A Multi-phase Backtracking Approach to Solving Scope-Resolved MRS

Eric Zinda

Demo

An Example World: A File System

- Individuals are files and folders
- Folders can contain files or other folders
- A user is in a “current folder”
- Individuals can be copied or deleted

What Might Be Said?

- Propositions and Yes/No Questions
  - Files are large.
  - Is a file in this folder not large?
- WH-Questions
  - Which files are in this folder?
  - What folder am I in?
- Commands
  - Delete <file>.
  - Copy <file> in <folder>.

How To Process and Respond?

- For all phrases: find a solution to a scope-resolved MRS

How To Process and Respond?

- For all phrases: find a solution to a scope-resolved MRS
- Propositions and Yes/No Questions: “Files are large”
  - True: “Yes, I agree!”
  - False: “No, that isn’t true”

How To Process and Respond?

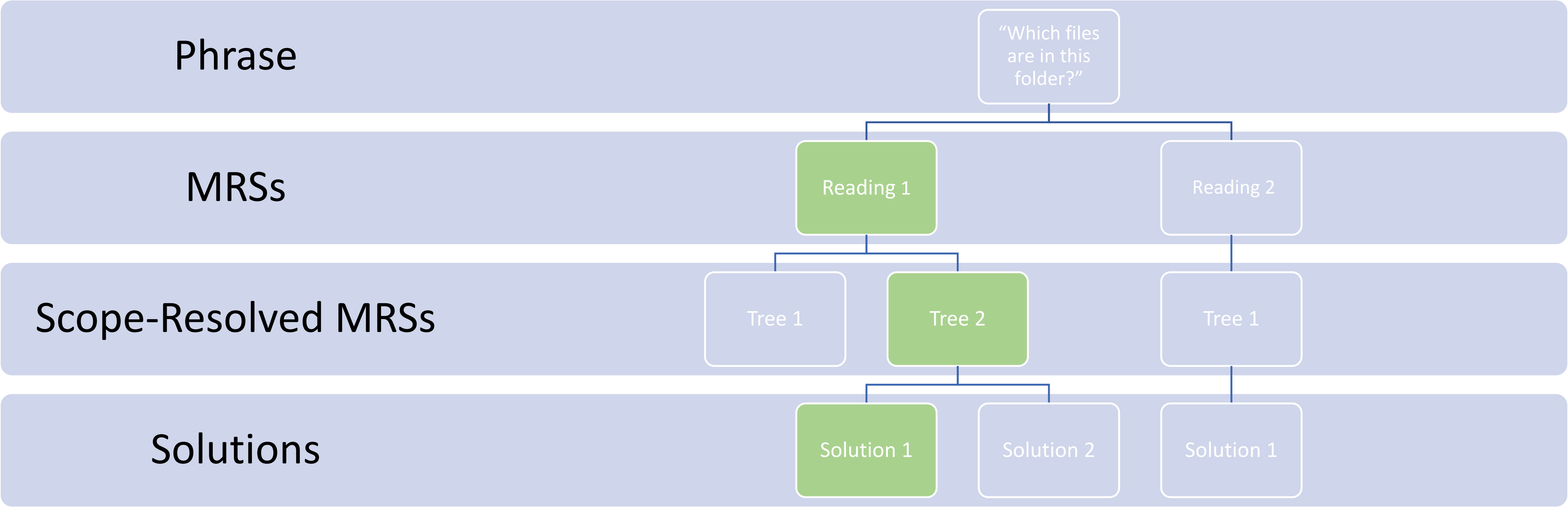
- For all phrases: find a solution to a scope-resolved MRS
- Propositions and Yes/No Questions: “Files are large”
  - True: “Yes, I agree!”
  - False: “No, that isn’t true”
- WH-Questions: “Which files are in this folder?”
  - Find the variable scoped by which_q(x, RSTR, BODY)
  - Print the values of x from the (arbitrarily first) solution

How To Process and Respond?

- For all phrases: find a solution to a scope-resolved MRS
- Propositions and Yes/No Questions: “Files are large”
    - True: “Yes, I agree!”
    - False: “No, that isn’t true”
- WH-Questions: “Which files are in this folder?”
    - Find the variable scoped by which_q(x, RSTR, BODY)
    - Print the values of x from the (arbitrarily first) solution
- Commands: “Delete <file>”
    - Perform operations* generated from the (arbitrarily first) solution

  * “operations” are extra data added to state by action verbs

Choosing the “Right” Solution



Phrase

“Which files are in this folder?”

MRSs

Reading 1

Reading 2

Scope-Resolved MRSs

Tree 1

Tree 2

Tree 1

Solutions

Solution 1

Solution 2

Solution 1

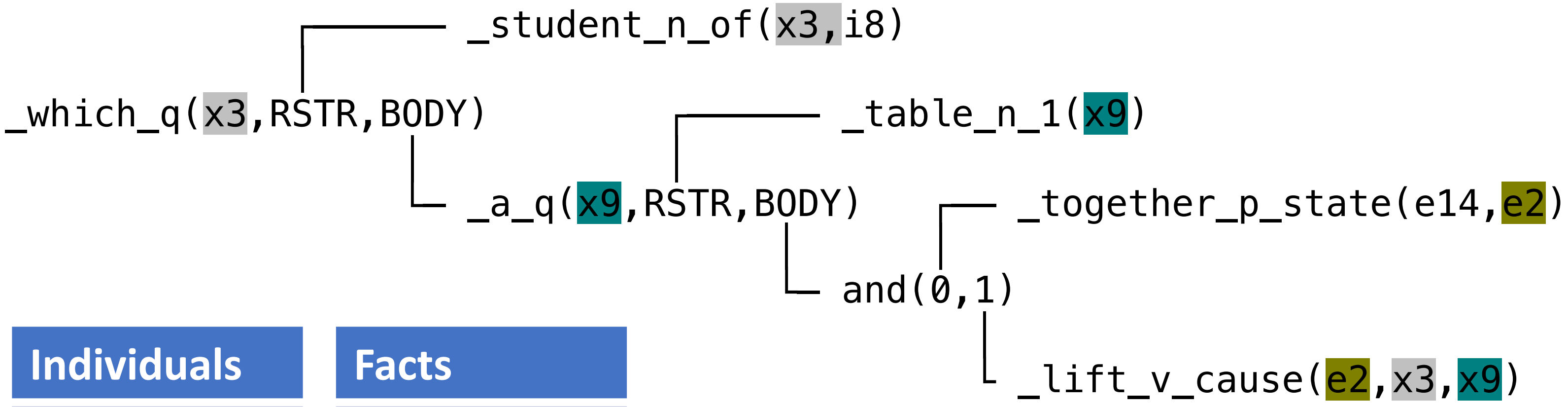
Heuristic: Respond using first solution that is found

… If no solution, respond with first failure

Solutions to Scope-Resolved MRS

“Which students are lifting a table together?”

- X variables always contain sets
- E variables are always a dictionary (name/value pairs)
- Scopal arguments form the shape of the tree



```
              ┌───── _student_n_of(x3,i8)
              │
_which_q(x3,RSTR,BODY)             ┌───── _table_n_1(x9)
                   │               │
                   └── _a_q(x9,RSTR,BODY)       ┌── _together_p_state(e14,e2)
                                        │       │
                                        └── and(0,1)
                                                  │
                                                  └── _lift_v_cause(e2,x3,x9)
```

Individuals
Amir
Wan
table1

Facts
Amir and Wan are lifting table1 at the same time

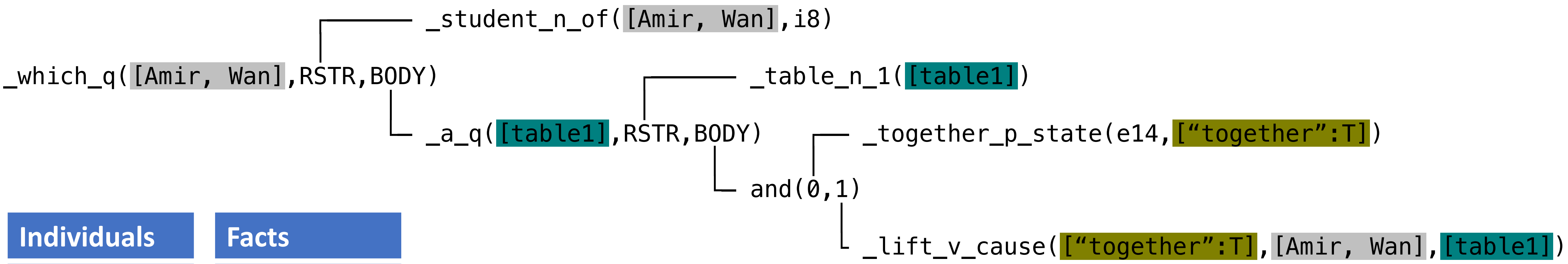
```
x3   [ x PERS: 3 NUM: pl IND: + ]
x9   [ x PERS: 3 NUM: sg IND: + ]
E2   [ e SF: ques TENSE: pres
         MOOD: indicative PROG: + PERF: - ]
```

Solutions to Scope-Resolved MRS

“Which students are lifting a table together?”

- X variables always contain sets
- E variables are always a dictionary (name/value pairs)
- Scopal arguments form the shape of the tree

Solution	
x3	[Amir, Wan]
x9	[table1]
e14	[“together”:T]



```
_which_q([Amir, Wan],RSTR,BODY)
    _student_n_of([Amir, Wan],i8)
    _a_q([table1],RSTR,BODY)
        _table_n_1([table1])
        and(0,1)
            _together_p_state(e14,[“together”:T])
            _lift_v_cause([“together”:T],[Amir, Wan],[table1])
```

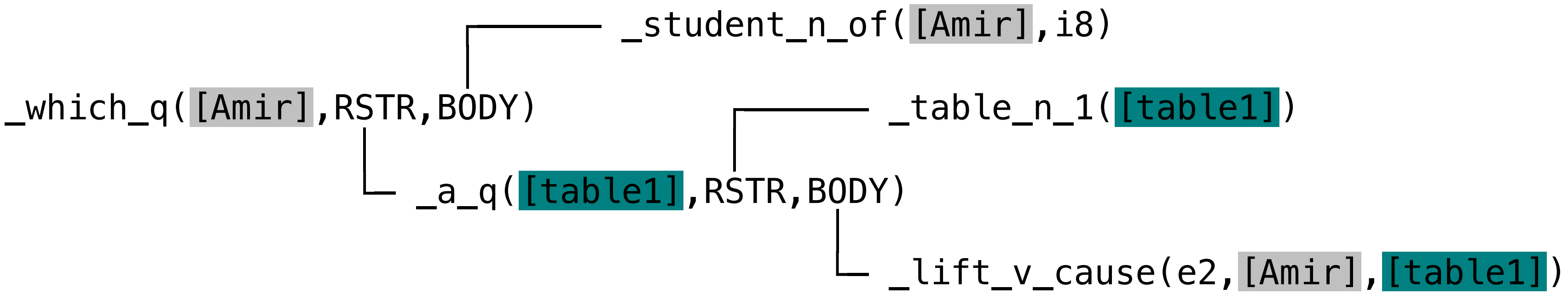
Individuals
Amir
Wan
table1

Facts
Amir and Wan are lifting table1 at the same time

```
x3   [ x PERS: 3 NUM: pl IND: + ]
x9   [ x PERS: 3 NUM: sg IND: + ]
E2   [ e SF: ques TENSE: pres
         MOOD: indicative PROG: + PERF: - ]
```

Solutions to Scope-Resolved MRS:

“Which students are lifting a table?”



```
                                          _student_n_of([Amir],i8)
_which_q([Amir],RSTR,BODY)
                                                            _table_n_1([table1])
                            _a_q([table1],RSTR,BODY)
                                                            _lift_v_cause(e2,[Amir],[table1])
```

Solution	
x3	[Amir]
x9	[table1]

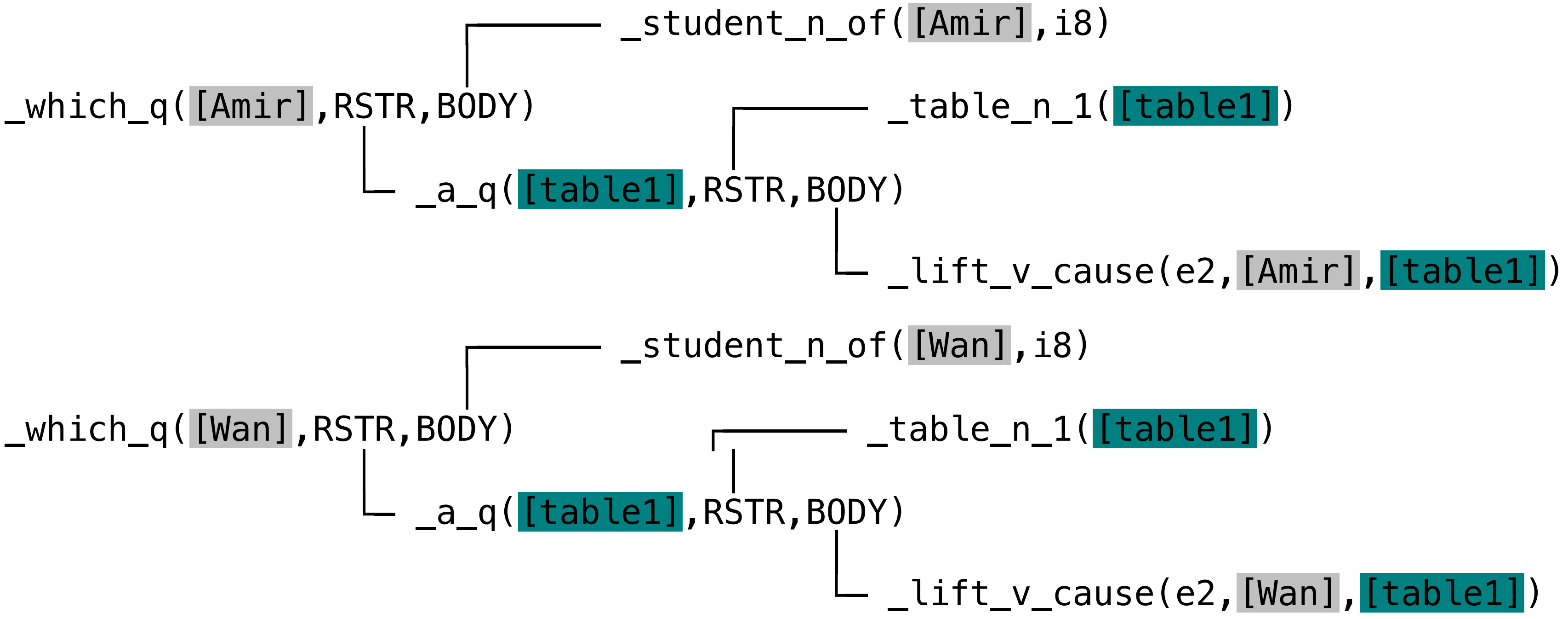
Individuals
Amir
Wan
table1

Facts
Amir and Wan are lifting table1 at the same time

```
x3  [ x PERS: 3 NUM: pl IND: + ]
x9  [ x PERS: 3 NUM: sg IND: + ]
E2  [ e SF: ques TENSE: pres
        MOOD: indicative PROG: + PERF: - ]
```

Solutions to Scope-Resolved MRS:

“Which students are lifting a table?”



```
                              _student_n_of([Amir],i8)
_which_q([Amir],RSTR,BODY)
                                           _table_n_1([table1])
                   _a_q([table1],RSTR,BODY)
                                           _lift_v_cause(e2,[Amir],[table1])
                              _student_n_of([Wan],i8)
_which_q([Wan],RSTR,BODY)
                                           _table_n_1([table1])
                   _a_q([table1],RSTR,BODY)
                                           _lift_v_cause(e2,[Wan],[table1])
```

Solution Group

Solution	
x3	[Amir]
x9	[table1]

Solution 2	
x3	[Wan]
x9	[table1]

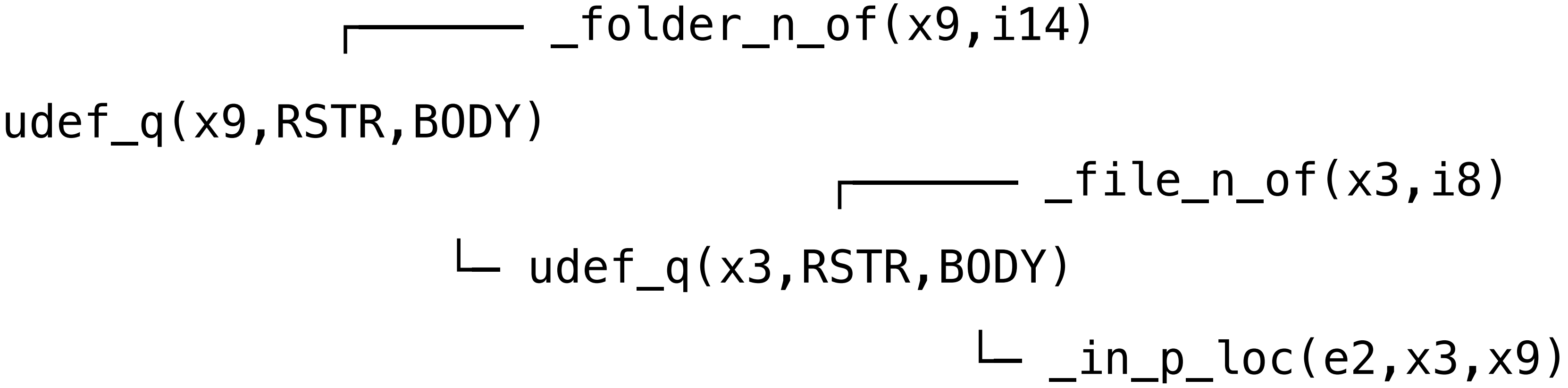
Individuals
Amir
Wan
table1

Facts
Amir and Wan are lifting table1 at the same time

```
x3  [ x PERS: 3 NUM: pl IND: + ]
x9  [ x PERS: 3 NUM: sg IND: + ]
E2  [ e SF: ques TENSE: pres
        MOOD: indicative PROG: + PERF: - ]
```

Finding Solutions to Scope-Resolved MRS:

A Multi-phase Backtracking Approach



```
                    ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                                   ┌────── _file_n_of(x3,i8)
                        └─ udef_q(x3,RSTR,BODY)
                                        └─ _in_p_loc(e2,x3,x9)
```

- Do a depth-first traversal of the tree
- Quantifiers try every individual in their scoped variable iteratively

Finding Solutions to Scope-Resolved MRS:

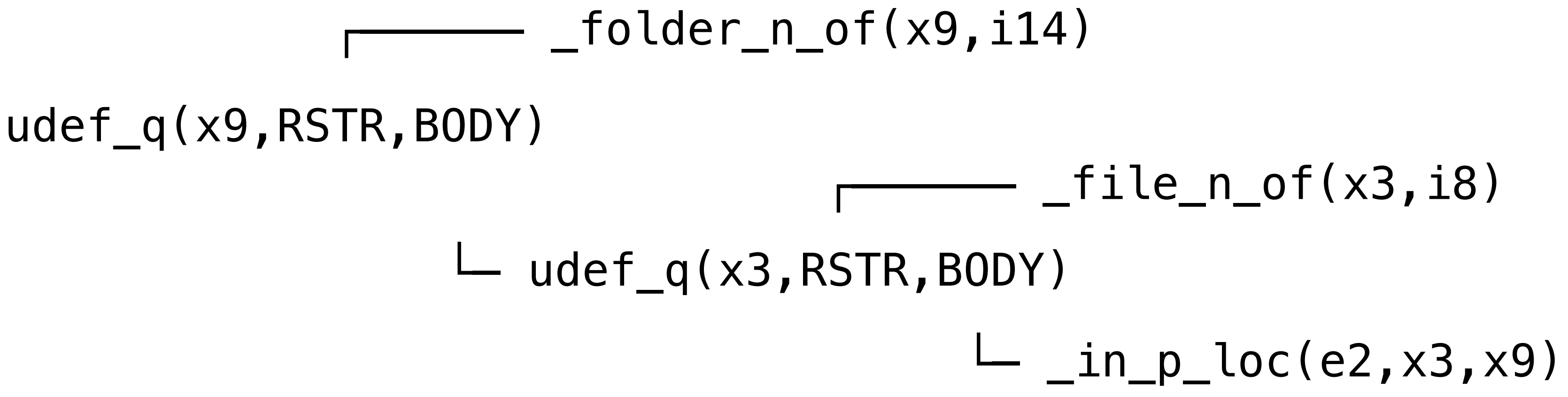
A Multi-phase Backtracking Approach

```
            ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                               ┌────── _file_n_of(x3,i8)
                 └─ udef_q(x3,RSTR,BODY)
                                    └─ _in_p_loc(e2,x3,x9)
```

- Do a depth-first traversal of the tree
- Quantifiers try every individual in their scoped variable iteratively
- Failures backtrack to the nearest quantifier and retry next individual

Finding Solutions to Scope-Resolved MRS:

A Multi-phase Backtracking Approach



```
                 ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                                  ┌────── _file_n_of(x3,i8)
                     └─ udef_q(x3,RSTR,BODY)
                                        └─ _in_p_loc(e2,x3,x9)
```

- Do a depth-first traversal of the tree
- Quantifiers try every individual in their scoped variable iteratively
- Failures backtrack to the nearest quantifier and retry next individual
- When all predications are true, we have a solution

Finding Solutions to Scope-Resolved MRS:

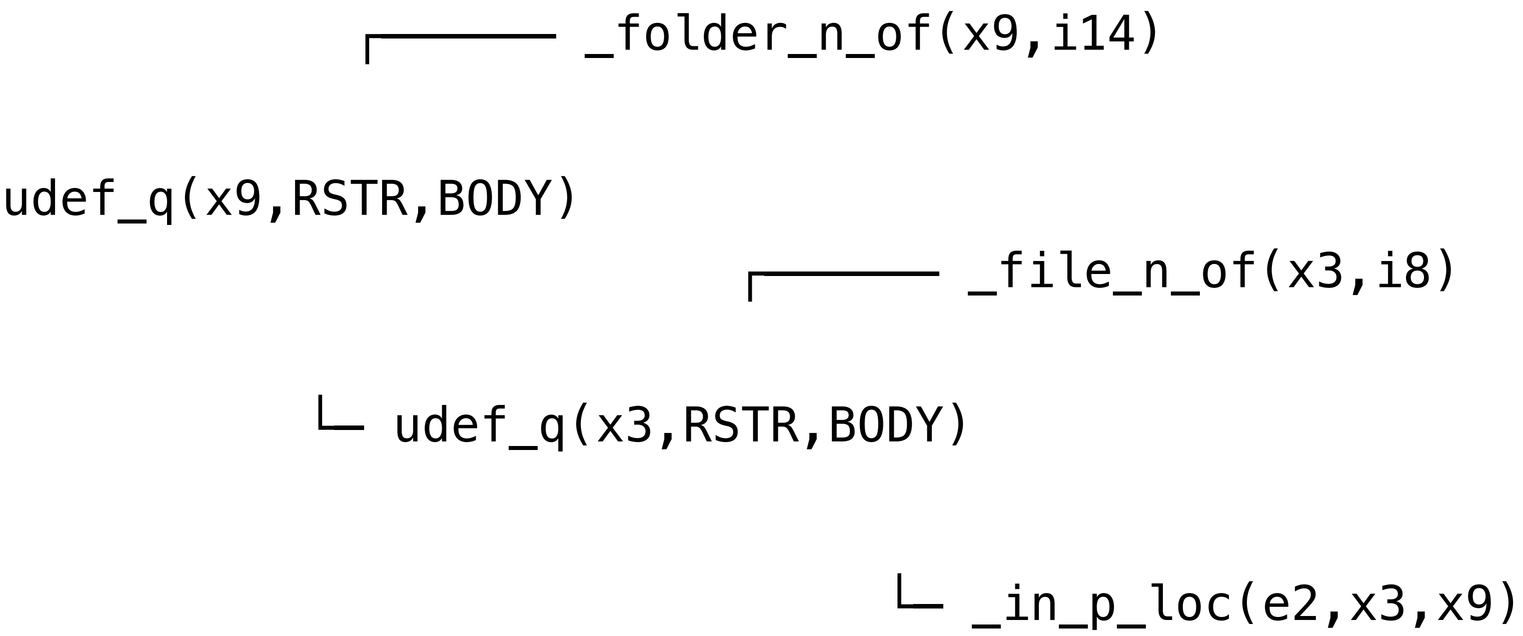
A Multi-phase Backtracking Approach

```
             ┌───── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                               ┌───── _file_n_of(x3,i8)
                └─ udef_q(x3,RSTR,BODY)
                                   └─ _in_p_loc(e2,x3,x9)
```

- Do a depth-first traversal of the tree
- Quantifiers try every individual in their scoped variable iteratively
- Failures backtrack to the nearest quantifier and retry next individual
- When all predications are true: a **solution**
- Global semantics (like plural counting) done in a second phase: a **solution group**

**many optimizations are available, but this is the base model*

Solving a Scope-Resolved MRS: Phase 1



_folder_n_of(x9,i14)

udef_q(x9,RSTR,BODY)

_file_n_of(x3,i8)

udef_q(x3,RSTR,BODY)

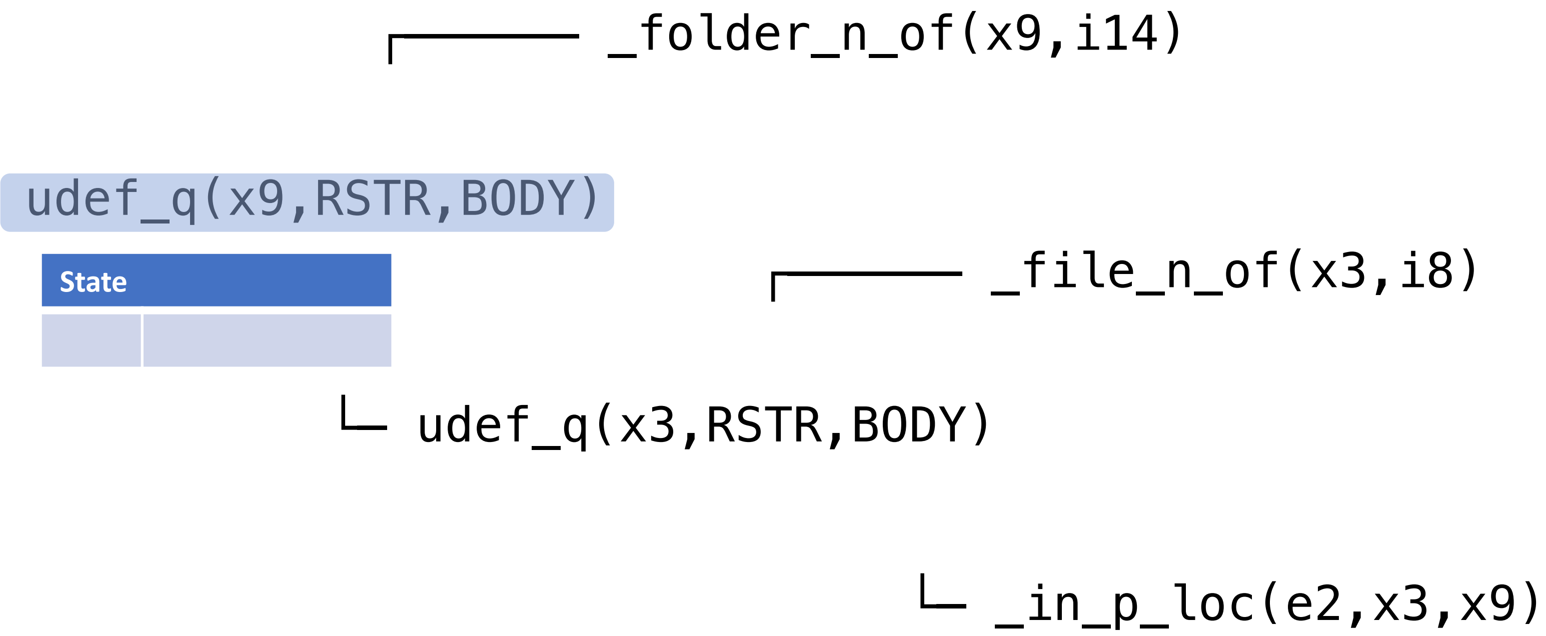
_in_p_loc(e2,x3,x9)

State	

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

x3 [x PERS: 3 NUM: pl IND: +]
x9 [x PERS: 3 NUM: sg IND: +]

Solving a Scope-Resolved MRS: Phase 1

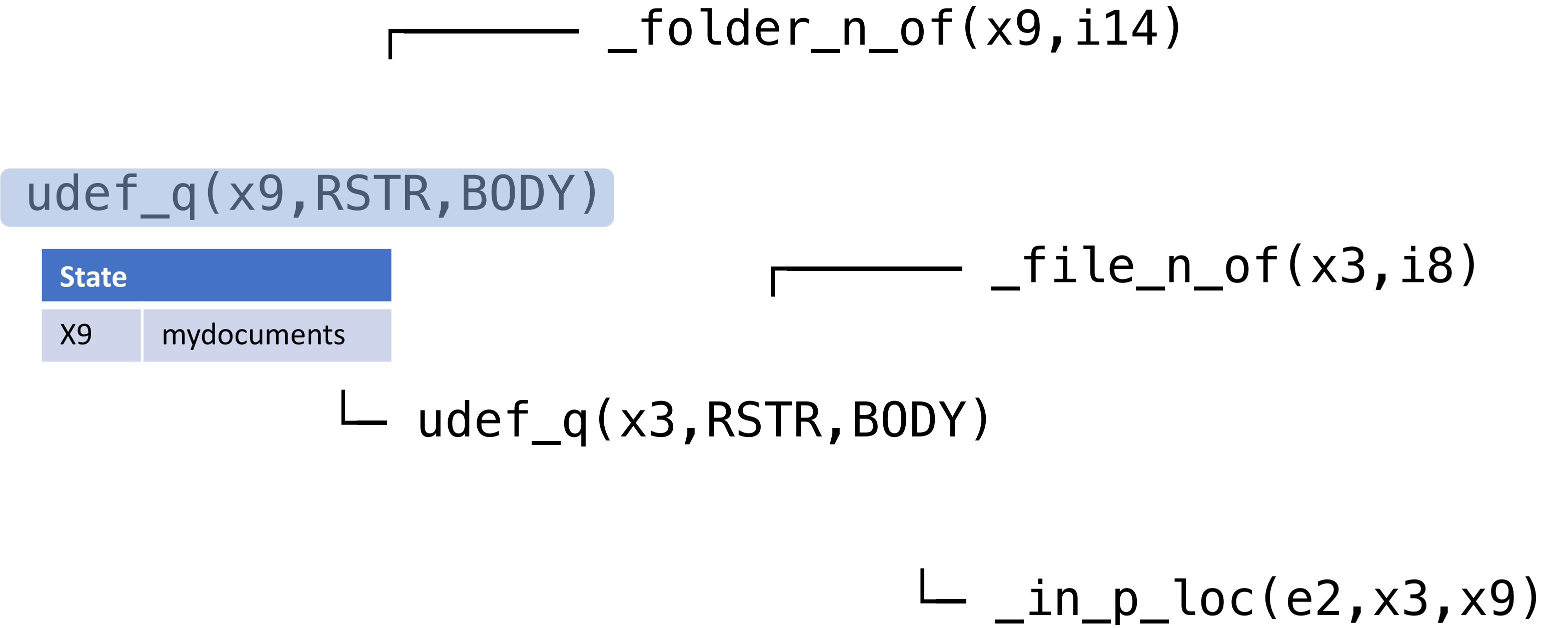


```
                    ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                              ┌──── _file_n_of(x3,i8)
                    └─ udef_q(x3,RSTR,BODY)
                                    └─ _in_p_loc(e2,x3,x9)
```

State	

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
            ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                          ┌────── _file_n_of(x3,i8)
            └─ udef_q(x3,RSTR,BODY)
                               └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1

```
                 ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                            ┌──── _file_n_of(x3,i8)
                └─ udef_q(x3,RSTR,BODY)
                                    └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments

All Individuals
mydocuments
File1.txt
File2.txt

Facts
mydocuments contains File1.txt and File2.txt

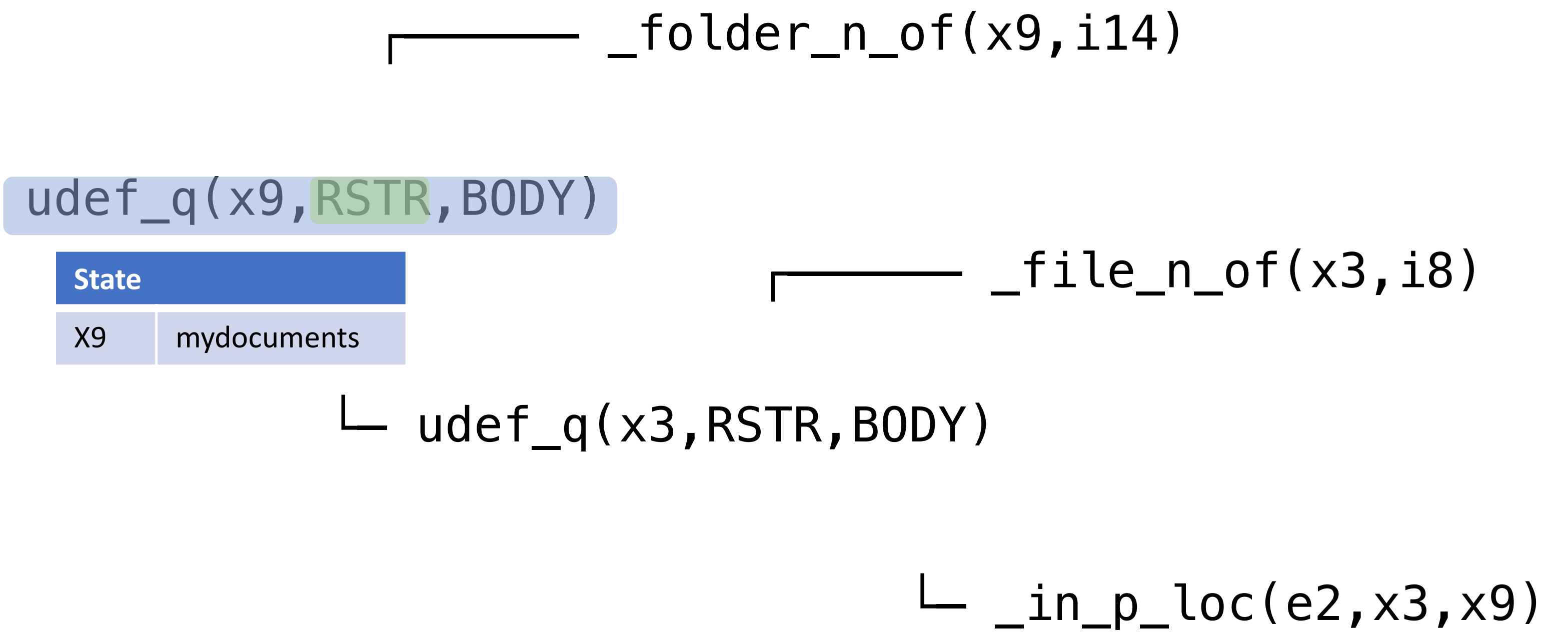
Solving a Scope-Resolved MRS: Phase 1

```
                  ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                             ┌────── _file_n_of(x3,i8)
                └─ udef_q(x3,RSTR,BODY)
                                      └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
                 ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                                    ┌──── _file_n_of(x3,i8)
                 └─ udef_q(x3,RSTR,BODY)
                                         └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1

```
           ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                    ┌──── _file_n_of(x3,i8)
           └─ udef_q(x3,RSTR,BODY)
                          └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments

All Individuals
mydocuments
File1.txt
File2.txt

Facts
mydocuments contains File1.txt and File2.txt

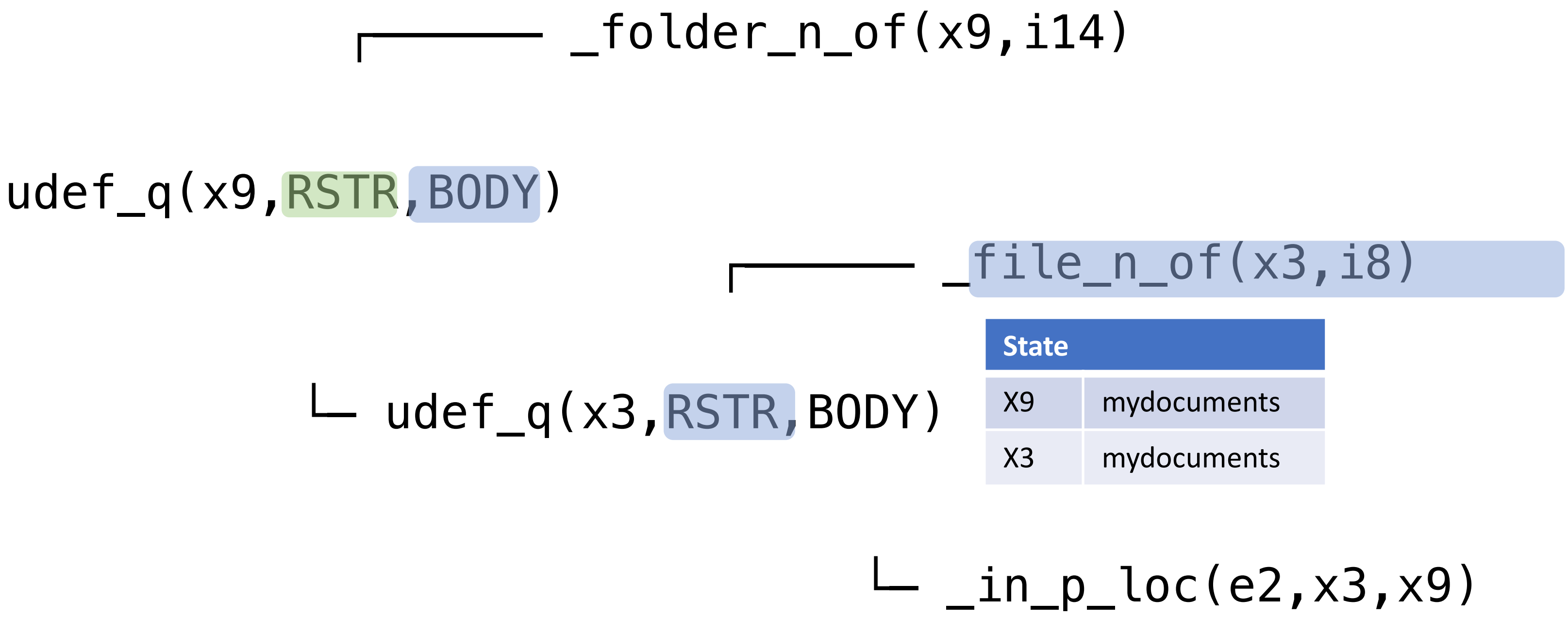
Solving a Scope-Resolved MRS: Phase 1

```
                    ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                              ┌────── _file_n_of(x3,i8)
                    └─ udef_q(x3,RSTR,BODY)
                                    └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
               ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                        ┌──── _file_n_of(x3,i8)
          └─ udef_q(x3,RSTR,BODY)
                        └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	mydocuments

All Individuals
mydocuments
File1.txt
File2.txt

Facts
mydocuments contains File1.txt and File2.txt

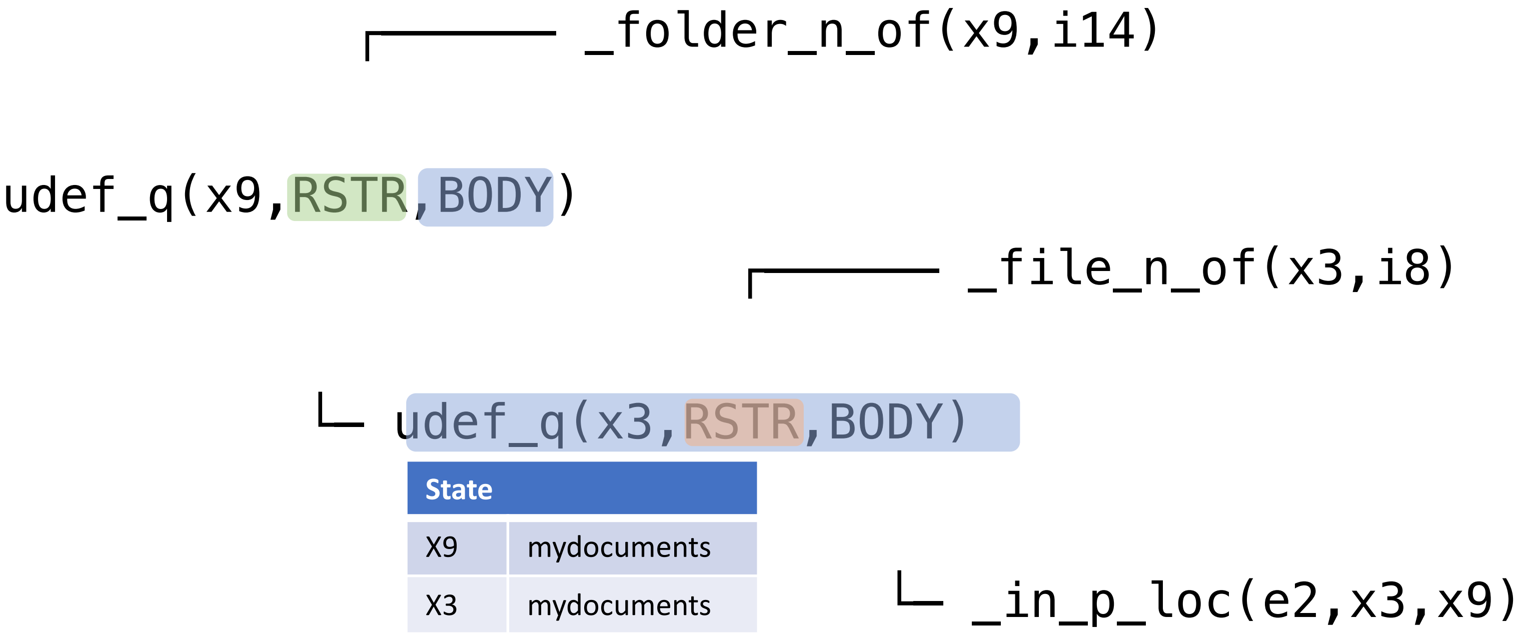
Solving a Scope-Resolved MRS: Phase 1

```
                  ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                            ┌──── _file_n_of(x3,i8)
                 └─ udef_q(x3,RSTR,BODY)
                                      └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
          ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                     ┌──── _file_n_of(x3,i8)
          └─ udef_q(x3,RSTR,BODY)
                     └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	mydocuments

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

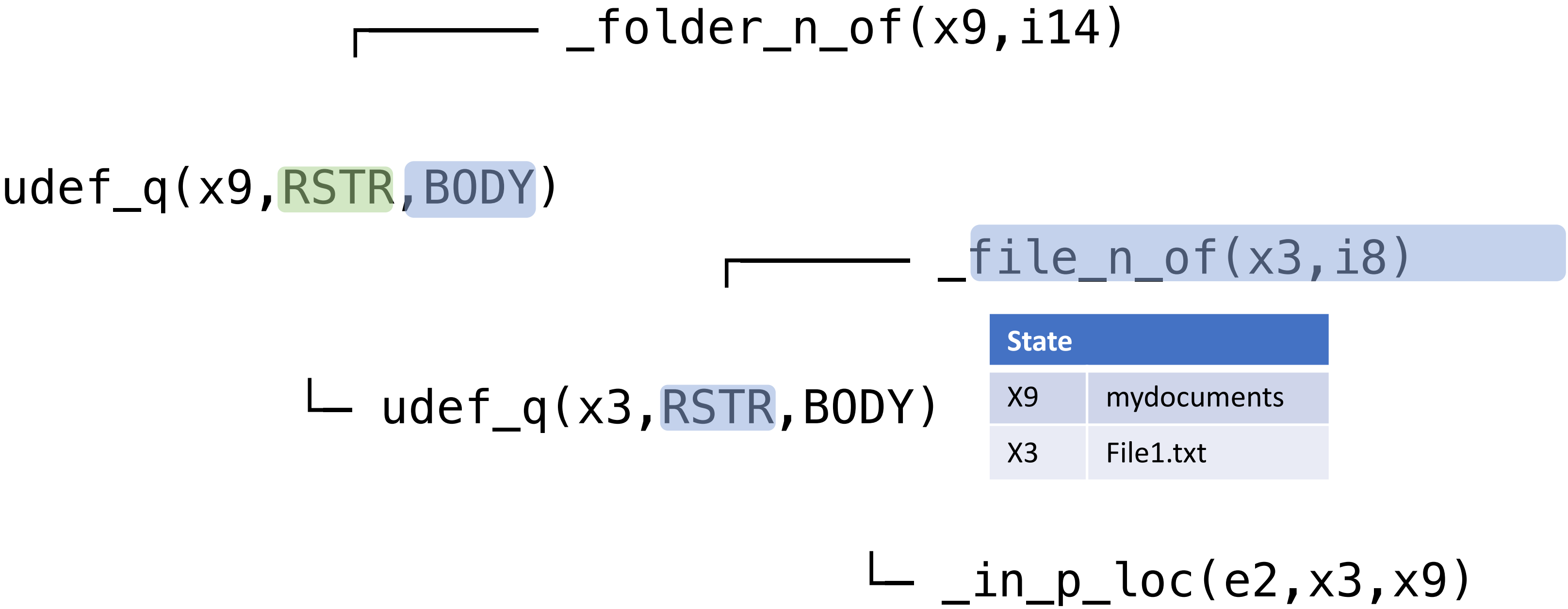
Solving a Scope-Resolved MRS: Phase 1

```
                  ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                              ┌──── _file_n_of(x3,i8)
                └─ udef_q(x3,RSTR,BODY)
                                   └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1

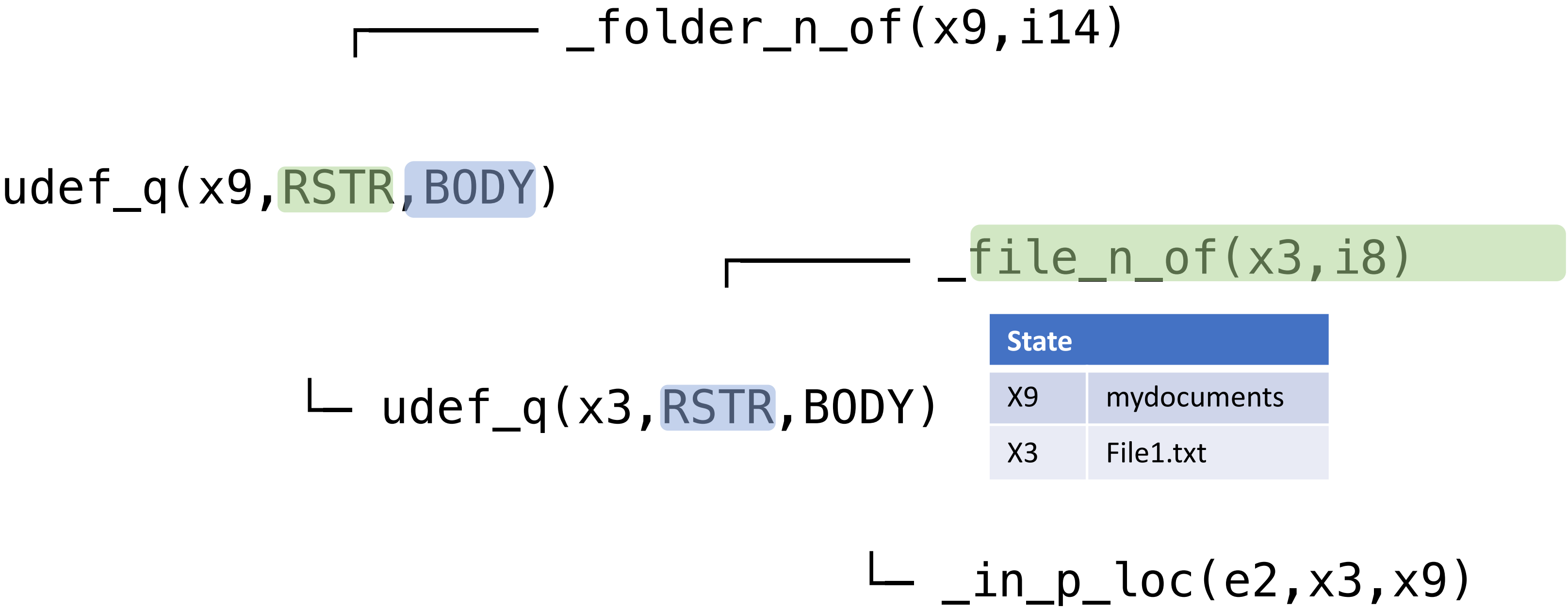


```
          ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                   ┌──── _file_n_of(x3,i8)
          └─ udef_q(x3,RSTR,BODY)
                          └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
               ┌──── _folder_n_of(x9,i14)

udef_q(x9,RSTR,BODY)
                      ┌──── _file_n_of(x3,i8)

           └─ udef_q(x3,RSTR,BODY)

                                 └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

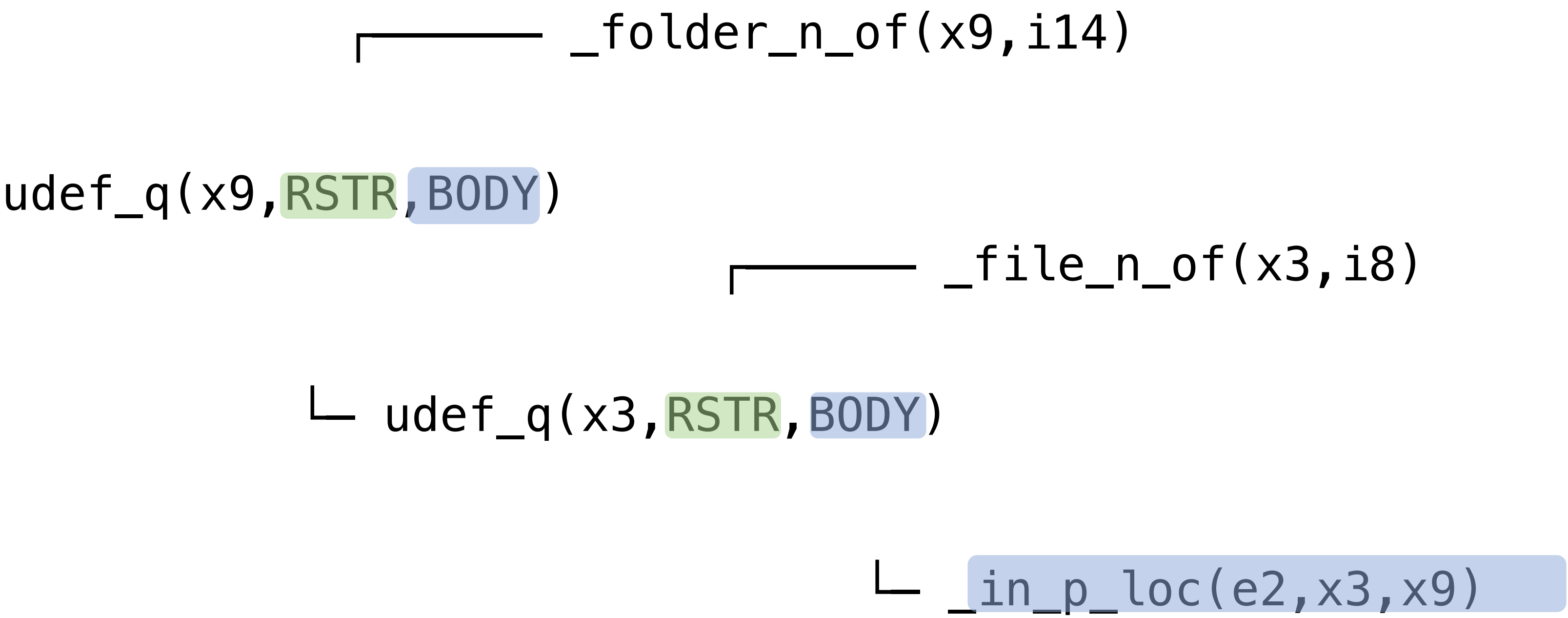
udef_q(x9,RSTR,BODY)

```
                ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                       ┌──── _file_n_of(x3,i8)
               └─ udef_q(x3,RSTR,BODY)
                            └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

udef_q(x9,RSTR,BODY)

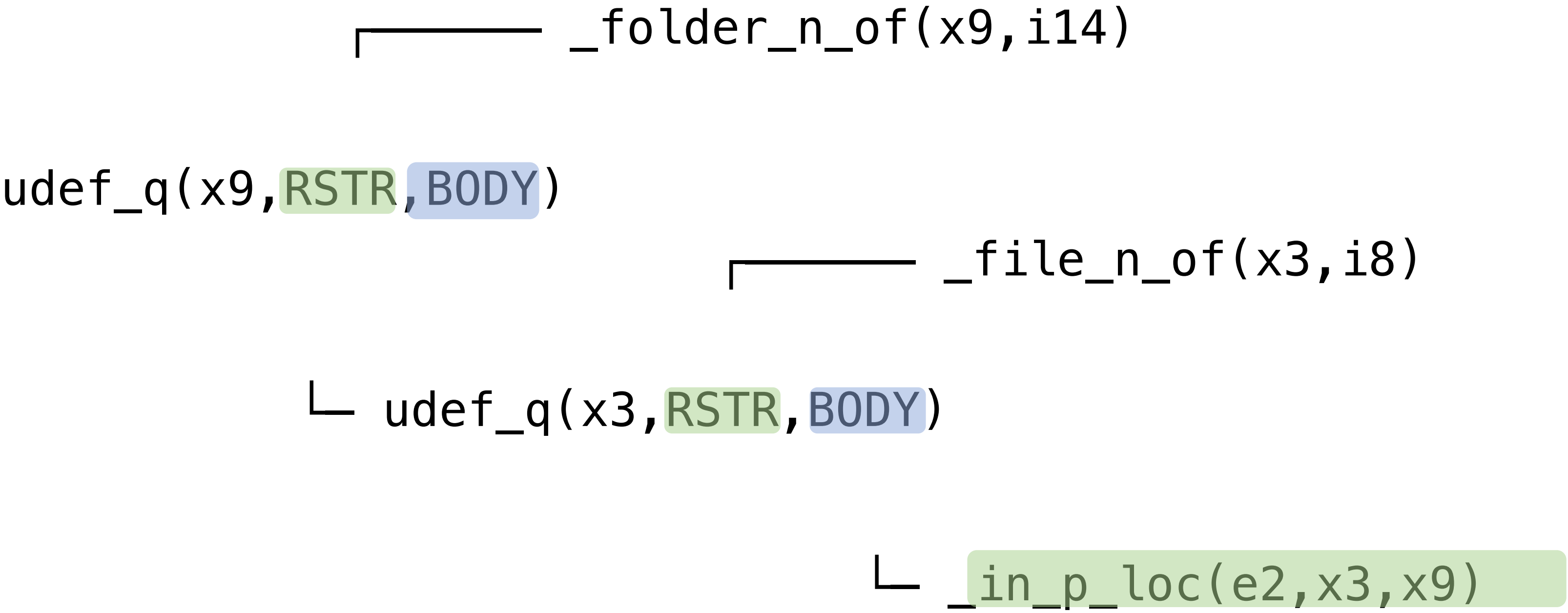


```
          ┌──────  _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                        ┌──────  _file_n_of(x3,i8)
          └─ udef_q(x3,RSTR,BODY)
                             └─ _in_p_loc(e2,x3,x9)
```

State	
X9	Mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
          ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                     ┌──── _file_n_of(x3,i8)
        └ udef_q(x3,RSTR,BODY)
                         └ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals
mydocuments
File1.txt
File2.txt

Facts
mydocuments contains File1.txt and File2.txt

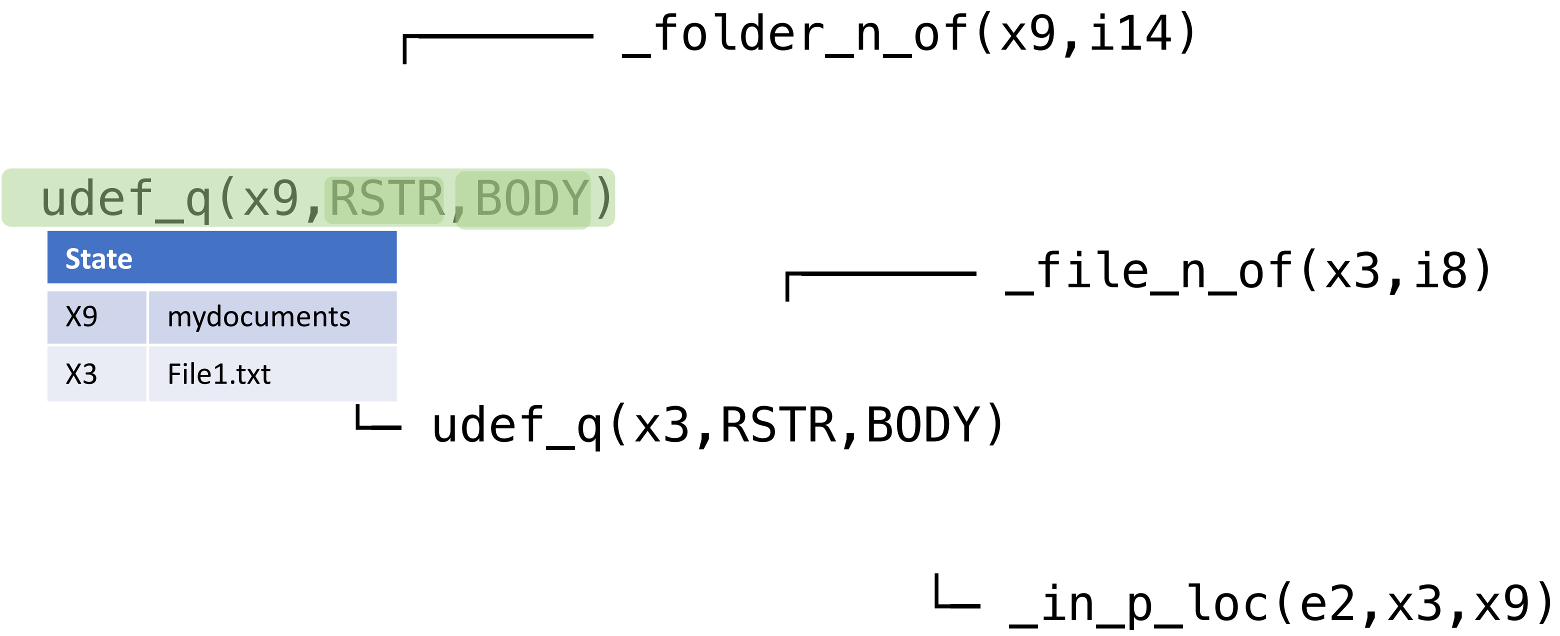
Solving a Scope-Resolved MRS: Phase 1

```
                  ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                              ┌────── _file_n_of(x3,i8)
                └─ udef_q(x3,RSTR,BODY)
                              └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
➡ File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1



```
            ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                        ┌──── _file_n_of(x3,i8)
            └─ udef_q(x3,RSTR,BODY)
                           └─ _in_p_loc(e2,x3,x9)
```

State	
X9	mydocuments
X3	File1.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1

Solution 1!

State	
X9	mydocuments
X3	File1.txt

```
          ┌──── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                     ┌──── _file_n_of(x3,i8)
         └─ udef_q(x3,RSTR,BODY)
                                └─ _in_p_loc(e2,x3,x9)
```

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Solving a Scope-Resolved MRS: Phase 1

```
                    ┌────── _folder_n_of(x9,i14)
udef_q(x9,RSTR,BODY)
                              ┌────── _file_n_of(x3,i8)
                   └─ udef_q(x3,RSTR,BODY)
                                   └─ _in_p_loc(e2,x3,x9)
```

Solution 1!

State	
X9	mydocuments
X3	File1.txt

Solution 2!

State	
X9	mydocuments
X3	File2.txt

All Individuals	Facts
mydocuments	mydocuments contains File1.txt and File2.txt
File1.txt	
File2.txt	

Handling Plurals

- Both files (x3) and folders (x9) are plural
- Plurals, counting and other global (i.e. cross-solution) constraints are handled *after* solutions are found in Phase 2

Phase 2

All Solutions:

	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

Start with the full set of solutions

Phase 2

All Solutions:

	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

Group Being Tested:

	X9	X3
Solution 1	Mydocuments	File1.txt

Examine every combination of solutions to find those that meet constraints

Phase 2

All Solutions:

	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

Group Being Tested:

X9 greater than 1

X3 greater than 1

	X9	X3
Solution 1	Mydocuments	File1.txt

Phase 2

All Solutions:

	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

Group Being Tested:

X9 greater than 1

X3 greater than 1

	X9	X3
Solution 2	Mydocuments	File2.txt

Phase 2

All Solutions:

	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

Group Being Tested:

	X9 greater than 1	X3 greater than 1
	X9	X3
Solution 1	Mydocuments	File1.txt
Solution 2	Mydocuments	File2.txt

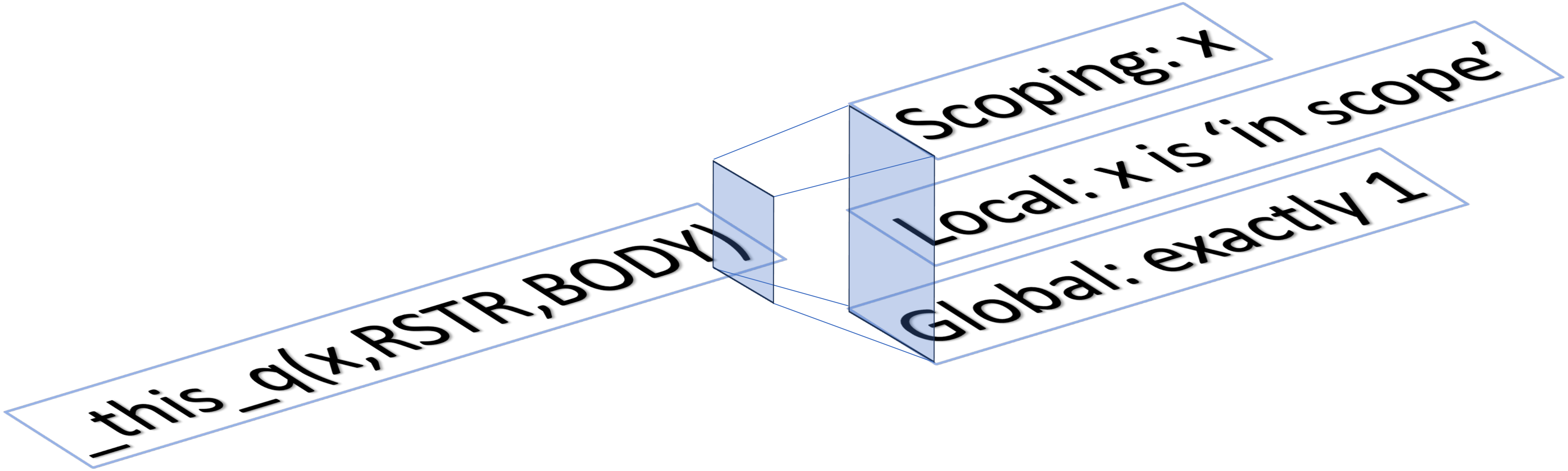
Response to “files are in folders”

- “files are not in folders”
  - … because “folders” is plural and no solution groups have > 1 folder
- Successful Groups
  - Create *solution groups* that meet the constraints
  - Solution groups are the actual solution to the scope-resolved MRS

Predication “Semantic Layers”

_this_q(x,RSTR,BODY)

Predication “Semantic Layers”



_this_q(x,RSTR,BODY)

Scoping: x

Local: x is ‘in scope’

Global: exactly 1

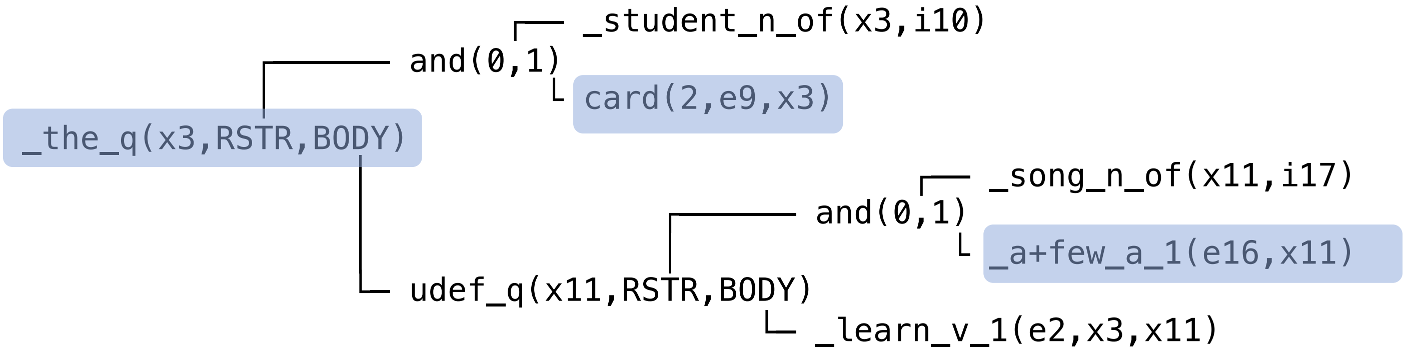
Predication “Semantic Layers”

Predications can contribute any combination of:
scoping, local or global constraints

Predicate	Scoping	Local	Global
_large_a_1(e,x)	<none>	True for “large” x	<none>
udef_q(x,RSTR,BODY)	x	<none>	<none>
_a_q(x,RSTR,BODY)	x	<none>	Exactly 1
_the_q(x,RSTR,BODY)* *… one of several meanings	x	<none>	1 or more Where all rstr satisfy the body
_this_q(x,RSTR,BODY)	x	True if x is “in scope”* *… among other meanings	Exactly 1
card(CARG,e,x)	<none>	<none>	At least CARG
a_few_a_1(e,x)	<none>	<none>	Between 3 and 5* *top value is debatable
and_c(x,x1,x2)	<none>	<none>	Exactly N where N is total of x1 and x2

Phase 1: Solve After Removing Global Constraints

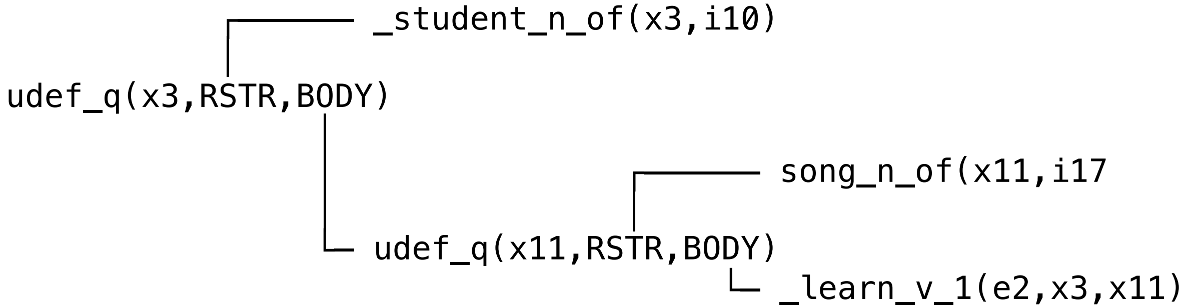
“The two students learned a few songs”



```
                                 ┌─ _student_n_of(x3,i10)
                  and(0,1)
                                 └ card(2,e9,x3)
_the_q(x3,RSTR,BODY)
                                              ┌─ _song_n_of(x11,i17)
                                 and(0,1)
                                              └ _a+few_a_1(e16,x11)
                  udef_q(x11,RSTR,BODY)
                                 └─ _learn_v_1(e2,x3,x11)
```

MRS: x3 [NUM: pl] and x11 [NUM: pl]

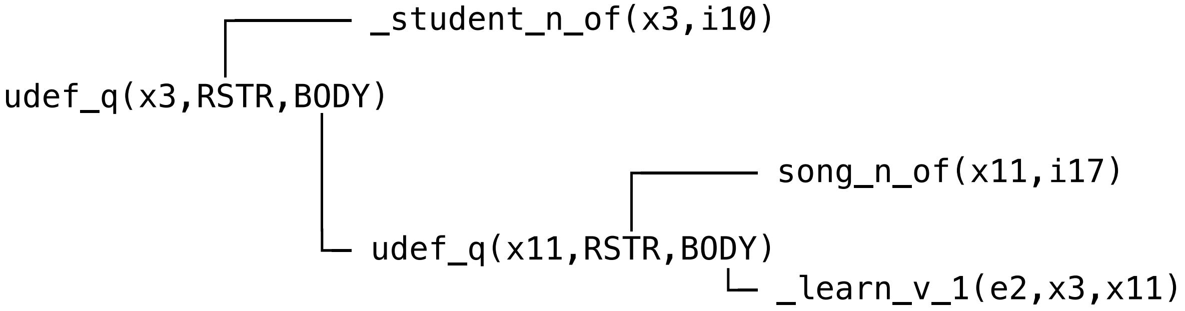
”student(s) learned song(s)”



```
                  _student_n_of(x3,i10)
udef_q(x3,RSTR,BODY)
                                    song_n_of(x11,i17
                  udef_q(x11,RSTR,BODY)
                                 └─ _learn_v_1(e2,x3,x11)
```

Phase 1: Solve After Removing Global Constraints

” student(s) learned song(s)”



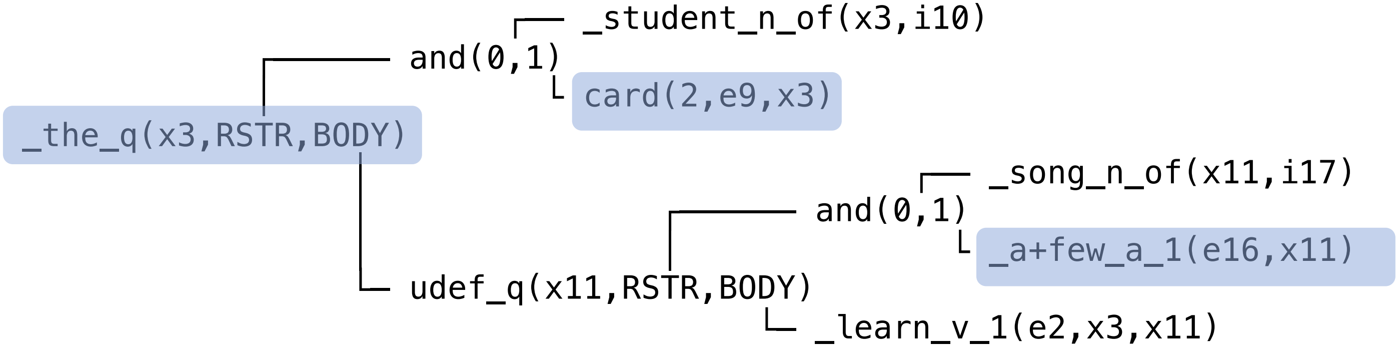
```
                 ┌────── _student_n_of(x3,i10)
udef_q(x3,RSTR,BODY)
                    │                   ┌────── song_n_of(x11,i17)
                    └─ udef_q(x11,RSTR,BODY)
                                           └─ _learn_v_1(e2,x3,x11)
```

All Solutions:

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Phase 2: Find Global Constraints For Variables

”The two students learned a few songs”

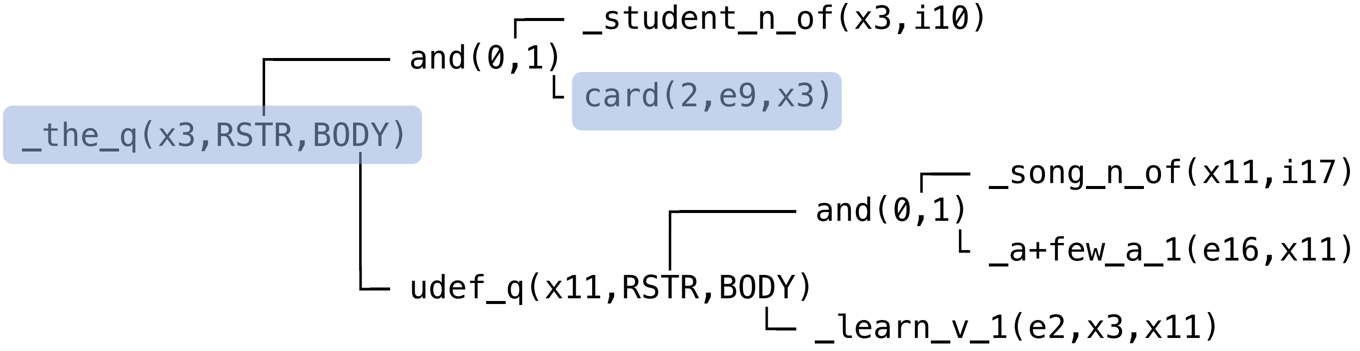


```
                        ┌── _student_n_of(x3,i10)
           ┌──── and(0,1)
           │            └ card(2,e9,x3)
_the_q(x3,RSTR,BODY)
              │                          ┌── _song_n_of(x11,i17)
              │              ┌──── and(0,1)
              │              │          └ _a+few_a_1(e16,x11)
              └── udef_q(x11,RSTR,BODY)
                                  └── _learn_v_1(e2,x3,x11)
```

MRS: x3 [NUM: pl] and x11 [NUM: pl]

Phase 2: Find Global Constraints For Variables

”The two students learned a few songs”



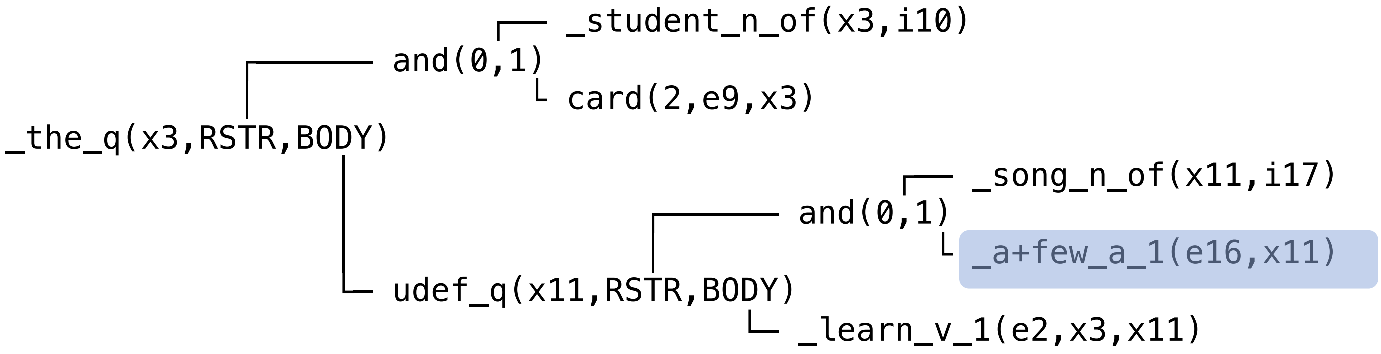
```
                         ┌── _student_n_of(x3,i10)
           ┌──── and(0,1)
           │             └ card(2,e9,x3)
_the_q(x3,RSTR,BODY)
                │                          ┌── _song_n_of(x11,i17)
                │             ┌──── and(0,1)
                │             │            └ _a+few_a_1(e16,x11)
                └─ udef_q(x11,RSTR,BODY)
                                     └─ _learn_v_1(e2,x3,x11)
```

MRS: x3 [NUM: pl] and x11 [NUM: pl]

X3 Predications	Constraints
card(2,e9,x3)	Min=2, Max=inf
_the_q(x3,RSTR,BODY)	Min=1, Max=inf and all_rstr_meet_body
x3 [NUM: pl]	Min=2, Max=inf
	Min=2, Max=inf and all_rstr_meet_body

Phase 2: Find Global Constraints For Variables

”The two students learned a few songs”



```
                               ┌── _student_n_of(x3,i10)
               ┌────── and(0,1)
               │              └ card(2,e9,x3)
_the_q(x3,RSTR,BODY)
                    │                        ┌── _song_n_of(x11,i17)
                    │          ┌────── and(0,1)
                    │          │             └ _a+few_a_1(e16,x11)
                    └─ udef_q(x11,RSTR,BODY)
                                         └─ _learn_v_1(e2,x3,x11)
```

MRS: x3 [NUM: pl] and x11 [NUM: pl]

X11 Predications	Constraints
_a+few_a_1(e16,x11)	Min=3, Max=5
x11 [NUM: pl]	Min=2, Max=inf
	Min=3, Max=5

Phase 2: Find Solution Groups that Meet Constraints

”The two students learned a few songs”

All Solutions:

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Group Being Tested:

X3: Min=2, Max=inf and all_rstr_meet_body

X11: Min=3, Max=5

	X3	X11
Solution 1	Diya	”Words Get in the Way”

*All other single solution groups fail for the same reason, not shown

Phase 2: Find Solution Groups that Meet Constraints

”The two students learned a few songs”

All Solutions:

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Group Being Tested:

X3: Min=2, Max=inf and all_rstr_meet_body

X11: Min=3, Max=5

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 3	Wan	“Ten Thousand Words”

*All other double solution groups fail for various reasons, not shown

Phase 2: Find Solution Groups that Meet Constraints

”The two students learned a few songs”

All Solutions:

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Group Being Tested:

X3: Min=2, Max=inf and all_rstr_meet_body

X11: Min=3, Max=5

	X3	X11
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Collective, Distributive, Cumulative

”The two students learned a few songs”

- This solution group is a **cumulative** reading
- Phase 2 checks the constraints for all three types of readings

	X3: Min=2, Max=inf and all_rstr_meet_body	X11: Min=3, Max=5
	X3	**X11**
Solution 1	Diya	”Words Get in the Way”
Solution 2	Diya	“More than Words”
Solution 3	Wan	“Ten Thousand Words”

Collective, Distributive, Cumulative Math

- Variables have an order due to forward and reverse readings
- Count *this* variable per *the previous variable*
- *How* to count depends on type
- First variable always counted the same (since no previous variable)

	X3	X11

X3: Min=2, Max=inf
and
all_rstr_meet_body

First Variable:
Count unique individuals across all x3 values

X11: Min=3, Max=5

Counting math depends on type

Distributive: “The two students sang a few songs”

- Students: must be grouped distributively into subgroups:
  - More than one subgroup
  - The total of students across the subgroups must add up to two
- Songs: Each student subgroup must be singing a few songs

	X3	X11
Solution	[Diya]	[”Words Get in the Way”]
Solution	[Diya]	[“More than Words”]
Solution	[Diya]	[“Ten Thousand Words”]
Solution	[Wan]	[“Word Up”]
Solution	[Wan]	[“Paperback Writer”]
Solution	[Wan]	[“Unwritten”]

X3: Min=2, Max=inf and all_rstr_meet_body

First Variable Counting: Count unique individuals across all x3 values

X11: Min=3, Max=5

Distributive Counting: 3-5 x11 *individuals* per x3 *set value*

Collective: “The two students sang a few songs”

- Students must be grouped collectively:
  - Exactly 1 “subgroup” that contains the entire set of students
- Songs : Each student subgroup must be singing a few songs

	X3	X11
Solution	[Diya, Wan]	[”Words Get in the Way”]
Solution	[Diya, Wan]	[“More than Words”]
Solution	[Diya, Wan]	[“Ten Thousand Words”]

	X3	X11
Solution	[Diya, Wan]	[”Words Get in the Way”, “More than Words”]
Solution	[Diya, Wan]	[“Ten Thousand Words”]

	X3	X11
Solution	[Diya, Wan]	[”Words Get in the Way”, “More than Words”, “Ten Thousand Words”]

X3: Min=2, Max=inf
and
all_rstr_meet_body

First Variable Counting: Count unique individuals across all x3 values

X11: Min=3, Max=5

Collective Counting:
- *exactly one* x3 set
- 3-5 x11 *individuals* for the one x3 *set*

Cumulative: “The two students sang a few songs”

- Students: must be grouped distributively into subgroups, which means:
  - More than one subgroup
  - The total of students across the subgroups must add up to two
- Songs: The total of songs across all subgroups must be a few

	X3	X11
Solution	[Diya]	[”Words Get in the Way”]
Solution	[Diya]	[“More than Words”]
Solution	[Wan]	[“Ten Thousand Words”]

	X3	X11
Solution	[Diya]	[”Words Get in the Way”, “More than Words”]
Solution	[Wan]	[“Ten Thousand Words”]

X3: Min=2, Max=inf and all_rstr_meet_body

First Variable Counting: Count unique individuals across all x3 values

X11: Min=3, Max=5

Cumulative Counting: 3-5 x11 *individuals* across all x3 subgroups

Summary

- Phase 0: Generate MRS and scope-resolved MRS
- Phase 1: Find solutions using scope-resolved MRS minus global constraints
- Phase 2: Find solution groups using global constraints
- Phase 3: Respond using the first solution group (or error)
    - If no solutions: “<error>”
    - Propositions and Yes/No Questions: “Files are large”
        - “Yes, I agree!”
    - WH-Questions: “Which files are in this folder?”
        - Find the variable scoped by which_q(x, RSTR, BODY)
        - Respond with the values of x from the solution group
    - Commands: “Delete <file>”
        - Perform operations* generated from the solution group

          * “operations” are extra data added to state by action verbs

*optimization: Individual solutions are pulled through the phases in a pipeline

Questions?