TECHNISCHE UNIVERSITÄT DRESDEN

International Center for Computational Logic

Alex Ivliev, Lukas Gerlach, Simon Meusel, Jakob Steinberg, Markus Krötzsch

Knowledge-Based Systems, TU Dresden

Nemo: A Scalable and Versatile Datalog Engine

Datalog 2.0, 11.10.2024

Introducing Nemo

- Datalog engine written in Rust
- Free and open source
- More than 10 contributors

Design Goals

Introducing Nemo

- Datalog engine written in Rust
- Free and open source
- More than 10 contributors

Nemo
Graph Rule Engine

Design Goals

Scalability

Introducing Nemo

- Datalog engine written in Rust
- Free and open source
- More than 10 contributors

Nemo
Graph Rule Engine

Design Goals

Scalability

Versatility

Introducing Nemo

- Datalog engine written in Rust
- Free and open source
- More than 10 contributors

Nemo
Graph Rule Engine

Design Goals

Scalability

Versatility

Usablity

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1   head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3   @import data :- csv {resource="data.csv"}.
4   @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6   age("Alice", 24).
7
8   result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10  message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12  negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14  summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16  human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1  head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3  @import data :- csv {resource="data.csv"}.
4  @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6  age("Alice", 24).
7
8  result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10 message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12 negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14 summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16 human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1   head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3   @import data :- csv {resource="data.csv"}.
4   @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6   age("Alice", 24).
7
8   result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10  message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12  negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14  summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16  human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1   head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3   @import data :- csv {resource="data.csv"}.
4   @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6   age("Alice", 24).
7
8   result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10  message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12  negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14  summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16  human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1   head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3   @import data :- csv {resource="data.csv"}.
4   @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6   age("Alice", 24).
7
8   result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10  message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12  negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14  summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16  human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1   head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3   @import data :- csv {resource="data.csv"}.
4   @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6   age("Alice", 24).
7
8   result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10  message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12  negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14  summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16  human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

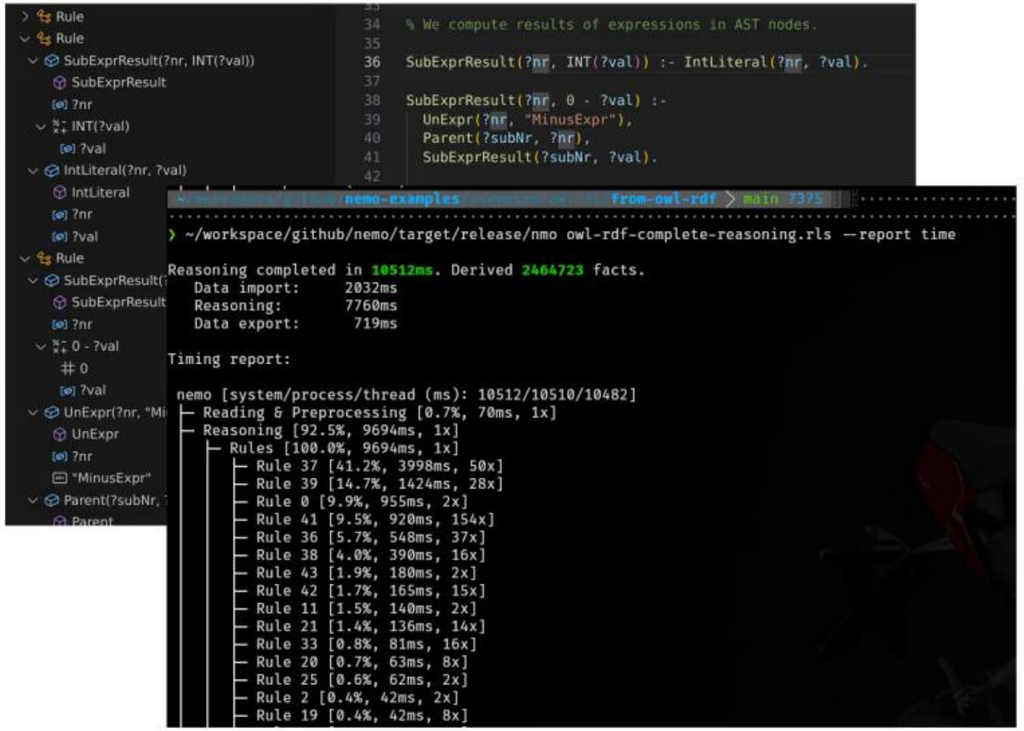
```
1  head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3  @import data :- csv {resource="data.csv"}.
4  @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6  age("Alice", 24).
7
8  result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10 message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12 negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14 summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16 human(!P), parent(?H, !P) :- human(?H) .
```

Language Features

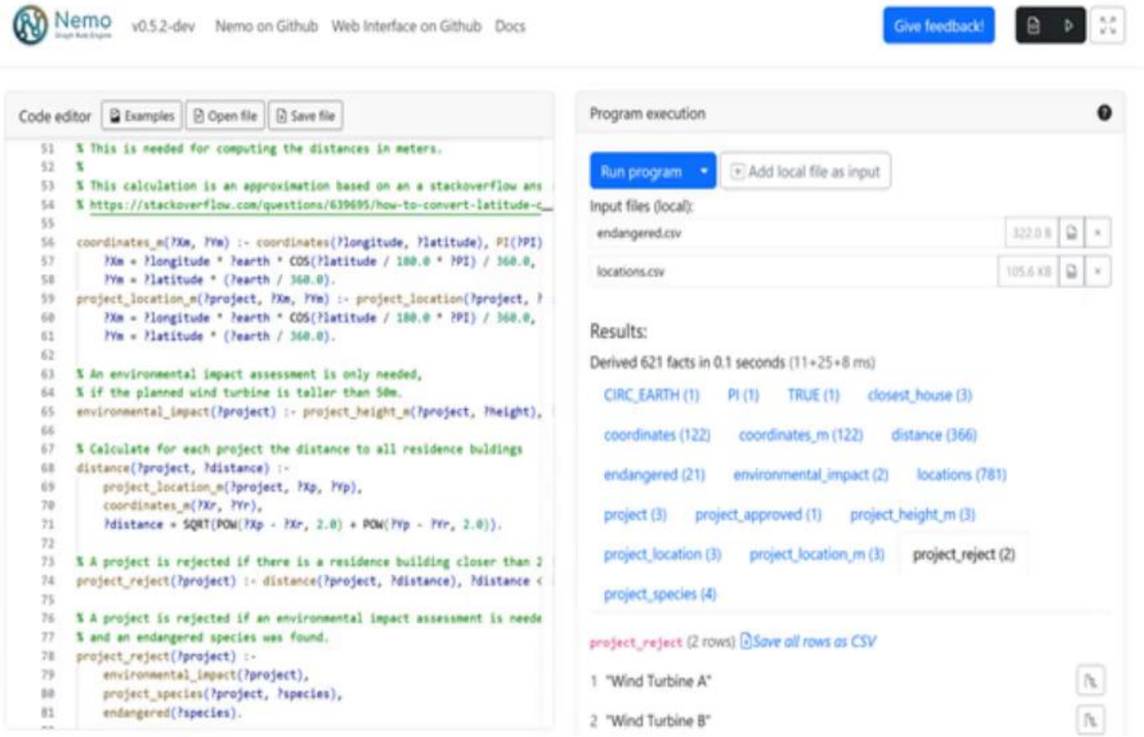
- Basis: Recursive Datalog
- Import/Export (CSV, TSV, N-Triples, Turtle)
- Datatypes
- Arithmetic Expressions
- Builtin Functions
- Stratified Negation
- Stratified Aggregation
- Value Invention

```
1  head(?X, ?Z) :- body(?X, ?Y), head(?Y, ?Z).
2
3  @import data :- csv {resource="data.csv"}.
4  @import web :- rdf {resource="http://abc.com/data.ttl"}.
5
6  age("Alice", 24).
7
8  result(?X + ?Y) :- data(?X, ?Y), ?X < ?Y.
9
10 message(CONCAT("Result: ", STR(?X))) :- result(?X).
11
12 negation(?X) :- data(?X, ?Y), ~result(?Y).
13
14 summed(?X, #sum(?Y)) :- data(?X, ?Y).
15
16 human(!P), parent(?H, !P) :- human(?H) .
```

The Nemo Toolkit



VSCode Extension and CLI



Web Application

User Feedback

```
[202] Error: unsafe variable used in rule head: `?projekt`
    ┌─[test.rls:77:22]
    │
 77 │ environmental_impact(?projekt) :- project_height_m(?project, ?height), ?height >= 50.
    │                      ─────┬─
    │                           └── unsafe variable used in rule head: `?projekt`
    │
    │ Help: a variable with a similar name exists: `project`
    │
    │ Note: every universal variable in the head must occur at a safe position in the body
    └────
```

Nemo Demo: Wind Turbines

Rules for placing wind turbines

- Minimal distance from buildings
- If certain height: no endangered species

Architecture

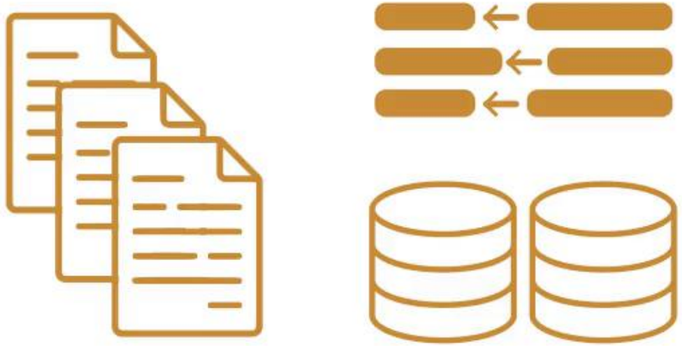
Physical

- Query Engine
- Dictionary
- Data Structures

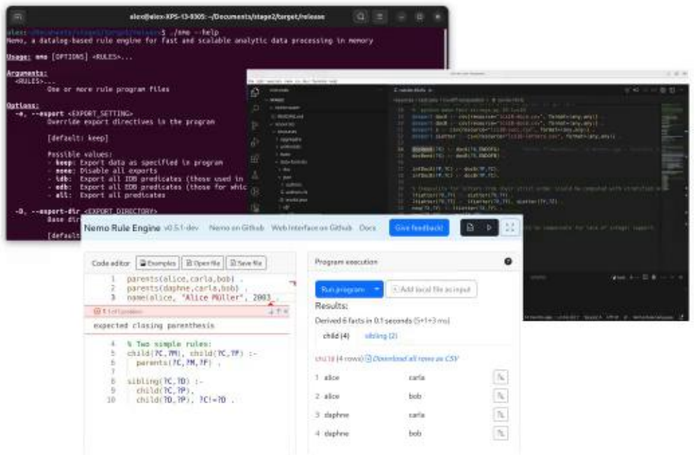
- Operations

Logical

- Planner
- Execution Engine



Application



Data structure

Eve	Chuck	Alice
Eve	Alice	Bob
Alice	Chuck	David
Eve	Frank	Chuck
Alice	Chuck	Bob
Alice	Bob	David
Eve	Bob	Chuck

Data structure

5	3	1
5	1	2
1	3	4
5	6	3
1	3	2
1	2	4
5	2	3

Data structure

5	3	1
5	1	2
1	3	4
5	6	3
1	3	2
1	2	4
5	2	3

Alice → 1

Bob → 2

Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure

1	2	4
1	3	2
1	3	4
5	1	2
5	2	3
5	3	1
5	6	3

Alice → 1

Bob → 2

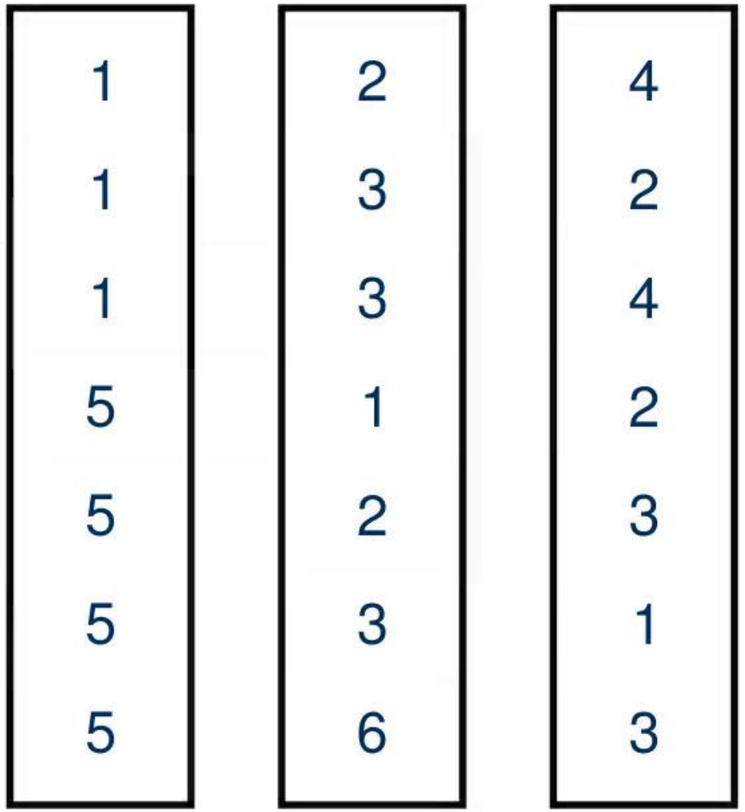
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



1	2	4
1	3	2
1	3	4
5	1	2
5	2	3
5	3	1
5	6	3

Alice → 1

Bob → 2

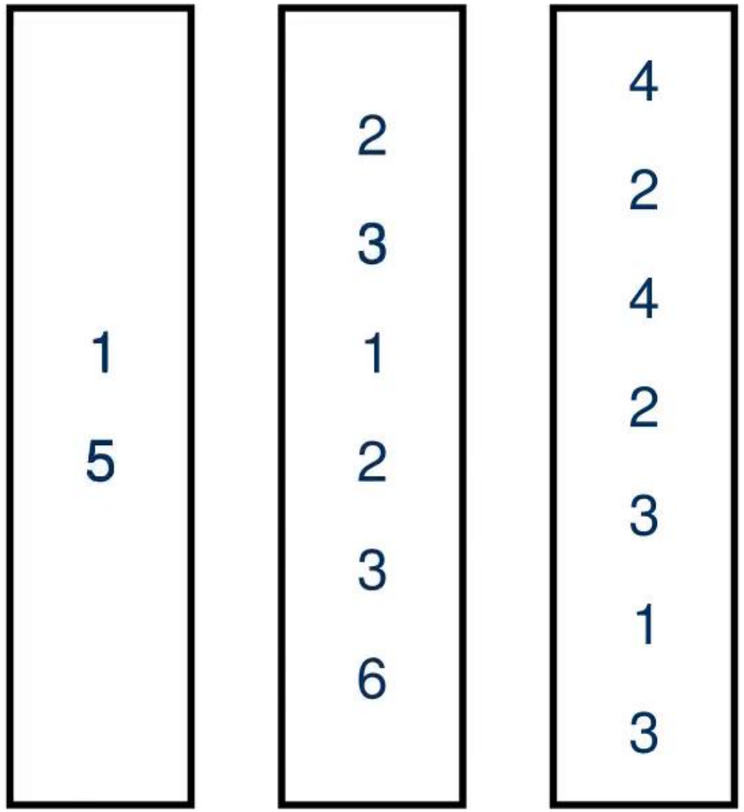
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

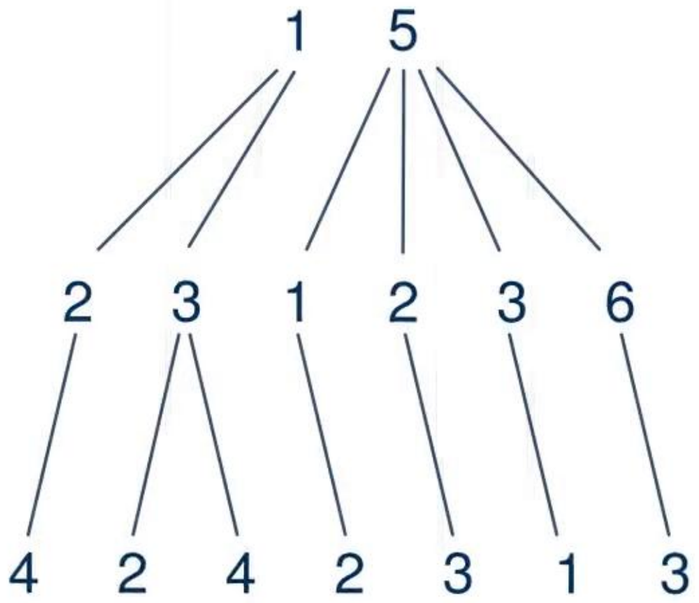
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

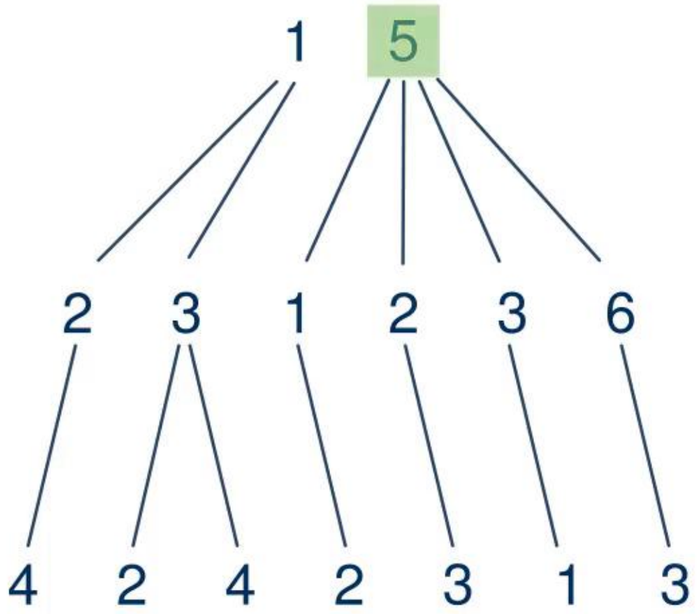
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

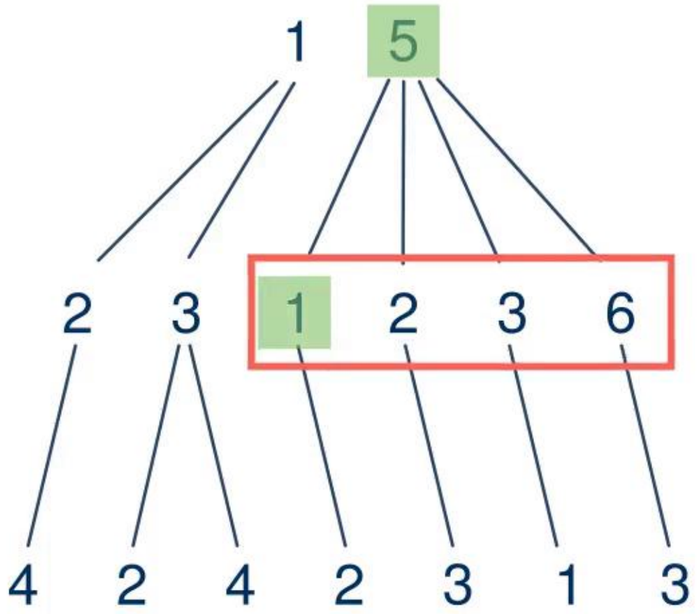
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

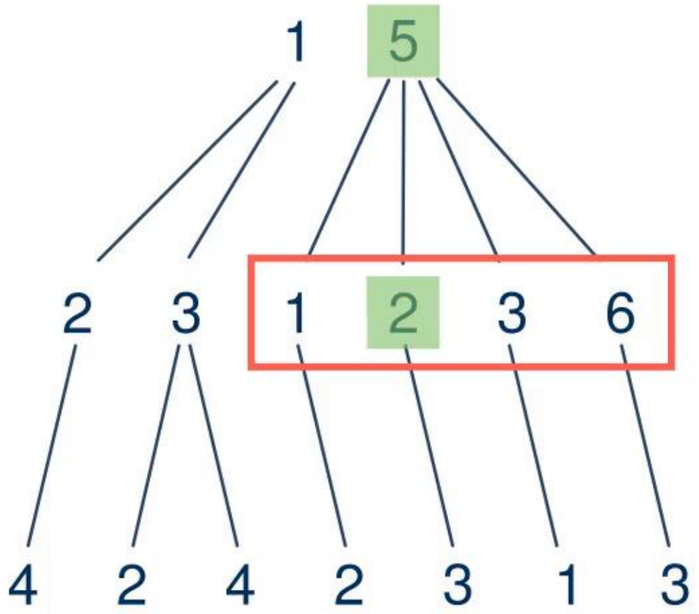
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

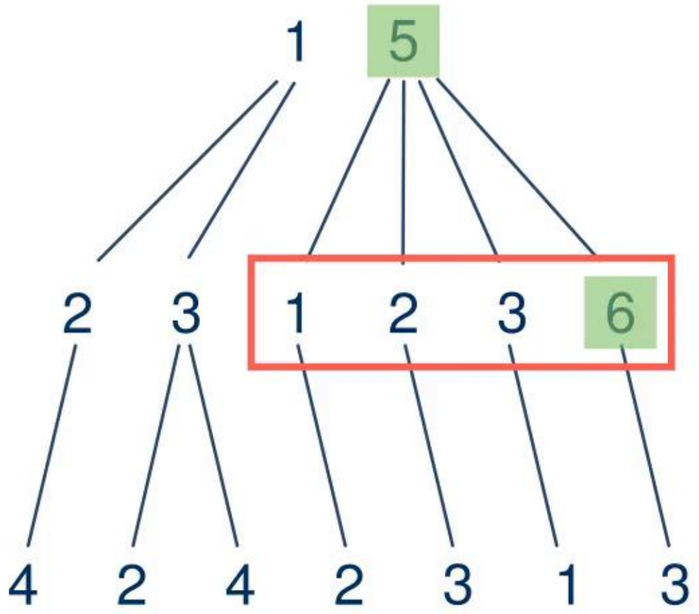
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

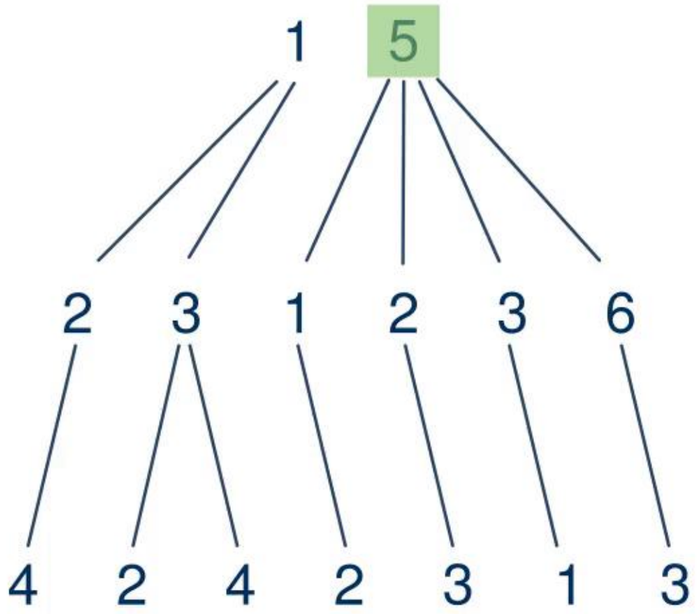
Chuck → 3

David → 4

Eve → 5

Frank → 6

Data structure



Alice → 1

Bob → 2

Chuck → 3

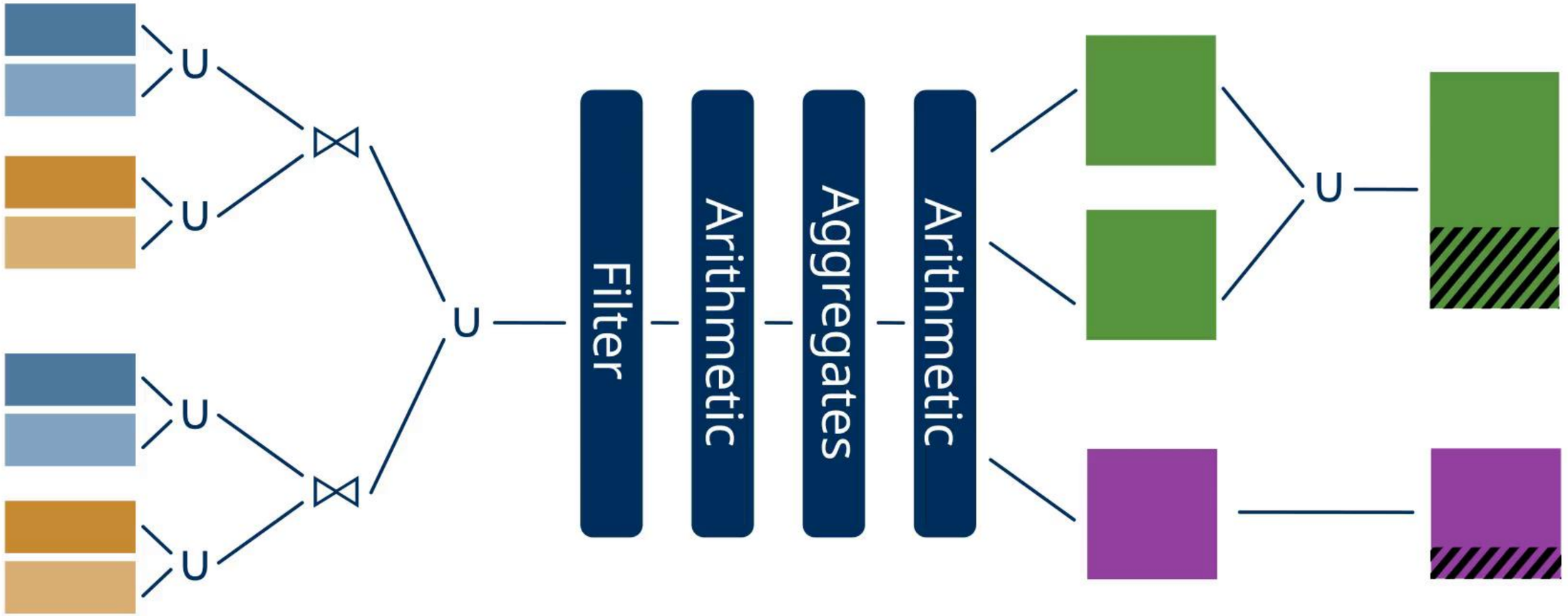
David → 4

Eve → 5

Frank → 6

Pipeline

a(x, y), b(y, x), y > 2x, z = x+y, a = #min(z), r = a/2 ➔ p(x, r), p(y, r), q(z)



Performance Evaluation

Compared Engines

VLog

Runtime

- EL Reasoning on medical ontologies
- Existential Rule Benchmarks: Benchmarking the Chase

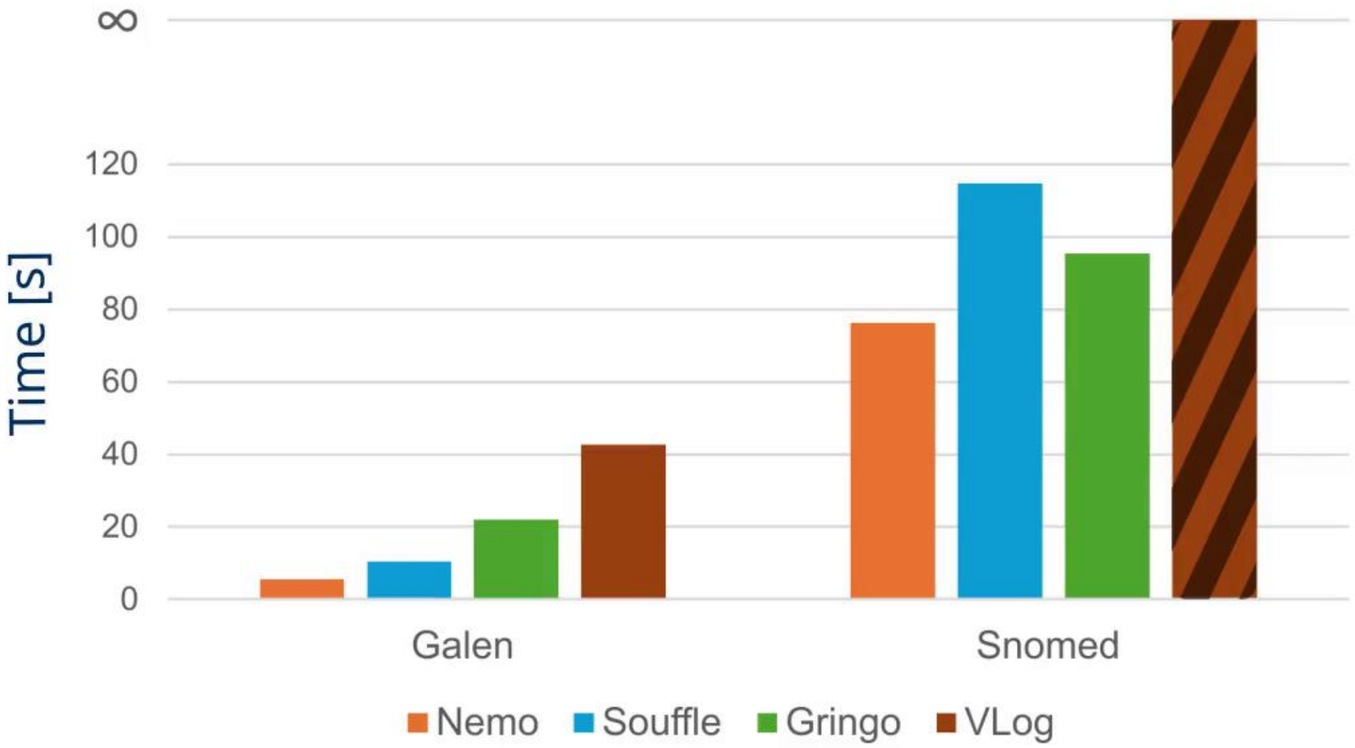
Scalability

- Reasoning Task on Wikidata

Evaluation: EL Reasoning

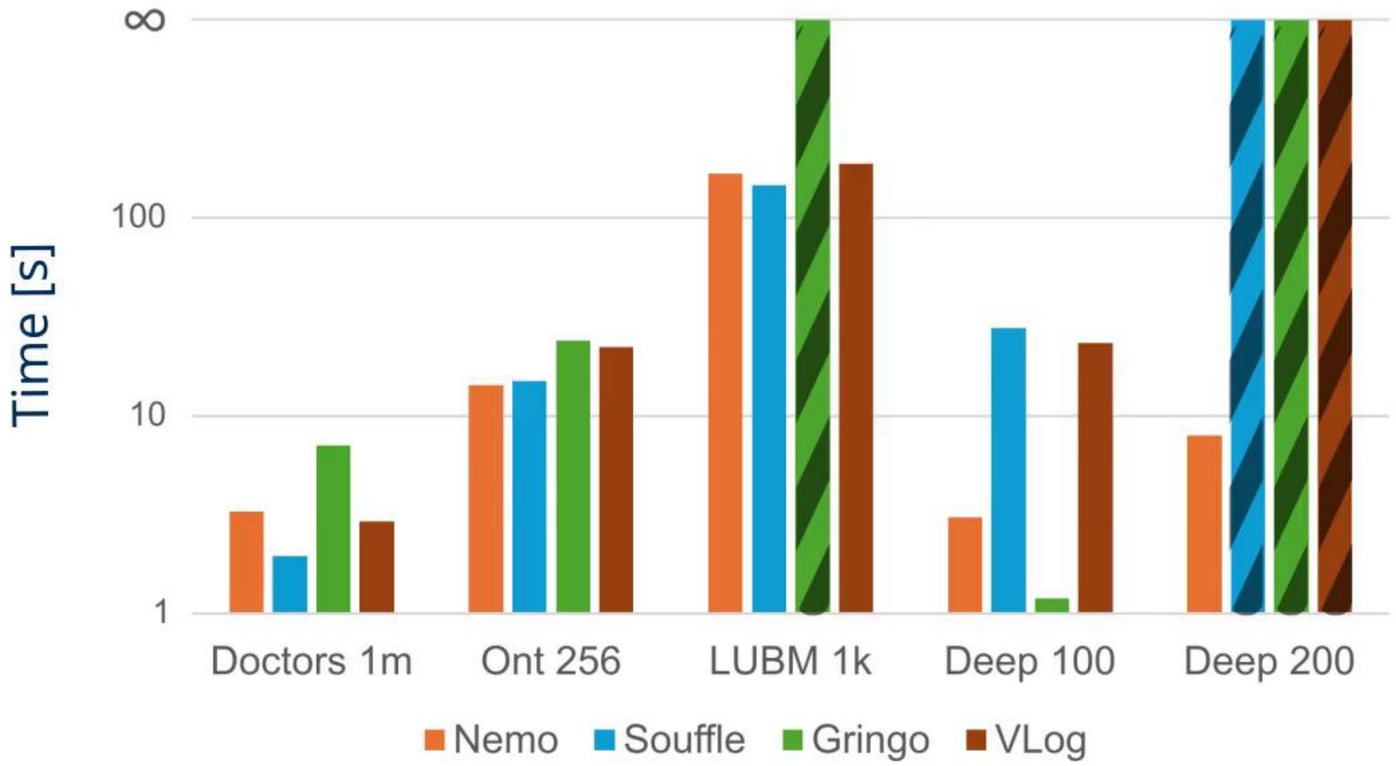
EL Reasoning with Datalog (12 rules, ≤ 5 body atoms, arity ≤ 3)

Input: Galen (facts: 140K) and SNOMED (facts: 1.5M)



Evaluation: Benchmarking the Chase

Established Benchmarks for the Chase [Benedikt et al., PODS'2017]



Evaluation: Wikidata

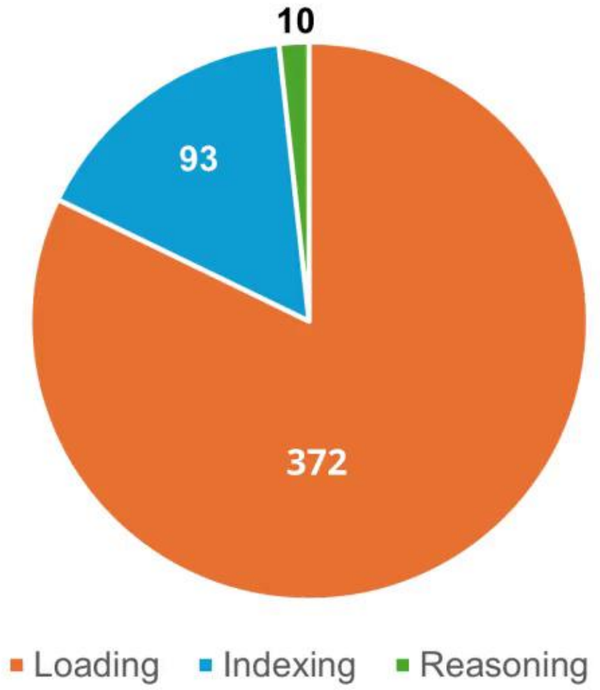
Large-Scale reasoning on (Truthy) Wikidata

Input

- Processing task in creating YAGO KB [Suchanek et. al., SIGIR'24]
- 8 billion triples
- 61GB (compressen N-Triples)

Result

- Time: 8 hours
- Memory: 220GB



Summary

Nemo

- Versatile and scalable Datalog engine
- Free and open-source
- Advanced tool support

Upcoming

- Language extensions
  - List, Sets, Maps
  - Modules and user-defined functions
- Better explainability
- Performance improvements

https://tools.iccl.inf.tu-dresden.de/nemo/