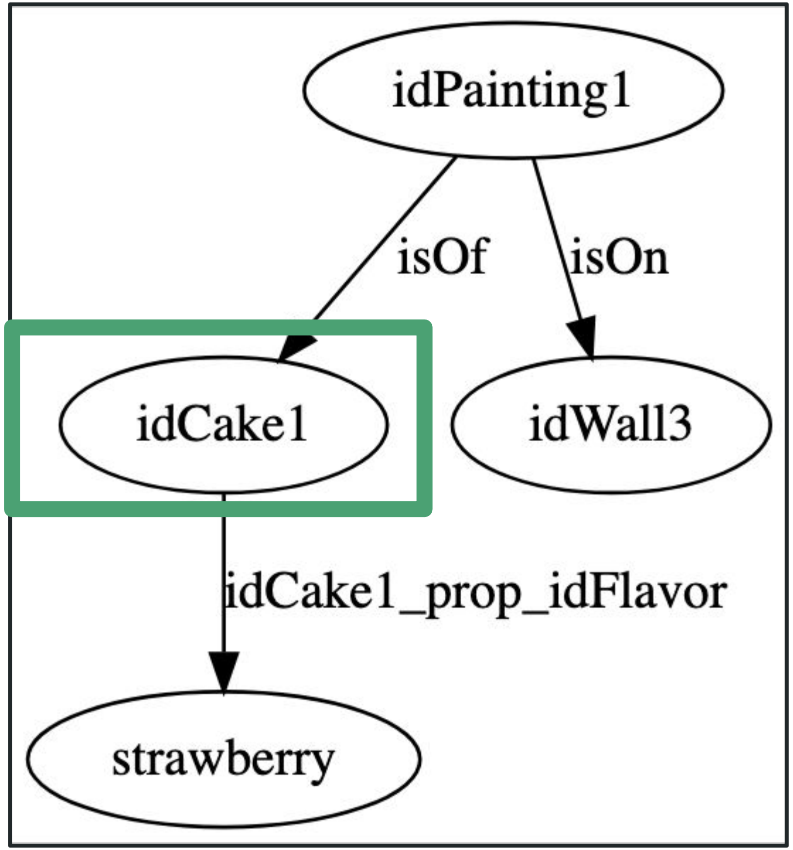


A painting of a strawberry cake on the wall

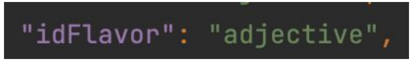
1. Graph-to-MRS algorithm starts recursion at root, **idPainting1**
2. Recurse to **idCake1**
3. Recurse to **strawberry**
    a. Produce MRS fragment for *strawberry*
        i. Consult lexicon
        ii. 

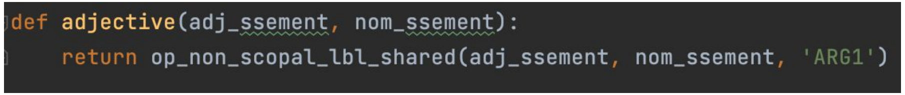
```
"strawberry": "_strawberry_a_1"
```

        iii. Produce MRS fragment with EP for _strawberry_a_1

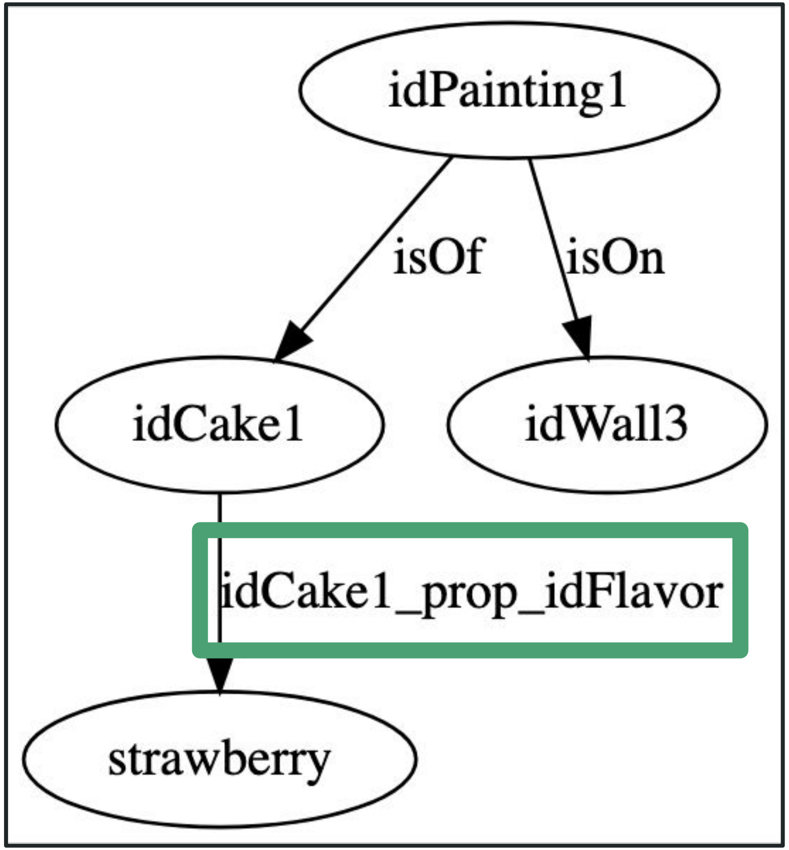


A painting of a strawberry cake on the wall

1. Graph-to-MRS algorithm starts recursion at root, **idPainting1**
2. Recurse to **idCake1**
3. Recurse to **strawberry**
4. Return up call stack
   a. Perform composition between *cake* and *strawberry*
      i. Consult lexicon
      ii. 
      ```
      "idFlavor": "adjective",
      ```
      iii. Consult composition library



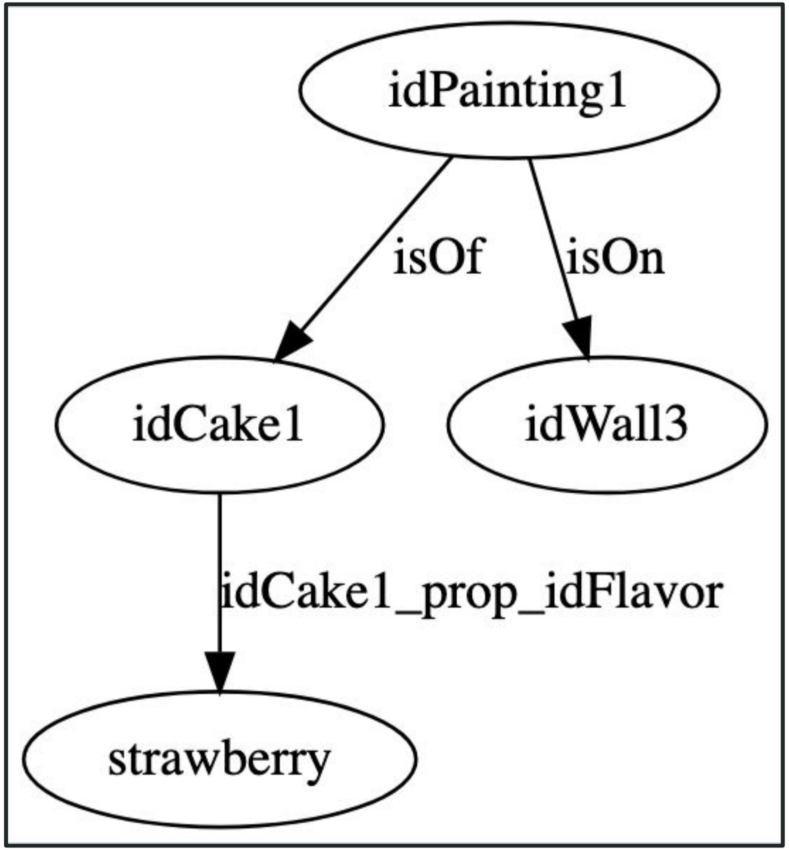
```
def adjective(adj_ssement, nom_ssement):
    return op_non_scopal_lbl_shared(adj_ssement, nom_ssement, 'ARG1')
```



A painting of a strawberry cake on the wall

1. Graph-to-MRS algorithm starts recursion at root, **idPainting1**
2. Recurse to **idCake1**
3. Recurse to **strawberry**
4. Return up call stack
5. etc.

The role of each component should now be clear



Evaluation

Evaluation Metrics

- Graph Coverage — how many graphs result in English strings?
- Node Coverage — how many nodes in a graph produced an MRS segment?
- Edge Coverage — how many edges in a graph were successfully used for composition?
- Node Inclusion — how many nodes contributed semantic information to the final MRS?
- Edge Inclusion — how many edges contributed semantic information to the final MRS?

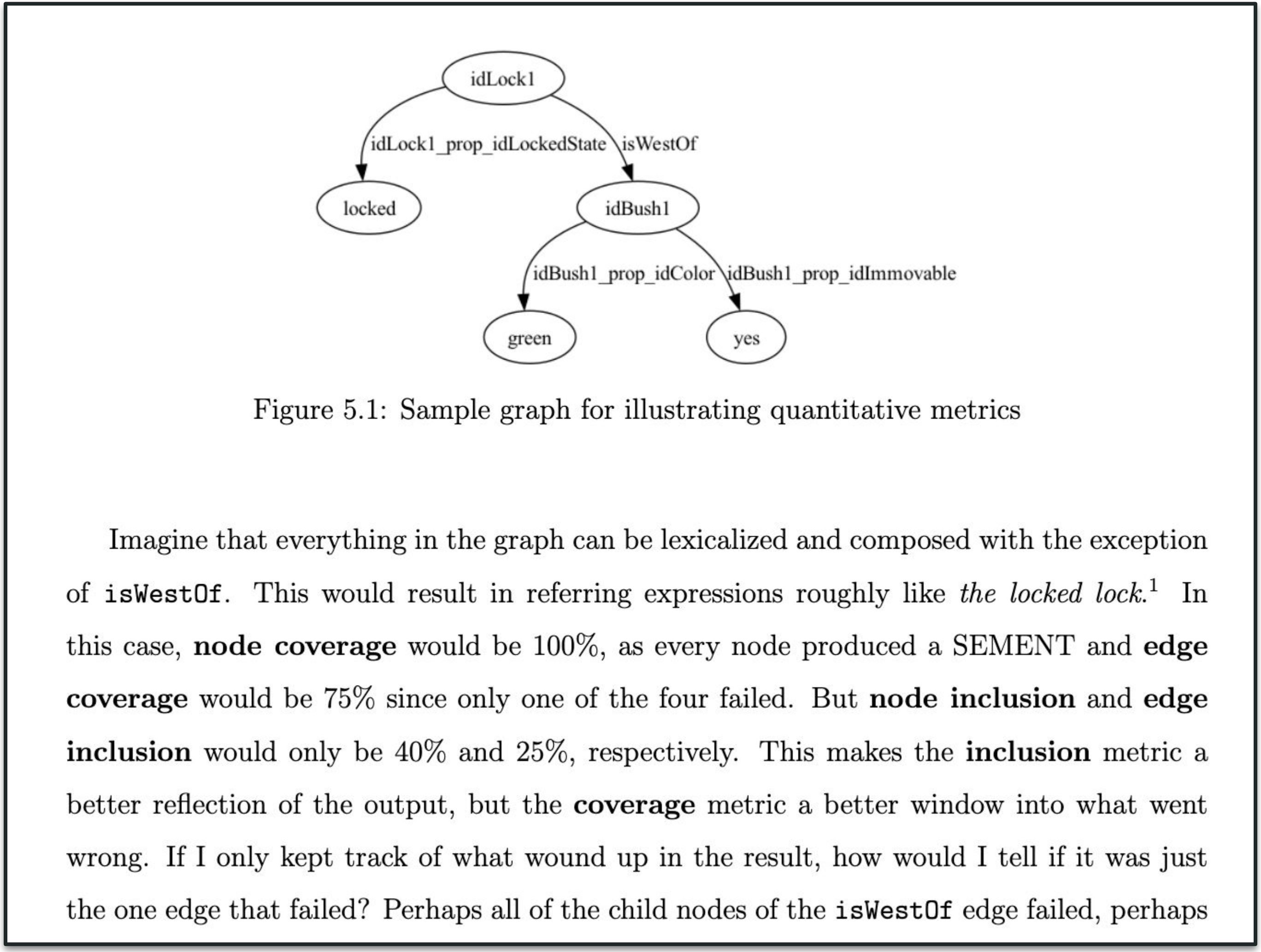
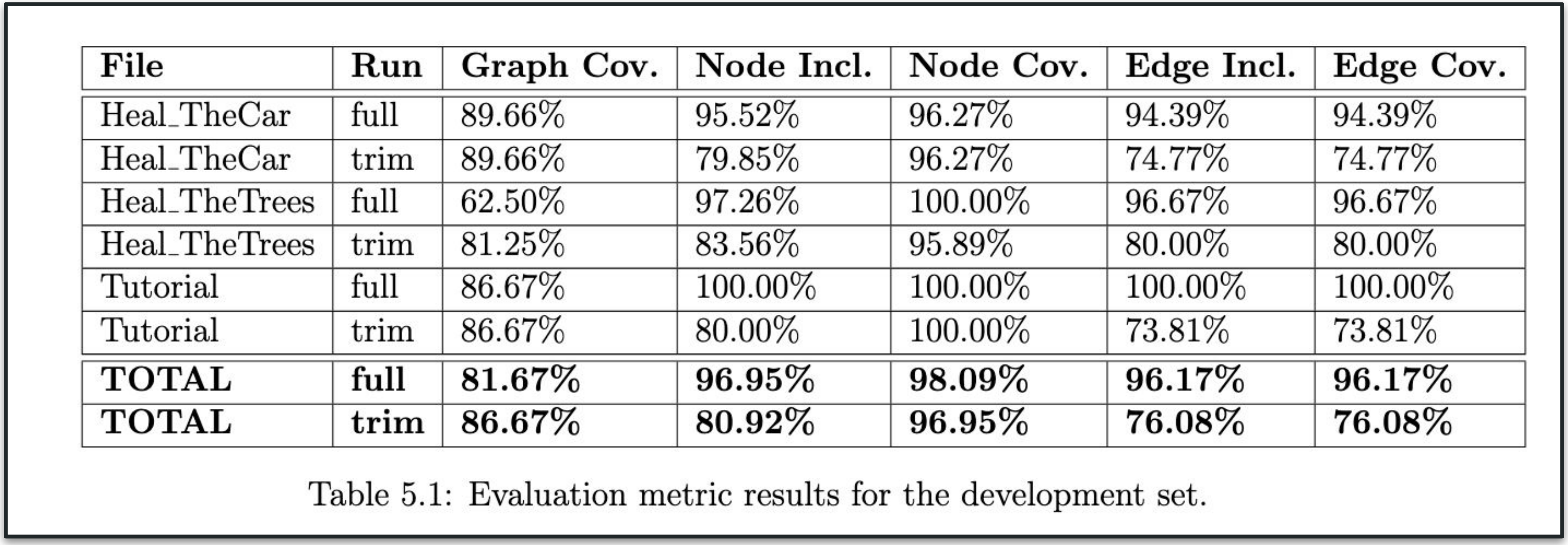

idLock1
idLock1_prop_idLockedState
isWestOf
locked
idBush1
idBush1_prop_idColor
idBush1_prop_idImmovable
green
yes

Figure 5.1: Sample graph for illustrating quantitative metrics

Imagine that everything in the graph can be lexicalized and composed with the exception of `isWestOf`. This would result in referring expressions roughly like *the locked lock.*[1] In this case, **node coverage** would be 100%, as every node produced a SEMENT and **edge coverage** would be 75% since only one of the four failed. But **node inclusion** and **edge inclusion** would only be 40% and 25%, respectively. This makes the **inclusion** metric a better reflection of the output, but the **coverage** metric a better window into what went wrong. If I only kept track of what wound up in the result, how would I tell if it was just the one edge that failed? Perhaps all of the child nodes of the `isWestOf` edge failed, perhaps

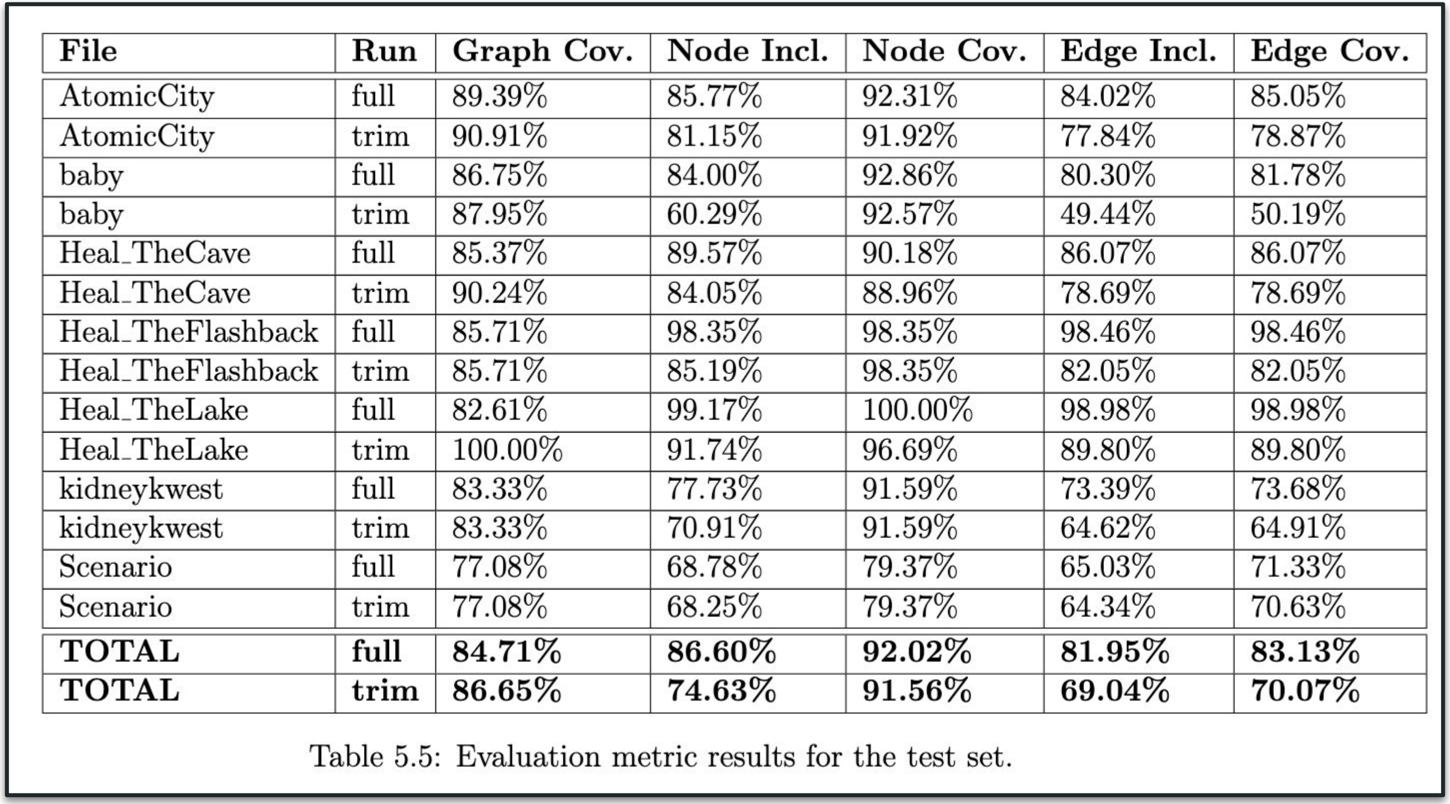
Results



File	**Run**	**Graph Cov.**	**Node Incl.**	**Node Cov.**	**Edge Incl.**	**Edge Cov.**
Heal_TheCar	full	89.66%	95.52%	96.27%	94.39%	94.39%
Heal_TheCar	trim	89.66%	79.85%	96.27%	74.77%	74.77%
Heal_TheTrees	full	62.50%	97.26%	100.00%	96.67%	96.67%
Heal_TheTrees	trim	81.25%	83.56%	95.89%	80.00%	80.00%
Tutorial	full	86.67%	100.00%	100.00%	100.00%	100.00%
Tutorial	trim	86.67%	80.00%	100.00%	73.81%	73.81%
TOTAL	**full**	**81.67%**	**96.95%**	**98.09%**	**96.17%**	**96.17%**
TOTAL	**trim**	**86.67%**	**80.92%**	**96.95%**	**76.08%**	**76.08%**

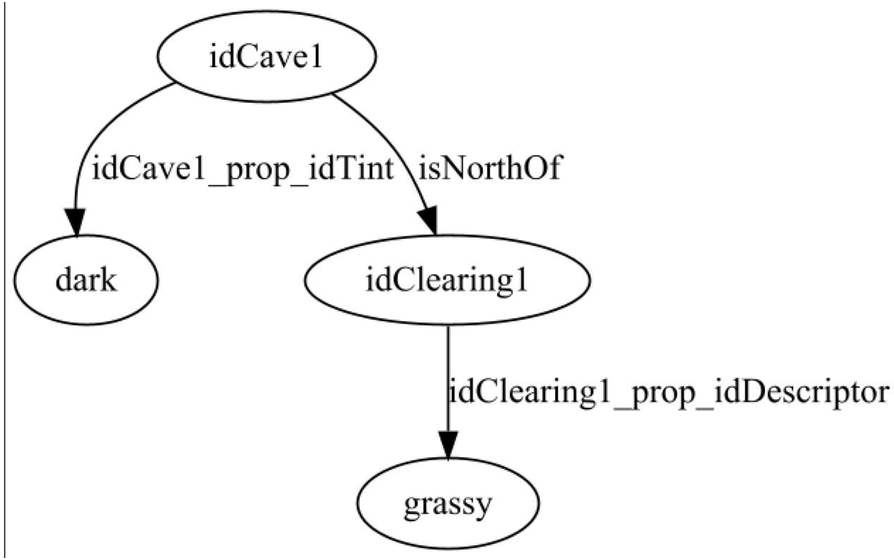
Table 5.1: Evaluation metric results for the development set.

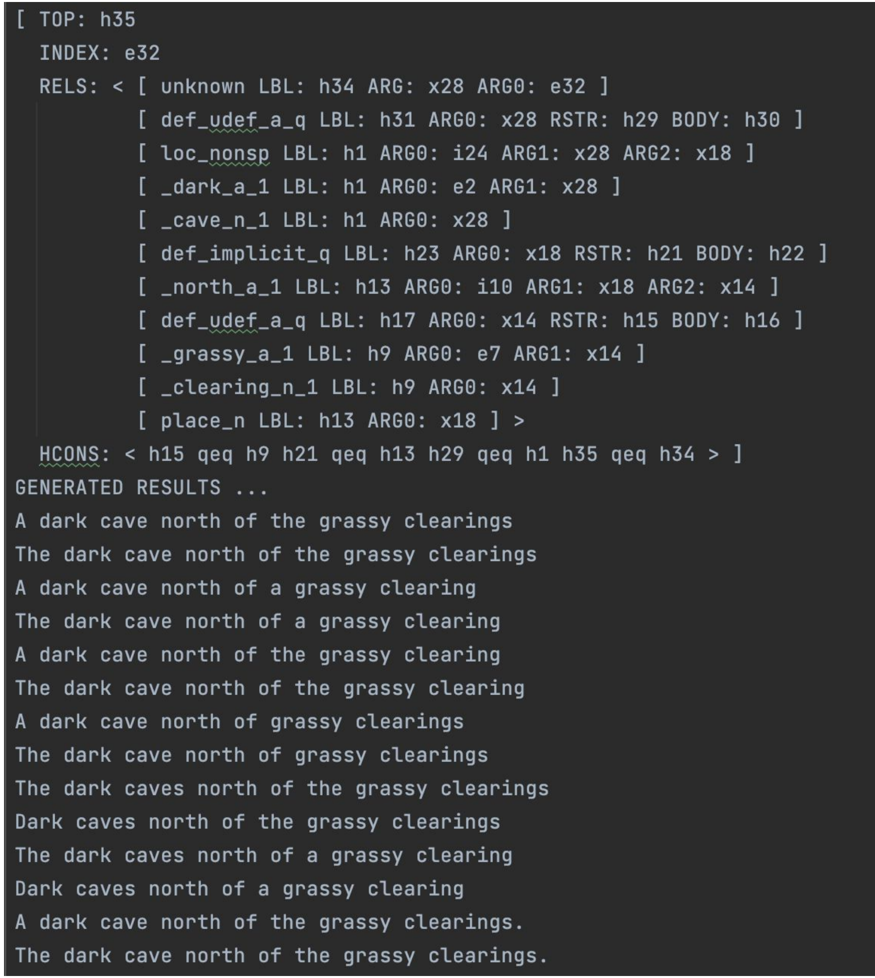
Results



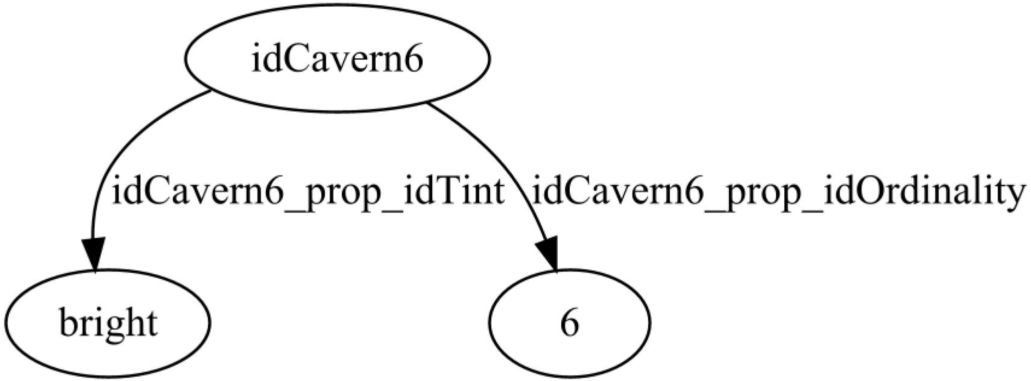
File	Run	Graph Cov.	Node Incl.	Node Cov.	Edge Incl.	Edge Cov.
AtomicCity	full	89.39%	85.77%	92.31%	84.02%	85.05%
AtomicCity	trim	90.91%	81.15%	91.92%	77.84%	78.87%
baby	full	86.75%	84.00%	92.86%	80.30%	81.78%
baby	trim	87.95%	60.29%	92.57%	49.44%	50.19%
Heal_TheCave	full	85.37%	89.57%	90.18%	86.07%	86.07%
Heal_TheCave	trim	90.24%	84.05%	88.96%	78.69%	78.69%
Heal_TheFlashback	full	85.71%	98.35%	98.35%	98.46%	98.46%
Heal_TheFlashback	trim	85.71%	85.19%	98.35%	82.05%	82.05%
Heal_TheLake	full	82.61%	99.17%	100.00%	98.98%	98.98%
Heal_TheLake	trim	100.00%	91.74%	96.69%	89.80%	89.80%
kidneykwest	full	83.33%	77.73%	91.59%	73.39%	73.68%
kidneykwest	trim	83.33%	70.91%	91.59%	64.62%	64.91%
Scenario	full	77.08%	68.78%	79.37%	65.03%	71.33%
Scenario	trim	77.08%	68.25%	79.37%	64.34%	70.63%
TOTAL	**full**	**84.71%**	**86.60%**	**92.02%**	**81.95%**	**83.13%**
TOTAL	**trim**	**86.65%**	**74.63%**	**91.56%**	**69.04%**	**70.07%**

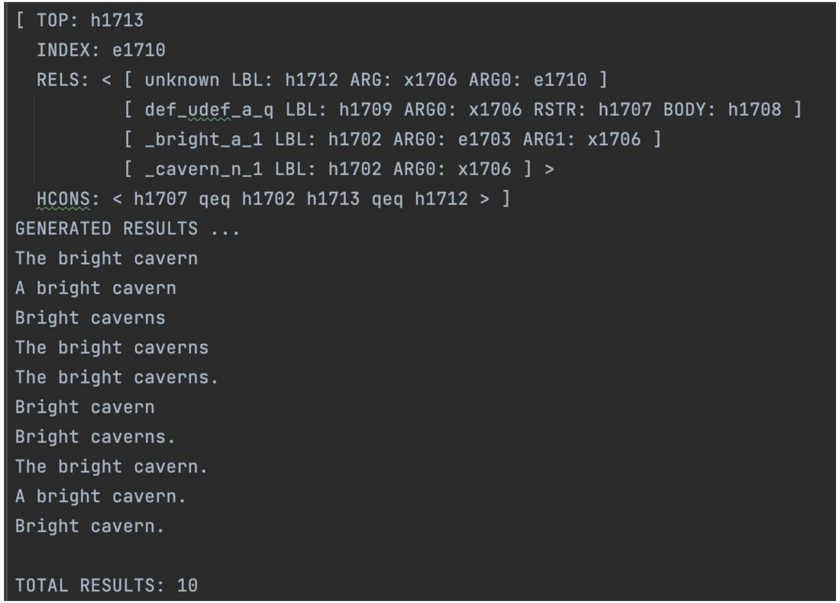
Table 5.5: Evaluation metric results for the test set.



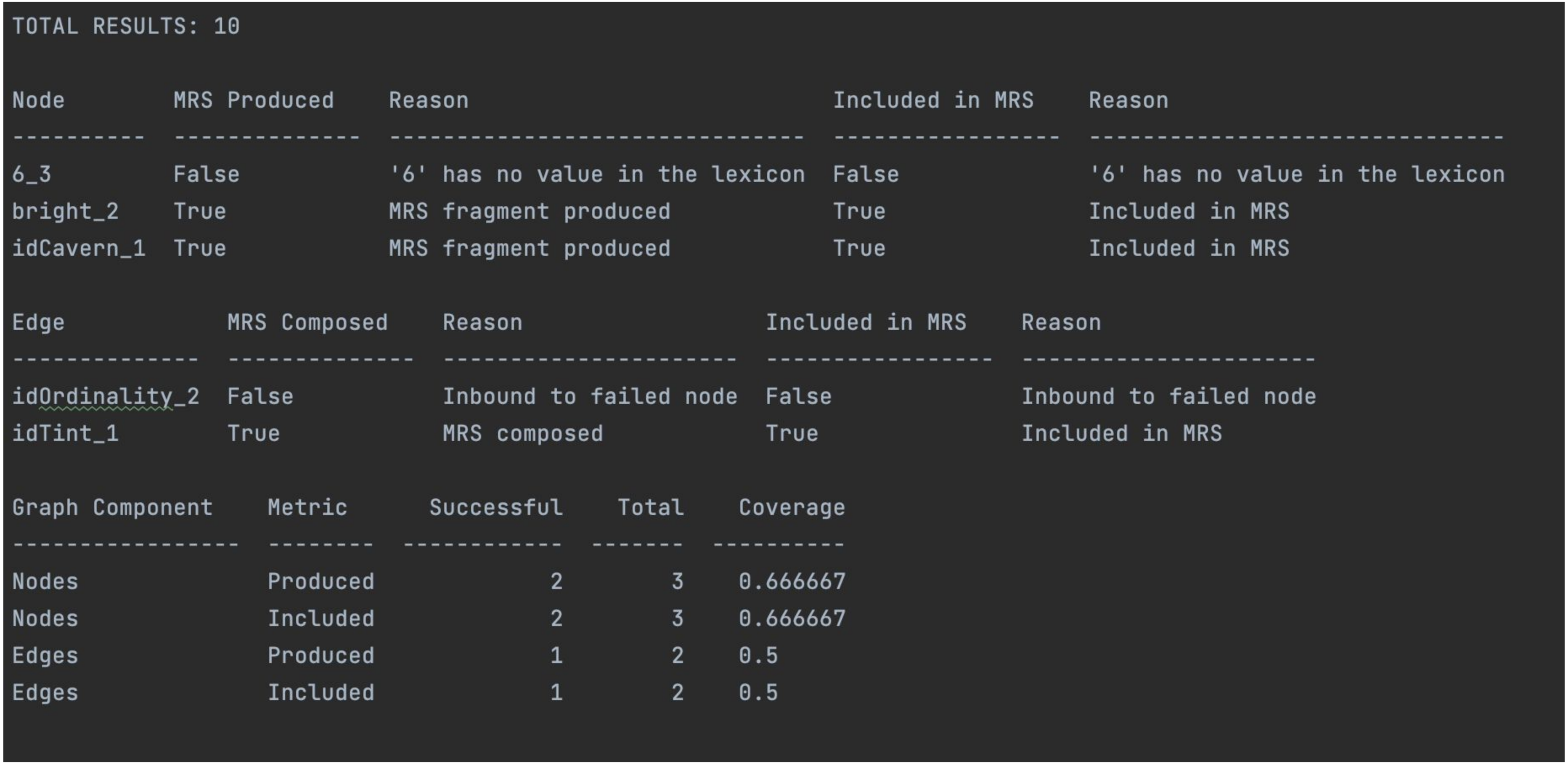


```
[ TOP: h35
  INDEX: e32
  RELS: < [ unknown LBL: h34 ARG: x28 ARG0: e32 ]
          [ def_udef_a_q LBL: h31 ARG0: x28 RSTR: h29 BODY: h30 ]
          [ loc_nonsp LBL: h1 ARG0: i24 ARG1: x28 ARG2: x18 ]
          [ _dark_a_1 LBL: h1 ARG0: e2 ARG1: x28 ]
          [ _cave_n_1 LBL: h1 ARG0: x28 ]
          [ def_implicit_q LBL: h23 ARG0: x18 RSTR: h21 BODY: h22 ]
          [ _north_a_1 LBL: h13 ARG0: i10 ARG1: x18 ARG2: x14 ]
          [ def_udef_a_q LBL: h17 ARG0: x14 RSTR: h15 BODY: h16 ]
          [ _grassy_a_1 LBL: h9 ARG0: e7 ARG1: x14 ]
          [ _clearing_n_1 LBL: h9 ARG0: x14 ]
          [ place_n LBL: h13 ARG0: x18 ] >
  HCONS: < h15 qeq h9 h21 qeq h13 h29 qeq h1 h35 qeq h34 > ]
GENERATED RESULTS ...
A dark cave north of the grassy clearings
The dark cave north of the grassy clearings
A dark cave north of a grassy clearing
The dark cave north of a grassy clearing
A dark cave north of the grassy clearing
The dark cave north of the grassy clearing
A dark cave north of grassy clearings
The dark cave north of grassy clearings
The dark caves north of the grassy clearings
Dark caves north of the grassy clearings
The dark caves north of a grassy clearing
Dark caves north of a grassy clearing
A dark cave north of the grassy clearings.
The dark cave north of the grassy clearings.
```

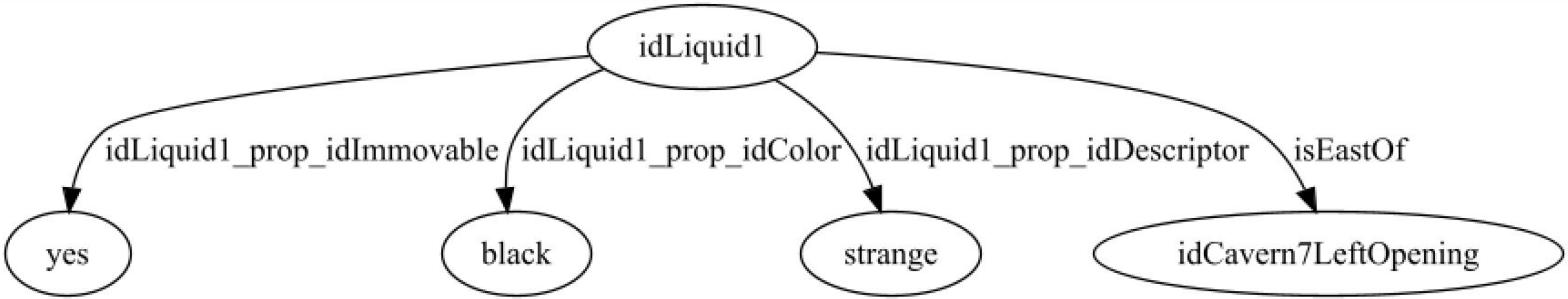


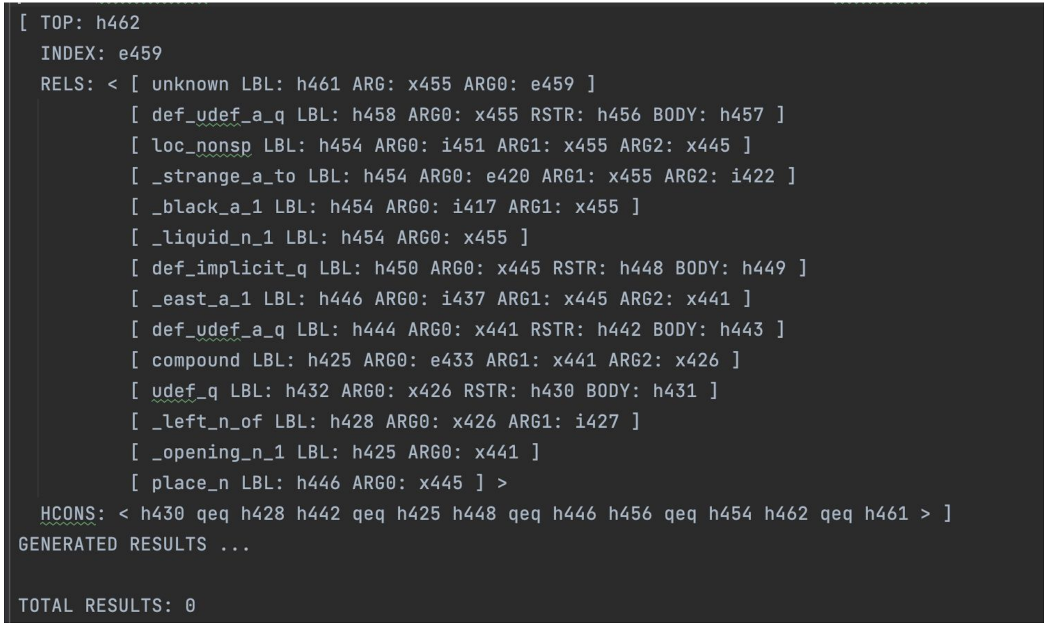


```
[ TOP: h1713
  INDEX: e1710
  RELS: < [ unknown LBL: h1712 ARG: x1706 ARG0: e1710 ]
          [ def_udef_a_q LBL: h1709 ARG0: x1706 RSTR: h1707 BODY: h1708 ]
          [ _bright_a_1 LBL: h1702 ARG0: e1703 ARG1: x1706 ]
          [ _cavern_n_1 LBL: h1702 ARG0: x1706 ] >
  HCONS: < h1707 qeq h1702 h1713 qeq h1712 > ]
GENERATED RESULTS ...
The bright cavern
A bright cavern
Bright caverns
The bright caverns
The bright caverns.
Bright cavern
Bright caverns.
The bright cavern.
A bright cavern.
Bright cavern.

TOTAL RESULTS: 10
```

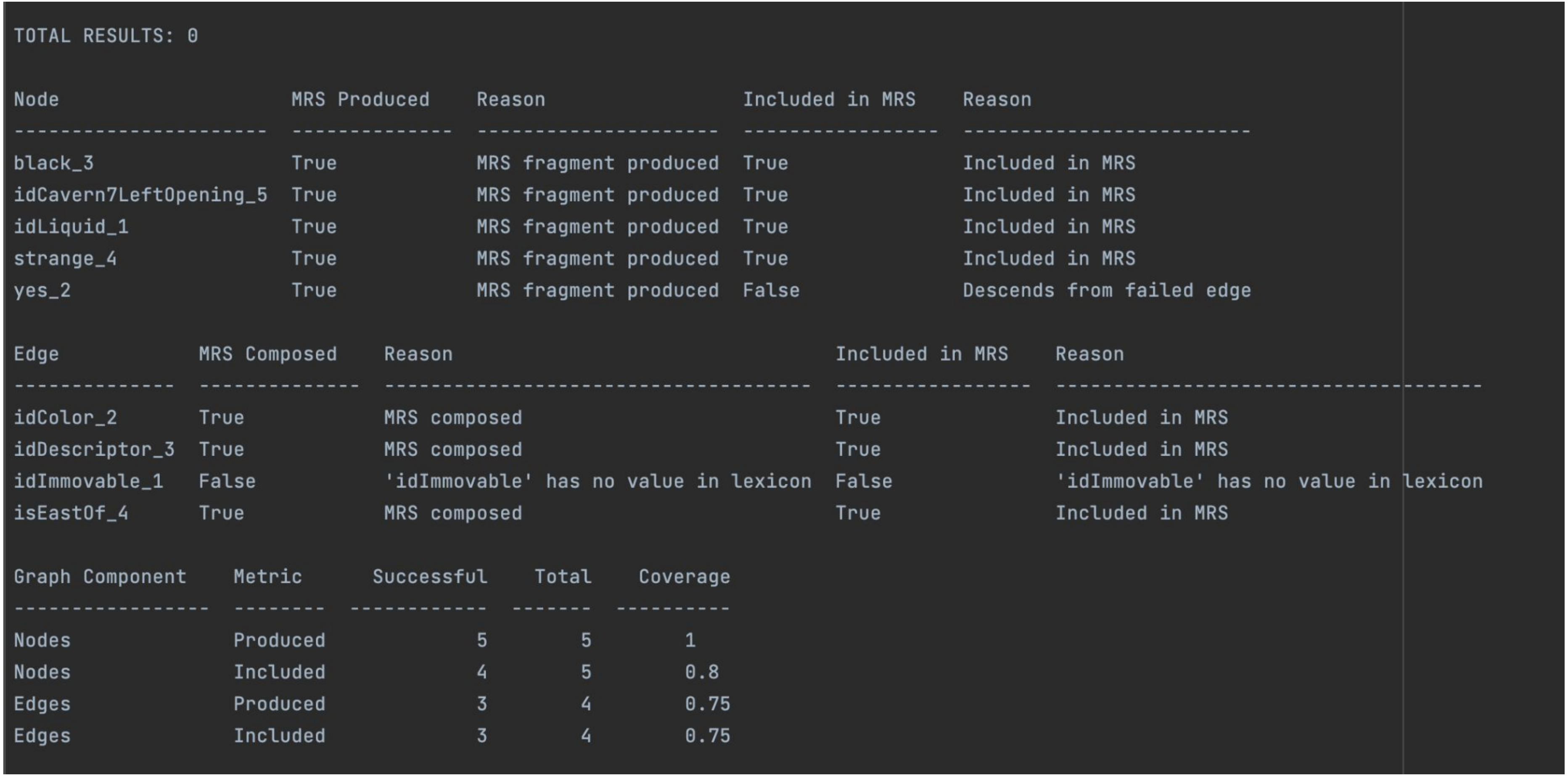


```
TOTAL RESULTS: 10

Node        MRS Produced    Reason                           Included in MRS    Reason
----------  --------------  -------------------------------  -----------------  -------------------------------
6_3         False           '6' has no value in the lexicon  False              '6' has no value in the lexicon
bright_2    True            MRS fragment produced            True               Included in MRS
idCavern_1  True            MRS fragment produced            True               Included in MRS

Edge            MRS Composed    Reason                  Included in MRS    Reason
--------------  --------------  ----------------------  -----------------  ----------------------
idOrdinality_2  False           Inbound to failed node  False              Inbound to failed node
idTint_1        True            MRS composed            True               Included in MRS

Graph Component    Metric      Successful    Total    Coverage
-----------------  --------  ------------  -------  ----------
Nodes              Produced             2        3    0.666667
Nodes              Included             2        3    0.666667
Edges              Produced             1        2    0.5
Edges              Included             1        2    0.5
```

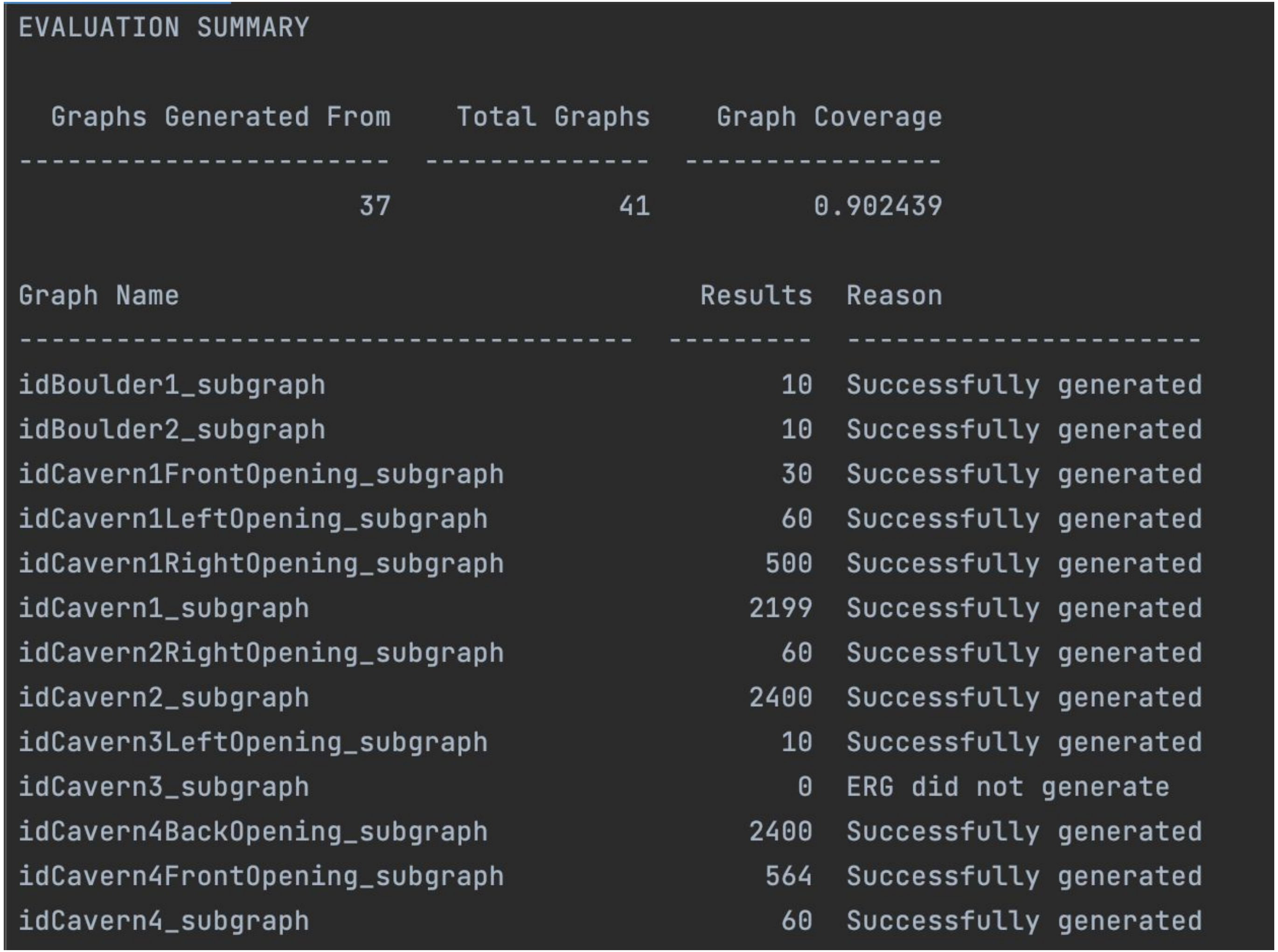




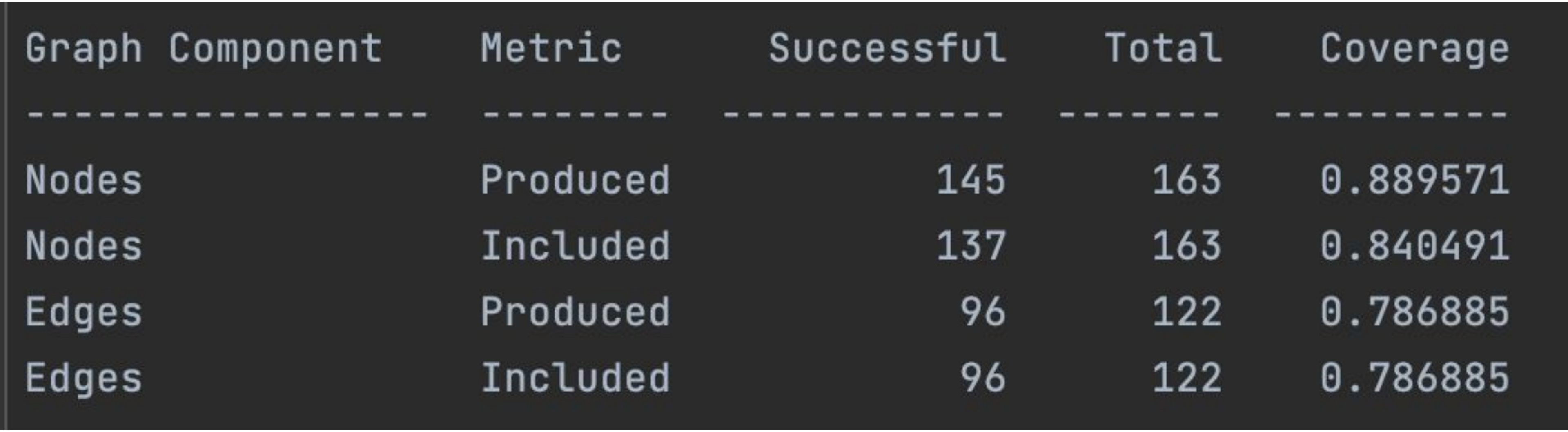
```
[ TOP: h462
  INDEX: e459
  RELS: < [ unknown LBL: h461 ARG: x455 ARG0: e459 ]
          [ def_udef_a_q LBL: h458 ARG0: x455 RSTR: h456 BODY: h457 ]
          [ loc_nonsp LBL: h454 ARG0: i451 ARG1: x455 ARG2: x445 ]
          [ _strange_a_to LBL: h454 ARG0: e420 ARG1: x455 ARG2: i422 ]
          [ _black_a_1 LBL: h454 ARG0: i417 ARG1: x455 ]
          [ _liquid_n_1 LBL: h454 ARG0: x455 ]
          [ def_implicit_q LBL: h450 ARG0: x445 RSTR: h448 BODY: h449 ]
          [ _east_a_1 LBL: h446 ARG0: i437 ARG1: x445 ARG2: x441 ]
          [ def_udef_a_q LBL: h444 ARG0: x441 RSTR: h442 BODY: h443 ]
          [ compound LBL: h425 ARG0: e433 ARG1: x441 ARG2: x426 ]
          [ udef_q LBL: h432 ARG0: x426 RSTR: h430 BODY: h431 ]
          [ _left_n_of LBL: h428 ARG0: x426 ARG1: i427 ]
          [ _opening_n_1 LBL: h425 ARG0: x441 ]
          [ place_n LBL: h446 ARG0: x445 ] >
  HCONS: < h430 qeq h428 h442 qeq h425 h448 qeq h446 h456 qeq h454 h462 qeq h461 > ]
GENERATED RESULTS ...

TOTAL RESULTS: 0
```



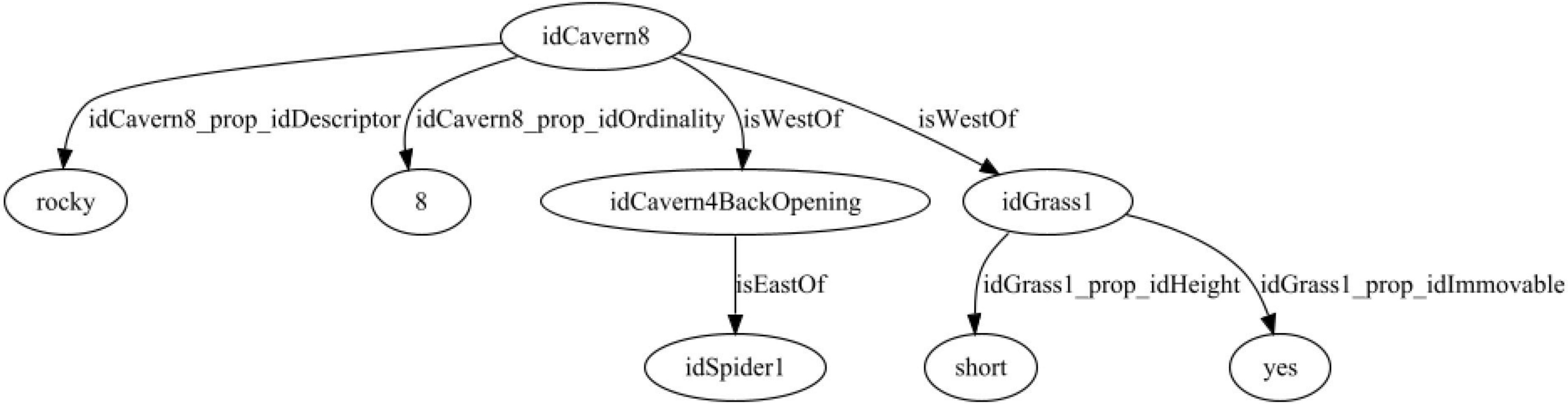
```
TOTAL RESULTS: 0

Node                    MRS Produced    Reason                 Included in MRS    Reason
----------------------  --------------  ---------------------  -----------------  -------------------------
black_3                 True            MRS fragment produced  True               Included in MRS
idCavern7LeftOpening_5  True            MRS fragment produced  True               Included in MRS
idLiquid_1              True            MRS fragment produced  True               Included in MRS
strange_4               True            MRS fragment produced  True               Included in MRS
yes_2                   True            MRS fragment produced  False              Descends from failed edge

Edge            MRS Composed    Reason                                 Included in MRS    Reason
--------------  --------------  -------------------------------------  -----------------  -------------------------------------
idColor_2       True            MRS composed                           True               Included in MRS
idDescriptor_3  True            MRS composed                           True               Included in MRS
idImmovable_1   False           'idImmovable' has no value in lexicon  False              'idImmovable' has no value in lexicon
isEastOf_4      True            MRS composed                           True               Included in MRS

Graph Component    Metric      Successful    Total    Coverage
-----------------  --------  ------------  -------  ----------
Nodes              Produced             5        5        1
Nodes              Included             4        5        0.8
Edges              Produced             3        4        0.75
Edges              Included             3        4        0.75
```

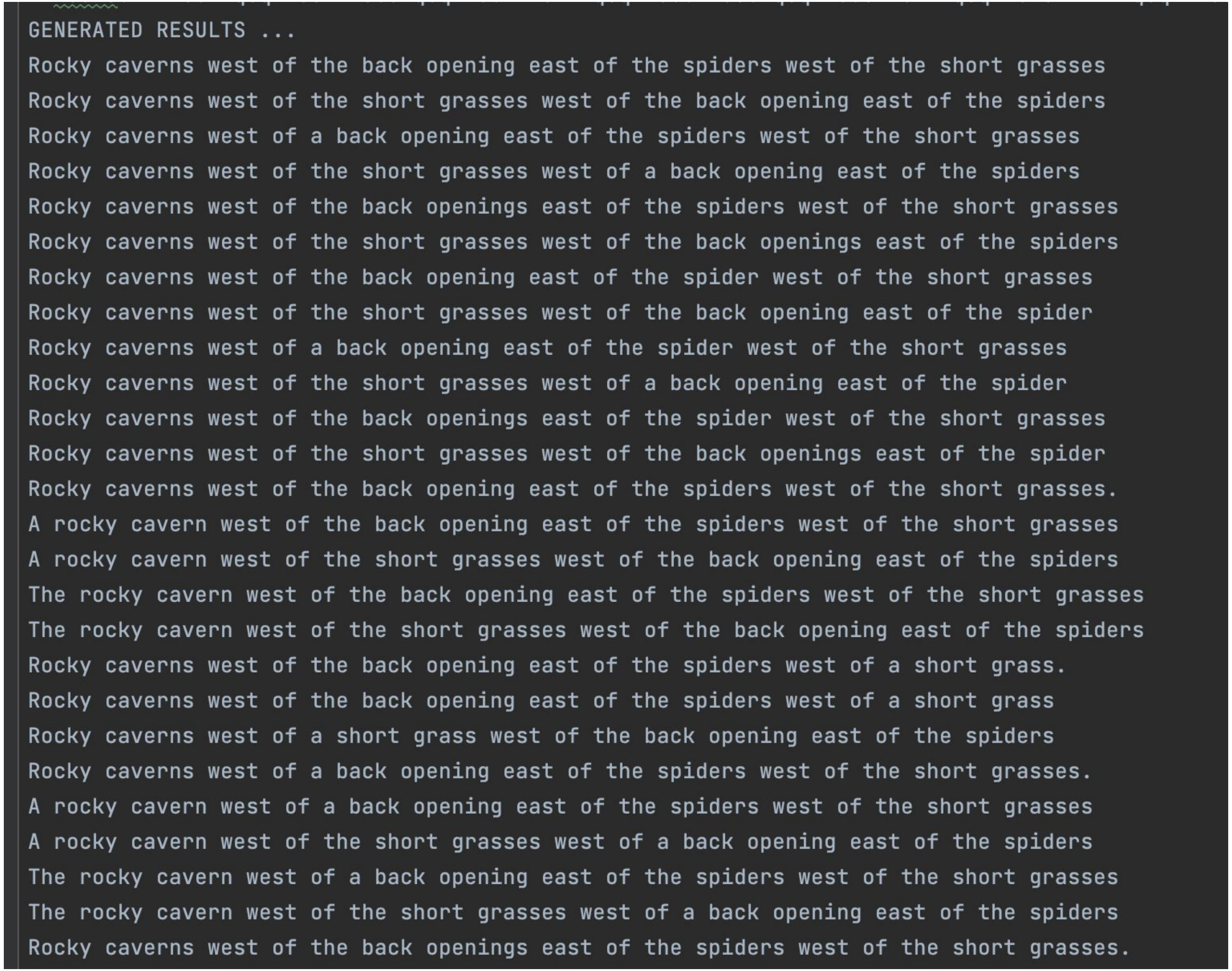


```
EVALUATION SUMMARY

  Graphs Generated From    Total Graphs    Graph Coverage
-----------------------  --------------  ----------------
                     37              41          0.902439

Graph Name                                Results  Reason
--------------------------------------  ---------  ----------------------
idBoulder1_subgraph                            10  Successfully generated
idBoulder2_subgraph                            10  Successfully generated
idCavern1FrontOpening_subgraph                 30  Successfully generated
idCavern1LeftOpening_subgraph                  60  Successfully generated
idCavern1RightOpening_subgraph                500  Successfully generated
idCavern1_subgraph                           2199  Successfully generated
idCavern2RightOpening_subgraph                 60  Successfully generated
idCavern2_subgraph                           2400  Successfully generated
idCavern3LeftOpening_subgraph                  10  Successfully generated
idCavern3_subgraph                              0  ERG did not generate
idCavern4BackOpening_subgraph                2400  Successfully generated
idCavern4FrontOpening_subgraph                564  Successfully generated
idCavern4_subgraph                             60  Successfully generated
```

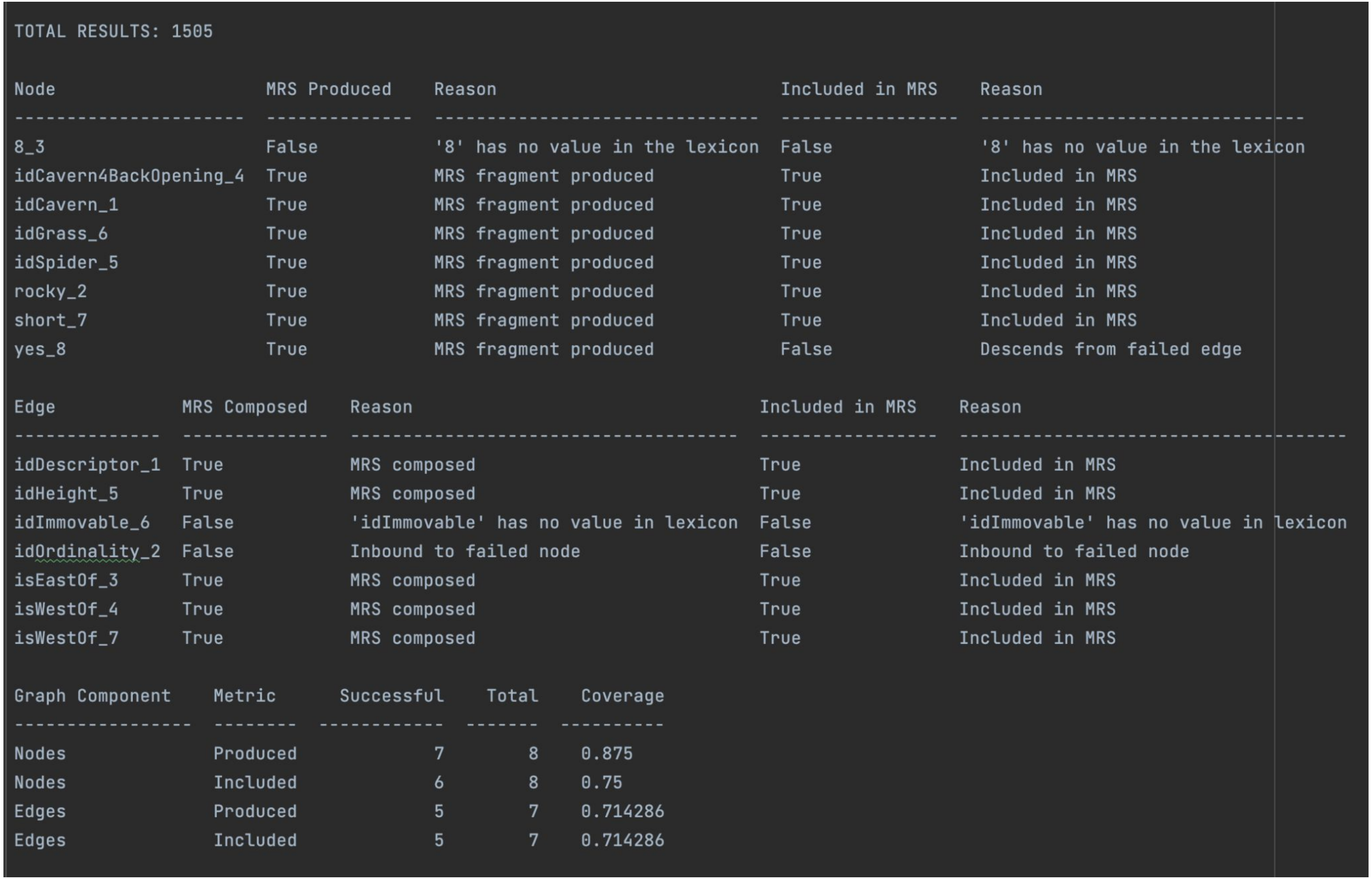


```
Graph Component    Metric      Successful    Total    Coverage
-----------------  --------  ------------  -------  ----------
Nodes              Produced           145      163    0.889571
Nodes              Included           137      163    0.840491
Edges              Produced            96      122    0.786885
Edges              Included            96      122    0.786885
```


idCavern8
idCavern8_prop_idDescriptor
idCavern8_prop_idOrdinality
isWestOf
isWestOf
rocky
8
idCavern4BackOpening
idGrass1
isEastOf
idGrass1_prop_idHeight
idGrass1_prop_idImmovable
idSpider1
short
yes

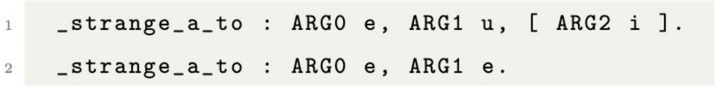


```
GENERATED RESULTS ...
Rocky caverns west of the back opening east of the spiders west of the short grasses
Rocky caverns west of the short grasses west of the back opening east of the spiders
Rocky caverns west of a back opening east of the spiders west of the short grasses
Rocky caverns west of the short grasses west of a back opening east of the spiders
Rocky caverns west of the back openings east of the spiders west of the short grasses
Rocky caverns west of the short grasses west of the back openings east of the spiders
Rocky caverns west of the back opening east of the spider west of the short grasses
Rocky caverns west of the short grasses west of the back opening east of the spider
Rocky caverns west of a back opening east of the spider west of the short grasses
Rocky caverns west of the short grasses west of a back opening east of the spider
Rocky caverns west of the back openings east of the spider west of the short grasses
Rocky caverns west of the short grasses west of the back openings east of the spider
Rocky caverns west of the back opening east of the spiders west of the short grasses.
A rocky cavern west of the back opening east of the spiders west of the short grasses
A rocky cavern west of the short grasses west of the back opening east of the spiders
The rocky cavern west of the back opening east of the spiders west of the short grasses
The rocky cavern west of the short grasses west of the back opening east of the spiders
Rocky caverns west of the back opening east of the spiders west of a short grass.
Rocky caverns west of the back opening east of the spiders west of a short grass
Rocky caverns west of a short grass west of the back opening east of the spiders
Rocky caverns west of a back opening east of the spiders west of the short grasses.
A rocky cavern west of a back opening east of the spiders west of the short grasses
A rocky cavern west of the short grasses west of a back opening east of the spiders
The rocky cavern west of a back opening east of the spiders west of the short grasses
The rocky cavern west of the short grasses west of a back opening east of the spiders
Rocky caverns west of the back openings east of the spiders west of the short grasses.
```



```
TOTAL RESULTS: 1505

Node                    MRS Produced    Reason                           Included in MRS    Reason
----------------------  --------------  -------------------------------  -----------------  -------------------------------
8_3                     False           '8' has no value in the lexicon  False              '8' has no value in the lexicon
idCavern4BackOpening_4  True            MRS fragment produced            True               Included in MRS
idCavern_1              True            MRS fragment produced            True               Included in MRS
idGrass_6               True            MRS fragment produced            True               Included in MRS
idSpider_5              True            MRS fragment produced            True               Included in MRS
rocky_2                 True            MRS fragment produced            True               Included in MRS
short_7                 True            MRS fragment produced            True               Included in MRS
yes_8                   True            MRS fragment produced            False              Descends from failed edge

Edge            MRS Composed    Reason                                 Included in MRS    Reason
--------------  --------------  -------------------------------------  -----------------  -------------------------------------
idDescriptor_1  True            MRS composed                           True               Included in MRS
idHeight_5      True            MRS composed                           True               Included in MRS
idImmovable_6   False           'idImmovable' has no value in lexicon  False              'idImmovable' has no value in lexicon
idOrdinality_2  False           Inbound to failed node                 False              Inbound to failed node
isEastOf_3      True            MRS composed                           True               Included in MRS
isWestOf_4      True            MRS composed                           True               Included in MRS
isWestOf_7      True            MRS composed                           True               Included in MRS

Graph Component    Metric      Successful    Total    Coverage
-----------------  --------  ------------  -------  ----------
Nodes              Produced             7        8    0.875
Nodes              Included             6        8    0.75
Edges              Produced             5        7    0.714286
Edges              Included             5        7    0.714286
```

Error Analysis

- Types of errors:
    - Phenomena unaccounted for
        - proper names, numbers (cardinal and ordinal), nouns that take complements
    - Issues with algebra implementation
        - dropping holes that are required later
            - *un- + locked*
                - since un- is the semantic functor, the holes from *locked* are dropped but they are needed later because I have to specify what it is that is unlocked
    - ERG won’t generate for unknown reason
        - e.g. *the go button*
    - Selecting the wrong synopsis
        - 

```
_strange_a_to : ARG0 e, ARG1 u, [ ARG2 i ].
_strange_a_to : ARG0 e, ARG1 e.
```

Questions

Questions (from me to you)

- Why is it that in the algebra it works out to pass up the slots from both the functor and argument?
  - Any ideas for how I can address this for my use case where I am not relying on syntax?
- Can you provide more clarity on the distinction between an MRS and a SEMENT?
  - DELPH-IN QA conversation: https://delphinqa.ling.washington.edu/t/differences-between-mrs-and-sement/1055/5

Questions (from you to me)

Any questions? :)