openCypher Specification

Gábor Szárnyas, József Marton

2016/10/16 01:11

Contents

Executive Summary

This document is generated on each commit of the ingraph repository[1]. The document has two goals.

1. Chapter 1 introduces the theoretical foundations of the openCypher language.
2. Chapter 2 contains the acceptance tests defined in the openCypher Technology Compliance Kit[2].

[1]https://github.com/FTSRG/ingraph

[2]https://github.com/opencypher/openCypher/tree/master/tck

Chapter 1

Theoretical Foundations

This chapter is based on our submission to the BTW 2017 conference.[1]

1.1 Introduction

Context. Graphs are a well-known formalism, widely used for describing and analysing systems. Graphs provide an intuitive formalism for modelling real-world scenarios, as the human mind tends to interpret the world in terms of objects (*vertices*) and their respective relationships to one another (*edges*) [16].

The *property graph* data model [18] extends graphs by adding labels and properties for both vertices and edges. This gives a rich set of features for users to model their specific domain in a natural way. Graph databases are able to store property graphs and query their contents with complex graph patterns, which, otherwise would be are cumbersome to define and/or inefficient to evaluate on traditional relational databases and query technologies.

Neo4j [13], a popular NoSQL property graph database, offers the Cypher [12] query language to specify graph patterns. Cypher is a high-level declarative query language which can be optimised by the query engine. The openCypher project [14] is an initiative of Neo Technology, the company behind Neo4j, to deliver an open specification of Cypher.

Problem and objectives. The openCypher project features a formal specification of the grammar of the query language (Section 1.3) and a set of acceptance tests that define the behaviour of various Cypher features. However, there is no mathematical formalisation for any of the language features. In ambiguous cases, the user is advised to consult Neo4j's Cypher documentation or to experiment with Neo4j's Cypher query engine and follow its behaviour. Our goal is to provide a formal specification for the core features of openCypher.

Contributions. In this paper, we use a formal definition of the property graph data model [8] and an extended version of relational algebra, operating on multisets (bags) and featuring additional operators [7]. These allow us to construct a concise formal specification for the core features in the openCypher grammar, which can then serve as a basis for implementing an openCypher-compliant query engine.

1.2 Preliminaries

This section defines the mathematical concepts used in the paper. Our notation closely follows [8] and is similar to [18][2].

1.2.1 Property Graph Data Model

A *property graph* is defined as $G = (V, E, \mathsf{src\_trg}, L_v, L_e, l_v, l_e, P_v, P_e)$, where V is a set of vertices, E is a set of directed edges, $\mathsf{src\_trg} : E \to V \times V$ assigns the source and target vertices to edges. The graph is labelled (or typed):

- L_v is a set of vertex labels, $l_v : V \to 2^{L_v}$ assigns *a set of labels* to each vertex.
- L_e is a set of edge labels, $l_e : E \to L_e$ assigns *a single label* to each edge.

Furthermore, graph G has properties (*attributed graph*). Let D be a set of atomic domains.

- P_v is a set of vertex properties. A vertex property $p_i \in P_v$ is a function $p_i : V \to D_i \cup \{\varepsilon\}$, which assigns a property value from a domain $D_i \in D$ to a vertex $v \in V$, if v has property p_i, otherwise $p_i(v)$ returns ε.

[1]`http://btw2017.informatik.uni-stuttgart.de/?pageId=Start&language=en`

[2]The formalism presented in [18] lacks the notion of *vertex labels*.

- P_e is a set of edge properties. An edge property $p_j \in P_e$ is a function $p_j : E \to D_j \cup \{\varepsilon\}$, which assigns a property value from a domain $D_j \in D$ to an edge $e \in E$, if e has property p_j, otherwise $p_j(e)$ returns ε.

Running example. Fig. 1.1 presents an example inspired by the Movie Database dataset[3].

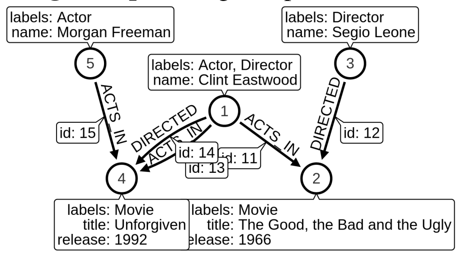


Fig. 1.1: Example movie graph.

$V = \{1,2,3,4,5\}; E = \{11,12,13,14,15\};$
$\mathsf{src\_trg}(11) = \langle 1,2\rangle; \mathsf{src\_trg}(12) = \langle 3,2\rangle; \ldots$
$L_v = \{\mathsf{Actor}, \mathsf{Director}, \mathsf{Movie}\};$
$L_e = \{\mathsf{ACTS\_IN}, \mathsf{DIRECTED}\};$
$l_v(1) = \{\mathsf{Actor}, \mathsf{Director}\}; l_v(2) = \{\mathsf{Movie}\}; \ldots;$
$l_e(11) = \mathsf{ACTS\_IN}; l_e(12) = \mathsf{DIRECTED}; \ldots;$
$P_v = \{\mathsf{name}, \mathsf{title}, \mathsf{release}\}; P_e = \{\};$
$\mathsf{name}(1) = \mathsf{'Clint\ Eastwood'}; \mathsf{name}(2) = \varepsilon; \ldots$
$\mathsf{title}(1) = \varepsilon; \mathsf{title}(2) = \mathsf{'The\ Good,\ the\ Bad\ and\ the\ Ugly'}; \ldots$
$\mathsf{release}(1) = \varepsilon; \mathsf{release}(2) = 1966; \ldots$

Fig. 1.2: The dataset represented as a property graph.

In the context of this paper, we define a *relation* as a bag (*multiset*) of tuples: a tuple can occur more than once in the relation [7]. Given a property graph G, relation r is a *graph relation* if the following holds:

$$\forall A \in \operatorname{attr}(r) : \operatorname{dom}(A) \subseteq V \cup E \cup D,$$

where $\operatorname{attr}(r)$ is the set of attributes of r, $\operatorname{dom}(A)$ is the domain of attribute A. The schema of r, sch(r) is a list containing the attribute names. For schema transformations, the *append* operator is denoted by $\|$, the *remove* operator is denoted by $-$.

1.2.2 Foundations of Relational Algebra

Basic operators of relational algebra. We give a brief summary of the operators in relational algebra. A more detailed discussion is available in database textbooks, e.g. [4].

Unary operators. The *projection* operator π keeps a specific set of attributes in the relation: $t = \pi_{\mathsf{A_1,\ldots,A_n}}(r)$. Note that the tuples are not deduplicated by default, i.e. the results will have the same number of tuples as the input relation r. The projection operator can also rename the attributes, e.g. $\pi_{\mathsf{v1} \to \mathsf{v2}}(r)$ renames $\mathsf{v1}$ to $\mathsf{v2}$. The *selection* operator σ filters the incoming relation according to some criteria. Formally, $t = \sigma_\theta(r)$, where predicate θ is a propositional formula. The operator selects all tuples in r for which θ holds.

Binary operators. The $\cup$ operator produces the set union of two relations, while the $\uplus$ operator produces the *bag union* of two operators, e.g. $\{\langle 1,2\rangle, \langle 1,2\rangle, \langle 3,4\rangle\} \uplus \{\langle 1,2\rangle\} = \{\langle 1,2\rangle, \langle 1,2\rangle, \langle 1,2\rangle, \langle 3,4\rangle\}$. For both the *union* and *bag union* operators, the schema of the operands must have the same number of attributes. Some authors also require that they share a common schema, i.e. have the same set of attributes [7].

The $\times$ operator produces the *Cartesian product* $t = r \times s$. The result of the *natural join* operator $\bowtie$ is determined by creating the Cartesian product of the relations, then filtering those tuples which are equal on the attributes that share a common name. The combined tuples are projected: from the attributes present in both of the two input relations, we only keep the ones in r and drop the ones in s. Thus, the join operator is defined as

$$r \bowtie s = \pi_{R \cup S}\left(\sigma_{\mathsf{r.A_1=s.A_1} \wedge \ldots \wedge \mathsf{r.A_n=s.A_n}}(r \times s)\right),$$

where $\{A_1, \ldots, A_n\}$ is the set of attributes that occur both in R and S, i.e. $R \cap S = \{A_1, \ldots, A_n\}$. Note that if the set of common attributes is empty, the *natural join* operator is equivalent to the Cartesian product of the relations. The join operator is both commutative and associative: $r \bowtie s = s \bowtie r$ and $(r \bowtie s) \bowtie t = r \bowtie (s \bowtie t)$, respectively. The *antijoin* operator $\rhd$ (also known as *left anti semijoin*) collects the tuples from the left relation r which have no matching pair in the right relation s:

$$t = r \rhd s = r \setminus \pi_R(r \bowtie s),$$

where π_R denotes a projection operator, which only keeps the attributes of the schema over relation r. The antijoin operator is not commutative and not associative. The *left outer join* $⟕$ pads tuples from the left relation that did not match any from the right relation with ε values and adds them to the result of the *natural join* [20].

1.2.3 Common Extensions to Relational Algebra

Most textbooks also define *extended operators* of relational algebra [7]:

- The *duplicate-elimination* operator δ eliminates duplicate tuples in a bag.

[3] `https://neo4j.com/developer/movie-database/`

- The *grouping* operator γ groups tuples according to their value in one or more attributes and aggregates the remaining attributes.
- The *sorting* operator τ transforms a bag relation of tuples to a list of tuples by ordering them. The ordering is defined by specified attributes of the tuples with an ordering direction (ascending $\Uparrow$/descending $\Downarrow$) for each attribute, e.g. $\tau_{\Uparrow \mathsf{v1}, \Downarrow \mathsf{v2}}(r)$.

The *top* operator λ_l^s (adapted from [11]) takes a list as its input, skips the top s tuples and returns the next l tuples.[4]

1.2.4 Graph-Specific Extensions to Relational Algebra

We adapted graph-specific operators from [8][5] and propose an additional operator.

The *get-vertices* nullary operator $\bigcirc_{(\mathtt{v}:\ \mathsf{t}_1 \wedge \ldots \wedge \mathsf{t}_n)}$ returns a graph relation of a single attribute v that contains the ID of all vertices that have *all* of labels $t_1, \ldots, t_n$.

The *expand-both* unary operator $\updownarrow_{(\mathtt{E})}^{(\mathtt{l}_1 \vee \ldots \vee \mathtt{l}_k * \mathtt{min} \ldots \mathtt{max}:\ \mathtt{v})} \left[\mathtt{w}:\ \mathsf{t}_1 \wedge \ldots \wedge \mathsf{t}_n\right](r)$ adds (1) a new attribute w to r containing the IDs of vertices having *all* labels $t_1, \ldots, t_n$ that can be reached from vertices of attribute v by traversing edges having *any* labels $l_1, \ldots, l_n$, and (2) a new attribute E for the edges of the path from v to w. The operator may use at least min and at most max hops, both defaulting to 1 if omitted. The *expand-in* operator $\downarrow$ and *expand-out* operator $\uparrow$ only consider directed paths from w to v and from v to w, respectively.

We propose the *all-different* operator to guarantee the uniqueness of edges (see the remark on *uniqueness of edges* in Section 1.3). The *all-different* operator $\dot{\neq}_{\mathsf{E}_1,\mathsf{E}_2,\mathsf{E}_3,\ldots}(r)$ filters r to keep tuples where the variables in $\bigcup_i E_i$ are pairwise different.[6] It can be expressed as a *selection*:

$$\dot{\neq}_{\mathsf{E}_1,\mathsf{E}_2,\mathsf{E}_3,\ldots}(r) = \sigma_{\bigwedge_{\mathsf{e}_1,\mathsf{e}_2 \in \bigcup_i \mathsf{E}_i \ \wedge\ \mathsf{e}_1 \neq \mathsf{e}_2} \mathsf{r.e}_1 \neq \mathsf{r.e}_2}(r)$$

Property access. Assuming that x is an attribute of a graph relation, we use the notation $x.a$ in (1) attribute lists for projections and (2) selection conditions to express the access to the corresponding value of property a in the property graph [8].

Summary. Table 1.1 provides an overview of the operators of relational graph algebra.

ops	operator	name	prop.	output for			schema
				set	**bag**	**list**	
0	$\bigcirc_{(\mathtt{v})}$	*get-vertices*	set	set	set	set	$\langle \mathsf{v} \rangle$
1	$\pi_{\mathsf{v}_1,\mathsf{v}_2,\ldots}(r)$	*projection*	i	bag	bag	list	$\langle \mathsf{v}_1, \mathsf{v}_2, \ldots \rangle$
	$\sigma_{\mathsf{condition}}(r)$	*selection*	i	set	bag	list	sch(r)
	$\updownarrow_{(\mathtt{v})}^{(\mathtt{w})} [\mathsf{e}](r)$	*expand-both*	–	set	bag	list	sch(r) ‖ $\langle \mathsf{e}, \mathsf{w} \rangle$
	$\dot{\neq}_{\mathsf{variables}}(r)$	*all-different*	i	set	bag	list	sch(r)
	$\delta(r)$	*duplicate-elimination*	i	set	set	list	sch(r)
	$\tau_{\Downarrow \mathsf{v}_1, \Uparrow \mathsf{v}_2,\ldots}(r)$	*sorting*	i	list	list	list	sch(r)
	$\gamma_{\mathsf{v}_1,\mathsf{v}_2,\ldots}(r)$	*grouping*	i	set	set	set	$\langle \mathsf{v}_1, \mathsf{v}_2, \ldots \rangle$
	$\lambda(r)$	*top*	–	list	list	list	sch(r)
2	$r \cup s, r \setminus s$	*union, minus*	–	set	set	set	sch(r)
	$r \uplus s$	*bag union*	c, a	bag	bag	bag	sch(r)
	$r \times s$	*Cartesian product*	c, a	set	bag	bag	sch(r) ‖ sch(s)
	$r \bowtie s$	*natural join*	c, a	set	bag	bag	sch(r) ‖ (sch(s) \ sch(r))
	$r ⟕ s$	*left outer join*	–	set	bag	bag	sch(r) ‖ (sch(s) \ sch(r))
	$r \rhd s$	*antijoin*	c, a	set	bag	bag	sch(r)

Table 1.1: Properties of relational graph algebra operators. A unary operator α is idempotent (i), iff $\alpha(x) = \alpha(\alpha(x))$ for all inputs. A binary operator β is commutative (c), iff $x \ \beta \ y = y \ \beta \ x$ and associative (a), iff $(x \ \beta \ y) \ \beta \ z = x \ \beta \ (y \ \beta \ z)$.

[4] SQL implementations offer the `OFFSET` and the `LIMIT`/`TOP` keywords.

[5] The GETNODES operator introduced in [8] and did not support labels. We extended it by allowing the specification of vertex labels and renamed it to *get-vertices* to be consistent with the rest of the definitions. We also extended the EXPANDIN and EXPANDOUT operators to allow it to return a set of edges, and introduced the *expand-both* operator to allow navigation to both directions.

[6] Should e.g. E_2 be a set of the single variable e_2, the variable name can be used as a shorthand instead, so $\dot{\neq}_{\mathsf{E}_1,\mathsf{e}_2,\mathsf{E}_3,\ldots}(r) \equiv \dot{\neq}_{\mathsf{E}_1,\{\mathsf{e}_2\},\mathsf{E}_3,\ldots}(r)$

1.3 The openCypher Query Language

Language. As the primary query language of Neo4j [13], Cypher [12] was designed to read easily. It allows users to specify the graph pattern by a syntax resembling an actual graph. The goal of the openCypher project [14] is to provide a standardised specification of the Cypher language. **??** 1.1 shows an openCypher query, which returns all people who (1) are both actors and directors and (2) have acted in a movie together with Clint Eastwood.

```
MATCH (a1)-[:ACTS_IN]->(:Movie)<-[:ACTS_IN]-(a2:Actor:Director)
WHERE a1.name = "Clint Eastwood"
RETURN a2
```

Listing 1.1: Get people who are both actors and directors and acted in a movie with Clint Eastwood.

The query returns with a bag of vertices that have both the labels Actor and Director and share a common Movie neighbor through ACTS_IN edges. Cypher guarantees that these edges are only traversed once, so the vertex of Clint Eastwood is not returned (see the section on the uniqueness of edges).

Implementation. While Neo4j uses a parsing expression grammar (PEG) [6] for specifying the grammar rules of Cypher, openCypher aims to achieve an implementation-agnostic specification by only providing a context-free grammar. The parser can be implemented using any capable parser technology, e.g. ANTLR4 [15] or Xtext [5].

Legacy grammar rules. It is not a goal of the openCypher project to fully cover the features of Neo4j's Cypher language: "Not all grammar rules of the Cypher language will be standardised in their current form, meaning that they will not be part of openCypher as-is. Therefore, the openCypher grammar will not include some well-known Cypher constructs; these are called 'legacy'."[7] The *legacy rules* include commands (`CREATE INDEX`, `CREATE UNIQUE CONSTRAINT`, etc.), pre-parser rules (`EXPLAIN`, `PROFILE`) and deprecated constructs (`START`). A detailed description is provided in the openCypher specification. In our work, we focused on the *standard core* of the language and ignored legacy rules.

Uniqueness for edges. In an openCypher query, a `MATCH` clause defines a graph pattern. A query can be composed of multiple patterns spanning multiple `MATCH` clauses. For the matches of a pattern within a single `MATCH` clause, edges are required to be unique. However, matches for multiple `MATCH` clauses can share edges. This uniqueness criterium can be expressed in a compact way with the *all-different* operator introduced in Section 1.2.4. For vertices, this restriction does not apply.

Aggregation. It indeed makes sense to calculate aggregation over graph pattern matches, though, its result will not necessarily be pattern match with vertices and edges. Based on some *grouping criteria*, matches are put into categories, and values for the grouping criteria as well as grouping functions[8] over the groups, the aggregations are evaluated in a single tuple for each and every category. In the SQL query language, grouping criteria is explicitly given by using the `GROUP BY` clause. In openCypher, however, this is done implicitly in the `RETURN` as well as in `WITH` clauses: vertices, edges and their properties that appear outside the grouping functions become the *grouping criteria*.[9]

Subqueries. One can compose an openCypher query of multiple subqueries. Subqueries, written subsequently, mostly begin by a `MATCH` clause and end at (including) a `RETURN` or `WITH` clause, the latter having an optional `WHERE` clause to follow. The `WITH` and `RETURN` clauses determine the resulting schema of the subquery by specifiying the vertices, edges, attributes and aggregates of the result. When `WITH` has the optional `WHERE` clause, it applies an other filter on the subquery result. [10] The last subquery must be ended by `RETURN`, whereas all the previous ones must be ended by `WITH`. If a query is composed by more than one subqueries, their results are joined together using *natural join* or *left outer join* operators.

1.4 Mapping openCypher Queries to Relational Graph Algebra

In this section, we first give the mapping algorithm of openCypher queries to relational graph algebra, then we give a more detailed listing of the compilation rules for the query language constructs in Table 1.2. We follow the bottom-up approach to build the relational graph algebra tree based on the openCypher query. The algorithm is as follows. Join operations always use all common variables to match the two inputs (see *natural join* in Section 1.2.4).

1. A single pattern is turned left-to-right to a *get-vertices* for the first vertex and a chain of *expand-in*, *expand-out* or *expand-both* operators for inbound, outbound or undirected relationships, respectively.

[7] `https://github.com/opencypher/openCypher/tree/master/grammar`

[8] For example, `count`, `avg`, `sum`, `max`, `min`, `stdDev`, `stdDevP`, `collect`. The `collect` function is an exception as it does not return a single scalar value but returns a collection (list).

[9] This approach is also used by some SQL code assistant IDEs generating the `GROUP BY` clause for a query.

[10] This is much like the `HAVING` construct of the SQL language with the major difference that it is also allowed in openCypher in case no aggregation has been done.

2. Patterns in the same `MATCH` clause are joined by *natural join*.
3. Append an *all-different* operator for all edge variables that appear in the `MATCH` clause because of the non-repeating edges language rule.
4. Process the `WHERE` clause. Note that according to the grammar, `WHERE` is bound to a `MATCH` clause.
5. Several `MATCH` clauses are connected to a left deep tree of *natural join*. If `MATCH` has the `OPTIONAL` modifier, *left outer join* is used instead of *natural join*.
6. If there is a positive or negative pattern deferred from `WHERE` processing, append it as a *natural join* or *antijoin* operator, respectively.
7. Append *grouping*, if `RETURN` or `WITH` clause has grouping functions inside
8. Append *projection* operator based on the `RETURN` or `WITH` clause. This operator will also handle the renaming (i.e. `AS`).
9. Append *duplicate-elimination* operator, if the `RETURN` or `WITH` clause has the `DISTINCT` modifier.
10. Append a *selection* operator in case the `WITH` had the optional `WHERE` clause.
11. If this is not the first subquery, join to the relational graph algebra tree using *natural join* or *left outer join*.
12. Assemble a *union* operation from the query parts[11]. As the *union* operator is technically a binary operator, the *union* of more than two query parts are represented as a left deep tree of `UNION` operators.

Example. The example query in **??** 1.1 can be formalized as:

$$\pi_{\mathsf{a2}}\left(\sigma_{\mathsf{a1.name='C.E.'}}\left(\not\doteq_{\mathsf{\_e1,\_e2}} \downarrow^{\mathsf{(a2:\ Actor \wedge Director)}}_{\mathsf{(a1)}} [\mathsf{\_e1:\ ACTS\_IN}] \uparrow^{\mathsf{(:\ Movie)}}_{\mathsf{(a1)}} [\mathsf{\_e2:\ ACTS\_IN}] \left(\bigcirc_{\mathsf{(a1:\ Actor)}}\right)\right)\right)$$

Note that the $\not\doteq$ guarantees the uniqueness constraint for the edges (Section 1.3), which prevents the query from returning the vertex Clint Eastwood.

Optimisations. Queries with negative conditions for patterns can also be expressed using the *antijoin* operator. For example, `MATCH «p1» WHERE NOT «p2»` can be formalized as

$$\not\doteq_{\mathsf{edges\ of\ p1}}\left(\mathsf{p_1}\right) \rhd \not\doteq_{\mathsf{edges\ of\ p2}}\left(\mathsf{p_2}\right)$$

Limitations. Our mapping does not completely cover the openCypher language. As discussed in Section 1.3, some constructs are defined as legacy and thus were omitted. Also, we did not formalize expressions (e.g. conditions in selections), collections (arrays and maps), which are required for both path variables[12] and the `UNWIND` operator. The mapping does not cover parameters and data manipulation operations, e.g. `CREATE`, `DELETE`, `SET` and `MERGE`.

1.5 Related Work

The TinkerPop framework [1] aims to provide a standard data model for property graphs, along with Gremlin, a high-level graph-traversal language [17] and the Gremlin Structure API, a low-level programming interface.

Besides property graphs, graph queries can be formalized on different graph-like data models and even relational databases.

EMF. The Eclipse Modeling Framework (EMF) is an object-oriented modelling framework widely used in model-driven engineering. Henshin [2] provides a visual language for defining patterns, while Epsilon [9] and VIATRA Query [3] provide high-level declarative (textual) query languages, Epsilon Pattern Language and VIATRA Query Language.

RDF. The Resource Description Framework (RDF) [22] aims to describe entities of the semantic web. RDF assumes sparse, ever-growing and incomplete data stored as triples that can be queried using the SPARQL [23] graph pattern language.

[11] In this context, query parts refer to those parts of the query connected by the `UNION` openCypher keyword.
[12] `MATCH p=(:Person)-[:FRIEND*1..2]->(:Person)`

Language construct	**Relational algebra expression**
Vertex, edge and path patterns	
`()`	$\bigcirc_{(\_v)}$
`(:«t1»:«t2»)`	$\bigcirc_{(\_v:\ t1 \land t2 \land \ldots)}$
`(«v»:«t1»:«t2»)`	$\bigcirc_{(v:\ t1 \land t2 \land \ldots)}$
`«p»-[«e»:«l1|...»]-(«w»:«t1»:«t2»...)`	$\updownarrow^{(w:\ t1 \land t2 \land \ldots)}_{(v)} [e:\ l1 \lor \ldots] (p)$
`«p»<-[«e»:«l1|...»]->(«w»:«t1»:«t2»...)`	
`«p»-[«e»:«l1|...»]->(«w»:«t1»:«t2»...)`	$\uparrow^{(w:\ t1 \land t2 \land \ldots)}_{(v)} [e:\ l1 \lor \ldots] (p)$
`«p»<-[«e»:«l1|...»]-(«w»:«t1»:«t2»...)`	$\downarrow^{(w:\ t1 \land t2 \land \ldots)}_{(v)} [e:\ l1 \lor \ldots] (p)$
`«p»-[«l1|...»*«min»..«max»]->(«w»:«t2»)`	$\uparrow^{(w:\ t1 \land t2 \land \ldots)}_{(v)} [E:\ l1 \lor \ldots] (p)$
Combining and filtering pattern matches	
`MATCH «p»`	$\dot{\neq}_{\text{edges of p}} (p)$
`MATCH «p1», «p2»`	$\dot{\neq}_{\text{edges of p1 and p2}} \left(p_1 \bowtie p_2\right)$

`MATCH «p1»`
 `MATCH «p2»`
	$\dot{\neq}_{\text{edges of p1}} \left(p_1\right) \bowtie \dot{\neq}_{\text{edges of p2}} \left(p_2\right)$

`MATCH «p1»`
 `OPTIONAL MATCH «p2»`
	$\dot{\neq}_{\text{edges of p1}} \left(p_1\right) ⟕ \dot{\neq}_{\text{edges of p2}} \left(p_2\right)$

`MATCH «p»`
 `WHERE «condition»`
	$\sigma_{\text{condition}}(r)$, where condition may specify patterns and arithmetic constraints on existing variables

Result and sub-result operations. Rules for `RETURN` also apply to `WITH`.	
`RETURN «variables»`	$\pi_{\text{variables}}(r)$
`RETURN «v1» AS «alias1» ...`	$\pi_{\text{v1} \rightarrow \text{alias1}, \ldots}(r)$
`RETURN DISTINCT «variables»`	$\delta\left(\pi_{\text{variables}}(r)\right)$
`RETURN «variables», «aggregates»`	$\gamma_{\text{variables,aggregates}}(r)$
List operations	
`ORDER BY «v1» [ASC|DESC] ...`	$\tau_{\Uparrow/\Downarrow \text{v1}, \ldots}(r)$
`LIMIT «l»`	$\lambda_l(r)$
`SKIP «s»`	$\lambda^s(r)$
`SKIP «s» LIMIT «l»`	$\lambda_l^s(r)$
Combining results	
`«query1» UNION «query2»`	$r_1 \cup r_2$
`«query1» UNION ALL «query2»`	$r_1 \uplus r_2$

Table 1.2: Mapping from openCypher constructs to relational algebra.

SQL. In general, relational databases offer limited support for graph queries: recursive queries are supported by PostgreSQL using the `WITH RECURSIVE` keyword and by the Oracle Database using the `CONNECT BY` keyword. Graph queries are supported in SAP HANA Graph Scale-Out Extension prototype [19], through a SQL-based language [10].

1.6 Conclusion and Future Work

In this paper, we presented a formal specification for a subset of the openCypher query language. This provides the theoretical foundations to use openCypher as a language for graph query engines. Using the proposed mapping, an openCypher-compliant query engine could be built on any relational database engine to (1) store property graphs as graph relations and to (2) efficiently evaluate the extended operators of relational graph algebra.

As a future work, we will give formal specification of the operators for incremental query evaluation, which requires us to define *maintenance operations* to keep their result in sync with the latest set of changes. Our long-term research objective is to design and prototype a *distributed, incremental graph query engine* [21] for the property graph data model.

Bibliography

[1] Apache. Tinkerpop3. `http://tinkerpop.apache.org/`, 2016.

[2] T. Arendt, E. Biermann, S. Jurack, C. Krause, and G. Taentzer. pages 121–135, 2010.

[3] G. Bergmann et al. Incremental evaluation of model queries over EMF models. In *MODELS*, pages 76–90, 2010.

[4] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems, 3rd Edition*. Addison-Wesley-Longman, 2000.

[5] M. Eysholdt and H. Behrens. Xtext: implement your language faster than the quick and dirty way. In *SIGPLAN, SPLASH/OOPSLA*, pages 307–309, 2010.

[6] B. Ford. Parsing expression grammars: a recognition-based syntactic foundation. In *SIGPLAN-SIGACT, POPL*, pages 111–122. ACM, 2004.

[7] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database systems – The complete book*. Pearson Education, 2nd edition, 2009.

[8] J. Hölsch and M. Grossniklaus. An algebra and equivalences to transform graph patterns in neo4j. In *EDBT/ICDT*, 2016.

[9] D. S. Kolovos, R. F. Paige, and F. Polack. The epsilon transformation language. pages 46–60, 2008.

[10] C. Krause et al. An SQL-based query language and engine for graph pattern matching. In *ICGT*, pages 153–169, 2016.

[11] C. Li, K. C. Chang, I. F. Ilyas, and S. Song. RankSQL: Query algebra and optimization for relational top-k queries. In *SIGMOD*, pages 131–142, 2005.

[12] Neo Technology. Cypher query language. `https://neo4j.com/docs/developer-manual/current/cypher/`, 2016.

[13] Neo Technology. Neo4j. `http://neo4j.org/`, 2016.

[14] Neo Technology. openCypher project. `http://www.opencypher.org/`, 2016.

[15] T. Parr. *The Definitive ANTLR 4 Reference*. Pragmatic Bookshelf, 2nd edition, 2013.

[16] M. A. Rodriguez. A collectively generated model of the world. In *Collective intelligence: creating a prosperous world at peace*. Oakton: Earth Intelligence Network, 2008.

[17] M. A. Rodriguez. The gremlin graph traversal machine and language (invited talk). DBPL 2015, pages 1–10, New York, NY, USA, 2015. ACM.

[18] M. A. Rodriguez and P. Neubauer. The graph traversal pattern. In *Graph Data Management: Techniques and Applications.*, pages 29–46. IGI Global, 2011.

[19] M. Rudolf, M. Paradies, C. Bornhövd, and W. Lehner. The graph story of the SAP HANA database. In *BTW*, volume 214 of *LNI*, pages 403–420. GI, 2013.

[20] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts, 5th Edition*. McGraw-Hill Book Company, 2005.

[21] G. Szárnyas et al. IncQuery-D: A distributed incremental model query framework in the cloud. In *MODELS*, pages 653–669, 2014.

[22] World Wide Web Consortium. Resource Description Framework (RDF). `http://www.w3.org/standards/techs/rdf/`.

[23] World Wide Web Consortium. SPARQL query language for RDF. `http://www.w3.org/TR/rdf-sparql-query/`.

Chapter 2

TCK Acceptance Tests

2.1 AggregationAcceptance

2.1.1 Support multiple divisions in aggregate function

Query specification

```
MATCH (n)
RETURN count(n) / 60 / 60 AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.2 Support column renaming for aggregates as well

Query specification

```
MATCH ()
RETURN count(*) AS columnName
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.3 Aggregates inside normal functions

Query specification

```
MATCH (a)
RETURN size(collect(a))
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.4 Handle aggregates inside non-aggregate expressions

Query specification

```
MATCH (a {name: 'Andres'})<-[:FATHER]-(child)
RETURN {foo: a.name='Andres', kids: collect(child.name)}
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.5 Count nodes

Query specification

```
MATCH (a:L)-[rel]->(b)
RETURN a, count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.6 Sort on aggregate function and normal property

Query specification

```
MATCH (n)
RETURN n.division, count(*)
ORDER BY count(*) DESC, n.division ASC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.7 Aggregate on property

Query specification

```
MATCH (n)
RETURN n.x, count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.8 Count non-null values

Query specification

```
MATCH (n)
RETURN n.y, count(n.x)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.9 Sum non-null values

Query specification

```
MATCH (n)
RETURN n.y, sum(n.x)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.10 Handle aggregation on functions

Query specification

```
MATCH p=(a:L)-[*]->(b)
RETURN b, avg(length(p))
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.11 Distinct on unbound node

Query specification

```
OPTIONAL MATCH (a)
RETURN count(DISTINCT a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.12 Distinct on null

Query specification

```
MATCH (a)
RETURN count(DISTINCT a.foo)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.13 Collect distinct nulls

Query specification

```
UNWIND [null, null] AS x
RETURN collect(DISTINCT x) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.14 Collect distinct values mixed with nulls

Query specification

```
UNWIND [null, 1, null] AS x
RETURN collect(DISTINCT x) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.15 Aggregate on list values

Query specification

```
MATCH (a)
RETURN DISTINCT a.color, count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.16 Aggregates in aggregates

Query specification

```
RETURN count(count(*))
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.17 Aggregates with arithmetics

Query specification

```
MATCH ()
RETURN count(*) * 10 AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.18 Aggregates ordered by arithmetics

Query specification

```
MATCH (a:A), (b:X)
RETURN count(a) * 10 + count(b) * 5 AS x
ORDER BY x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.19 Multiple aggregates on same variable

Query specification

```
MATCH (n)
RETURN count(n), collect(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.20 Simple counting of nodes

Query specification

```
MATCH ()
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.21 Aggregation of named paths

Query specification

```
MATCH p = (a)-[*]->(b)
RETURN collect(nodes(p)) AS paths, length(p) AS l
ORDER BY l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.22 Aggregation with ‘min()‘

Query specification

```
MATCH p = (a:T {name: ’a’})-[:R*]->(other:T)
WHERE other <> a
WITH a, other, min(length(p)) AS len
RETURN a.name AS name, collect(other.name) AS others, len
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.23 Handle subexpression in aggregation also occurring as standalone expression with nested aggregation in a literal map

Query specification

```
MATCH (a:A), (b:B)
RETURN coalesce(a.prop, b.prop) AS foo,
  b.prop AS bar,
  {y: count(b)} AS baz
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.24 Projection during aggregation in WITH before MERGE and after WITH with predicate

Query specification

```
UNWIND [42] AS props
WITH props WHERE props > 32
WITH DISTINCT props AS p
MERGE (a:A {prop: p})
RETURN a.prop AS prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.25 No overflow during summation

Query specification

```
UNWIND range(1000000, 2000000) AS i
WITH i
LIMIT 3000
RETURN sum(i)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.26 Counting with loops

Query specification

```
MATCH ()-[r]-()
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.27 ‘max()‘ should aggregate strings

Query specification

```
UNWIND ['a', 'b', 'B', null, 'abc', 'abc1'] AS i
RETURN max(i)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.1.28 ‘min()‘ should aggregate strings

Query specification

```
UNWIND ['a', 'b', 'B', null, 'abc', 'abc1'] AS i
RETURN min(i)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.2 ColumnNameAcceptance

2.2.1 Keeping used expression 1

Query specification

```
MATCH (n)
RETURN cOuNt( * )
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.2.2 Keeping used expression 2

Query specification

```
MATCH p = (n)-->(b)
RETURN nOdEs( p )
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.2.3 Keeping used expression 3

Query specification

```
MATCH p = (n)-->(b)
RETURN coUnt( dIstInct p )
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.2.4 Keeping used expression 4

Query specification

```
MATCH p = (n)-->(b)
RETURN aVg(    n.aGe     )
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.3 ComparisonOperatorAcceptance

2.3.1 Handling numerical ranges 1

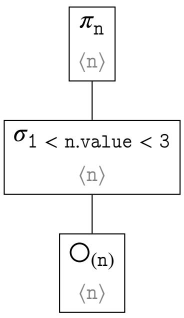
Query specification

```
MATCH (n)
WHERE 1 < n.value < 3
RETURN n.value
```

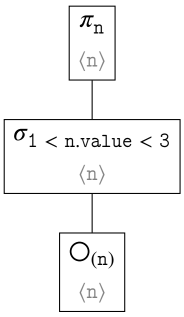
Relational algebra expression

$\pi_{\mathtt{n}}\left(\sigma_{1\,<\,\mathtt{n.value}\,<\,3}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{1 < \mathtt{n.value} < 3}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{1 < \mathtt{n.value} < 3}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.3.2 Handling numerical ranges 2

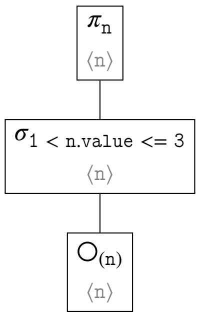
Query specification

```
MATCH (n)
WHERE 1 < n.value <= 3
RETURN n.value
```

Relational algebra expression

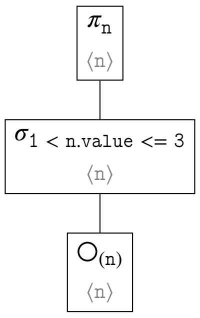
$\pi_{\mathtt{n}}\left(\sigma_{1\ <\ \mathtt{n.value}\ <=\ 3}\left(\dot{\not\div}\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$

Relational algebra tree



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{1\ <\ \mathtt{n.value}\ <=\ 3}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{1\ <\ \mathtt{n.value}\ <=\ 3}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.3.3 Handling numerical ranges 3

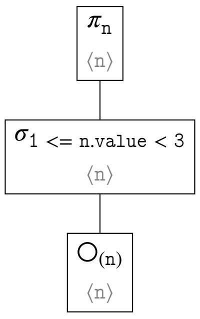
Query specification

```
MATCH (n)
WHERE 1 <= n.value < 3
RETURN n.value
```

Relational algebra expression

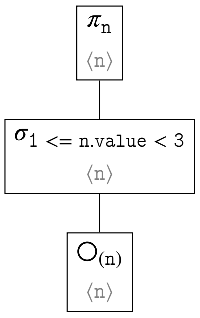
$\pi_{\mathtt{n}}\left(\sigma_{1\ <=\ \mathtt{n.value}\ <\ 3}\left(\dot{\not\div}\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$

Relational algebra tree



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{1\ <=\ \mathtt{n.value}\ <\ 3}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



π_{n}
⟨n⟩
$\sigma_{1 \text{ <= n.value < } 3}$
⟨n⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩

2.3.4 Handling numerical ranges 4

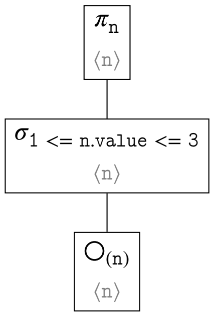
Query specification

```
MATCH (n)
WHERE 1 <= n.value <= 3
RETURN n.value
```

Relational algebra expression

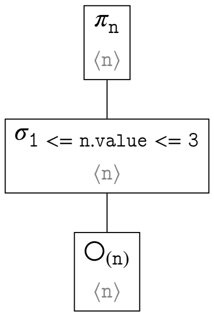
$\pi_{\mathrm{n}}\left(\sigma_{1 \text{ <= n.value <= } 3}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathrm{n})}\right)\right)\right)$

Relational algebra tree



π_{n}
⟨n⟩
$\sigma_{1 \text{ <= n.value <= } 3}$
⟨n⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩

Relational algebra tree for incremental queries



π_{n}
⟨n⟩
$\sigma_{1 \text{ <= n.value <= } 3}$
⟨n⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩

2.3.5 Handling string ranges 1

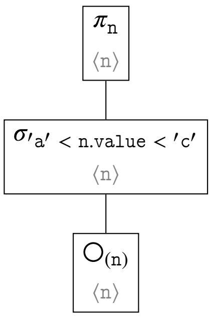
Query specification

```
MATCH (n)
WHERE 'a' < n.value < 'c'
RETURN n.value
```

Relational algebra expression

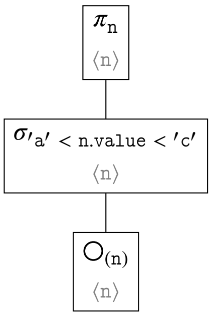
$\pi_{\mathrm{n}}\left(\sigma_{'\mathrm{a}' \text{ < n.value < } '\mathrm{c}'}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathrm{n})}\right)\right)\right)$

Relational algebra tree



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{'\mathtt{a}' < \mathtt{n.value} < '\mathtt{c}'}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{'\mathtt{a}' < \mathtt{n.value} < '\mathtt{c}'}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.3.6 Handling string ranges 2

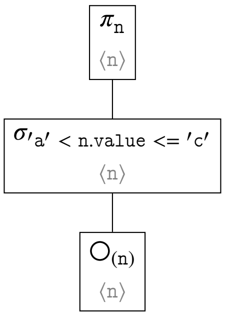
Query specification

```
MATCH (n)
WHERE 'a' < n.value <= 'c'
RETURN n.value
```

Relational algebra expression

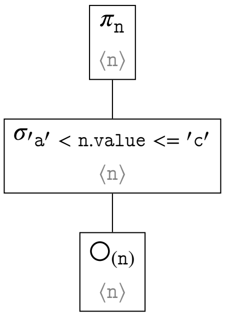
$$\pi_{\mathtt{n}}\left(\sigma_{'\mathtt{a}' < \mathtt{n.value} <= '\mathtt{c}'}\left(\not\doteq\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{'\mathtt{a}' < \mathtt{n.value} <= '\mathtt{c}'}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{'\mathtt{a}' < \mathtt{n.value} <= '\mathtt{c}'}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.3.7 Handling string ranges 3

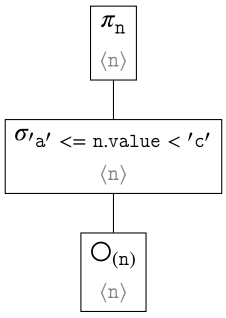
Query specification

```
MATCH (n)
WHERE 'a' <= n.value < 'c'
RETURN n.value
```

Relational algebra expression

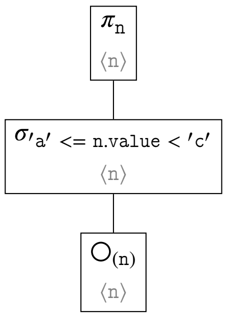
$\pi_{\texttt{n}} \left(\sigma_{'\texttt{a}' \ \texttt{<=}\ \texttt{n.value}\ \texttt{<}\ '\texttt{c}'} \left(\dot{\not\equiv} \left(\bigcirc_{(\texttt{n})} \right) \right) \right)$

Relational algebra tree



$\pi_{\texttt{n}}$
⟨n⟩
$\sigma_{'\texttt{a}' \ \texttt{<=}\ \texttt{n.value}\ \texttt{<}\ '\texttt{c}'}$
⟨n⟩
$\bigcirc_{(\texttt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\texttt{n}}$
⟨n⟩
$\sigma_{'\texttt{a}' \ \texttt{<=}\ \texttt{n.value}\ \texttt{<}\ '\texttt{c}'}$
⟨n⟩
$\bigcirc_{(\texttt{n})}$
⟨n⟩

2.3.8 Handling string ranges 4

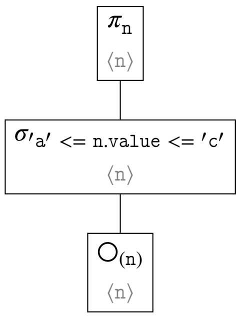
Query specification

```
MATCH (n)
WHERE 'a' <= n.value <= 'c'
RETURN n.value
```

Relational algebra expression

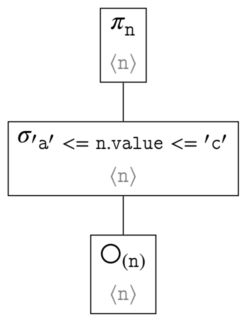
$\pi_{\texttt{n}} \left(\sigma_{'\texttt{a}' \ \texttt{<=}\ \texttt{n.value}\ \texttt{<=}\ '\texttt{c}'} \left(\dot{\not\equiv} \left(\bigcirc_{(\texttt{n})} \right) \right) \right)$

Relational algebra tree



$\pi_{\texttt{n}}$
⟨n⟩
$\sigma_{'\texttt{a}' \ \texttt{<=}\ \texttt{n.value}\ \texttt{<=}\ '\texttt{c}'}$
⟨n⟩
$\bigcirc_{(\texttt{n})}$
⟨n⟩

Relational algebra tree for incremental queries



2.3.9 Handling empty range

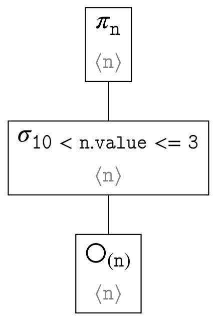
Query specification

```
MATCH (n)
WHERE 10 < n.value <= 3
RETURN n.value
```

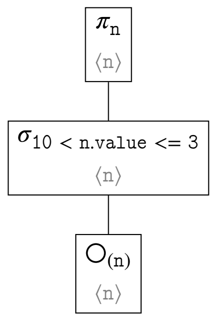
Relational algebra expression

$\pi_{n}\left(\sigma_{10 < n.value <= 3}\left(\not\equiv\left(\bigcirc_{(n)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.3.10 Handling long chains of operators

Query specification

```
MATCH (n)-->(m)
WHERE n.prop1 < m.prop1 = n.prop2 <> m.prop2
RETURN labels(m)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.4 Create

2.4.1 Creating a node

Query specification

```
CREATE ()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.4.2 Creating two nodes

Query specification

```
CREATE (), ()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.4.3 Creating two nodes and a relationship

Query specification

```
CREATE ()-[:TYPE]->()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.4.4 Creating a node with a label

Query specification

```
CREATE (:Label)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.4.5 Creating a node with a property

Query specification

```
CREATE ({created: true})
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5 CreateAcceptance

2.5.1 Create a single node

Query specification

```
CREATE ()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.2 Create a single node with a single label

Query specification

```
CREATE (:A)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.3 Create a single node with multiple labels

Query specification

```
CREATE (:A:B:C:D)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.4 Combine MATCH and CREATE

Query specification

```
MATCH ()
CREATE ()
```

Relational algebra expression

$\not\fallingdotseq \left(\bigcirc_{\texttt{(\_e1)}}\right)$

Relational algebra tree

$$\boxed{\begin{array}{c}\bigcirc_{\texttt{(\_e1)}}\\ \langle\texttt{\_e1}\rangle\end{array}}$$

Relational algebra tree for incremental queries

$$\boxed{\begin{array}{c}\bigcirc_{\texttt{(\_e1)}}\\ \langle\texttt{\_e1}\rangle\end{array}}$$

2.5.5 Combine MATCH, WITH and CREATE

Query specification

```
MATCH ()
CREATE ()
WITH *
MATCH ()
CREATE ()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.6 Newly-created nodes not visible to preceding MATCH

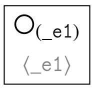
Query specification

```
MATCH ()
CREATE ()
```

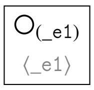
Relational algebra expression

$\not\fallingdotseq\left(\bigcirc_{(\_e1)}\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.5.7 Create a single node with properties

Query specification

```
CREATE (n {prop: 'foo'})
RETURN n.prop AS p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.8 Creating a node with null properties should not return those properties

Query specification

```
CREATE (n {id: 12, property: null})
RETURN n.id AS id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.9 Creating a relationship with null properties should not return those properties

Query specification

```
CREATE ()-[r:X {id: 12, property: null}]->()
RETURN r.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.10 Create a simple pattern

Query specification

```
CREATE ()-[:R]->()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.11 Create a self loop

Query specification

```
CREATE (root:R)-[:LINK]->(root)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.12 Create a self loop using MATCH

Query specification

```
MATCH (root:R)
CREATE (root)-[:LINK]->(root)
```

Relational algebra expression

$\not\fallingdotseq \left(\bigcirc_{(\texttt{root}\,:\,\textsf{R})}\right)$

Relational algebra tree

$\boxed{\begin{array}{c} \bigcirc_{(\texttt{root}\,:\,\textsf{R})} \\ \langle\texttt{root}\rangle \end{array}}$

Relational algebra tree for incremental queries

$\boxed{\begin{array}{c} \bigcirc_{(\texttt{root}\,:\,\textsf{R})} \\ \langle\texttt{root}\rangle \end{array}}$

2.5.13 Create nodes and relationships

Query specification

```
CREATE (a), (b),
       (a)-[:R]->(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.14 Create a relationship with a property

Query specification

```
CREATE ()-[:R {prop: 42}]->()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.15 Create a relationship with the correct direction

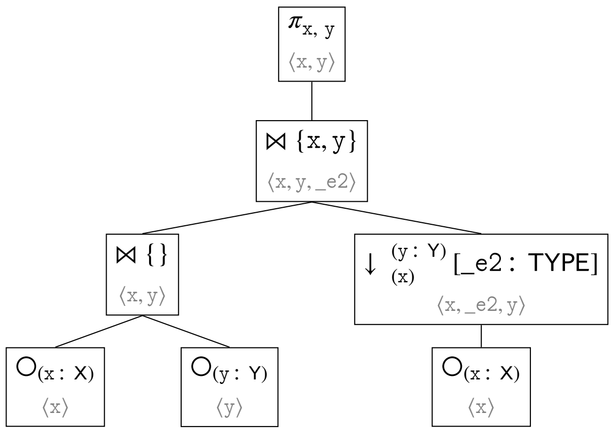
Query specification

```
MATCH (x:X), (y:Y)
CREATE (x)<-[:TYPE]-(y)MATCH (x:X)<-[:TYPE]-(y:Y)
RETURN x, y
```

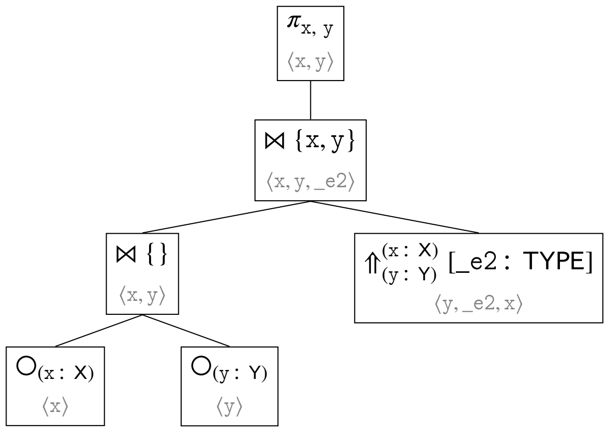
Relational algebra expression

$$\pi_{\mathrm{x,\ y}}\left(\not\doteqdot\left(\bigcirc_{(\mathrm{x:\ X})} \bowtie \{\}\bigcirc_{(\mathrm{y:\ Y})}\right) \bowtie \{\mathrm{x,y}\} \not\doteqdot\left(\downarrow\,_{(\mathrm{x})}^{(\mathrm{y:\ Y})}\,[\_\mathrm{e2:\ TYPE}]\left(\bigcirc_{(\mathrm{x:\ X})}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.5.16 Create a relationship and an end node from a matched starting node

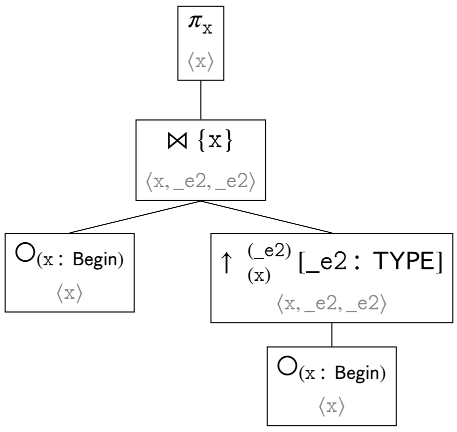
Query specification

```
MATCH (x:Begin)
CREATE (x)-[:TYPE]->(:End)MATCH (x:Begin)-[:TYPE]->()
RETURN x
```

Relational algebra expression

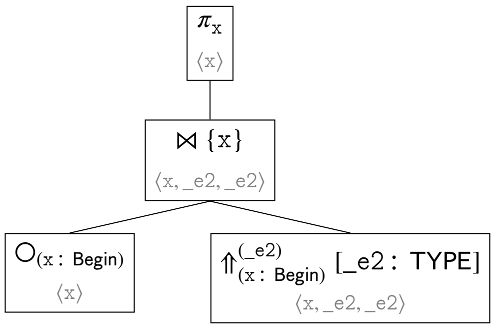
$$\pi_{\mathrm{x}}\left(\not\doteq\left(\bigcirc_{(\mathrm{x}:\ \mathrm{Begin})}\right)\bowtie\{\mathrm{x}\}\not\doteq\left(\uparrow^{(\_\mathrm{e2})}_{(\mathrm{x})}[\_\mathrm{e2}:\ \mathrm{TYPE}]\left(\bigcirc_{(\mathrm{x}:\ \mathrm{Begin})}\right)\right)\right)$$

Relational algebra tree



π_{x}
⟨x⟩
$\bowtie\{\mathrm{x}\}$
⟨x, _e2, _e2⟩
$\bigcirc_{(\mathrm{x}:\ \mathrm{Begin})}$
⟨x⟩
$\uparrow^{(\_\mathrm{e2})}_{(\mathrm{x})}[\_\mathrm{e2}:\ \mathrm{TYPE}]$
⟨x, _e2, _e2⟩
$\bigcirc_{(\mathrm{x}:\ \mathrm{Begin})}$
⟨x⟩

Relational algebra tree for incremental queries



π_{x}
⟨x⟩
$\bowtie\{\mathrm{x}\}$
⟨x, _e2, _e2⟩
$\bigcirc_{(\mathrm{x}:\ \mathrm{Begin})}$
⟨x⟩
$\Uparrow^{(\_\mathrm{e2})}_{(\mathrm{x}:\ \mathrm{Begin})}[\_\mathrm{e2}:\ \mathrm{TYPE}]$
⟨x, _e2, _e2⟩

2.5.17 Create a single node after a WITH

Query specification

```
MATCH ()
CREATE ()
WITH *
CREATE ()
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.18 Create a relationship with a reversed direction

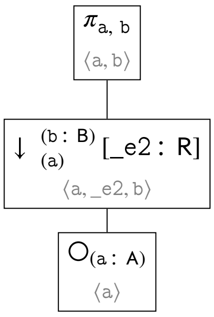
Query specification

```
CREATE (:A)<-[:R]-(:B)MATCH (a:A)<-[:R]-(b:B)
RETURN a, b
```

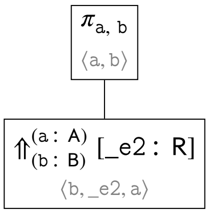
Relational algebra expression

$$\pi_{\mathtt{a,\ b}}\left(\dot{\not\equiv}\left(\downarrow\,^{(\mathtt{b}\,:\,\mathsf{B})}_{(\mathtt{a})}\,[\_\mathsf{e2}:\,\mathsf{R}]\left(\bigcirc_{(\mathtt{a}\,:\,\mathsf{A})}\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{a,\ b}}$
$\langle a, b\rangle$
$\downarrow\,^{(\mathtt{b}\,:\,\mathsf{B})}_{(\mathtt{a})}\,[\_\mathsf{e2}:\,\mathsf{R}]$
$\langle a, \_e2, b\rangle$
$\bigcirc_{(\mathtt{a}\,:\,\mathsf{A})}$
$\langle a\rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{a,\ b}}$
$\langle a, b\rangle$
$\Uparrow^{(\mathtt{a}\,:\,\mathsf{A})}_{(\mathtt{b}\,:\,\mathsf{B})}\,[\_\mathsf{e2}:\,\mathsf{R}]$
$\langle b, \_e2, a\rangle$

2.5.19 Create a pattern with multiple hops

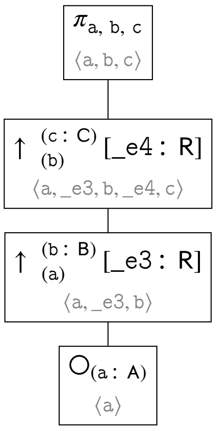
Query specification

```
CREATE (:A)-[:R]->(:B)-[:R]->(:C)MATCH (a:A)-[:R]->(b:B)-[:R]->(c:C)
RETURN a, b, c
```

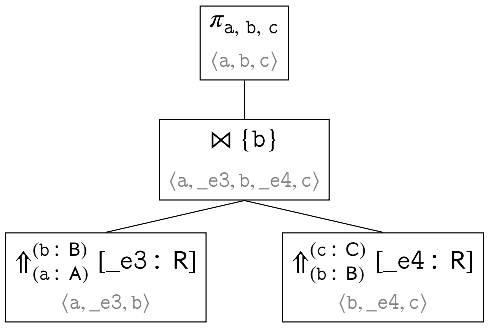
Relational algebra expression

$$\pi_{\mathtt{a,\ b,\ c}}\left(\dot{\not\equiv}\left(\uparrow\,^{(\mathtt{c}\,:\,\mathsf{C})}_{(\mathtt{b})}\,[\_\mathsf{e4}:\,\mathsf{R}]\left(\uparrow\,^{(\mathtt{b}\,:\,\mathsf{B})}_{(\mathtt{a})}\,[\_\mathsf{e3}:\,\mathsf{R}]\left(\bigcirc_{(\mathtt{a}\,:\,\mathsf{A})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{a,\ b,\ c}}$
$\langle a, b, c\rangle$
$\uparrow\,^{(\mathtt{c}\,:\,\mathsf{C})}_{(\mathtt{b})}\,[\_\mathsf{e4}:\,\mathsf{R}]$
$\langle a, \_e3, b, \_e4, c\rangle$
$\uparrow\,^{(\mathtt{b}\,:\,\mathsf{B})}_{(\mathtt{a})}\,[\_\mathsf{e3}:\,\mathsf{R}]$
$\langle a, \_e3, b\rangle$
$\bigcirc_{(\mathtt{a}\,:\,\mathsf{A})}$
$\langle a\rangle$

Relational algebra tree for incremental queries



$\pi_{a, b, c}$
⟨a, b, c⟩
⋈ {b}
⟨a, _e3, b, _e4, c⟩
$\Uparrow^{(b:B)}_{(a:A)}$ [_e3 : R]
⟨a, _e3, b⟩
$\Uparrow^{(c:C)}_{(b:B)}$ [_e4 : R]
⟨b, _e4, c⟩

2.5.20 Create a pattern with multiple hops in the reverse direction

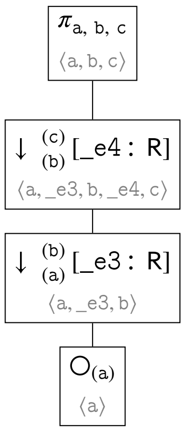
Query specification

```
CREATE (:A)<-[:R]-(:B)<-[:R]-(:C)MATCH (a)<-[:R]-(b)<-[:R]-(c)
RETURN a, b, c
```

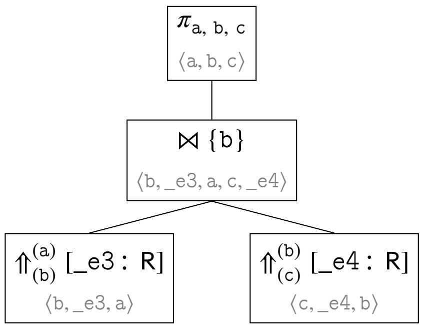
Relational algebra expression

$$\pi_{a,\ b,\ c}\left(\not\equiv\left(\downarrow\ ^{(c)}_{(b)}[\_e4:R]\left(\downarrow\ ^{(b)}_{(a)}[\_e3:R]\left(\bigcirc_{(a)}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.5.21 Create a pattern with multiple hops in varying directions

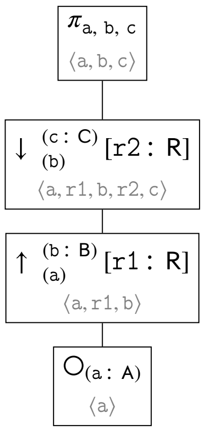
Query specification

```
CREATE (:A)-[:R]->(:B)<-[:R]-(:C)MATCH (a:A)-[r1:R]->(b:B)<-[r2:R]-(c:C)
RETURN a, b, c
```

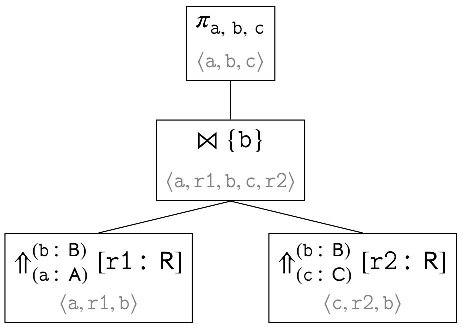
Relational algebra expression

$$\pi_{\mathrm{a,\ b,\ c}}\left(\not\equiv\left(\downarrow\,^{(\mathrm{c}:\,\mathrm{C})}_{(\mathrm{b})}[\mathrm{r2}:\ \mathrm{R}]\left(\uparrow\,^{(\mathrm{b}:\,\mathrm{B})}_{(\mathrm{a})}[\mathrm{r1}:\ \mathrm{R}]\left(\bigcirc_{(\mathrm{a}:\ \mathrm{A})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.5.22 Create a pattern with multiple hops with multiple types and varying directions

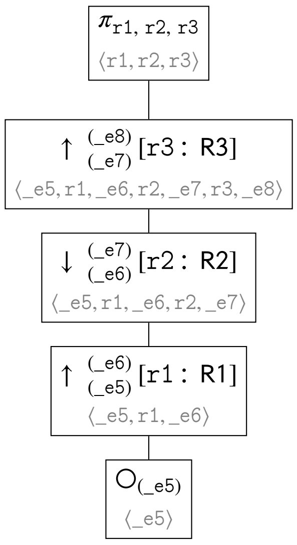
Query specification

```
CREATE ()-[:R1]->()<-[:R2]-()-[:R3]->()MATCH ()-[r1:R1]->()<-[r2:R2]-()-[r3:R3]->()
RETURN r1, r2, r3
```

Relational algebra expression

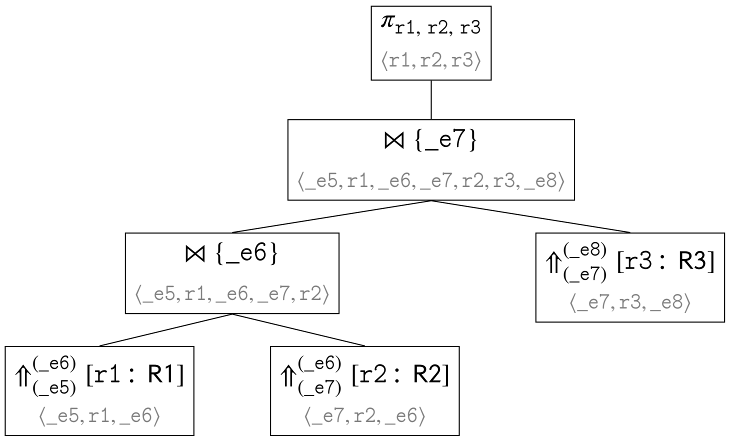
$$\pi_{\mathrm{r1,\ r2,\ r3}}\left(\not\equiv\left(\uparrow\,^{(\_\mathrm{e8})}_{(\_\mathrm{e7})}[\mathrm{r3}:\ \mathrm{R3}]\left(\downarrow\,^{(\_\mathrm{e7})}_{(\_\mathrm{e6})}[\mathrm{r2}:\ \mathrm{R2}]\left(\uparrow\,^{(\_\mathrm{e6})}_{(\_\mathrm{e5})}[\mathrm{r1}:\ \mathrm{R1}]\left(\bigcirc_{(\_\mathrm{e5})}\right)\right)\right)\right)\right)$$

Relational algebra tree



$\pi_{\text{r1, r2, r3}}$
⟨r1, r2, r3⟩
$\uparrow^{(\_e8)}_{(\_e7)}$ [r3 : R3]
⟨_e5, r1, _e6, r2, _e7, r3, _e8⟩
$\downarrow^{(\_e7)}_{(\_e6)}$ [r2 : R2]
⟨_e5, r1, _e6, r2, _e7⟩
$\uparrow^{(\_e6)}_{(\_e5)}$ [r1 : R1]
⟨_e5, r1, _e6⟩
$\bigcirc_{(\_e5)}$
⟨_e5⟩

Relational algebra tree for incremental queries



$\pi_{\text{r1, r2, r3}}$
⟨r1, r2, r3⟩
⋈ {_e7}
⟨_e5, r1, _e6, _e7, r2, r3, _e8⟩
⋈ {_e6}
⟨_e5, r1, _e6, _e7, r2⟩
$\Uparrow^{(\_e8)}_{(\_e7)}$ [r3 : R3]
⟨_e7, r3, _e8⟩
$\Uparrow^{(\_e6)}_{(\_e5)}$ [r1 : R1]
⟨_e5, r1, _e6⟩
$\Uparrow^{(\_e6)}_{(\_e7)}$ [r2 : R2]
⟨_e7, r2, _e6⟩

2.5.23 Nodes are not created when aliases are applied to variable names

Query specification

```
MATCH (n)
MATCH (m)
WITH n AS a, m AS b
CREATE (a)-[:T]->(b)
RETURN a, b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.24 Only a single node is created when an alias is applied to a variable name

Query specification

```
MATCH (n)
WITH n AS a
CREATE (a)-[:T]->()
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.25 Nodes are not created when aliases are applied to variable names multiple times

Query specification

```
MATCH (n)
MATCH (m)
WITH n AS a, m AS b
CREATE (a)-[:T]->(b)
WITH a AS x, b AS y
CREATE (x)-[:T]->(y)
RETURN x, y
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.26 Only a single node is created when an alias is applied to a variable name multiple times

Query specification

```
MATCH (n)
WITH n AS a
CREATE (a)-[:T]->()
WITH a AS x
CREATE (x)-[:T]->()
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.27 A bound node should be recognized after projection with WITH + WITH

Query specification

```
CREATE (a)
WITH a
WITH *
CREATE (b)
CREATE (a)<-[:T]-(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.28 A bound node should be recognized after projection with WITH + UNWIND

Query specification

```
CREATE (a)
WITH a
UNWIND [0] AS i
CREATE (b)
CREATE (a)<-[:T]-(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.29 A bound node should be recognized after projection with WITH + MERGE node

Query specification

```
CREATE (a)
WITH a
MERGE ()
CREATE (b)
CREATE (a)<-[:T]-(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.30 A bound node should be recognized after projection with WITH + MERGE pattern

Query specification

```
CREATE (a)
WITH a
MERGE (x)
MERGE (y)
MERGE (x)-[:T]->(y)
CREATE (b)
CREATE (a)<-[:T]-(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.5.31 Creating a pattern with multiple hops and changing directions

Query specification

```
CREATE (:A)<-[:R1]-(:B)-[:R2]->(:C)MATCH (a:A)<-[r1:R1]-(b:B)-[r2:R2]->(c:C) RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6 DeleteAcceptance

2.6.1 Delete nodes

Query specification

```
MATCH (n)
DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.2 Detach delete node

Query specification

```
MATCH (n)
DETACH DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.3 Delete relationships

Query specification

```
MATCH ()-[r]-()
DELETE r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.4 Deleting connected nodes

Query specification

```
MATCH (n:X)
DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.5 Detach deleting connected nodes and relationships

Query specification

```
MATCH (n:X)
DETACH DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.6 Detach deleting paths

Query specification

```
MATCH p = (:X)-->()-->()-->()
DETACH DELETE p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.7 Undirected expand followed by delete and count

Query specification

```
MATCH (a)-[r]-(b)
DELETE r, a, b
RETURN count(*) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.8 Undirected variable length expand followed by delete and count

Query specification

```
MATCH (a)-[*]-(b)
DETACH DELETE a, b
RETURN count(*) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.9 Create and delete in same query

Query specification

```
MATCH ()
CREATE (n)
DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.10 Delete optionally matched relationship

Query specification

```
MATCH (n)
OPTIONAL MATCH (n)-[r]-()
DELETE n, r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.11 Delete on null node

Query specification

```
OPTIONAL MATCH (n)
DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.12 Detach delete on null node

Query specification

```
OPTIONAL MATCH (n)
DETACH DELETE n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.13 Delete on null path

Query specification

```
OPTIONAL MATCH p = ()-->()
DETACH DELETE p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.14 Delete node from a list

Query specification

```
MATCH (:User)-[:FRIEND]->(n)
WITH collect(n) AS friends
DETACH DELETE friends[$friendIndex]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.15 Delete node from a list

Query specification

```
MATCH (:User)-[:FRIEND]->(n)
WITH collect(n) AS friends
DETACH DELETE friends[$friendIndex]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.16 Delete relationship from a list

Query specification

```
MATCH (:User)-[r:FRIEND]->()
WITH collect(r) AS friendships
DETACH DELETE friendships[$friendIndex]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.17 Delete nodes from a map

Query specification

```
MATCH (u:User)
WITH {key: u} AS nodes
DELETE nodes.key
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.18 Delete relationships from a map

Query specification

```
MATCH (:User)-[r]->(:User)
WITH {key: r} AS rels
DELETE rels.key
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.19 Detach delete nodes from nested map/list

Query specification

```
MATCH (u:User)
WITH {key: collect(u)} AS nodeMap
DETACH DELETE nodeMap.key[0]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.20 Delete relationships from nested map/list

Query specification

```
MATCH (:User)-[r]->(:User)
WITH {key: {key: collect(r)}} AS rels
DELETE rels.key.key[0]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.6.21 Delete paths from nested map/list

Query specification

```
MATCH p = (:User)-[r]->(:User)
WITH {key: collect(p)} AS pathColls
DELETE pathColls.key[0], pathColls.key[1]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.7 EqualsAcceptance

2.7.1 Number-typed integer comparison

Query specification

```
WITH collect([0, 0.0]) AS numbers
UNWIND numbers AS arr
WITH arr[0] AS expected
MATCH (n) WHERE toInteger(n.id) = expected
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.7.2 Number-typed float comparison

Query specification

```
WITH collect([0.5, 0]) AS numbers
UNWIND numbers AS arr
WITH arr[0] AS expected
MATCH (n) WHERE toInteger(n.id) = expected
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.7.3 Any-typed string comparison

Query specification

```
WITH collect(['0', 0]) AS things
UNWIND things AS arr
WITH arr[0] AS expected
MATCH (n) WHERE toInteger(n.id) = expected
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.7.4 Comparing nodes to nodes

Query specification

```
MATCH (a)
WITH a
MATCH (b)
WHERE a = b
RETURN count(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.7.5 Comparing relationships to relationships

Query specification

```
MATCH ()-[a]->()
WITH a
MATCH ()-[b]->()
WHERE a = b
RETURN count(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8 ExpressionAcceptance

2.8.1 Execute n[0]

Query specification

```
RETURN [1, 2, 3][0] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.2 Execute n['name'] in read queries

Query specification

```
MATCH (n {name: 'Apa'})
RETURN n['nam' + 'e'] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.3 Execute n['name'] in update queries

Query specification

```
CREATE (n {name: 'Apa'})
RETURN n['nam' + 'e'] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.4 Use dynamic property lookup based on parameters when there is no type information

Query specification

```
WITH $expr AS expr, $idx AS idx
RETURN expr[idx] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.5 Use dynamic property lookup based on parameters when there is lhs type information

Query specification

```
CREATE (n {name: 'Apa'})
RETURN n[$idx] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.6 Use dynamic property lookup based on parameters when there is rhs type information

Query specification

```
WITH $expr AS expr, $idx AS idx
RETURN expr[toString(idx)] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.7 Use collection lookup based on parameters when there is no type information

Query specification

```
WITH $expr AS expr, $idx AS idx
RETURN expr[idx] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.8 Use collection lookup based on parameters when there is lhs type information

Query specification

```
WITH ['Apa'] AS expr
RETURN expr[$idx] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.8.9 Use collection lookup based on parameters when there is rhs type information

Query specification

```
WITH $expr AS expr, $idx AS idx
RETURN expr[toInteger(idx)] AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9 FunctionsAcceptance

2.9.1 Run coalesce

Query specification

```
MATCH (a)
RETURN coalesce(a.title, a.name)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.2 Functions should return null if they get path containing unbound

Query specification

```
WITH null AS a
OPTIONAL MATCH p = (a)-[r]->()
RETURN length(nodes(p)), type(r), nodes(p), relationships(p)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.3 ‘split()‘

Query specification

```
UNWIND split(’one1two’, ’1’) AS item
RETURN count(item) AS item
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.4 ‘properties()‘ on a node

Query specification

```
MATCH (p:Person)
RETURN properties(p) AS m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.5 ‘properties()‘ on a relationship

Query specification

```
MATCH ()-[r:R]->()
RETURN properties(r) AS m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.6 ‘properties()‘ on a map

Query specification

```
RETURN properties({name: ’Popeye’, level: 9001}) AS m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.7 ‘properties()‘ failing on an integer literal

Query specification

```
RETURN properties(1)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.8 ‘properties()‘ failing on a string literal

Query specification

```
RETURN properties('Cypher')
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.9 ‘properties()‘ failing on a list of booleans

Query specification

```
RETURN properties([true, false])
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.10 ‘properties()‘ on null

Query specification

```
RETURN properties(null)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.11 ‘reverse()‘

Query specification

```
RETURN reverse('raks0')
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.12 ‘exists()‘ with dynamic property lookup

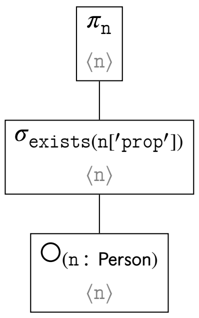
Query specification

```
MATCH (n:Person)
WHERE exists(n['prop'])
RETURN n
```

Relational algebra expression

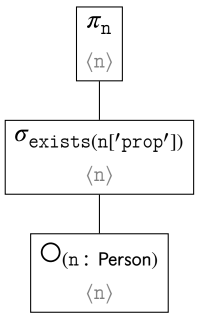
$\pi_{\mathtt{n}}\left(\sigma_{\mathtt{exists(n['prop'])}}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Person})}\right)\right)\right)$

Relational algebra tree



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{\mathtt{exists(n['prop'])}}$
⟨n⟩
$\bigcirc_{(\mathtt{n}:\ \mathsf{Person})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{\mathtt{exists(n['prop'])}}$
⟨n⟩
$\bigcirc_{(\mathtt{n}:\ \mathsf{Person})}$
⟨n⟩

2.9.13 ‘percentileDisc()‘ failing in more involved query

Query specification

```
MATCH (n:S)
WITH n, size([(n)-->() | 1]) AS deg
WHERE deg > 2
WITH deg
LIMIT 100
RETURN percentileDisc(0.90, deg), deg
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.14 ‘type()‘

Query specification

```
MATCH ()-[r]->()
RETURN type(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.15 ‘type()‘ on two relationships

Query specification

```
MATCH ()-[r1]->()-[r2]->()
RETURN type(r1), type(r2)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.16 ‘type()‘ on null relationship

Query specification

```
MATCH (a)
OPTIONAL MATCH (a)-[r:NOT_THERE]->()
RETURN type(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.17 ‘type()‘ on mixed null and non-null relationships

Query specification

```
MATCH (a)
OPTIONAL MATCH (a)-[r:T]->()
RETURN type(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.18 ‘type()‘ handling Any type

Query specification

```
MATCH (a)-[r]->()
WITH [r, 1] AS list
RETURN type(list[0])
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.19 ‘labels()‘ should accept type Any

Query specification

```
MATCH (a)
WITH [a, 1] AS list
RETURN labels(list[0]) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.20 ‘labels()‘ should accept type Any

Query specification

```
MATCH p = (a)
RETURN labels(p) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.21 ‘labels()‘ should accept type Any

Query specification

```
MATCH (a)
WITH [a, 1] AS list
RETURN labels(list[1]) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.9.22 'exists()' is case insensitive

Query specification

```
MATCH (n:X)
RETURN n, EXIsTS(n.prop) AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.10 ValueHashJoinAcceptance

2.10.1 Find friends of others

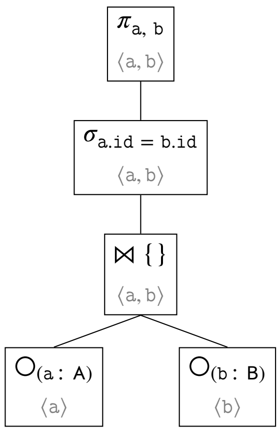
Query specification

```
MATCH (a:A), (b:B)
WHERE a.id = b.id
RETURN a, b
```

Relational algebra expression

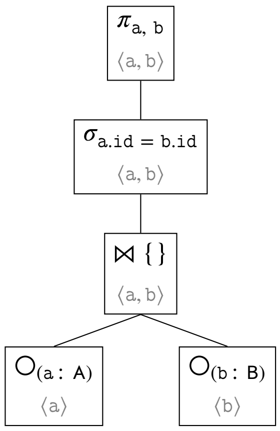
$\pi_{\text{a, b}} \left(\sigma_{\text{a.id = b.id}} \left(\not\equiv \left(\bigcirc_{(\text{a}:\ \text{A})} \bowtie \{\} \bigcirc_{(\text{b}:\ \text{B})} \right) \right) \right)$

Relational algebra tree



$\pi_{\text{a, b}}$
$\langle \text{a}, \text{b} \rangle$
$\sigma_{\text{a.id = b.id}}$
$\langle \text{a}, \text{b} \rangle$
$\bowtie \{\}$
$\langle \text{a}, \text{b} \rangle$
$\bigcirc_{(\text{a}:\ \text{A})}$
$\langle \text{a} \rangle$
$\bigcirc_{(\text{b}:\ \text{B})}$
$\langle \text{b} \rangle$

Relational algebra tree for incremental queries



2.10.2 Should only join when matching

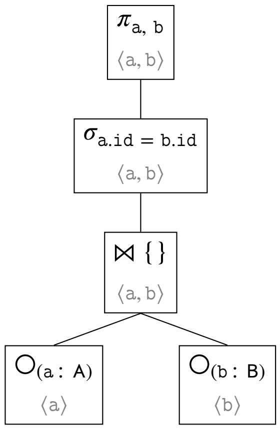
Query specification

```
MATCH (a:A), (b:B)
WHERE a.id = b.id
RETURN a, b
```

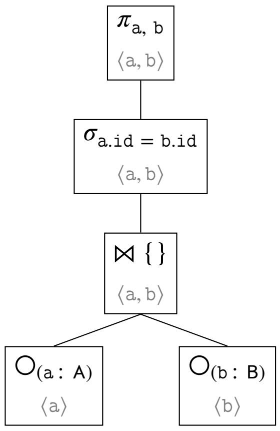
Relational algebra expression

$\pi_{a,\ b}\left(\sigma_{a.id\ =\ b.id}\left(\not\equiv\left(\bigcirc_{(a:\ A)} \bowtie \{\} \bigcirc_{(b:\ B)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.11 KeysAcceptance

2.11.1 Using ‘keys()‘ on a single node, non-empty result

Query specification

```
MATCH (n)
UNWIND keys(n) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.2 Using ‘keys()‘ on multiple nodes, non-empty result

Query specification

```
MATCH (n)
UNWIND keys(n) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.3 Using ‘keys()‘ on a single node, empty result

Query specification

```
MATCH (n)
UNWIND keys(n) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.4 Using ‘keys()‘ on an optionally matched node

Query specification

```
OPTIONAL MATCH (n)
UNWIND keys(n) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.5 Using ‘keys()‘ on a relationship, non-empty result

Query specification

```
MATCH ()-[r:KNOWS]-()
UNWIND keys(r) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.6 Using ‘keys()‘ on a relationship, empty result

Query specification

```
MATCH ()-[r:KNOWS]-()
UNWIND keys(r) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.7 Using ‘keys()‘ on an optionally matched relationship

Query specification

```
OPTIONAL MATCH ()-[r:KNOWS]-()
UNWIND keys(r) AS x
RETURN DISTINCT x AS theProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.8 Using ‘keys()‘ on a literal map

Query specification

```
RETURN keys({name: ’Alice’, age: 38, address: {city: ’London’, residential: true}}) AS k
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.11.9 Using ‘keys()‘ on a parameter map

Query specification

```
RETURN keys($param) AS k
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12 LabelsAcceptance

2.12.1 Adding a single label

Query specification

```
MATCH (n)
SET n:Foo
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.2 Ignore space before colon

Query specification

```
MATCH (n)
SET n :Foo
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.3 Adding multiple labels

Query specification

```
MATCH (n)
SET n:Foo:Bar
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.4 Ignoring intermediate whitespace 1

Query specification

```
MATCH (n)
SET n :Foo :Bar
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.5 Ignoring intermediate whitespace 2

Query specification

```
MATCH (n)
SET n :Foo:Bar
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.6 Creating node without label

Query specification

```
CREATE (node)
RETURN labels(node)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.7 Creating node with two labels

Query specification

```
CREATE (node:Foo:Bar {name: 'Mattias'})
RETURN labels(node)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.8 Ignore space when creating node with labels

Query specification

```
CREATE (node :Foo:Bar)
RETURN labels(node)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.9 Create node with label in pattern

Query specification

```
CREATE (n:Person)-[:OWNS]->(:Dog)
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.10 Using 'labels()' in return clauses

Query specification

```
MATCH (n)
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.11 Removing a label

Query specification

```
MATCH (n)
REMOVE n:Foo
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.12.12 Removing a non-existent label

Query specification

```
MATCH (n)
REMOVE n:Bar
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.13 LargeCreateQuery

2.13.1 Generate the movie graph correctly

Query specification

```
CREATE (theMatrix:Movie {title: ’The Matrix’, released: 1999, tagline: ’Welcome to the Real World’})
CREATE (keanu:Person {name: ’Keanu Reeves’, born: 1964})
CREATE (carrie:Person {name: ’Carrie-Anne Moss’, born: 1967})
CREATE (laurence:Person {name: ’Laurence Fishburne’, born: 1961})
CREATE (hugo:Person {name: ’Hugo Weaving’, born: 1960})
CREATE (andyW:Person {name: ’Andy Wachowski’, born: 1967})
CREATE (lanaW:Person {name: ’Lana Wachowski’, born: 1965})
CREATE (joelS:Person {name: ’Joel Silver’, born: 1952})
CREATE
  (keanu)-[:ACTED_IN {roles: [’Neo’]}]->(theMatrix),
  (carrie)-[:ACTED_IN {roles: [’Trinity’]}]->(theMatrix),
  (laurence)-[:ACTED_IN {roles: [’Morpheus’]}]->(theMatrix),
  (hugo)-[:ACTED_IN {roles: [’Agent Smith’]}]->(theMatrix),
  (andyW)-[:DIRECTED]->(theMatrix),
  (lanaW)-[:DIRECTED]->(theMatrix),
  (joelS)-[:PRODUCED]->(theMatrix)

CREATE (emil:Person {name: ’Emil Eifrem’, born: 1978})
CREATE (emil)-[:ACTED_IN {roles: [’Emil’]}]->(theMatrix)

CREATE (theMatrixReloaded:Movie {title: ’The Matrix Reloaded’, released: 2003,
        tagline: ’Free your mind’})
CREATE
  (keanu)-[:ACTED_IN {roles: [’Neo’] }]->(theMatrixReloaded),
  (carrie)-[:ACTED_IN {roles: [’Trinity’]}]->(theMatrixReloaded),
  (laurence)-[:ACTED_IN {roles: [’Morpheus’]}]->(theMatrixReloaded),
  (hugo)-[:ACTED_IN {roles: [’Agent Smith’]}]->(theMatrixReloaded),
  (andyW)-[:DIRECTED]->(theMatrixReloaded),
  (lanaW)-[:DIRECTED]->(theMatrixReloaded),
  (joelS)-[:PRODUCED]->(theMatrixReloaded)

CREATE (theMatrixRevolutions:Movie {title: ’The Matrix Revolutions’, released: 2003,
  tagline: ’Everything that has a beginning has an end’})
CREATE
  (keanu)-[:ACTED_IN {roles: [’Neo’]}]->(theMatrixRevolutions),
  (carrie)-[:ACTED_IN {roles: [’Trinity’]}]->(theMatrixRevolutions),
  (laurence)-[:ACTED_IN {roles: [’Morpheus’]}]->(theMatrixRevolutions),
  (hugo)-[:ACTED_IN {roles: [’Agent Smith’]}]->(theMatrixRevolutions),
  (andyW)-[:DIRECTED]->(theMatrixRevolutions),
  (lanaW)-[:DIRECTED]->(theMatrixRevolutions),
  (joelS)-[:PRODUCED]->(theMatrixRevolutions)

CREATE (theDevilsAdvocate:Movie {title: ’The Devil\’s Advocate’, released: 1997,
  tagline: ’Evil has its winning ways’})
CREATE (charlize:Person {name: ’Charlize Theron’, born: 1975})
CREATE (al:Person {name: ’Al Pacino’, born: 1940})
CREATE (taylor:Person {name: ’Taylor Hackford’, born: 1944})
CREATE
  (keanu)-[:ACTED_IN {roles: [’Kevin Lomax’]}]->(theDevilsAdvocate),
  (charlize)-[:ACTED_IN {roles: [’Mary Ann Lomax’]}]->(theDevilsAdvocate),
  (al)-[:ACTED_IN {roles: [’John Milton’]}]->(theDevilsAdvocate),
  (taylor)-[:DIRECTED]->(theDevilsAdvocate)

CREATE (aFewGoodMen:Movie {title: ’A Few Good Men’, released: 1992,
  tagline: ’Deep within the heart of the nation\’s capital, one man will stop at nothing to keep his
    honor, ...’})
CREATE (tomC:Person {name: ’Tom Cruise’, born: 1962})
CREATE (jackN:Person {name: ’Jack Nicholson’, born: 1937})
CREATE (demiM:Person {name: ’Demi Moore’, born: 1962})
CREATE (kevinB:Person {name: ’Kevin Bacon’, born: 1958})
CREATE (kieferS:Person {name: ’Kiefer Sutherland’, born: 1966})
CREATE (noahW:Person {name: ’Noah Wyle’, born: 1971})
CREATE (cubaG:Person {name: ’Cuba Gooding Jr.’, born: 1968})
CREATE (kevinP:Person {name: ’Kevin Pollak’, born: 1957})
CREATE (jTW:Person {name: ’J.T. Walsh’, born: 1943})
CREATE (jamesM:Person {name: 'James Marshall', born: 1967})
CREATE (christopherG:Person {name: 'Christopher Guest', born: 1948})
CREATE (robR:Person {name: 'Rob Reiner', born: 1947})
CREATE (aaronS:Person {name: 'Aaron Sorkin', born: 1961})
CREATE
  (tomC)-[:ACTED_IN {roles: ['Lt. Daniel Kaffee']}]->(aFewGoodMen),
  (jackN)-[:ACTED_IN {roles: ['Col. Nathan R. Jessup']}]->(aFewGoodMen),
  (demiM)-[:ACTED_IN {roles: ['Lt. Cdr. JoAnne Galloway']}]->(aFewGoodMen),
  (kevinB)-[:ACTED_IN {roles: ['Capt. Jack Ross']}]->(aFewGoodMen),
  (kieferS)-[:ACTED_IN {roles: ['Lt. Jonathan Kendrick']}]->(aFewGoodMen),
  (noahW)-[:ACTED_IN {roles: ['Cpl. Jeffrey Barnes']}]->(aFewGoodMen),
  (cubaG)-[:ACTED_IN {roles: ['Cpl. Carl Hammaker']}]->(aFewGoodMen),
  (kevinP)-[:ACTED_IN {roles: ['Lt. Sam Weinberg']}]->(aFewGoodMen),
  (jTW)-[:ACTED_IN {roles: ['Lt. Col. Matthew Andrew Markinson']}]->(aFewGoodMen),
  (jamesM)-[:ACTED_IN {roles: ['Pfc. Louden Downey']}]->(aFewGoodMen),
  (christopherG)-[:ACTED_IN {roles: ['Dr. Stone']}]->(aFewGoodMen),
  (aaronS)-[:ACTED_IN {roles: ['Bar patron']}]->(aFewGoodMen),
  (robR)-[:DIRECTED]->(aFewGoodMen),
  (aaronS)-[:WROTE]->(aFewGoodMen)

CREATE (topGun:Movie {title: 'Top Gun', released: 1986,
    tagline: 'I feel the need, the need for speed.'})
CREATE (kellyM:Person {name: 'Kelly McGillis', born: 1957})
CREATE (valK:Person {name: 'Val Kilmer', born: 1959})
CREATE (anthonyE:Person {name: 'Anthony Edwards', born: 1962})
CREATE (tomS:Person {name: 'Tom Skerritt', born: 1933})
CREATE (megR:Person {name: 'Meg Ryan', born: 1961})
CREATE (tonyS:Person {name: 'Tony Scott', born: 1944})
CREATE (jimC:Person {name: 'Jim Cash', born: 1941})
CREATE
  (tomC)-[:ACTED_IN {roles: ['Maverick']}]->(topGun),
  (kellyM)-[:ACTED_IN {roles: ['Charlie']}]->(topGun),
  (valK)-[:ACTED_IN {roles: ['Iceman']}]->(topGun),
  (anthonyE)-[:ACTED_IN {roles: ['Goose']}]->(topGun),
  (tomS)-[:ACTED_IN {roles: ['Viper']}]->(topGun),
  (megR)-[:ACTED_IN {roles: ['Carole']}]->(topGun),
  (tonyS)-[:DIRECTED]->(topGun),
  (jimC)-[:WROTE]->(topGun)

CREATE (jerryMaguire:Movie {title: 'Jerry Maguire', released: 2000,
    tagline: 'The rest of his life begins now.'})
CREATE (reneeZ:Person {name: 'Renee Zellweger', born: 1969})
CREATE (kellyP:Person {name: 'Kelly Preston', born: 1962})
CREATE (jerryO:Person {name: 'Jerry O\'Connell', born: 1974})
CREATE (jayM:Person {name: 'Jay Mohr', born: 1970})
CREATE (bonnieH:Person {name: 'Bonnie Hunt', born: 1961})
CREATE (reginaK:Person {name: 'Regina King', born: 1971})
CREATE (jonathanL:Person {name: 'Jonathan Lipnicki', born: 1996})
CREATE (cameronC:Person {name: 'Cameron Crowe', born: 1957})
CREATE
  (tomC)-[:ACTED_IN {roles: ['Jerry Maguire']}]->(jerryMaguire),
  (cubaG)-[:ACTED_IN {roles: ['Rod Tidwell']}]->(jerryMaguire),
  (reneeZ)-[:ACTED_IN {roles: ['Dorothy Boyd']}]->(jerryMaguire),
  (kellyP)-[:ACTED_IN {roles: ['Avery Bishop']}]->(jerryMaguire),
  (jerryO)-[:ACTED_IN {roles: ['Frank Cushman']}]->(jerryMaguire),
  (jayM)-[:ACTED_IN {roles: ['Bob Sugar']}]->(jerryMaguire),
  (bonnieH)-[:ACTED_IN {roles: ['Laurel Boyd']}]->(jerryMaguire),
  (reginaK)-[:ACTED_IN {roles: ['Marcee Tidwell']}]->(jerryMaguire),
  (jonathanL)-[:ACTED_IN {roles: ['Ray Boyd']}]->(jerryMaguire),
  (cameronC)-[:DIRECTED]->(jerryMaguire),
  (cameronC)-[:PRODUCED]->(jerryMaguire),
  (cameronC)-[:WROTE]->(jerryMaguire)

CREATE (standByMe:Movie {title: 'Stand-By-Me', released: 1986,
    tagline: 'The last real taste of innocence'})
CREATE (riverP:Person {name: 'River Phoenix', born: 1970})
CREATE (coreyF:Person {name: 'Corey Feldman', born: 1971})
CREATE (wilW:Person {name: 'Wil Wheaton', born: 1972})
CREATE (johnC:Person {name: 'John Cusack', born: 1966})
CREATE (marshallB:Person {name: 'Marshall Bell', born: 1942})
CREATE
  (wilW)-[:ACTED_IN {roles: ['Gordie Lachance']}]->(standByMe),
  (riverP)-[:ACTED_IN {roles: ['Chris Chambers']}]->(standByMe),
  (jerryO)-[:ACTED_IN {roles: ['Vern Tessio']}]->(standByMe),
  (coreyF)-[:ACTED_IN {roles: ['Teddy Duchamp']}]->(standByMe),
  (johnC)-[:ACTED_IN {roles: ['Denny Lachance']}]->(standByMe),
  (kieferS)-[:ACTED_IN {roles: ['Ace Merrill']}]->(standByMe),
  (marshallB)-[:ACTED_IN {roles: ['Mr. Lachance']}]->(standByMe),
  (robR)-[:DIRECTED]->(standByMe)

CREATE (asGoodAsItGets:Movie {title: 'As-good-as-it-gets', released: 1997,
    tagline: 'A comedy from the heart that goes for the throat'})
CREATE (helenH:Person {name: 'Helen Hunt', born: 1963})
CREATE (gregK:Person {name: 'Greg Kinnear', born: 1963})
CREATE (jamesB:Person {name: 'James L. Brooks', born: 1940})
CREATE
  (jackN)-[:ACTED_IN {roles: ['Melvin Udall']}]->(asGoodAsItGets),
  (helenH)-[:ACTED_IN {roles: ['Carol Connelly']}]->(asGoodAsItGets),
  (gregK)-[:ACTED_IN {roles: ['Simon Bishop']}]->(asGoodAsItGets),
  (cubaG)-[:ACTED_IN {roles: ['Frank Sachs']}]->(asGoodAsItGets),
  (jamesB)-[:DIRECTED]->(asGoodAsItGets)

CREATE (whatDreamsMayCome:Movie {title: 'What Dreams May Come', released: 1998,
    tagline: 'After life there is more. The end is just the beginning.'})
CREATE (annabellaS:Person {name: 'Annabella Sciorra', born: 1960})
CREATE (maxS:Person {name: 'Max von Sydow', born: 1929})
CREATE (wernerH:Person {name: 'Werner Herzog', born: 1942})
CREATE (robin:Person {name: 'Robin Williams', born: 1951})
CREATE (vincentW:Person {name: 'Vincent Ward', born: 1956})
CREATE
  (robin)-[:ACTED_IN {roles: ['Chris Nielsen']}]->(whatDreamsMayCome),
  (cubaG)-[:ACTED_IN {roles: ['Albert Lewis']}]->(whatDreamsMayCome),
  (annabellaS)-[:ACTED_IN {roles: ['Annie Collins-Nielsen']}]->(whatDreamsMayCome),
  (maxS)-[:ACTED_IN {roles: ['The Tracker']}]->(whatDreamsMayCome),
  (wernerH)-[:ACTED_IN {roles: ['The Face']}]->(whatDreamsMayCome),
  (vincentW)-[:DIRECTED]->(whatDreamsMayCome)

CREATE (snowFallingonCedars:Movie {title: 'Snow-Falling-on-Cedars', released: 1999,
  tagline: 'First loves last. Forever.'})
CREATE (ethanH:Person {name: 'Ethan Hawke', born: 1970})
CREATE (rickY:Person {name: 'Rick Yune', born: 1971})
CREATE (jamesC:Person {name: 'James Cromwell', born: 1940})
CREATE (scottH:Person {name: 'Scott Hicks', born: 1953})
CREATE
  (ethanH)-[:ACTED_IN {roles: ['Ishmael Chambers']}]->(snowFallingonCedars),
  (rickY)-[:ACTED_IN {roles: ['Kazuo Miyamoto']}]->(snowFallingonCedars),
  (maxS)-[:ACTED_IN {roles: ['Nels Gudmundsson']}]->(snowFallingonCedars),
  (jamesC)-[:ACTED_IN {roles: ['Judge Fielding']}]->(snowFallingonCedars),
  (scottH)-[:DIRECTED]->(snowFallingonCedars)

CREATE (youveGotMail:Movie {title: 'You\'ve Got Mail', released: 1998,
    tagline: 'At-odds-in-life, in-love-on-line'})
CREATE (parkerP:Person {name: 'Parker Posey', born: 1968})
CREATE (daveC:Person {name: 'Dave Chappelle', born: 1973})
CREATE (steveZ:Person {name: 'Steve Zahn', born: 1967})
CREATE (tomH:Person {name: 'Tom Hanks', born: 1956})
CREATE (noraE:Person {name: 'Nora Ephron', born: 1941})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Joe Fox']}]->(youveGotMail),
  (megR)-[:ACTED_IN {roles: ['Kathleen Kelly']}]->(youveGotMail),
  (gregK)-[:ACTED_IN {roles: ['Frank Navasky']}]->(youveGotMail),
  (parkerP)-[:ACTED_IN {roles: ['Patricia Eden']}]->(youveGotMail),
  (daveC)-[:ACTED_IN {roles: ['Kevin Jackson']}]->(youveGotMail),
  (steveZ)-[:ACTED_IN {roles: ['George Pappas']}]->(youveGotMail),
  (noraE)-[:DIRECTED]->(youveGotMail)

CREATE (sleeplessInSeattle:Movie {title: 'Sleepless-in-Seattle', released: 1993,
    tagline: 'What if someone you never met, someone you never saw, someone you never knew was the
     only someone for you?'})
CREATE (ritaW:Person {name: 'Rita Wilson', born: 1956})
CREATE (billPull:Person {name: 'Bill Pullman', born: 1953})
CREATE (victorG:Person {name: 'Victor Garber', born: 1949})
CREATE (rosieO:Person {name: 'Rosie O\'Donnell', born: 1962})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Sam Baldwin']}]->(sleeplessInSeattle),
  (megR)-[:ACTED_IN {roles: ['Annie Reed']}]->(sleeplessInSeattle),
  (ritaW)-[:ACTED_IN {roles: ['Suzy']}]->(sleeplessInSeattle),
  (billPull)-[:ACTED_IN {roles: ['Walter']}]->(sleeplessInSeattle),
  (victorG)-[:ACTED_IN {roles: ['Greg']}]->(sleeplessInSeattle),
  (rosieO)-[:ACTED_IN {roles: ['Becky']}]->(sleeplessInSeattle),
  (noraE)-[:DIRECTED]->(sleeplessInSeattle)

CREATE (joeVersustheVolcano:Movie {title: 'Joe-Versus-the-Volcano', released: 1990,
    tagline: 'A story of love'})
CREATE (johnS:Person {name: 'John Patrick Stanley', born: 1950})
CREATE (nathan:Person {name: 'Nathan Lane', born: 1956})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Joe Banks']}]->(joeVersustheVolcano),
  (megR)-[:ACTED_IN {roles: ['DeDe', 'Angelica Graynamore', 'Patricia Graynamore']}]->(
     joeVersustheVolcano),
  (nathan)-[:ACTED_IN {roles: ['Baw']}]->(joeVersustheVolcano),
  (johnS)-[:DIRECTED]->(joeVersustheVolcano)

CREATE (whenHarryMetSally:Movie {title: 'When-Harry-Met-Sally', released: 1998,
    tagline: 'When-Harry-Met-Sally'})
CREATE (billyC:Person {name: 'Billy Crystal', born: 1948})
CREATE (carrieF:Person {name: 'Carrie Fisher', born: 1956})
CREATE (brunoK:Person {name: 'Bruno Kirby', born: 1949})
CREATE
  (billyC)-[:ACTED_IN {roles: ['Harry Burns']}]->(whenHarryMetSally),
  (megR)-[:ACTED_IN {roles: ['Sally Albright']}]->(whenHarryMetSally),
  (carrieF)-[:ACTED_IN {roles: ['Marie']}]->(whenHarryMetSally),
  (brunoK)-[:ACTED_IN {roles: ['Jess']}]->(whenHarryMetSally),
  (robR)-[:DIRECTED]->(whenHarryMetSally),
  (robR)-[:PRODUCED]->(whenHarryMetSally),
  (noraE)-[:PRODUCED]->(whenHarryMetSally),
  (noraE)-[:WROTE]->(whenHarryMetSally)

CREATE (thatThingYouDo:Movie {title: 'That-Thing-You-Do', released: 1996,
    tagline: 'There comes a time...'})
CREATE (livT:Person {name: 'Liv Tyler', born: 1977})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Mr. White']}]->(thatThingYouDo),
  (livT)-[:ACTED_IN {roles: ['Faye Dolan']}]->(thatThingYouDo),
  (charlize)-[:ACTED_IN {roles: ['Tina']}]->(thatThingYouDo),
  (tomH)-[:DIRECTED]->(thatThingYouDo)

CREATE (theReplacements:Movie {title: 'The Replacements', released: 2000,
    tagline: 'Pain heals, Chicks dig scars... Glory lasts forever'})
CREATE (brooke:Person {name: 'Brooke Langton', born: 1970})
CREATE (gene:Person {name: 'Gene Hackman', born: 1930})
CREATE (orlando:Person {name: 'Orlando Jones', born: 1968})
CREATE (howard:Person {name: 'Howard Deutch', born: 1950})
CREATE
  (keanu)-[:ACTED_IN {roles: ['Shane Falco']}]->(theReplacements),
  (brooke)-[:ACTED_IN {roles: ['Annabelle Farrell']}]->(theReplacements),
  (gene)-[:ACTED_IN {roles: ['Jimmy McGinty']}]->(theReplacements),
  (orlando)-[:ACTED_IN {roles: ['Clifford Franklin']}]->(theReplacements),
  (howard)-[:DIRECTED]->(theReplacements)

CREATE (rescueDawn:Movie {title: 'RescueDawn', released: 2006,
    tagline: 'The extraordinary true story'})
CREATE (christianB:Person {name: 'Christian Bale', born: 1974})
CREATE (zachG:Person {name: 'Zach Grenier', born: 1954})
CREATE
  (marshallB)-[:ACTED_IN {roles: ['Admiral']}]->(rescueDawn),
  (christianB)-[:ACTED_IN {roles: ['Dieter Dengler']}]->(rescueDawn),
  (zachG)-[:ACTED_IN {roles: ['Squad Leader']}]->(rescueDawn),
  (steveZ)-[:ACTED_IN {roles: ['Duane']}]->(rescueDawn),
  (wernerH)-[:DIRECTED]->(rescueDawn)

CREATE (theBirdcage:Movie {title: 'The-Birdcage', released: 1996, tagline: 'Come-as-you-are'})
CREATE (mikeN:Person {name: 'Mike Nichols', born: 1931})
CREATE
  (robin)-[:ACTED_IN {roles: ['Armand Goldman']}]->(theBirdcage),
  (nathan)-[:ACTED_IN {roles: ['Albert Goldman']}]->(theBirdcage),
  (gene)-[:ACTED_IN {roles: ['Sen. Kevin Keeley']}]->(theBirdcage),
  (mikeN)-[:DIRECTED]->(theBirdcage)

CREATE (unforgiven:Movie {title: 'Unforgiven', released: 1992,
    tagline: 'It\'s a hell of a thing, killing a man'})
CREATE (richardH:Person {name: 'Richard Harris', born: 1930})
CREATE (clintE:Person {name: 'Clint Eastwood', born: 1930})
CREATE
  (richardH)-[:ACTED_IN {roles: ['English Bob']}]->(unforgiven),
  (clintE)-[:ACTED_IN {roles: ['Bill Munny']}]->(unforgiven),
  (gene)-[:ACTED_IN {roles: ['Little Bill Daggett']}]->(unforgiven),
  (clintE)-[:DIRECTED]->(unforgiven)

CREATE (johnnyMnemonic:Movie {title: 'Johnny-Mnemonic', released: 1995,
    tagline: 'The-hottest-data-in-the-coolest-head'})
CREATE (takeshi:Person {name: 'Takeshi Kitano', born: 1947})
CREATE (dina:Person {name: 'Dina Meyer', born: 1968})
CREATE (iceT:Person {name: 'Ice-T', born: 1958})
CREATE (robertL:Person {name: 'Robert Longo', born: 1953})
CREATE
  (keanu)-[:ACTED_IN {roles: ['Johnny Mnemonic']}]->(johnnyMnemonic),
  (takeshi)-[:ACTED_IN {roles: ['Takahashi']}]->(johnnyMnemonic),
  (dina)-[:ACTED_IN {roles: ['Jane']}]->(johnnyMnemonic),
  (iceT)-[:ACTED_IN {roles: ['J-Bone']}]->(johnnyMnemonic),
  (robertL)-[:DIRECTED]->(johnnyMnemonic)

CREATE (cloudAtlas:Movie {title: 'Cloud Atlas', released: 2012, tagline: 'Everything is connected'})
CREATE (halleB:Person {name: 'Halle Berry', born: 1966})
CREATE (jimB:Person {name: 'Jim Broadbent', born: 1949})
CREATE (tomT:Person {name: 'Tom Tykwer', born: 1965})
CREATE (davidMitchell:Person {name: 'David Mitchell', born: 1969})
CREATE (stefanArndt:Person {name: 'Stefan Arndt', born: 1961})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Zachry', 'Dr. Henry Goose', 'Isaac Sachs', 'Dermot Hoggins']}]->(
    cloudAtlas),
  (hugo)-[:ACTED_IN {roles: ['Bill Smoke', 'Haskell Moore', 'Tadeusz Kesselring', 'Nurse Noakes', '
    Boardman Mephi', 'Old Georgie']}]->(cloudAtlas),
  (halleB)-[:ACTED_IN {roles: ['Luisa Rey', 'Jocasta Ayrs', 'Ovid', 'Meronym']}]->(cloudAtlas),
  (jimB)-[:ACTED_IN {roles: ['Vyvyan Ayrs', 'Captain Molyneux', 'Timothy Cavendish']}]->(cloudAtlas),
  (tomT)-[:DIRECTED]->(cloudAtlas),
  (andyW)-[:DIRECTED]->(cloudAtlas),
  (lanaW)-[:DIRECTED]->(cloudAtlas),
  (davidMitchell)-[:WROTE]->(cloudAtlas),
  (stefanArndt)-[:PRODUCED]->(cloudAtlas)

CREATE (theDaVinciCode:Movie {title: 'The Da Vinci Code', released: 2006, tagline: 'Break The Codes'})
CREATE (ianM:Person {name: 'Ian McKellen', born: 1939})
CREATE (audreyT:Person {name: 'Audrey Tautou', born: 1976})
CREATE (paulB:Person {name: 'Paul Bettany', born: 1971})
CREATE (ronH:Person {name: 'Ron Howard', born: 1954})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Dr. Robert Langdon']}]->(theDaVinciCode),
  (ianM)-[:ACTED_IN {roles: ['Sir Leight Teabing']}]->(theDaVinciCode),
  (audreyT)-[:ACTED_IN {roles: ['Sophie Neveu']}]->(theDaVinciCode),
  (paulB)-[:ACTED_IN {roles: ['Silas']}]->(theDaVinciCode),
  (ronH)-[:DIRECTED]->(theDaVinciCode)

CREATE (vforVendetta:Movie {title: 'V for Vendetta', released: 2006, tagline: 'Freedom! Forever!'})
CREATE (natalieP:Person {name: 'Natalie Portman', born: 1981})
CREATE (stephenR:Person {name: 'Stephen Rea', born: 1946})
CREATE (johnH:Person {name: 'John Hurt', born: 1940})
CREATE (benM:Person {name: 'Ben Miles', born: 1967})
CREATE
  (hugo)-[:ACTED_IN {roles: ['V']}]->(vforVendetta),
  (natalieP)-[:ACTED_IN {roles: ['Evey Hammond']}]->(vforVendetta),
  (stephenR)-[:ACTED_IN {roles: ['Eric Finch']}]->(vforVendetta),
  (johnH)-[:ACTED_IN {roles: ['High Chancellor Adam Sutler']}]->(vforVendetta),
  (benM)-[:ACTED_IN {roles: ['Dascomb']}]->(vforVendetta),
  (jamesM)-[:DIRECTED]->(vforVendetta),
  (andyW)-[:PRODUCED]->(vforVendetta),
  (lanaW)-[:PRODUCED]->(vforVendetta),
  (joelS)-[:PRODUCED]->(vforVendetta),
  (andyW)-[:WROTE]->(vforVendetta),
  (lanaW)-[:WROTE]->(vforVendetta)

CREATE (speedRacer:Movie {title: 'Speed Racer', released: 2008, tagline: 'Speed has no limits'})
CREATE (emileH:Person {name: 'Emile Hirsch', born: 1985})
CREATE (johnG:Person {name: 'John Goodman', born: 1960})
CREATE (susanS:Person {name: 'Susan Sarandon', born: 1946})
CREATE (matthewF:Person {name: 'Matthew Fox', born: 1966})
CREATE (christinaR:Person {name: 'Christina Ricci', born: 1980})
CREATE (rain:Person {name: 'Rain', born: 1982})
CREATE
  (emileH)-[:ACTED_IN {roles: ['Speed Racer']}]->(speedRacer),
  (johnG)-[:ACTED_IN {roles: ['Pops']}]->(speedRacer),
  (susanS)-[:ACTED_IN {roles: ['Mom']}]->(speedRacer),
  (matthewF)-[:ACTED_IN {roles: ['Racer X']}]->(speedRacer),
  (christinaR)-[:ACTED_IN {roles: ['Trixie']}]->(speedRacer),
  (rain)-[:ACTED_IN {roles: ['Taejo Togokahn']}]->(speedRacer),
  (benM)-[:ACTED_IN {roles: ['Cass Jones']}]->(speedRacer),
  (andyW)-[:DIRECTED]->(speedRacer),
  (lanaW)-[:DIRECTED]->(speedRacer),
  (andyW)-[:WROTE]->(speedRacer),
  (lanaW)-[:WROTE]->(speedRacer),
  (joelS)-[:PRODUCED]->(speedRacer)

CREATE (ninjaAssassin:Movie {title: 'Ninja Assassin', released: 2009,
    tagline: 'Prepare to enter a secret world of assassins'})
CREATE (naomieH:Person {name: 'Naomie Harris'})
CREATE
  (rain)-[:ACTED_IN {roles: ['Raizo']}]->(ninjaAssassin),
  (naomieH)-[:ACTED_IN {roles: ['Mika Coretti']}]->(ninjaAssassin),
  (rickY)-[:ACTED_IN {roles: ['Takeshi']}]->(ninjaAssassin),
  (benM)-[:ACTED_IN {roles: ['Ryan Maslow']}]->(ninjaAssassin),
  (jamesM)-[:DIRECTED]->(ninjaAssassin),
  (andyW)-[:PRODUCED]->(ninjaAssassin),
  (lanaW)-[:PRODUCED]->(ninjaAssassin),
  (joelS)-[:PRODUCED]->(ninjaAssassin)

CREATE (theGreenMile:Movie {title: 'The Green Mile', released: 1999,
    tagline: 'Walk a mile you\'ll never forget.'})
CREATE (michaelD:Person {name: 'Michael Clarke Duncan', born: 1957})
CREATE (davidM:Person {name: 'David Morse', born: 1953})
CREATE (samR:Person {name: 'Sam Rockwell', born: 1968})
CREATE (garyS:Person {name: 'Gary Sinise', born: 1955})
CREATE (patriciaC:Person {name: 'Patricia Clarkson', born: 1959})
CREATE (frankD:Person {name: 'Frank Darabont', born: 1959})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Paul Edgecomb']}]->(theGreenMile),
  (michaelD)-[:ACTED_IN {roles: ['John Coffey']}]->(theGreenMile),
  (davidM)-[:ACTED_IN {roles: ['Brutus Brutal Howell']}]->(theGreenMile),
  (bonnieH)-[:ACTED_IN {roles: ['Jan Edgecomb']}]->(theGreenMile),
  (jamesC)-[:ACTED_IN {roles: ['Warden Hal Moores']}]->(theGreenMile),
  (samR)-[:ACTED_IN {roles: ['Wild Bill Wharton']}]->(theGreenMile),
  (garyS)-[:ACTED_IN {roles: ['Burt Hammersmith']}]->(theGreenMile),
  (patriciaC)-[:ACTED_IN {roles: ['Melinda Moores']}]->(theGreenMile),
  (frankD)-[:DIRECTED]->(theGreenMile)

CREATE (frostNixon:Movie {title: 'Frost/Nixon', released: 2008,
    tagline: '400 million people were waiting for the truth.'})
CREATE (frankL:Person {name: 'Frank Langella', born: 1938})
CREATE (michaelS:Person {name: 'Michael Sheen', born: 1969})
CREATE (oliverP:Person {name: 'Oliver Platt', born: 1960})
CREATE
  (frankL)-[:ACTED_IN {roles: ['Richard Nixon']}]->(frostNixon),
  (michaelS)-[:ACTED_IN {roles: ['David Frost']}]->(frostNixon),
  (kevinB)-[:ACTED_IN {roles: ['Jack Brennan']}]->(frostNixon),
  (oliverP)-[:ACTED_IN {roles: ['Bob Zelnick']}]->(frostNixon),
  (samR)-[:ACTED_IN {roles: ['James Reston, Jr.']}]->(frostNixon),
  (ronH)-[:DIRECTED]->(frostNixon)

CREATE (hoffa:Movie {title: 'Hoffa', released: 1992, tagline: "He didn't want law. He wanted justice
    ."})
CREATE (dannyD:Person {name: 'Danny DeVito', born: 1944})
CREATE (johnR:Person {name: 'John C. Reilly', born: 1965})
CREATE
  (jackN)-[:ACTED_IN {roles: ['Hoffa']}]->(hoffa),
  (dannyD)-[:ACTED_IN {roles: ['Robert Bobby Ciaro']}]->(hoffa),
  (jTW)-[:ACTED_IN {roles: ['Frank Fitzsimmons']}]->(hoffa),
  (johnR)-[:ACTED_IN {roles: ['Peter Connelly']}]->(hoffa),
  (dannyD)-[:DIRECTED]->(hoffa)

CREATE (apollo13:Movie {title: 'Apollo 13', released: 1995, tagline: 'Houston, we have a problem.'})
CREATE (edH:Person {name: 'Ed Harris', born: 1950})
CREATE (billPax:Person {name: 'Bill Paxton', born: 1955})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Jim Lovell']}]->(apollo13),
  (kevinB)-[:ACTED_IN {roles: ['Jack Swigert']}]->(apollo13),
  (edH)-[:ACTED_IN {roles: ['Gene Kranz']}]->(apollo13),
  (billPax)-[:ACTED_IN {roles: ['Fred Haise']}]->(apollo13),
  (garyS)-[:ACTED_IN {roles: ['Ken Mattingly']}]->(apollo13),
  (ronH)-[:DIRECTED]->(apollo13)

CREATE (twister:Movie {title: 'Twister', released: 1996, tagline: 'Don\'t Breathe. Don\'t Look Back
    .'})
CREATE (philipH:Person {name: 'Philip Seymour Hoffman', born: 1967})
CREATE (janB:Person {name: 'Jan de Bont', born: 1943})
CREATE
  (billPax)-[:ACTED_IN {roles: ['Bill Harding']}]->(twister),
  (helenH)-[:ACTED_IN {roles: ['Dr. Jo Harding']}]->(twister),
  (zachG)-[:ACTED_IN {roles: ['Eddie']}]->(twister),
  (philipH)-[:ACTED_IN {roles: ['Dustin Davis']}]->(twister),
  (janB)-[:DIRECTED]->(twister)

CREATE (castAway:Movie {title: 'Cast Away', released: 2000,
    tagline: 'At the edge of the world, his journey begins.'})
CREATE (robertZ:Person {name: 'Robert Zemeckis', born: 1951})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Chuck Noland']}]->(castAway),
  (helenH)-[:ACTED_IN {roles: ['Kelly Frears']}]->(castAway),
  (robertZ)-[:DIRECTED]->(castAway)

CREATE (oneFlewOvertheCuckoosNest:Movie {title: 'One Flew Over the Cuckoo\'s Nest', released: 1975,
    tagline: 'If he is crazy, what does that make you?'})
CREATE (milosF:Person {name: 'Milos Forman', born: 1932})
CREATE
  (jackN)-[:ACTED_IN {roles: ['Randle McMurphy']}]->(oneFlewOvertheCuckoosNest),
  (dannyD)-[:ACTED_IN {roles: ['Martini']}]->(oneFlewOvertheCuckoosNest),
  (milosF)-[:DIRECTED]->(oneFlewOvertheCuckoosNest)

CREATE (somethingsGottaGive:Movie {title: 'Something\'s Gotta Give', released: 2003})
CREATE (dianeK:Person {name: 'Diane Keaton', born: 1946})
CREATE (nancyM:Person {name: 'Nancy Meyers', born: 1949})
CREATE
  (jackN)-[:ACTED_IN {roles: ['Harry Sanborn']}]->(somethingsGottaGive),
  (dianeK)-[:ACTED_IN {roles: ['Erica Barry']}]->(somethingsGottaGive),
  (keanu)-[:ACTED_IN {roles: ['Julian Mercer']}]->(somethingsGottaGive),
  (nancyM)-[:DIRECTED]->(somethingsGottaGive),
  (nancyM)-[:PRODUCED]->(somethingsGottaGive),
  (nancyM)-[:WROTE]->(somethingsGottaGive)

CREATE (bicentennialMan:Movie {title: 'Bicentennial Man', released: 1999,
    tagline: 'One robot\'s 200 year journey to become an ordinary man.'})
CREATE (chrisC:Person {name: 'Chris Columbus', born: 1958})
CREATE
  (robin)-[:ACTED_IN {roles: ['Andrew Marin']}]->(bicentennialMan),
  (oliverP)-[:ACTED_IN {roles: ['Rupert Burns']}]->(bicentennialMan),
  (chrisC)-[:DIRECTED]->(bicentennialMan)

CREATE (charlieWilsonsWar:Movie {title: 'Charlie Wilson\'s War', released: 2007,
    tagline: 'A stiff drink. A little mascara. A lot of nerve. Who said they could not bring down the
     Soviet empire.'})
CREATE (juliaR:Person {name: 'Julia Roberts', born: 1967})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Rep. Charlie Wilson']}]->(charlieWilsonsWar),
  (juliaR)-[:ACTED_IN {roles: ['Joanne Herring']}]->(charlieWilsonsWar),
  (philipH)-[:ACTED_IN {roles: ['Gust Avrakotos']}]->(charlieWilsonsWar),
  (mikeN)-[:DIRECTED]->(charlieWilsonsWar)

CREATE (thePolarExpress:Movie {title: 'The Polar Express', released: 2004,
    tagline: 'This Holiday Season... Believe'})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Hero Boy', 'Father', 'Conductor', 'Hobo', 'Scrooge', 'Santa Claus']}]->(
     thePolarExpress),
  (robertZ)-[:DIRECTED]->(thePolarExpress)

CREATE (aLeagueofTheirOwn:Movie {title: 'A League of Their Own', released: 1992,
    tagline: 'A league of their own'})
CREATE (madonna:Person {name: 'Madonna', born: 1954})
CREATE (geenaD:Person {name: 'Geena Davis', born: 1956})
CREATE (loriP:Person {name: 'Lori Petty', born: 1963})
CREATE (pennyM:Person {name: 'Penny Marshall', born: 1943})
CREATE
  (tomH)-[:ACTED_IN {roles: ['Jimmy Dugan']}]->(aLeagueofTheirOwn),
  (geenaD)-[:ACTED_IN {roles: ['Dottie Hinson']}]->(aLeagueofTheirOwn),
  (loriP)-[:ACTED_IN {roles: ['Kit Keller']}]->(aLeagueofTheirOwn),
  (rosieO)-[:ACTED_IN {roles: ['Doris Murphy']}]->(aLeagueofTheirOwn),
  (madonna)-[:ACTED_IN {roles: ['Mae Mordabito']}]->(aLeagueofTheirOwn),
  (billPax)-[:ACTED_IN {roles: ['Bob Hinson']}]->(aLeagueofTheirOwn),
  (pennyM)-[:DIRECTED]->(aLeagueofTheirOwn)

CREATE (paulBlythe:Person {name: 'Paul Blythe'})
CREATE (angelaScope:Person {name: 'Angela Scope'})
CREATE (jessicaThompson:Person {name: 'Jessica Thompson'})
CREATE (jamesThompson:Person {name: 'James Thompson'})

```

```
CREATE
  (jamesThompson)-[:FOLLOWS]->(jessicaThompson),
  (angelaScope)-[:FOLLOWS]->(jessicaThompson),
  (paulBlythe)-[:FOLLOWS]->(angelaScope)

CREATE
  (jessicaThompson)-[:REVIEWED {summary: 'An amazing journey', rating: 95}]->(cloudAtlas),
  (jessicaThompson)-[:REVIEWED {summary: 'Silly, but fun', rating: 65}]->(theReplacements),
  (jamesThompson)-[:REVIEWED {summary: 'The coolest football movie ever', rating: 100}]->(
    theReplacements),
  (angelaScope)-[:REVIEWED {summary: 'Pretty funny at times', rating: 62}]->(theReplacements),
  (jessicaThompson)-[:REVIEWED {summary: 'Dark, but compelling', rating: 85}]->(unforgiven),
  (jessicaThompson)-[:REVIEWED {summary: 'Slapstick', rating: 45}]->(theBirdcage),
  (jessicaThompson)-[:REVIEWED {summary: 'A solid romp', rating: 68}]->(theDaVinciCode),
  (jamesThompson)-[:REVIEWED {summary: 'Fun, but a little far fetched', rating: 65}]->(theDaVinciCode)
    ,
  (jessicaThompson)-[:REVIEWED {summary: 'You had me at Jerry', rating: 92}]->(jerryMaguire)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.13.2 Many CREATE clauses

Query specification

```
CREATE (hf:School {name: 'Hilly Fields Technical College'})
CREATE (hf)-[:STAFF]->(mrb:Teacher {name: 'Mr Balls'})
CREATE (hf)-[:STAFF]->(mrspb:Teacher {name: 'Ms Packard-Bell'})
CREATE (hf)-[:STAFF]->(mrs:Teacher {name: 'Mr Smith'})
CREATE (hf)-[:STAFF]->(mrsa:Teacher {name: 'Mrs Adenough'})
CREATE (hf)-[:STAFF]->(mrvdg:Teacher {name: 'Mr Van der Graaf'})
CREATE (hf)-[:STAFF]->(msn:Teacher {name: 'Ms Noethe'})
CREATE (hf)-[:STAFF]->(mrsn:Teacher {name: 'Mrs Noakes'})
CREATE (hf)-[:STAFF]->(mrm:Teacher {name: 'Mr Marker'})
CREATE (hf)-[:STAFF]->(msd:Teacher {name: 'Ms Delgado'})
CREATE (hf)-[:STAFF]->(mrsg:Teacher {name: 'Mrs Glass'})
CREATE (hf)-[:STAFF]->(mrf:Teacher {name: 'Mr Flint'})
CREATE (hf)-[:STAFF]->(mrk:Teacher {name: 'Mr Kearney'})
CREATE (hf)-[:STAFF]->(msf:Teacher {name: 'Mrs Forrester'})
CREATE (hf)-[:STAFF]->(mrsf:Teacher {name: 'Mrs Fischer'})
CREATE (hf)-[:STAFF]->(mrj:Teacher {name: 'Mr Jameson'})

CREATE (hf)-[:STUDENT]->(_001:Student {name: 'Portia Vasquez'})
CREATE (hf)-[:STUDENT]->(_002:Student {name: 'Andrew Parks'})
CREATE (hf)-[:STUDENT]->(_003:Student {name: 'Germane Frye'})
CREATE (hf)-[:STUDENT]->(_004:Student {name: 'Yuli Gutierrez'})
CREATE (hf)-[:STUDENT]->(_005:Student {name: 'Kamal Solomon'})
CREATE (hf)-[:STUDENT]->(_006:Student {name: 'Lysandra Porter'})
CREATE (hf)-[:STUDENT]->(_007:Student {name: 'Stella Santiago'})
CREATE (hf)-[:STUDENT]->(_008:Student {name: 'Brenda Torres'})
CREATE (hf)-[:STUDENT]->(_009:Student {name: 'Heidi Dunlap'})

CREATE (hf)-[:STUDENT]->(_010:Student {name: 'Halee Taylor'})
CREATE (hf)-[:STUDENT]->(_011:Student {name: 'Brennan Crosby'})
CREATE (hf)-[:STUDENT]->(_012:Student {name: 'Rooney Cook'})
CREATE (hf)-[:STUDENT]->(_013:Student {name: 'Xavier Morrison'})
CREATE (hf)-[:STUDENT]->(_014:Student {name: 'Zelenia Santana'})
CREATE (hf)-[:STUDENT]->(_015:Student {name: 'Eaton Bonner'})
CREATE (hf)-[:STUDENT]->(_016:Student {name: 'Leilani Bishop'})
CREATE (hf)-[:STUDENT]->(_017:Student {name: 'Jamalia Pickett'})
CREATE (hf)-[:STUDENT]->(_018:Student {name: 'Wynter Russell'})
CREATE (hf)-[:STUDENT]->(_019:Student {name: 'Liberty Melton'})

CREATE (hf)-[:STUDENT]->(_020:Student {name: 'MacKensie Obrien'})
CREATE (hf)-[:STUDENT]->(_021:Student {name: 'Oprah Maynard'})
CREATE (hf)-[:STUDENT]->(_022:Student {name: 'Lyle Parks'})
CREATE (hf)-[:STUDENT]->(_023:Student {name: 'Madonna Justice'})
CREATE (hf)-[:STUDENT]->(_024:Student {name: 'Herman Frederick'})
CREATE (hf)-[:STUDENT]->(_025:Student {name: 'Preston Stevenson'})
CREATE (hf)-[:STUDENT]->(_026:Student {name: 'Drew Carrillo'})
CREATE (hf)-[:STUDENT]->(_027:Student {name: 'Hamilton Woodward'})
CREATE (hf)-[:STUDENT]->(_028:Student {name: 'Buckminster Bradley'})
CREATE (hf)-[:STUDENT]->(_029:Student {name: 'Shea Cote'})

CREATE (hf)-[:STUDENT]->(_030:Student {name: 'Raymond Leonard'})
CREATE (hf)-[:STUDENT]->(_031:Student {name: 'Gavin Branch'})
CREATE (hf)-[:STUDENT]->(_032:Student {name: 'Kylan Powers'})
CREATE (hf)-[:STUDENT]->(_033:Student {name: 'Hedy Bowers'})
CREATE (hf)-[:STUDENT]->(_034:Student {name: 'Derek Church'})
CREATE (hf)-[:STUDENT]->(_035:Student {name: 'Silas Santiago'})
CREATE (hf)-[:STUDENT]->(_036:Student {name: 'Elton Bright'})
CREATE (hf)-[:STUDENT]->(_037:Student {name: 'Dora Schmidt'})
CREATE (hf)-[:STUDENT]->(_038:Student {name: 'Julian Sullivan'})
CREATE (hf)-[:STUDENT]->(_039:Student {name: 'Willow Morton'})

CREATE (hf)-[:STUDENT]->(_040:Student {name: 'Blaze Hines'})
CREATE (hf)-[:STUDENT]->(_041:Student {name: 'Felicia Tillman'})
CREATE (hf)-[:STUDENT]->(_042:Student {name: 'Ralph Webb'})
CREATE (hf)-[:STUDENT]->(_043:Student {name: 'Roth Gilmore'})
CREATE (hf)-[:STUDENT]->(_044:Student {name: 'Dorothy Burgess'})
CREATE (hf)-[:STUDENT]->(_045:Student {name: 'Lana Sandoval'})
CREATE (hf)-[:STUDENT]->(_046:Student {name: 'Nevada Strickland'})
CREATE (hf)-[:STUDENT]->(_047:Student {name: 'Lucian Franco'})
CREATE (hf)-[:STUDENT]->(_048:Student {name: 'Jasper Talley'})
CREATE (hf)-[:STUDENT]->(_049:Student {name: 'Madaline Spears'})

CREATE (hf)-[:STUDENT]->(_050:Student {name: 'Upton Browning'})
CREATE (hf)-[:STUDENT]->(_051:Student {name: 'Cooper Leon'})
CREATE (hf)-[:STUDENT]->(_052:Student {name: 'Celeste Ortega'})
CREATE (hf)-[:STUDENT]->(_053:Student {name: 'Willa Hewitt'})
CREATE (hf)-[:STUDENT]->(_054:Student {name: 'Rooney Bryan'})
CREATE (hf)-[:STUDENT]->(_055:Student {name: 'Nayda Hays'})
CREATE (hf)-[:STUDENT]->(_056:Student {name: 'Kadeem Salazar'})
CREATE (hf)-[:STUDENT]->(_057:Student {name: 'Halee Allen'})
CREATE (hf)-[:STUDENT]->(_058:Student {name: 'Odysseus Mayo'})
CREATE (hf)-[:STUDENT]->(_059:Student {name: 'Kato Merrill'})

CREATE (hf)-[:STUDENT]->(_060:Student {name: 'Halee Juarez'})
CREATE (hf)-[:STUDENT]->(_061:Student {name: 'Chloe Charles'})
CREATE (hf)-[:STUDENT]->(_062:Student {name: 'Abel Montoya'})
CREATE (hf)-[:STUDENT]->(_063:Student {name: 'Hilda Welch'})
CREATE (hf)-[:STUDENT]->(_064:Student {name: 'Britanni Bean'})
CREATE (hf)-[:STUDENT]->(_065:Student {name: 'Joelle Beach'})
CREATE (hf)-[:STUDENT]->(_066:Student {name: 'Ciara Odom'})
CREATE (hf)-[:STUDENT]->(_067:Student {name: 'Zia Williams'})
CREATE (hf)-[:STUDENT]->(_068:Student {name: 'Darrel Bailey'})
CREATE (hf)-[:STUDENT]->(_069:Student {name: 'Lance Mcdowell'})

CREATE (hf)-[:STUDENT]->(_070:Student {name: 'Clayton Bullock'})
CREATE (hf)-[:STUDENT]->(_071:Student {name: 'Roanna Mosley'})
CREATE (hf)-[:STUDENT]->(_072:Student {name: 'Amethyst Mcclure'})
CREATE (hf)-[:STUDENT]->(_073:Student {name: 'Hanae Mann'})
CREATE (hf)-[:STUDENT]->(_074:Student {name: 'Graiden Haynes'})
CREATE (hf)-[:STUDENT]->(_075:Student {name: 'Marcia Byrd'})
CREATE (hf)-[:STUDENT]->(_076:Student {name: 'Yoshi Joyce'})
CREATE (hf)-[:STUDENT]->(_077:Student {name: 'Gregory Sexton'})
CREATE (hf)-[:STUDENT]->(_078:Student {name: 'Nash Carey'})
CREATE (hf)-[:STUDENT]->(_079:Student {name: 'Rae Stevens'})

CREATE (hf)-[:STUDENT]->(_080:Student {name: 'Blossom Fulton'})
CREATE (hf)-[:STUDENT]->(_081:Student {name: 'Lev Curry'})
CREATE (hf)-[:STUDENT]->(_082:Student {name: 'Margaret Gamble'})
CREATE (hf)-[:STUDENT]->(_083:Student {name: 'Rylee Patterson'})
CREATE (hf)-[:STUDENT]->(_084:Student {name: 'Harper Perkins'})
CREATE (hf)-[:STUDENT]->(_085:Student {name: 'Kennan Murphy'})
CREATE (hf)-[:STUDENT]->(_086:Student {name: 'Hilda Coffey'})
CREATE (hf)-[:STUDENT]->(_087:Student {name: 'Marah Reed'})
CREATE (hf)-[:STUDENT]->(_088:Student {name: 'Blaine Wade'})
CREATE (hf)-[:STUDENT]->(_089:Student {name: 'Geraldine Sanders'})

CREATE (hf)-[:STUDENT]->(_090:Student {name: 'Kerry Rollins'})
CREATE (hf)-[:STUDENT]->(_091:Student {name: 'Virginia Sweet'})
CREATE (hf)-[:STUDENT]->(_092:Student {name: 'Sophia Merrill'})
CREATE (hf)-[:STUDENT]->(_093:Student {name: 'Hedda Carson'})
CREATE (hf)-[:STUDENT]->(_094:Student {name: 'Tamekah Charles'})
CREATE (hf)-[:STUDENT]->(_095:Student {name: 'Knox Barton'})
CREATE (hf)-[:STUDENT]->(_096:Student {name: 'Ariel Porter'})
CREATE (hf)-[:STUDENT]->(_097:Student {name: 'Berk Wooten'})
CREATE (hf)-[:STUDENT]->(_098:Student {name: 'Galena Glenn'})
CREATE (hf)-[:STUDENT]->(_099:Student {name: 'Jolene Anderson'})

CREATE (hf)-[:STUDENT]->(_100:Student {name: 'Leonard Hewitt'})
CREATE (hf)-[:STUDENT]->(_101:Student {name: 'Maris Salazar'})
CREATE (hf)-[:STUDENT]->(_102:Student {name: 'Brian Frost'})
CREATE (hf)-[:STUDENT]->(_103:Student {name: 'Zane Moses'})
CREATE (hf)-[:STUDENT]->(_104:Student {name: 'Serina Finch'})
CREATE (hf)-[:STUDENT]->(_105:Student {name: 'Anastasia Fletcher'})
CREATE (hf)-[:STUDENT]->(_106:Student {name: 'Glenna Chapman'})
CREATE (hf)-[:STUDENT]->(_107:Student {name: 'Mufutau Gillespie'})
CREATE (hf)-[:STUDENT]->(_108:Student {name: 'Basil Guthrie'})
CREATE (hf)-[:STUDENT]->(_109:Student {name: 'Theodore Marsh'})

CREATE (hf)-[:STUDENT]->(_110:Student {name: 'Jaime Contreras'})
CREATE (hf)-[:STUDENT]->(_111:Student {name: 'Irma Poole'})
CREATE (hf)-[:STUDENT]->(_112:Student {name: 'Buckminster Bender'})
CREATE (hf)-[:STUDENT]->(_113:Student {name: 'Elton Morris'})
CREATE (hf)-[:STUDENT]->(_114:Student {name: 'Barbara Nguyen'})
CREATE (hf)-[:STUDENT]->(_115:Student {name: 'Tanya Kidd'})
CREATE (hf)-[:STUDENT]->(_116:Student {name: 'Kaden Hoover'})
CREATE (hf)-[:STUDENT]->(_117:Student {name: 'Christopher Bean'})
CREATE (hf)-[:STUDENT]->(_118:Student {name: 'Trevor Daugherty'})
CREATE (hf)-[:STUDENT]->(_119:Student {name: 'Rudyard Bates'})

CREATE (hf)-[:STUDENT]->(_120:Student {name: 'Stacy Monroe'})
CREATE (hf)-[:STUDENT]->(_121:Student {name: 'Kieran Keller'})
CREATE (hf)-[:STUDENT]->(_122:Student {name: 'Ivy Garrison'})
CREATE (hf)-[:STUDENT]->(_123:Student {name: 'Miranda Haynes'})
CREATE (hf)-[:STUDENT]->(_124:Student {name: 'Abigail Heath'})
CREATE (hf)-[:STUDENT]->(_125:Student {name: 'Margaret Santiago'})
CREATE (hf)-[:STUDENT]->(_126:Student {name: 'Cade Floyd'})
CREATE (hf)-[:STUDENT]->(_127:Student {name: 'Allen Crane'})
CREATE (hf)-[:STUDENT]->(_128:Student {name: 'Stella Gilliam'})
CREATE (hf)-[:STUDENT]->(_129:Student {name: 'Rashad Miller'})

CREATE (hf)-[:STUDENT]->(_130:Student {name: 'Francis Cox'})
CREATE (hf)-[:STUDENT]->(_131:Student {name: 'Darryl Rosario'})
CREATE (hf)-[:STUDENT]->(_132:Student {name: 'Michael Daniels'})
CREATE (hf)-[:STUDENT]->(_133:Student {name: 'Aretha Henderson'})
CREATE (hf)-[:STUDENT]->(_134:Student {name: 'Roth Barrera'})
CREATE (hf)-[:STUDENT]->(_135:Student {name: 'Yael Day'})
CREATE (hf)-[:STUDENT]->(_136:Student {name: 'Wynter Richmond'})
CREATE (hf)-[:STUDENT]->(_137:Student {name: 'Quyn Flowers'})
CREATE (hf)-[:STUDENT]->(_138:Student {name: 'Yvette Marquez'})
CREATE (hf)-[:STUDENT]->(_139:Student {name: 'Teagan Curry'})

CREATE (hf)-[:STUDENT]->(_140:Student {name: 'Brenden Bishop'})
CREATE (hf)-[:STUDENT]->(_141:Student {name: 'Montana Black'})
CREATE (hf)-[:STUDENT]->(_142:Student {name: 'Ramona Parker'})
CREATE (hf)-[:STUDENT]->(_143:Student {name: 'Merritt Hansen'})
CREATE (hf)-[:STUDENT]->(_144:Student {name: 'Melvin Vang'})
CREATE (hf)-[:STUDENT]->(_145:Student {name: 'Samantha Perez'})
CREATE (hf)-[:STUDENT]->(_146:Student {name: 'Thane Porter'})
CREATE (hf)-[:STUDENT]->(_147:Student {name: 'Vaughan Haynes'})
CREATE (hf)-[:STUDENT]->(_148:Student {name: 'Irma Miles'})
CREATE (hf)-[:STUDENT]->(_149:Student {name: 'Amery Jensen'})

CREATE (hf)-[:STUDENT]->(_150:Student {name: 'Montana Holman'})
CREATE (hf)-[:STUDENT]->(_151:Student {name: 'Kimberly Langley'})
CREATE (hf)-[:STUDENT]->(_152:Student {name: 'Ebony Bray'})
CREATE (hf)-[:STUDENT]->(_153:Student {name: 'Ishmael Pollard'})
CREATE (hf)-[:STUDENT]->(_154:Student {name: 'Illana Thompson'})
CREATE (hf)-[:STUDENT]->(_155:Student {name: 'Rhona Bowers'})
CREATE (hf)-[:STUDENT]->(_156:Student {name: 'Lilah Dotson'})
CREATE (hf)-[:STUDENT]->(_157:Student {name: 'Shelly Roach'})
CREATE (hf)-[:STUDENT]->(_158:Student {name: 'Celeste Woodward'})
CREATE (hf)-[:STUDENT]->(_159:Student {name: 'Christen Lynn'})

CREATE (hf)-[:STUDENT]->(_160:Student {name: 'Miranda Slater'})
CREATE (hf)-[:STUDENT]->(_161:Student {name: 'Lunea Clements'})
CREATE (hf)-[:STUDENT]->(_162:Student {name: 'Lester Francis'})
CREATE (hf)-[:STUDENT]->(_163:Student {name: 'David Fischer'})
CREATE (hf)-[:STUDENT]->(_164:Student {name: 'Kyra Bean'})
CREATE (hf)-[:STUDENT]->(_165:Student {name: 'Imelda Alston'})
CREATE (hf)-[:STUDENT]->(_166:Student {name: 'Finn Farrell'})
CREATE (hf)-[:STUDENT]->(_167:Student {name: 'Kirby House'})
CREATE (hf)-[:STUDENT]->(_168:Student {name: 'Amanda Zamora'})
CREATE (hf)-[:STUDENT]->(_169:Student {name: 'Rina Franco'})

CREATE (hf)-[:STUDENT]->(_170:Student {name: 'Sonia Lane'})
CREATE (hf)-[:STUDENT]->(_171:Student {name: 'Nora Jefferson'})
CREATE (hf)-[:STUDENT]->(_172:Student {name: 'Colton Ortiz'})
CREATE (hf)-[:STUDENT]->(_173:Student {name: 'Alden Munoz'})
CREATE (hf)-[:STUDENT]->(_174:Student {name: 'Ferdinand Cline'})
CREATE (hf)-[:STUDENT]->(_175:Student {name: 'Cynthia Prince'})
CREATE (hf)-[:STUDENT]->(_176:Student {name: 'Asher Hurst'})
CREATE (hf)-[:STUDENT]->(_177:Student {name: 'MacKensie Stevenson'})
CREATE (hf)-[:STUDENT]->(_178:Student {name: 'Sydnee Sosa'})
CREATE (hf)-[:STUDENT]->(_179:Student {name: 'Dante Callahan'})

CREATE (hf)-[:STUDENT]->(_180:Student {name: 'Isabella Santana'})
CREATE (hf)-[:STUDENT]->(_181:Student {name: 'Raven Bowman'})
CREATE (hf)-[:STUDENT]->(_182:Student {name: 'Kirby Bolton'})
CREATE (hf)-[:STUDENT]->(_183:Student {name: 'Peter Shaffer'})
CREATE (hf)-[:STUDENT]->(_184:Student {name: 'Fletcher Beard'})
CREATE (hf)-[:STUDENT]->(_185:Student {name: 'Irene Lowe'})
CREATE (hf)-[:STUDENT]->(_186:Student {name: 'Ella Talley'})
CREATE (hf)-[:STUDENT]->(_187:Student {name: 'Jorden Kerr'})
CREATE (hf)-[:STUDENT]->(_188:Student {name: 'Macey Delgado'})
CREATE (hf)-[:STUDENT]->(_189:Student {name: 'Ulysses Graves'})

CREATE (hf)-[:STUDENT]->(_190:Student {name: 'Declan Blake'})
CREATE (hf)-[:STUDENT]->(_191:Student {name: 'Lila Hurst'})
CREATE (hf)-[:STUDENT]->(_192:Student {name: 'David Rasmussen'})
CREATE (hf)-[:STUDENT]->(_193:Student {name: 'Desiree Cortez'})
CREATE (hf)-[:STUDENT]->(_194:Student {name: 'Myles Horton'})
CREATE (hf)-[:STUDENT]->(_195:Student {name: 'Rylee Willis'})
CREATE (hf)-[:STUDENT]->(_196:Student {name: 'Kelsey Yates'})
CREATE (hf)-[:STUDENT]->(_197:Student {name: 'Alika Stanton'})
CREATE (hf)-[:STUDENT]->(_198:Student {name: 'Ria Campos'})
CREATE (hf)-[:STUDENT]->(_199:Student {name: 'Elijah Hendricks'})

CREATE (hf)-[:STUDENT]->(_200:Student {name: 'Hayes House'})

CREATE (hf)-[:DEPARTMENT]->(md:Department {name: 'Mathematics'})
CREATE (hf)-[:DEPARTMENT]->(sd:Department {name: 'Science'})
CREATE (hf)-[:DEPARTMENT]->(ed:Department {name: 'Engineering'})

CREATE (pm:Subject {name: 'Pure Mathematics'})
CREATE (am:Subject {name: 'Applied Mathematics'})
CREATE (ph:Subject {name: 'Physics'})
CREATE (ch:Subject {name: 'Chemistry'})
CREATE (bi:Subject {name: 'Biology'})
CREATE (es:Subject {name: 'Earth Science'})
CREATE (me:Subject {name: 'Mechanical Engineering'})
CREATE (ce:Subject {name: 'Chemical Engineering'})
CREATE (se:Subject {name: 'Systems Engineering'})
CREATE (ve:Subject {name: 'Civil Engineering'})
CREATE (ee:Subject {name: 'Electrical Engineering'})

CREATE (sd)-[:CURRICULUM]->(ph)
CREATE (sd)-[:CURRICULUM]->(ch)
CREATE (sd)-[:CURRICULUM]->(bi)
CREATE (sd)-[:CURRICULUM]->(es)
CREATE (md)-[:CURRICULUM]->(pm)
CREATE (md)-[:CURRICULUM]->(am)
CREATE (ed)-[:CURRICULUM]->(me)
CREATE (ed)-[:CURRICULUM]->(se)
CREATE (ed)-[:CURRICULUM]->(ce)
CREATE (ed)-[:CURRICULUM]->(ee)
CREATE (ed)-[:CURRICULUM]->(ve)

CREATE (ph)-[:TAUGHT_BY]->(mrb)
CREATE (ph)-[:TAUGHT_BY]->(mrk)
CREATE (ch)-[:TAUGHT_BY]->(mrk)
CREATE (ch)-[:TAUGHT_BY]->(mrsn)
CREATE (bi)-[:TAUGHT_BY]->(mrsn)
CREATE (bi)-[:TAUGHT_BY]->(mrsf)
CREATE (es)-[:TAUGHT_BY]->(msn)
CREATE (pm)-[:TAUGHT_BY]->(mrf)
CREATE (pm)-[:TAUGHT_BY]->(mrm)
CREATE (pm)-[:TAUGHT_BY]->(mrvdg)
CREATE (am)-[:TAUGHT_BY]->(mrsg)
CREATE (am)-[:TAUGHT_BY]->(mrspb)
CREATE (am)-[:TAUGHT_BY]->(mrvdg)
CREATE (me)-[:TAUGHT_BY]->(mrj)
CREATE (ce)-[:TAUGHT_BY]->(mrsa)
CREATE (se)-[:TAUGHT_BY]->(mrs)
CREATE (ve)-[:TAUGHT_BY]->(msd)
CREATE (ee)-[:TAUGHT_BY]->(mrsf)

CREATE(_001)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_188)
CREATE(_002)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_198)
CREATE(_003)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_106)
CREATE(_004)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_029)
CREATE(_005)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_153)
CREATE(_006)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_061)
CREATE(_007)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_177)
CREATE(_008)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_115)
CREATE(_009)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_131)
CREATE(_010)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_142)
CREATE(_011)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_043)
CREATE(_012)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_065)
CREATE(_013)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_074)
CREATE(_014)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_165)
CREATE(_015)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_016)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_086)
CREATE(_017)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_062)
CREATE(_018)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_033)
CREATE(_019)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_171)
CREATE(_020)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_021)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_086)
CREATE(_022)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_121)
CREATE(_023)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_049)
CREATE(_024)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_152)
CREATE(_025)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_152)
CREATE(_026)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_085)
CREATE(_027)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_084)
CREATE(_028)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_143)
CREATE(_029)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_099)
CREATE(_030)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_094)
CREATE(_031)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_125)
CREATE(_032)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_024)
CREATE(_033)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_075)
CREATE(_034)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_161)
CREATE(_035)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_197)
CREATE(_036)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_067)
CREATE(_037)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_049)
CREATE(_038)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_038)
CREATE(_039)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_116)
CREATE(_040)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_149)
CREATE(_041)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_044)
CREATE(_042)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_150)
CREATE(_043)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_095)
CREATE(_044)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_016)
CREATE(_045)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_021)
CREATE(_046)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_047)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_189)
CREATE(_048)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_094)
CREATE(_049)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_161)
CREATE(_050)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_098)
CREATE(_051)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_052)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_148)
CREATE(_053)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_054)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_055)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_175)
CREATE(_056)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_010)
CREATE(_057)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_042)
CREATE(_058)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_059)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_067)
CREATE(_060)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_034)
CREATE(_061)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_062)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_088)
CREATE(_063)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_142)
CREATE(_064)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_88)
CREATE(_065)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_099)
CREATE(_066)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_178)
CREATE(_067)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_041)
CREATE(_068)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_022)
CREATE(_069)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_109)
CREATE(_070)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_045)
CREATE(_071)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_182)
CREATE(_072)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_144)
CREATE(_073)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_074)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_128)
CREATE(_075)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_149)
CREATE(_076)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_038)
CREATE(_077)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_104)
CREATE(_078)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_032)
CREATE(_079)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_080)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_081)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_174)
CREATE(_082)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_162)
CREATE(_083)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_011)
CREATE(_084)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_085)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_086)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_067)
CREATE(_087)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_173)
CREATE(_088)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_128)
CREATE(_089)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_177)
CREATE(_090)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_076)
CREATE(_091)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_137)
CREATE(_092)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_024)
CREATE(_093)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_156)
CREATE(_094)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_020)
CREATE(_095)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_112)
CREATE(_096)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_193)
CREATE(_097)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_006)
CREATE(_098)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_099)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_141)
CREATE(_100)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_001)
CREATE(_101)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_169)
CREATE(_102)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_161)
CREATE(_103)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_136)
CREATE(_104)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_125)
CREATE(_105)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_127)
CREATE(_106)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_095)
CREATE(_107)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_036)
CREATE(_108)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_074)
CREATE(_109)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_150)
CREATE(_110)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_191)
CREATE(_111)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_068)
CREATE(_112)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_019)
CREATE(_113)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_035)
CREATE(_114)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_061)
CREATE(_115)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_070)
CREATE(_116)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_069)
CREATE(_117)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_096)
CREATE(_118)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_107)
CREATE(_119)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_120)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_167)
CREATE(_121)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_122)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_090)
CREATE(_123)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_004)
CREATE(_124)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_083)
CREATE(_125)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_094)
CREATE(_126)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_174)
CREATE(_127)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_168)
CREATE(_128)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_084)
CREATE(_129)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_186)
CREATE(_130)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_090)
CREATE(_131)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_010)
CREATE(_132)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_031)
CREATE(_133)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_059)
CREATE(_134)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_037)
CREATE(_135)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_012)
CREATE(_136)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_197)
CREATE(_137)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_059)
CREATE(_138)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_065)
CREATE(_139)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_175)
CREATE(_140)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_170)
CREATE(_141)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_191)
CREATE(_142)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_139)
CREATE(_143)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_054)
CREATE(_144)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_176)
CREATE(_145)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_188)
CREATE(_146)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_072)
CREATE(_147)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_096)
CREATE(_148)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_108)
CREATE(_149)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_155)
CREATE(_150)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_151)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_076)
CREATE(_152)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_169)
CREATE(_153)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_179)
CREATE(_154)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_186)
CREATE(_155)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_058)
CREATE(_156)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_071)
CREATE(_157)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_073)
CREATE(_158)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_159)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_182)
CREATE(_160)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_199)
CREATE(_161)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_072)
CREATE(_162)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_014)
CREATE(_163)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_163)
CREATE(_164)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_038)
CREATE(_165)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_044)
CREATE(_166)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_136)
CREATE(_167)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_038)
CREATE(_168)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_110)
CREATE(_169)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_198)
CREATE(_170)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_178)
CREATE(_171)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_022)
CREATE(_172)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_020)
CREATE(_173)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_164)
CREATE(_174)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_075)
CREATE(_175)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_175)
CREATE(_176)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_177)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_178)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_006)
CREATE(_179)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_057)
CREATE(_180)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_185)
CREATE(_181)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_074)
CREATE(_182)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_183)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_131)
CREATE(_184)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_045)
CREATE(_185)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_186)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_187)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_150)
CREATE(_188)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_014)
CREATE(_189)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_096)
CREATE(_190)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_063)
CREATE(_191)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_079)
CREATE(_192)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_121)
CREATE(_193)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_194)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_029)
CREATE(_195)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_164)
CREATE(_196)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_083)
CREATE(_197)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_101)
CREATE(_198)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_039)
CREATE(_199)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_011)
CREATE(_200)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_073)
CREATE(_001)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_129)
CREATE(_002)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_078)
CREATE(_003)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_181)
CREATE(_004)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_162)
CREATE(_005)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_057)
CREATE(_006)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_111)
CREATE(_007)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_027)
CREATE(_008)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_009)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_132)
CREATE(_010)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_147)
CREATE(_011)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_083)
CREATE(_012)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_118)
CREATE(_013)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_099)
CREATE(_014)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_015)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_107)
CREATE(_016)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_116)
CREATE(_017)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_018)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_069)
CREATE(_019)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_024)
CREATE(_020)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_022)
CREATE(_021)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_184)
CREATE(_022)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_023)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_024)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_075)
CREATE(_025)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_087)
CREATE(_026)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_163)
CREATE(_027)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_115)
CREATE(_028)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_042)
CREATE(_029)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_058)
CREATE(_030)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_188)
CREATE(_031)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_032)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_015)
CREATE(_033)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_130)
CREATE(_034)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_141)
CREATE(_035)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_158)
CREATE(_036)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_020)
CREATE(_037)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_102)
CREATE(_038)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_184)
CREATE(_039)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_040)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_041)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_171)
CREATE(_042)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_050)
CREATE(_043)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_085)
CREATE(_044)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_025)
CREATE(_045)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_084)
CREATE(_046)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_118)
CREATE(_047)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_048)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_099)
CREATE(_049)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_071)
CREATE(_050)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_178)
CREATE(_051)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_052)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_059)
CREATE(_053)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_095)
CREATE(_054)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_185)
CREATE(_055)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_108)
CREATE(_056)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_083)
CREATE(_057)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_031)
CREATE(_058)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_054)
CREATE(_059)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_198)
CREATE(_060)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_138)
CREATE(_061)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_176)
CREATE(_062)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_086)
CREATE(_063)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_032)
CREATE(_064)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_101)
CREATE(_065)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_181)
CREATE(_066)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_153)
CREATE(_067)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_166)
CREATE(_068)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_069)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_027)
CREATE(_070)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_021)
CREATE(_071)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_193)
CREATE(_072)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_022)
CREATE(_073)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_108)
CREATE(_074)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_174)
CREATE(_075)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_019)
CREATE(_076)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_179)
CREATE(_077)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_005)
CREATE(_078)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_014)
CREATE(_079)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_017)
CREATE(_080)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_146)
CREATE(_081)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_098)
CREATE(_082)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_171)
CREATE(_083)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_099)
CREATE(_084)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_161)
CREATE(_085)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_098)
CREATE(_086)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_199)
CREATE(_087)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_057)
CREATE(_088)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_164)
CREATE(_089)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_064)
CREATE(_090)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_109)
CREATE(_091)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_077)
CREATE(_092)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_124)
CREATE(_093)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_181)
CREATE(_094)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_142)
CREATE(_095)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_191)
CREATE(_096)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_093)
CREATE(_097)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_031)
CREATE(_098)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_045)
CREATE(_099)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_182)
CREATE(_100)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_043)
CREATE(_101)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_146)
CREATE(_102)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_141)
CREATE(_103)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_040)
CREATE(_104)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_199)
CREATE(_105)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_063)
CREATE(_106)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_180)
CREATE(_107)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_010)
CREATE(_108)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_122)
CREATE(_109)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_111)
CREATE(_110)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_065)
CREATE(_111)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_199)
CREATE(_112)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_135)
CREATE(_113)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_172)
CREATE(_114)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_096)
CREATE(_115)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_028)
CREATE(_116)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_109)
CREATE(_117)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_191)
CREATE(_118)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_169)
CREATE(_119)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_101)
CREATE(_120)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_184)
CREATE(_121)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_032)
CREATE(_122)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_127)
CREATE(_123)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_129)
CREATE(_124)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_116)
CREATE(_125)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_150)
CREATE(_126)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_175)
CREATE(_127)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_018)
CREATE(_128)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_165)
CREATE(_129)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_130)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_066)
CREATE(_131)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_050)
CREATE(_132)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_197)
CREATE(_133)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_111)
CREATE(_134)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_125)
CREATE(_135)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_112)
CREATE(_136)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_173)
CREATE(_137)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_181)
CREATE(_138)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_072)
CREATE(_139)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_115)
CREATE(_140)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_013)
CREATE(_141)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_142)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_003)
CREATE(_143)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_144)
CREATE(_144)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_145)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_015)
CREATE(_146)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_061)
CREATE(_147)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_009)
CREATE(_148)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_149)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_176)
CREATE(_150)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_152)
CREATE(_151)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_055)
CREATE(_152)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_157)
CREATE(_153)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_090)
CREATE(_154)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_162)
CREATE(_155)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_146)
CREATE(_156)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_073)
CREATE(_157)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_044)
CREATE(_158)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_154)
CREATE(_159)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_160)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_168)
CREATE(_161)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_122)
CREATE(_162)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_015)
CREATE(_163)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_041)
CREATE(_164)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_087)
CREATE(_165)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_104)
CREATE(_166)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_116)
CREATE(_167)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_019)
CREATE(_168)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_021)
CREATE(_169)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_065)
CREATE(_170)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_183)
CREATE(_171)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_147)
CREATE(_172)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_045)
CREATE(_173)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_172)
CREATE(_174)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_137)
CREATE(_175)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_176)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_138)
CREATE(_177)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_078)
CREATE(_178)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_176)
CREATE(_179)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_062)
CREATE(_180)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_181)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_178)
CREATE(_182)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_173)
CREATE(_183)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_107)
CREATE(_184)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_198)
CREATE(_185)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_057)
CREATE(_186)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_041)
CREATE(_187)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_076)
CREATE(_188)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_132)
CREATE(_189)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_093)
CREATE(_190)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_191)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_183)
CREATE(_192)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_140)
CREATE(_193)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_194)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_117)
CREATE(_195)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_054)
CREATE(_196)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_197)
CREATE(_197)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_086)
CREATE(_198)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_190)
CREATE(_199)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_143)
CREATE(_200)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_144)
CREATE(_001)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_050)
CREATE(_002)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_024)
CREATE(_003)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_135)
CREATE(_004)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_094)
CREATE(_005)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_143)
CREATE(_006)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_066)
CREATE(_007)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_193)
CREATE(_008)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_022)
CREATE(_009)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_074)
CREATE(_010)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_166)
CREATE(_011)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_131)
CREATE(_012)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_036)
CREATE(_013)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_016)
CREATE(_014)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_108)
CREATE(_015)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_083)
CREATE(_016)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_017)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_016)
CREATE(_018)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_130)
CREATE(_019)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_013)
CREATE(_020)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_186)
CREATE(_021)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_026)
CREATE(_022)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_040)
CREATE(_023)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_064)
CREATE(_024)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_072)
CREATE(_025)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_017)
CREATE(_026)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_159)
CREATE(_027)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_076)
CREATE(_028)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_014)
CREATE(_029)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_089)
CREATE(_030)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_157)
CREATE(_031)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_029)
CREATE(_032)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_184)
CREATE(_033)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_131)
CREATE(_034)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_171)
CREATE(_035)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_051)
CREATE(_036)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_031)
CREATE(_037)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_038)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_057)
CREATE(_039)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_023)
CREATE(_040)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_109)
CREATE(_041)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_177)
CREATE(_042)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_020)
CREATE(_043)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_069)
CREATE(_044)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_068)
CREATE(_045)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_027)
CREATE(_046)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_018)
CREATE(_047)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_154)
CREATE(_048)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_090)
CREATE(_049)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_166)
CREATE(_050)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_150)
CREATE(_051)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_045)
CREATE(_052)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_123)
CREATE(_053)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_160)
CREATE(_054)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_088)
CREATE(_055)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_196)
CREATE(_056)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_057)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_110)
CREATE(_058)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_060)
CREATE(_059)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_084)
CREATE(_060)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_030)
CREATE(_061)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_170)
CREATE(_062)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_027)
CREATE(_063)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_018)
CREATE(_064)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_004)
CREATE(_065)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_138)
CREATE(_066)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_009)
CREATE(_067)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_172)
CREATE(_068)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_077)
CREATE(_069)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_112)
CREATE(_070)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_069)
CREATE(_071)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_018)
CREATE(_072)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_172)
CREATE(_073)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_053)
CREATE(_074)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_098)
CREATE(_075)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_068)
CREATE(_076)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_132)
CREATE(_077)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_134)
CREATE(_078)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_138)
CREATE(_079)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_080)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_125)
CREATE(_081)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_129)
CREATE(_082)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_048)
CREATE(_083)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_145)
CREATE(_084)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_101)
CREATE(_085)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_131)
CREATE(_086)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_011)
CREATE(_087)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_200)
CREATE(_088)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_070)
CREATE(_089)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_008)
CREATE(_090)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_107)
CREATE(_091)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_002)
CREATE(_092)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_180)
CREATE(_093)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_001)
CREATE(_094)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_120)
CREATE(_095)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_135)
CREATE(_096)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_116)
CREATE(_097)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_171)
CREATE(_098)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_122)
CREATE(_099)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_100)
CREATE(_100)-[:BUDDY]->(:StudyBuddy)<-[:BUDDY]-(_130)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.14 LargeIntegerEquality

2.14.1 Does not lose precision

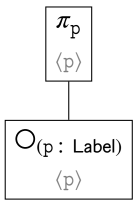
Query specification

```
MATCH (p:Label)
RETURN p.id
```

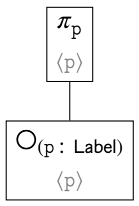
Relational algebra expression

$\pi_{\mathrm{p}}\left(\not\fallingdotseq\left(\bigcirc_{(\mathrm{p}\,:\,\mathsf{Label})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.14.2 Handling inlined equality of large integer

Query specification

```
MATCH (p:Label {id: 4611686018427387905})
RETURN p.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.14.3 Handling explicit equality of large integer

Query specification

```
MATCH (p:Label)
WHERE p.id = 4611686018427387905
RETURN p.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.14.4 Handling inlined equality of large integer, non-equal values

Query specification

```
MATCH (p:Label {id : 4611686018427387900})
RETURN p.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.14.5 Handling explicit equality of large integer, non-equal values

Query specification

```
MATCH (p:Label)
WHERE p.id = 4611686018427387900
RETURN p.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.15 ListComprehension

2.15.1 Returning a list comprehension

Query specification

```
MATCH p = (n)-->()
RETURN [x IN collect(p) | head(nodes(x))] AS p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.15.2 Using a list comprehension in a WITH

Query specification

```
MATCH p = (n:A)-->()
WITH [x IN collect(p) | head(nodes(x))] AS p, count(n) AS c
RETURN p, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.15.3 Using a list comprehension in a WHERE

Query specification

```
MATCH (n)-->(b)
WHERE n.prop IN [x IN labels(b) | lower(x)]
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16 Literals

2.16.1 Return an integer

Query specification

```
RETURN 1 AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.2 Return a float

Query specification

```
RETURN 1.0 AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.3 Return a float in exponent form

Query specification

```
RETURN -1e-9 AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.4 Return a boolean

Query specification

```
RETURN true AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.5 Return a single-quoted string

Query specification

```
RETURN '' AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.6 Return a double-quoted string

Query specification

```
RETURN "" AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.7 Return null

Query specification

```
RETURN null AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.8 Return an empty list

Query specification

```
RETURN [] AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.9 Return a nonempty list

Query specification

```
RETURN [0, 1, 2] AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.10 Return an empty map

Query specification

```
RETURN {} AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.16.11 Return a nonempty map

Query specification

```
RETURN {k1: 0, k2: 'string'} AS literal
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17 MatchAcceptance

2.17.1 Path query should return results in written order

Query specification

```
MATCH p = (a:Label1)<--(:Label2)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.2 Longer path query should return results in written order

Query specification

```
MATCH p = (a:Label1)<--(:Label2)--()
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.3 Use multiple MATCH clauses to do a Cartesian product

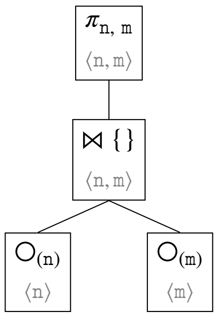
Query specification

```
MATCH (n), (m)
RETURN n.value AS n, m.value AS m
```

Relational algebra expression

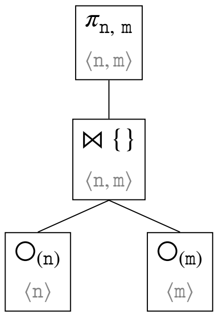
$\pi_{\texttt{n, m}} \left(\not\equiv \left(\bigcirc_{(\texttt{n})} \bowtie \{\} \bigcirc_{(\texttt{m})} \right) \right)$

Relational algebra tree



$\pi_{\texttt{n, m}}$
$\langle \texttt{n}, \texttt{m} \rangle$
$\bowtie \{\}$
$\langle \texttt{n}, \texttt{m} \rangle$
$\bigcirc_{(\texttt{n})}$
$\langle \texttt{n} \rangle$
$\bigcirc_{(\texttt{m})}$
$\langle \texttt{m} \rangle$

Relational algebra tree for incremental queries



$\pi_{\texttt{n, m}}$
$\langle \texttt{n}, \texttt{m} \rangle$
$\bowtie \{\}$
$\langle \texttt{n}, \texttt{m} \rangle$
$\bigcirc_{(\texttt{n})}$
$\langle \texttt{n} \rangle$
$\bigcirc_{(\texttt{m})}$
$\langle \texttt{m} \rangle$

2.17.4 Use params in pattern matching predicates

Query specification

```
MATCH (a)-[r]->(b)
WHERE r.foo = $param
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.5 Filter out based on node prop name

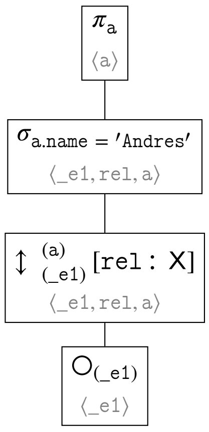
Query specification

```
MATCH ()-[rel:X]-(a)
WHERE a.name = 'Andres'
RETURN a
```

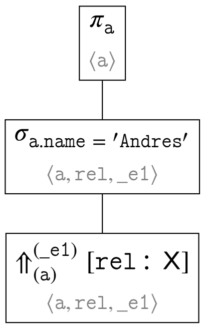
Relational algebra expression

$$\pi_{\texttt{a}}\left(\sigma_{\texttt{a.name} = \texttt{'Andres'}}\left(\not\equiv\left(\updownarrow{}^{(\texttt{a})}_{(\texttt{\_e1})}[\texttt{rel}:\texttt{X}]\left(\bigcirc_{(\texttt{\_e1})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.17.6 Honour the column name for RETURN items

Query specification

```
MATCH (a)
WITH a.name AS a
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.7 Filter based on rel prop name

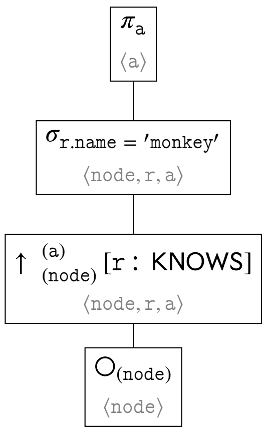
Query specification

```
MATCH (node)-[r:KNOWS]->(a)
WHERE r.name = 'monkey'
RETURN a
```

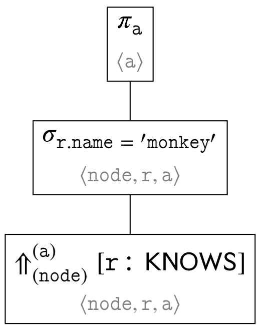
Relational algebra expression

$$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{r.name} = '\mathtt{monkey}'}\left(\not\equiv\left(\uparrow^{(\mathtt{a})}_{(\mathtt{node})}[\mathtt{r} : \mathsf{KNOWS}]\left(\bigcirc_{(\mathtt{node})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{r.name} = '\mathtt{monkey}'}$
⟨node, r, a⟩
$\uparrow^{(\mathtt{a})}_{(\mathtt{node})}[\mathtt{r} : \mathsf{KNOWS}]$
⟨node, r, a⟩
$\bigcirc_{(\mathtt{node})}$
⟨node⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{r.name} = '\mathtt{monkey}'}$
⟨node, r, a⟩
$\Uparrow^{(\mathtt{a})}_{(\mathtt{node})}[\mathtt{r} : \mathsf{KNOWS}]$
⟨node, r, a⟩

2.17.8 Cope with shadowed variables

Query specification

```
MATCH (n)
WITH n.name AS n
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.9 Get neighbours

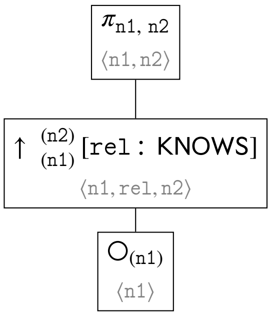
Query specification

```
MATCH (n1)-[rel:KNOWS]->(n2)
RETURN n1, n2
```

Relational algebra expression

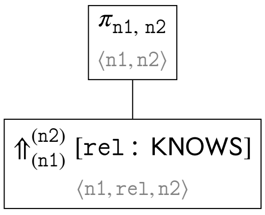
$$\pi_{\texttt{n1, n2}}\left(\not\equiv\left(\uparrow^{(\texttt{n2})}_{(\texttt{n1})}[\texttt{rel}:\ \textsf{KNOWS}]\left(\bigcirc_{(\texttt{n1})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\texttt{n1, n2}}$
$\langle \texttt{n1}, \texttt{n2} \rangle$
$\uparrow^{(\texttt{n2})}_{(\texttt{n1})}[\texttt{rel}:\ \textsf{KNOWS}]$
$\langle \texttt{n1}, \texttt{rel}, \texttt{n2} \rangle$
$\bigcirc_{(\texttt{n1})}$
$\langle \texttt{n1} \rangle$

Relational algebra tree for incremental queries



$\pi_{\texttt{n1, n2}}$
$\langle \texttt{n1}, \texttt{n2} \rangle$
$\Uparrow^{(\texttt{n2})}_{(\texttt{n1})}[\texttt{rel}:\ \textsf{KNOWS}]$
$\langle \texttt{n1}, \texttt{rel}, \texttt{n2} \rangle$

2.17.10 Get two related nodes

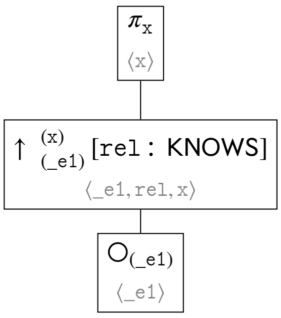
Query specification

```
MATCH ()-[rel:KNOWS]->(x)
RETURN x
```

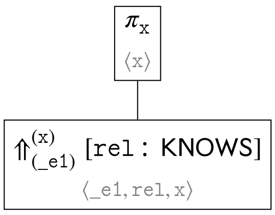
Relational algebra expression

$$\pi_{\texttt{x}}\left(\not\equiv\left(\uparrow^{(\texttt{x})}_{(\texttt{\_e1})}[\texttt{rel}:\ \textsf{KNOWS}]\left(\bigcirc_{(\texttt{\_e1})}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.17.11 Get related to related to

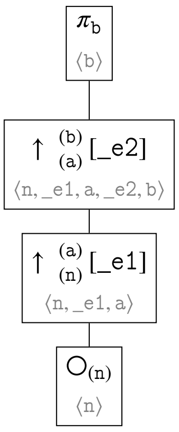
Query specification

```
MATCH (n)-->(a)-->(b)
RETURN b
```

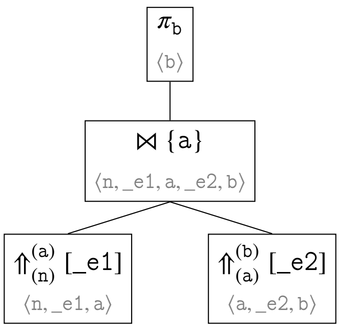
Relational algebra expression

$$\pi_{\mathrm{b}}\left(\not\equiv\left(\uparrow^{(\mathrm{b})}_{(\mathrm{a})}[\_\mathrm{e2}]\left(\uparrow^{(\mathrm{a})}_{(\mathrm{n})}[\_\mathrm{e1}]\left(\bigcirc_{(\mathrm{n})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.17.12 Handle comparison between node properties

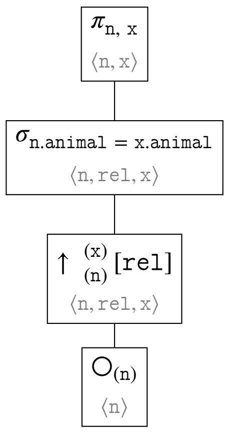
Query specification

```
MATCH (n)-[rel]->(x)
WHERE n.animal = x.animal
RETURN n, x
```

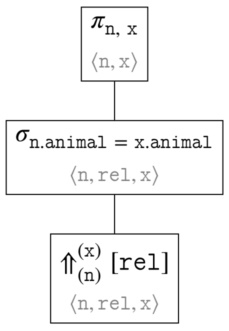
Relational algebra expression

$$\pi_{\texttt{n, x}}\left(\sigma_{\texttt{n.animal = x.animal}}\left(\not\equiv\left(\uparrow\,^{(\texttt{x})}_{(\texttt{n})}\,[\texttt{rel}]\left(\bigcirc_{(\texttt{n})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\texttt{n, x}}$
⟨n, x⟩
$\sigma_{\texttt{n.animal = x.animal}}$
⟨n, rel, x⟩
$\uparrow\,^{(\texttt{x})}_{(\texttt{n})}\,[\texttt{rel}]$
⟨n, rel, x⟩
$\bigcirc_{(\texttt{n})}$
⟨n⟩

Relational algebra tree for incremental queries


$\pi_{\texttt{n, x}}$
⟨n, x⟩
$\sigma_{\texttt{n.animal = x.animal}}$
⟨n, rel, x⟩
$\Uparrow^{(\texttt{x})}_{(\texttt{n})}\,[\texttt{rel}]$
⟨n, rel, x⟩

2.17.13 Return two subgraphs with bound undirected relationship

Query specification

```
MATCH (a)-[r {name: 'r'}]-(b)
RETURN a, b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.14 Return two subgraphs with bound undirected relationship and optional relationship

Query specification

```
MATCH (a)-[r {name: 'r1'}]-(b)
OPTIONAL MATCH (b)-[r2]-(c)
WHERE r <> r2
RETURN a, b, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.15 Rel type function works as expected

Query specification

```
MATCH (n {name: 'A'})-[r]->(x)
WHERE type(r) = 'KNOWS'
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.16 Walk alternative relationships

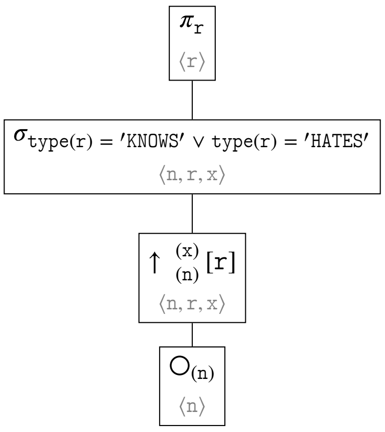
Query specification

```
MATCH (n)-[r]->(x)
WHERE type(r) = 'KNOWS' OR type(r) = 'HATES'
RETURN r
```

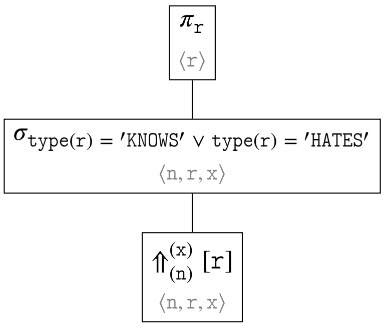
Relational algebra expression

$$\pi_{\mathtt{r}}\left(\sigma_{\mathtt{type(r)} = '\mathtt{KNOWS}' \lor \mathtt{type(r)} = '\mathtt{HATES}'}\left(\not\doteqdot\left(\uparrow\,{}^{(\mathtt{x})}_{(\mathtt{n})}\,[\mathtt{r}]\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\texttt{r}}$
$\langle \texttt{r} \rangle$
$\sigma_{\texttt{type(r)} = \texttt{'KNOWS'} \lor \texttt{type(r)} = \texttt{'HATES'}}$
$\langle \texttt{n}, \texttt{r}, \texttt{x} \rangle$
$\uparrow^{(\texttt{x})}_{(\texttt{n})} [\texttt{r}]$
$\langle \texttt{n}, \texttt{r}, \texttt{x} \rangle$
$\bigcirc_{(\texttt{n})}$
$\langle \texttt{n} \rangle$

Relational algebra tree for incremental queries


$\pi_{\texttt{r}}$
$\langle \texttt{r} \rangle$
$\sigma_{\texttt{type(r)} = \texttt{'KNOWS'} \lor \texttt{type(r)} = \texttt{'HATES'}}$
$\langle \texttt{n}, \texttt{r}, \texttt{x} \rangle$
$\Uparrow^{(\texttt{x})}_{(\texttt{n})} [\texttt{r}]$
$\langle \texttt{n}, \texttt{r}, \texttt{x} \rangle$

2.17.17 Handle OR in the WHERE clause

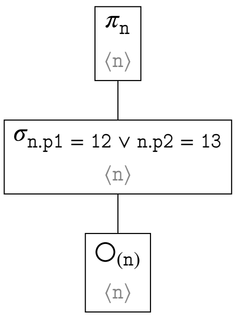
Query specification

```
MATCH (n)
WHERE n.p1 = 12 OR n.p2 = 13
RETURN n
```

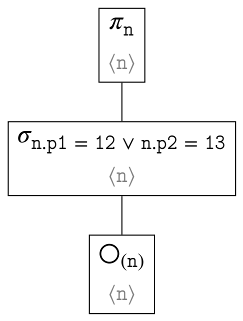
Relational algebra expression

$$\pi_{\texttt{n}} \left(\sigma_{\texttt{n.p1} = 12 \lor \texttt{n.p2} = 13} \left(\not\equiv \left(\bigcirc_{(\texttt{n})} \right) \right) \right)$$

Relational algebra tree


$\pi_{\texttt{n}}$
$\langle \texttt{n} \rangle$
$\sigma_{\texttt{n.p1} = 12 \lor \texttt{n.p2} = 13}$
$\langle \texttt{n} \rangle$
$\bigcirc_{(\texttt{n})}$
$\langle \texttt{n} \rangle$

Relational algebra tree for incremental queries



$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{\mathtt{n.p1} = 12 \lor \mathtt{n.p2} = 13}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.17.18 Return a simple path

Query specification

```
MATCH p = (a {name: 'A'})-->(b)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.19 Return a three node path

Query specification

```
MATCH p = (a {name: 'A'})-[rel1]->(b)-[rel2]->(c)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.20 Do not return anything because path length does not match

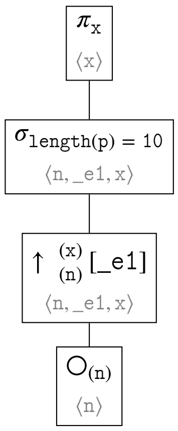
Query specification

```
MATCH p = (n)-->(x)
WHERE length(p) = 10
RETURN x
```

Relational algebra expression

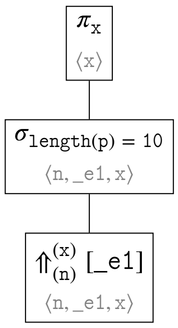
$$\pi_{\mathtt{x}}\left(\sigma_{\mathtt{length(p)} = 10}\left(\not\doteq\left(\uparrow\,{}^{(\mathtt{x})}_{(\mathtt{n})}\,[\mathtt{\_e1}]\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{x}}$
⟨x⟩
$\sigma_{\mathtt{length(p)} = 10}$
⟨n, _e1, x⟩
$\uparrow^{(x)}_{(n)} [\_e1]$
⟨n, _e1, x⟩
$\bigcirc_{(n)}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{x}}$
⟨x⟩
$\sigma_{\mathtt{length(p)} = 10}$
⟨n, _e1, x⟩
$\Uparrow^{(x)}_{(n)} [\_e1]$
⟨n, _e1, x⟩

2.17.21 Pass the path length test

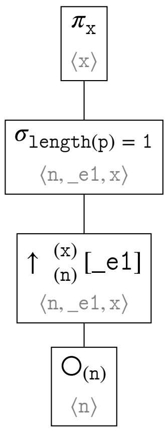
Query specification

```
MATCH p = (n)-->(x)
WHERE length(p) = 1
RETURN x
```

Relational algebra expression

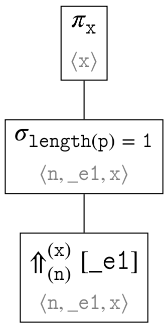
$$\pi_{\mathtt{x}}\left(\sigma_{\mathtt{length(p)} = 1}\left(\not\equiv\left(\uparrow^{(x)}_{(n)}[\_e1]\left(\bigcirc_{(n)}\right)\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{x}}$
⟨x⟩
$\sigma_{\mathtt{length(p)} = 1}$
⟨n, _e1, x⟩
$\uparrow^{(x)}_{(n)} [\_e1]$
⟨n, _e1, x⟩
$\bigcirc_{(n)}$
⟨n⟩

Relational algebra tree for incremental queries



2.17.22 Return relationships by fetching them from the path - starting from the end

Query specification

```
MATCH p = (a)-[:REL*2..2]->(b:End)
RETURN relationships(p)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.23 Return relationships by fetching them from the path

Query specification

```
MATCH p = (a:Start)-[:REL*2..2]->(b)
RETURN relationships(p)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.24 Return relationships by collecting them as a list - wrong way

Query specification

```
MATCH (a)-[r:REL*2..2]->(b:End)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.25 Return relationships by collecting them as a list - undirected

Query specification

```
MATCH (a)-[r:REL*2..2]-(b:End)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.26 Return relationships by collecting them as a list

Query specification

```
MATCH (a:Start)-[r:REL*2..2]-(b)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.27 Return a var length path

Query specification

```
MATCH p = (n {name: 'A'})-[:KNOWS*1..2]->(x)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.28 Return a var length path of length zero

Query specification

```
MATCH p = (a)-[*0..1]->(b)
RETURN a, b, length(p) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.29 Return a named var length path of length zero

Query specification

```
MATCH p = (a {name: 'A'})-[:KNOWS*0..1]->(b)-[:FRIEND*0..1]->(c)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.17.30 Accept skip zero

Query specification

```
MATCH (n)
WHERE 1 = 0
RETURN n SKIP 0
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18 MatchAcceptance2

2.18.1 Do not return non-existent nodes

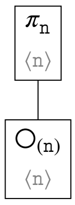
Query specification

```
MATCH (n)
RETURN n
```

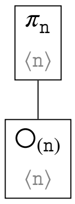
Relational algebra expression

$\pi_{\mathtt{n}} \left(\not\doteq \left(\bigcirc_{(\mathtt{n})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.2 Do not return non-existent relationships

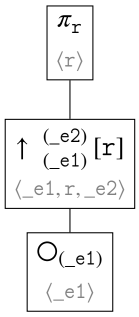
Query specification

```
MATCH ()-[r]->()
RETURN r
```

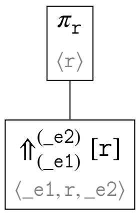
Relational algebra expression

$\pi_{\mathtt{r}} \left(\not\doteq \left(\uparrow^{(\_e2)}_{(\_e1)} [\mathtt{r}] \left(\bigcirc_{(\_e1)} \right) \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.3 Do not fail when evaluating predicates with illegal operations if the AND'ed predicate evaluates to false

Query specification

```
MATCH (:Root {name: 'x'})-->(i:TextNode)
WHERE i.id > 'te'
RETURN i
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.4 Do not fail when evaluating predicates with illegal operations if the OR'd predicate evaluates to true

Query specification

```
MATCH (:Root {name: 'x'})-->(i)
WHERE exists(i.id) OR i.id > 'te'
RETURN i
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.5 Aggregation with named paths

Query specification

```
MATCH p = ()-[*]->()
WITH count(*) AS count, p AS p
WITH nodes(p) AS nodes
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.6 Zero-length variable length pattern in the middle of the pattern

Query specification

```
MATCH (a {name: 'A'})-[:CONTAINS*0..1]->(b)-[:FRIEND*0..1]->(c)
RETURN a, b, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.7 Simple variable length pattern

Query specification

```
MATCH (a {name: 'A'})-[*]->(x)
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.8 Variable length relationship without lower bound

Query specification

```
MATCH p = ({name: 'A'})-[:KNOWS*..2]->()
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.9 Variable length relationship without bounds

Query specification

```
MATCH p = ({name: 'A'})-[:KNOWS*..]->()
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.10 Returning bound nodes that are not part of the pattern

Query specification

```
MATCH (a {name: 'A'}), (c {name: 'C'})
MATCH (a)-->(b)
RETURN a, b, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.11 Two bound nodes pointing to the same node

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MATCH (a)-->(x)<-->(b)
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.12 Three bound nodes pointing to the same node

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'}), (c {name: 'C'})
MATCH (a)-->(x), (b)-->(x), (c)-->(x)
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.13 Three bound nodes pointing to the same node with extra connections

Query specification

```
MATCH (a {name: 'a'}), (b {name: 'b'}), (c {name: 'c'})
MATCH (a)-->(x), (b)-->(x), (c)-->(x)
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.14 MATCH with OPTIONAL MATCH in longer pattern

Query specification

```
MATCH (a {name: 'A'})
OPTIONAL MATCH (a)-[:KNOWS]->()-[:KNOWS]->(foo)
RETURN foo
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.15 Optionally matching named paths

Query specification

```
MATCH (a {name: 'A'}), (x)
WHERE x.name IN ['B', 'C']
OPTIONAL MATCH p = (a)-->(x)
RETURN x, p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.16 Optionally matching named paths with single and variable length patterns

Query specification

```
MATCH (a {name: 'A'})
OPTIONAL MATCH p = (a)-->(b)-[*]->(c)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.17 Optionally matching named paths with variable length patterns

Query specification

```
MATCH (a {name: 'A'}), (x)
WHERE x.name IN ['B', 'C']
OPTIONAL MATCH p = (a)-[r*]->(x)
RETURN r, x, p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.18 Matching variable length patterns from a bound node

Query specification

```
MATCH (a:A)
MATCH (a)-[r*2]->()
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.19 Excluding connected nodes

Query specification

```
MATCH (a:A), (other:B)
OPTIONAL MATCH (a)-[r]->(other)
WITH other WHERE r IS NULL
RETURN other
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.20 Do not fail when predicates on optionally matched and missed nodes are invalid

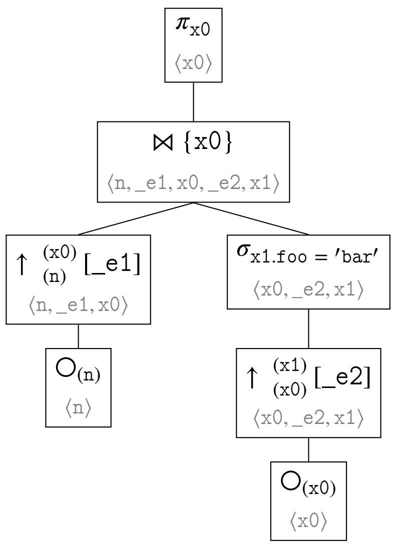
Query specification

```
MATCH (n)-->(x0)
OPTIONAL MATCH (x0)-->(x1)
WHERE x1.foo = 'bar'
RETURN x0.name
```

Relational algebra expression

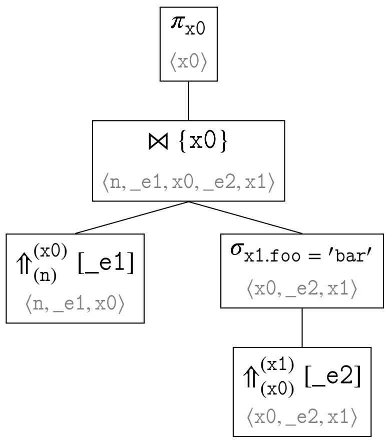
$$\pi_{\texttt{x0}}\left(\not\doteq\left(\uparrow\,{}^{(\texttt{x0})}_{(\texttt{n})}\,[\texttt{\_e1}]\left(\bigcirc_{(\texttt{n})}\right)\right)\bowtie\{\texttt{x0}\}\sigma_{\texttt{x1.foo}\,=\,'\texttt{bar}'}\left(\not\doteq\left(\uparrow\,{}^{(\texttt{x1})}_{(\texttt{x0})}\,[\texttt{\_e2}]\left(\bigcirc_{(\texttt{x0})}\right)\right)\right)\right)$$

Relational algebra tree



π_{x0}
$\langle x0 \rangle$
$\bowtie \{x0\}$
$\langle n, \_e1, x0, \_e2, x1 \rangle$
$\uparrow^{(x0)}_{(n)} [\_e1]$
$\langle n, \_e1, x0 \rangle$
$\sigma_{x1.foo = 'bar'}$
$\langle x0, \_e2, x1 \rangle$
$\bigcirc_{(n)}$
$\langle n \rangle$
$\uparrow^{(x1)}_{(x0)} [\_e2]$
$\langle x0, \_e2, x1 \rangle$
$\bigcirc_{(x0)}$
$\langle x0 \rangle$

Relational algebra tree for incremental queries



π_{x0}
$\langle x0 \rangle$
$\bowtie \{x0\}$
$\langle n, \_e1, x0, \_e2, x1 \rangle$
$\Uparrow^{(x0)}_{(n)} [\_e1]$
$\langle n, \_e1, x0 \rangle$
$\sigma_{x1.foo = 'bar'}$
$\langle x0, \_e2, x1 \rangle$
$\Uparrow^{(x1)}_{(x0)} [\_e2]$
$\langle x0, \_e2, x1 \rangle$

2.18.21 MATCH and OPTIONAL MATCH on same pattern

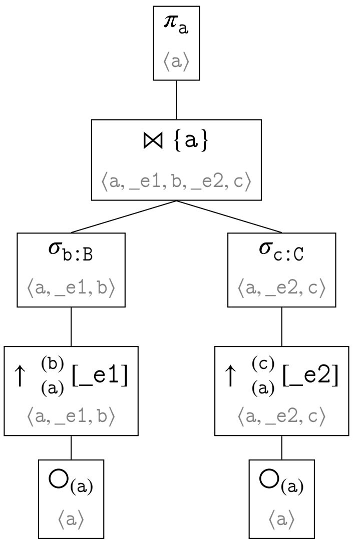
Query specification

```
MATCH (a)-->(b)
WHERE b:B
OPTIONAL MATCH (a)-->(c)
WHERE c:C
RETURN a.name
```

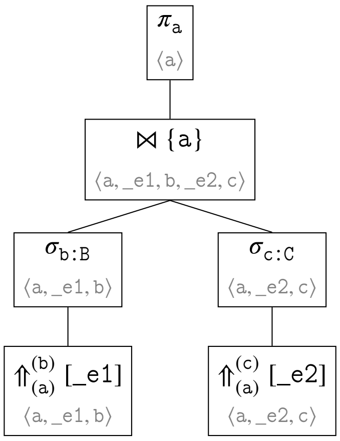
Relational algebra expression

$$\pi_{a}\left(\sigma_{b:B}\left(\not\doteq\left(\uparrow^{(b)}_{(a)}[\_e1]\left(\bigcirc_{(a)}\right)\right)\right) \bowtie \{a\}\sigma_{c:C}\left(\not\doteq\left(\uparrow^{(c)}_{(a)}[\_e2]\left(\bigcirc_{(a)}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.22 Matching using an undirected pattern

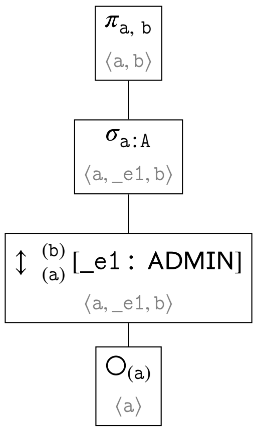
Query specification

```
MATCH (a)-[:ADMIN]-(b)
WHERE a:A
RETURN a.id, b.id
```

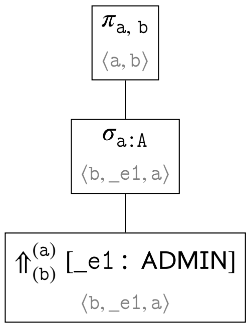
Relational algebra expression

$$\pi_{\mathtt{a},\ \mathtt{b}}\left(\sigma_{\mathtt{a:A}}\left(\not\equiv\left(\updownarrow\ {}^{(\mathsf{b})}_{(\mathsf{a})}\,[\_\mathsf{e1}:\ \mathsf{ADMIN}]\left(\bigcirc_{(\mathsf{a})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.23 Matching all nodes

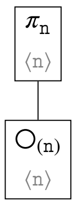
Query specification

```
MATCH (n)
RETURN n
```

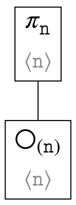
Relational algebra expression

$\pi_{n}\left(\not\equiv\left(\bigcirc_{(n)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.24 Comparing nodes for equality

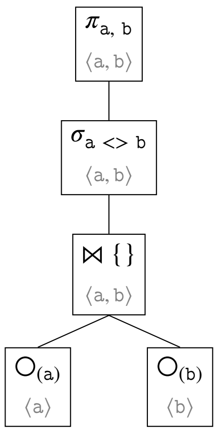
Query specification

```
MATCH (a), (b)
WHERE a <> b
RETURN a, b
```

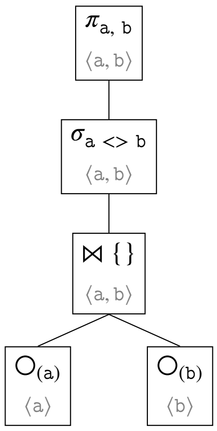
Relational algebra expression

$\pi_{\mathtt{a,\ b}}\left(\sigma_{\mathtt{a\ <>\ b}}\left(\not\equiv\left(\bigcirc_{(\mathsf{a})} \bowtie \{\} \bigcirc_{(\mathsf{b})}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{a,\ b}}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\sigma_{\mathtt{a\ <>\ b}}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\bowtie \{\}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\bigcirc_{(\mathsf{a})}$
$\langle \mathtt{a} \rangle$
$\bigcirc_{(\mathsf{b})}$
$\langle \mathtt{b} \rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{a,\ b}}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\sigma_{\mathtt{a\ <>\ b}}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\bowtie \{\}$
$\langle \mathtt{a}, \mathtt{b} \rangle$
$\bigcirc_{(\mathsf{a})}$
$\langle \mathtt{a} \rangle$
$\bigcirc_{(\mathsf{b})}$
$\langle \mathtt{b} \rangle$

2.18.25 Matching using self-referencing pattern returns no result

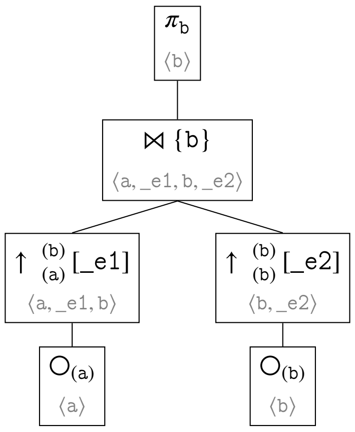
Query specification

```
MATCH (a)-->(b), (b)-->(b)
RETURN b
```

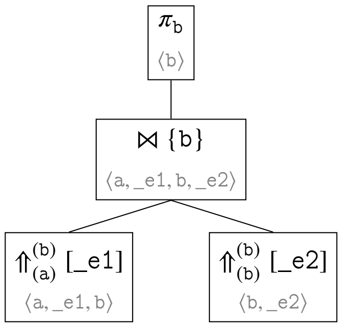
Relational algebra expression

$\pi_{\mathtt{b}}\left(\not\equiv\left(\uparrow^{(\mathsf{b})}_{(\mathsf{a})}[\_\mathsf{e1}]\left(\bigcirc_{(\mathsf{a})}\right) \bowtie \{\mathsf{b}\} \uparrow^{(\mathsf{b})}_{(\mathsf{b})}[\_\mathsf{e2}]\left(\bigcirc_{(\mathsf{b})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.26 Variable length relationship in OPTIONAL MATCH

Query specification

```
MATCH (a:A), (b:B)
OPTIONAL MATCH (a)-[r*]-(b)
WHERE r IS NULL
  AND a <> b
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.27 Matching using relationship predicate with multiples of the same type

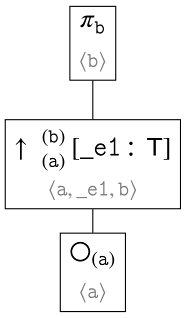
Query specification

```
MATCH (a)-[:T|:T]->(b)
RETURN b
```

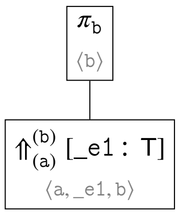
Relational algebra expression

$$\pi_{b}\left(\not\equiv\left(\uparrow\,^{(b)}_{(a)}[\_e1:\mathsf{T}]\left(\bigcirc_{(a)}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



π_{b}
$\langle b \rangle$
$\Uparrow^{(b)}_{(a)} [\_e1 : T]$
$\langle a, \_e1, b \rangle$

2.18.28 ORDER BY with LIMIT

Query specification

```
MATCH (a:A)-->(n)-->(m)
RETURN n.x, count(*)
  ORDER BY n.x
  LIMIT 1000
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.29 Simple node property predicate

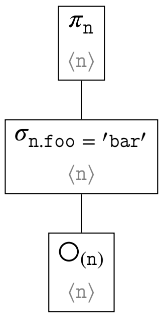
Query specification

```
MATCH (n)
WHERE n.foo = 'bar'
RETURN n
```

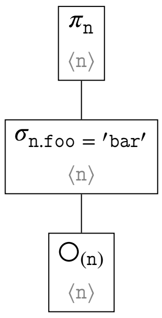
Relational algebra expression

$\pi_{n}\left(\sigma_{n.foo = 'bar'}\left(\not\equiv\left(\bigcirc_{(n)}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{\mathtt{n.foo} = \mathtt{'bar'}}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{n}}$
⟨n⟩
$\sigma_{\mathtt{n.foo} = \mathtt{'bar'}}$
⟨n⟩
$\bigcirc_{(\mathtt{n})}$
⟨n⟩

2.18.30 Handling direction of named paths

Query specification

```
MATCH p = (b)<--(a)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.31 Simple OPTIONAL MATCH on empty graph

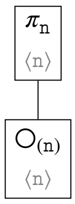
Query specification

```
OPTIONAL MATCH (n)
RETURN n
```

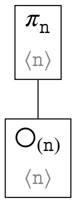
Relational algebra expression

$\pi_{\mathtt{n}} \left(\not\doteq \left(\bigcirc_{(\mathtt{n})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.32 OPTIONAL MATCH with previously bound nodes

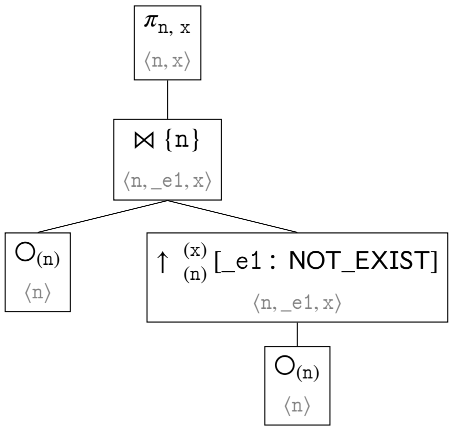
Query specification

```
MATCH (n)
OPTIONAL MATCH (n)-[:NOT_EXIST]->(x)
RETURN n, x
```

Relational algebra expression

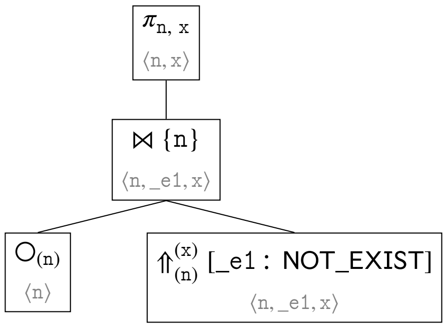
$$\pi_{\mathrm{n,\ x}}\left(\not\doteq\left(\bigcirc_{(\mathrm{n})}\right)\bowtie\{\mathrm{n}\}\not\doteq\left(\uparrow\,^{(\mathrm{x})}_{(\mathrm{n})}[\_\mathrm{e1}:\ \mathrm{NOT\_EXIST}]\left(\bigcirc_{(\mathrm{n})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathrm{n,\ x}}$
⟨n, x⟩
$\bowtie\{\mathrm{n}\}$
⟨n, _e1, x⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩
$\uparrow\,^{(\mathrm{x})}_{(\mathrm{n})}[\_\mathrm{e1}:\ \mathrm{NOT\_EXIST}]$
⟨n, _e1, x⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathrm{n,\ x}}$
⟨n, x⟩
$\bowtie\{\mathrm{n}\}$
⟨n, _e1, x⟩
$\bigcirc_{(\mathrm{n})}$
⟨n⟩
$\Uparrow^{(\mathrm{x})}_{(\mathrm{n})}[\_\mathrm{e1}:\ \mathrm{NOT\_EXIST}]$
⟨n, _e1, x⟩

2.18.33 ‘collect()‘ filtering nulls

Query specification

```
MATCH (n)
OPTIONAL MATCH (n)-[:NOT_EXIST]->(x)
RETURN n, collect(x)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.34 Multiple anonymous nodes in a pattern

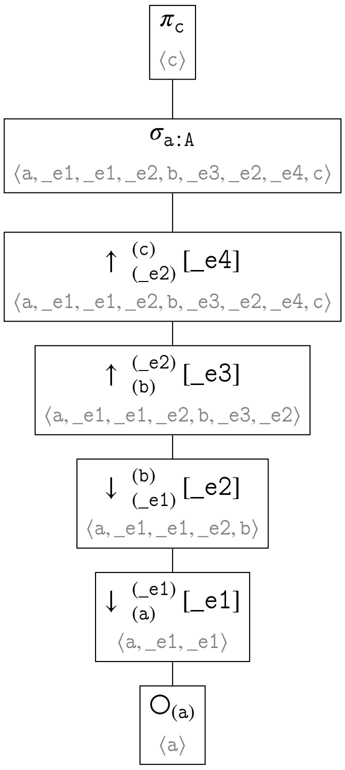
Query specification

```
MATCH (a)<--()<--(b)-->()-->(c)
WHERE a:A
RETURN c
```

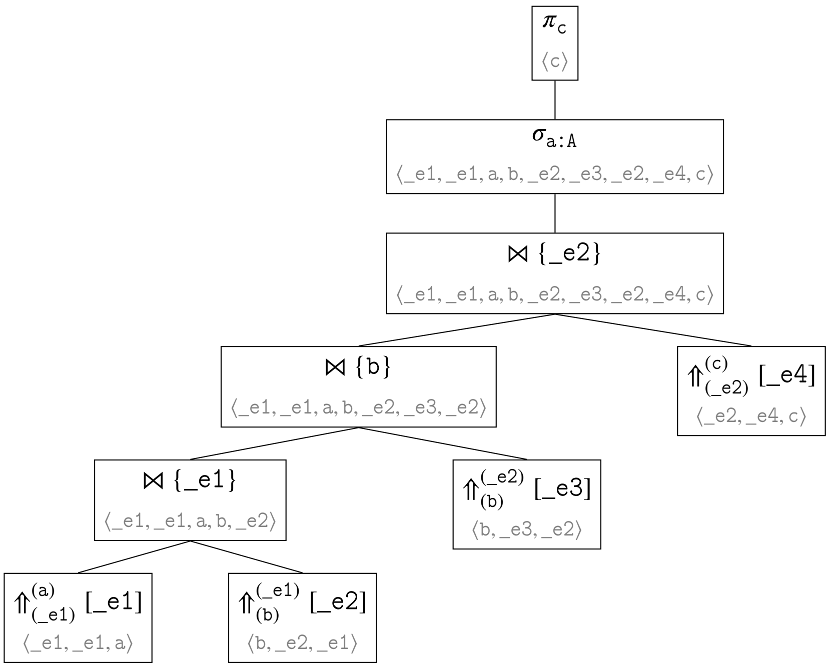
Relational algebra expression

$$\pi_{\mathtt{c}}\left(\sigma_{\mathtt{a:A}}\left(\not\equiv\left(\uparrow^{(\mathtt{c})}_{(\mathtt{\_e2})}[\mathtt{\_e4}]\left(\uparrow^{(\mathtt{\_e2})}_{(\mathtt{b})}[\mathtt{\_e3}]\left(\downarrow^{(\mathtt{b})}_{(\mathtt{\_e1})}[\mathtt{\_e2}]\left(\downarrow^{(\mathtt{\_e1})}_{(\mathtt{a})}[\mathtt{\_e1}]\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{c}}$
⟨c⟩
$\sigma_{\mathtt{a:A}}$
⟨a, _e1, _e1, _e2, b, _e3, _e2, _e4, c⟩
$\uparrow^{(\mathtt{c})}_{(\mathtt{\_e2})}[\mathtt{\_e4}]$
⟨a, _e1, _e1, _e2, b, _e3, _e2, _e4, c⟩
$\uparrow^{(\mathtt{\_e2})}_{(\mathtt{b})}[\mathtt{\_e3}]$
⟨a, _e1, _e1, _e2, b, _e3, _e2⟩
$\downarrow^{(\mathtt{b})}_{(\mathtt{\_e1})}[\mathtt{\_e2}]$
⟨a, _e1, _e1, _e2, b⟩
$\downarrow^{(\mathtt{\_e1})}_{(\mathtt{a})}[\mathtt{\_e1}]$
⟨a, _e1, _e1⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


π_{c}
⟨c⟩
$\sigma_{a:A}$
⟨_e1, _e1, a, b, _e2, _e3, _e2, _e4, c⟩
⋈ {_e2}
⟨_e1, _e1, a, b, _e2, _e3, _e2, _e4, c⟩
⋈ {b}
⟨_e1, _e1, a, b, _e2, _e3, _e2⟩
$\Uparrow^{(c)}_{(\_e2)}$ [_e4]
⟨_e2, _e4, c⟩
⋈ {_e1}
⟨_e1, _e1, a, b, _e2⟩
$\Uparrow^{(\_e2)}_{(b)}$ [_e3]
⟨b, _e3, _e2⟩
$\Uparrow^{(a)}_{(\_e1)}$ [_e1]
⟨_e1, _e1, a⟩
$\Uparrow^{(\_e1)}_{(b)}$ [_e2]
⟨b, _e2, _e1⟩

2.18.35 Matching a relationship pattern using a label predicate

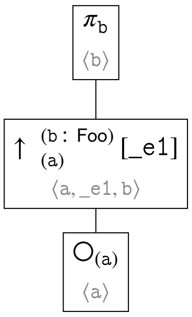
Query specification

```
MATCH (a)-->(b:Foo)
RETURN b
```

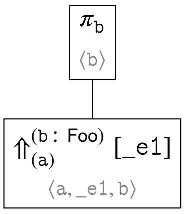
Relational algebra expression

$\pi_{b}\left(\not\equiv\left(\uparrow^{(b:\,Foo)}_{(a)}[\_e1]\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.36 Matching a relationship pattern using a label predicate on both sides

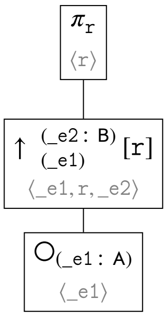
Query specification

```
MATCH (:A)-[r]->(:B)
RETURN r
```

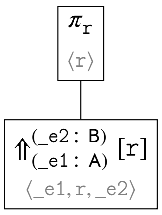
Relational algebra expression

$$\pi_{\mathtt{r}}\left(\not\doteqdot\left(\uparrow^{(\_e2:\,B)}_{(\_e1)}[\mathtt{r}]\left(\bigcirc_{(\_e1:\,A)}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.37 Matching nodes using multiple labels

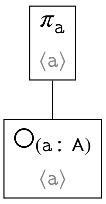
Query specification

```
MATCH (a:A:B:C)
RETURN a
```

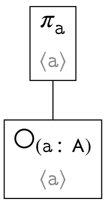
Relational algebra expression

$$\pi_{\mathtt{a}}\left(\not\doteqdot\left(\bigcirc_{(a:\,A)}\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.38 Returning label predicate expression

Query specification

```
MATCH (n)
RETURN (n:Foo)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.39 Matching with many predicates and larger pattern

Query specification

```
MATCH (advertiser)-[:ADV_HAS_PRODUCT]->(out)-[:AP_HAS_VALUE]->(red)<-[:AA_HAS_VALUE]-(a)
WHERE advertiser.id = $1
  AND a.id = $2
  AND red.name = 'red'
  AND out.name = 'product1'
RETURN out.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.40 Returning label predicate expression

Query specification

```
MATCH (n)
RETURN (n:Foo)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.41 Matching using a simple pattern with label predicate

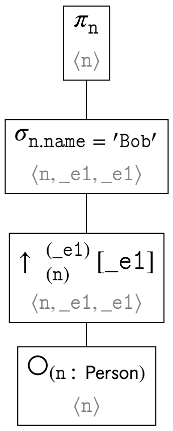
Query specification

```
MATCH (n:Person)-->()
WHERE n.name = 'Bob'
RETURN n
```

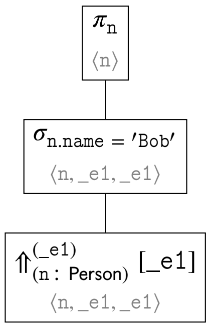
Relational algebra expression

$$\pi_{\mathtt{n}}\left(\sigma_{\mathtt{n.name} = \mathtt{'Bob'}}\left(\not\equiv\left(\uparrow^{(\_\mathtt{e1})}_{(\mathtt{n})}[\_\mathtt{e1}]\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Person})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{n}}$
$\langle \mathtt{n} \rangle$
$\sigma_{\mathtt{n.name} = \mathtt{'Bob'}}$
$\langle \mathtt{n}, \_\mathtt{e1}, \_\mathtt{e1} \rangle$
$\uparrow^{(\_\mathtt{e1})}_{(\mathtt{n})}[\_\mathtt{e1}]$
$\langle \mathtt{n}, \_\mathtt{e1}, \_\mathtt{e1} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Person})}$
$\langle \mathtt{n} \rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{n}}$
$\langle \mathtt{n} \rangle$
$\sigma_{\mathtt{n.name} = \mathtt{'Bob'}}$
$\langle \mathtt{n}, \_\mathtt{e1}, \_\mathtt{e1} \rangle$
$\Uparrow^{(\_\mathtt{e1})}_{(\mathtt{n}:\ \mathsf{Person})}[\_\mathtt{e1}]$
$\langle \mathtt{n}, \_\mathtt{e1}, \_\mathtt{e1} \rangle$

2.18.42 Matching disconnected patterns

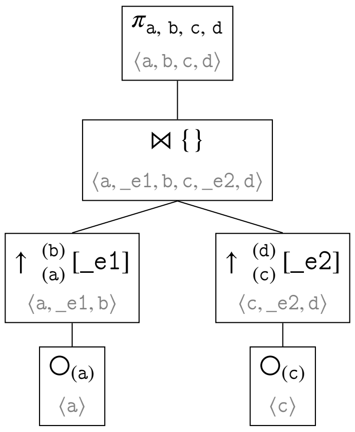
Query specification

```
MATCH (a)-->(b)
MATCH (c)-->(d)
RETURN a, b, c, d
```

Relational algebra expression

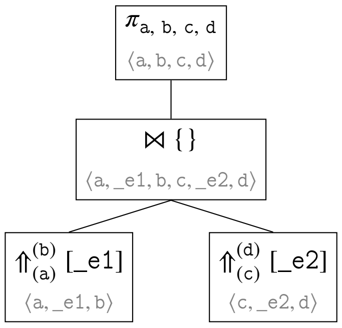
$$\pi_{\mathtt{a},\ \mathtt{b},\ \mathtt{c},\ \mathtt{d}}\left(\not\equiv\left(\uparrow^{(\mathtt{b})}_{(\mathtt{a})}[\_\mathtt{e1}]\left(\bigcirc_{(\mathtt{a})}\right)\right)\bowtie\{\}\not\equiv\left(\uparrow^{(\mathtt{d})}_{(\mathtt{c})}[\_\mathtt{e2}]\left(\bigcirc_{(\mathtt{c})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\texttt{a, b, c, d}}$
$\langle a, b, c, d \rangle$
$\bowtie \{\}$
$\langle a, \_e1, b, c, \_e2, d \rangle$
$\uparrow {}^{(b)}_{(a)} [\_e1]$
$\langle a, \_e1, b \rangle$
$\uparrow {}^{(d)}_{(c)} [\_e2]$
$\langle c, \_e2, d \rangle$
$\bigcirc_{(a)}$
$\langle a \rangle$
$\bigcirc_{(c)}$
$\langle c \rangle$

Relational algebra tree for incremental queries



$\pi_{\texttt{a, b, c, d}}$
$\langle a, b, c, d \rangle$
$\bowtie \{\}$
$\langle a, \_e1, b, c, \_e2, d \rangle$
$\Uparrow^{(b)}_{(a)} [\_e1]$
$\langle a, \_e1, b \rangle$
$\Uparrow^{(d)}_{(c)} [\_e2]$
$\langle c, \_e2, d \rangle$

2.18.43 Non-optional matches should not return nulls

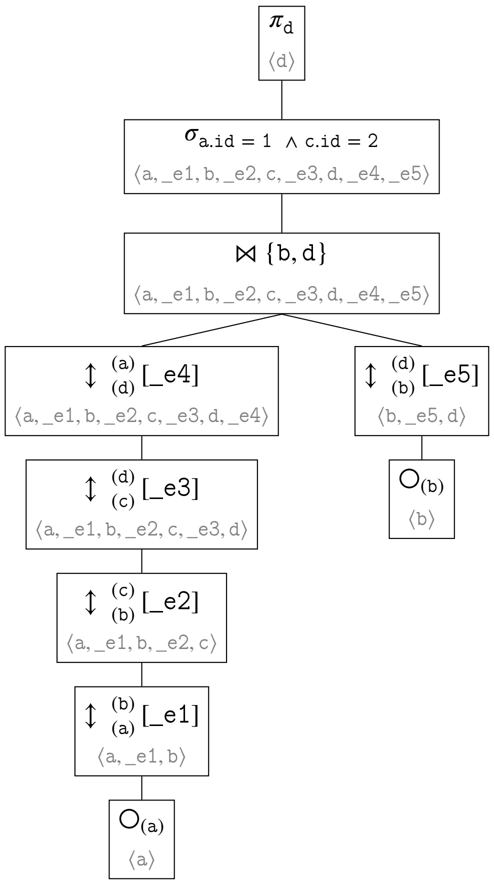
Query specification

```
MATCH (a)--(b)--(c)--(d)--(a), (b)--(d)
WHERE a.id = 1
  AND c.id = 2
RETURN d
```

Relational algebra expression

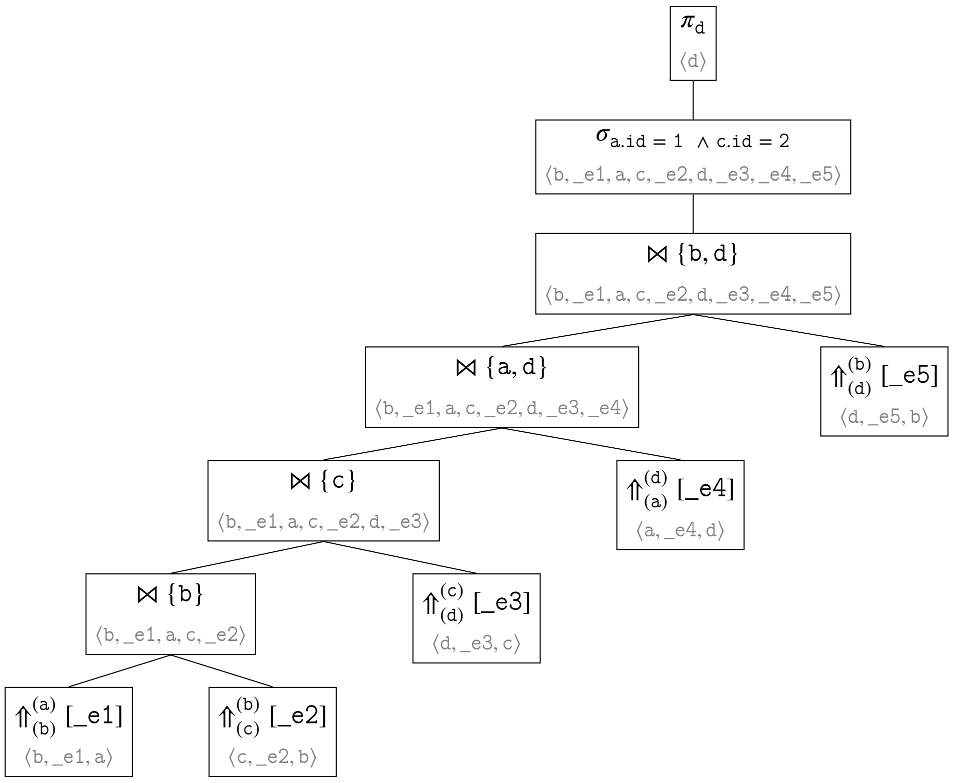
$$\pi_{\texttt{d}}\left(\sigma_{\texttt{a.id} = 1 \ \wedge \ \texttt{c.id} = 2}\left(\not\equiv\left(\updownarrow {}^{(a)}_{(d)} [\_e4] \left(\updownarrow {}^{(d)}_{(c)} [\_e3] \left(\updownarrow {}^{(c)}_{(b)} [\_e2] \left(\updownarrow {}^{(b)}_{(a)} [\_e1] \left(\bigcirc_{(a)}\right)\right)\right)\right) \bowtie \{b, d\} \updownarrow {}^{(d)}_{(b)} [\_e5] \left(\bigcirc_{(b)}\right)\right)\right)\right)$$

Relational algebra tree



π_{d}
⟨d⟩
$\sigma_{a.id = 1 \wedge c.id = 2}$
⟨a, _e1, b, _e2, c, _e3, d, _e4, _e5⟩
⋈ {b, d}
⟨a, _e1, b, _e2, c, _e3, d, _e4, _e5⟩
$\updownarrow^{(a)}_{(d)}$ [_e4]
⟨a, _e1, b, _e2, c, _e3, d, _e4⟩
$\updownarrow^{(d)}_{(b)}$ [_e5]
⟨b, _e5, d⟩
$\updownarrow^{(d)}_{(c)}$ [_e3]
⟨a, _e1, b, _e2, c, _e3, d⟩
$\bigcirc_{(b)}$
⟨b⟩
$\updownarrow^{(c)}_{(b)}$ [_e2]
⟨a, _e1, b, _e2, c⟩
$\updownarrow^{(b)}_{(a)}$ [_e1]
⟨a, _e1, b⟩
$\bigcirc_{(a)}$
⟨a⟩

Relational algebra tree for incremental queries



π_{d}
⟨d⟩
$\sigma_{a.id = 1 \wedge c.id = 2}$
⟨b, _e1, a, c, _e2, d, _e3, _e4, _e5⟩
⋈ {b, d}
⟨b, _e1, a, c, _e2, d, _e3, _e4, _e5⟩
⋈ {a, d}
⟨b, _e1, a, c, _e2, d, _e3, _e4⟩
$\Uparrow^{(b)}_{(d)}$ [_e5]
⟨d, _e5, b⟩
⋈ {c}
⟨b, _e1, a, c, _e2, d, _e3⟩
$\Uparrow^{(d)}_{(a)}$ [_e4]
⟨a, _e4, d⟩
⋈ {b}
⟨b, _e1, a, c, _e2⟩
$\Uparrow^{(c)}_{(d)}$ [_e3]
⟨d, _e3, c⟩
$\Uparrow^{(a)}_{(b)}$ [_e1]
⟨b, _e1, a⟩
$\Uparrow^{(b)}_{(c)}$ [_e2]
⟨c, _e2, b⟩

2.18.44 Handling cyclic patterns

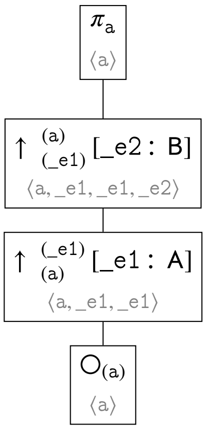
Query specification

```
MATCH (a)-[:A]->()-[:B]->(a)
RETURN a.name
```

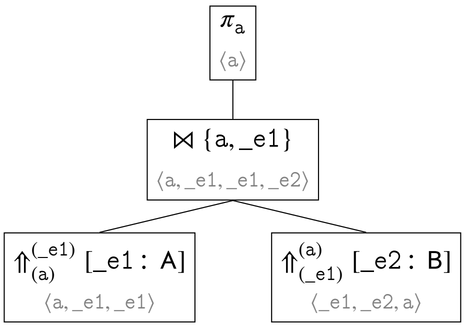
Relational algebra expression

$$\pi_{\mathsf{a}}\left(\not\equiv\left(\uparrow\,^{(\mathsf{a})}_{(\mathsf{\_e1})}\,[\mathsf{\_e2} : \mathsf{B}]\left(\uparrow\,^{(\mathsf{\_e1})}_{(\mathsf{a})}\,[\mathsf{\_e1} : \mathsf{A}]\left(\bigcirc_{(\mathsf{a})}\right)\right)\right)\right)$$

Relational algebra tree


π_{a}
$\langle$a$\rangle$
$\uparrow\,^{(\mathsf{a})}_{(\mathsf{\_e1})}\,[\mathsf{\_e2} : \mathsf{B}]$
$\langle$a, _e1, _e1, _e2$\rangle$
$\uparrow\,^{(\mathsf{\_e1})}_{(\mathsf{a})}\,[\mathsf{\_e1} : \mathsf{A}]$
$\langle$a, _e1, _e1$\rangle$
$\bigcirc_{(\mathsf{a})}$
$\langle$a$\rangle$

Relational algebra tree for incremental queries


π_{a}
$\langle$a$\rangle$
$\bowtie\{\mathsf{a}, \mathsf{\_e1}\}$
$\langle$a, _e1, _e1, _e2$\rangle$
$\Uparrow^{(\mathsf{\_e1})}_{(\mathsf{a})}\,[\mathsf{\_e1} : \mathsf{A}]$
$\langle$a, _e1, _e1$\rangle$
$\Uparrow^{(\mathsf{a})}_{(\mathsf{\_e1})}\,[\mathsf{\_e2} : \mathsf{B}]$
$\langle$_e1, _e2, a$\rangle$

2.18.45 Handling cyclic patterns when separated into two parts

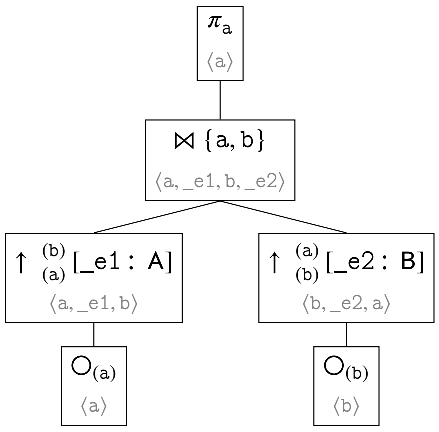
Query specification

```
MATCH (a)-[:A]->(b), (b)-[:B]->(a)
RETURN a.name
```

Relational algebra expression

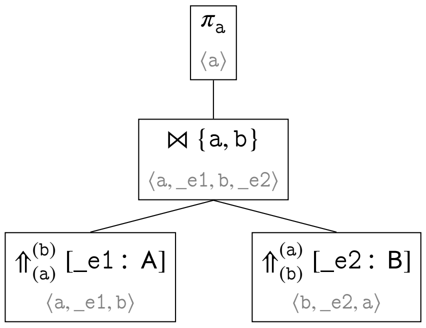
$$\pi_{\mathsf{a}}\left(\not\equiv\left(\uparrow\,^{(\mathsf{b})}_{(\mathsf{a})}\,[\mathsf{\_e1} : \mathsf{A}]\left(\bigcirc_{(\mathsf{a})}\right)\bowtie\{\mathsf{a}, \mathsf{b}\}\uparrow\,^{(\mathsf{a})}_{(\mathsf{b})}\,[\mathsf{\_e2} : \mathsf{B}]\left(\bigcirc_{(\mathsf{b})}\right)\right)\right)$$

Relational algebra tree



π_a
$\langle a \rangle$
$\bowtie \{a, b\}$
$\langle a, \_e1, b, \_e2 \rangle$
$\uparrow^{(b)}_{(a)} [\_e1 : A]$
$\langle a, \_e1, b \rangle$
$\uparrow^{(a)}_{(b)} [\_e2 : B]$
$\langle b, \_e2, a \rangle$
$\bigcirc_{(a)}$
$\langle a \rangle$
$\bigcirc_{(b)}$
$\langle b \rangle$

Relational algebra tree for incremental queries



π_a
$\langle a \rangle$
$\bowtie \{a, b\}$
$\langle a, \_e1, b, \_e2 \rangle$
$\Uparrow^{(b)}_{(a)} [\_e1 : A]$
$\langle a, \_e1, b \rangle$
$\Uparrow^{(a)}_{(b)} [\_e2 : B]$
$\langle b, \_e2, a \rangle$

2.18.46 Handling fixed-length variable length pattern

Query specification

```
MATCH (a)-[r*1..1]->(b)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.47 Matching from null nodes should return no results owing to finding no matches

Query specification

```
OPTIONAL MATCH (a)
WITH a
MATCH (a)-->(b)
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.48 Matching from null nodes should return no results owing to matches being filtered out

Query specification

```
OPTIONAL MATCH (a:Label)
WITH a
MATCH (a)-->(b)
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.49 Optionally matching from null nodes should return null

Query specification

```
OPTIONAL MATCH (a)
WITH a
OPTIONAL MATCH (a)-->(b)
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.50 OPTIONAL MATCH returns null

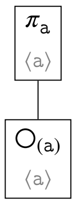
Query specification

```
OPTIONAL MATCH (a)
RETURN a
```

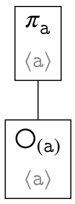
Relational algebra expression

$\pi_{\mathsf{a}}\left(\not\doteqdot\left(\bigcirc_{(\mathsf{a})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.51 Zero-length named path

Query specification

```
MATCH p = (a)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.52 Variable-length named path

Query specification

```
MATCH p = ()-[*0..]->()
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.53 Matching with aggregation

Query specification

```
MATCH (n)
RETURN n.prop AS n, count(n) AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.54 Matching using a relationship that is already bound

Query specification

```
MATCH ()-[r1]->()
WITH r1 AS r2
MATCH ()-[r2]->()
RETURN r2 AS rel
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.55 Matching using a relationship that is already bound, in conjunction with aggregation

Query specification

```
MATCH ()-[r1]->()
WITH r1 AS r2, count(*) AS c
  ORDER BY c
MATCH ()-[r2]->()
RETURN r2 AS rel
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.56 Matching using a relationship that is already bound, in conjunction with aggregation and ORDER BY

Query specification

```
MATCH (a)-[r]->(b)
WITH a, r, b, count(*) AS c
  ORDER BY c
MATCH (a)-[r]->(b)
RETURN r AS rel
  ORDER BY rel.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.57 Matching with LIMIT and optionally matching using a relationship that is already bound

Query specification

```
MATCH ()-[r]->()
WITH r
  LIMIT 1
OPTIONAL MATCH (a2)-[r]->(b2)
RETURN a2, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.58 Matching with LIMIT and optionally matching using a relationship and node that are both already bound

Query specification

```
MATCH (a1)-[r]->()
WITH r, a1
  LIMIT 1
OPTIONAL MATCH (a1)-[r]->(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.59 Matching with LIMIT, then matching again using a relationship and node that are both already bound along with an additional predicate

Query specification

```
MATCH (a1)-[r]->()
WITH r, a1
  LIMIT 1
MATCH (a1:X)-[r]->(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.60 Matching with LIMIT and predicates, then matching again using a relationship and node that are both already bound along with a duplicate predicate

Query specification

```
MATCH (a1:X:Y)-[r]->()
WITH r, a1
  LIMIT 1
MATCH (a1:Y)-[r]->(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.61 Matching twice with conflicting relationship types on same relationship

Query specification

```
MATCH (a1)-[r:T]->()
WITH r, a1
  LIMIT 1
MATCH (a1)-[r:Y]->(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.62 Matching twice with duplicate relationship types on same relationship

Query specification

```
MATCH (a1)-[r:T]->() WITH r, a1
LIMIT 1
MATCH (a1)-[r:T]->(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.63 Matching relationships into a list and matching variable length using the list

Query specification

```
MATCH ()-[r1]->()-[r2]->()
WITH [r1, r2] AS rs
  LIMIT 1
MATCH (first)-[rs*]->(second)
RETURN first, second
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.64 Matching relationships into a list and matching variable length using the list, with bound nodes

Query specification

```
MATCH (a)-[r1]->()-[r2]->(b)
WITH [r1, r2] AS rs, a AS first, b AS second
  LIMIT 1
MATCH (first)-[rs*]->(second)
RETURN first, second
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.65 Matching relationships into a list and matching variable length using the list, with bound nodes, wrong direction

Query specification

```
MATCH (a)-[r1]->()-[r2]->(b)
WITH [r1, r2] AS rs, a AS second, b AS first
  LIMIT 1
MATCH (first)-[rs*]->(second)
RETURN first, second
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.66 Matching and optionally matching with bound nodes in reverse direction

Query specification

```
MATCH (a1)-[r]->()
WITH r, a1
  LIMIT 1
OPTIONAL MATCH (a1)<-[r]-(b2)
RETURN a1, r, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.67 Matching and optionally matching with unbound nodes and equality predicate in reverse direction

Query specification

```
MATCH (a1)-[r]->()
WITH r, a1
  LIMIT 1
OPTIONAL MATCH (a2)<-[r]-(b2)
WHERE a1 = a2
RETURN a1, r, b2, a2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.68 Matching and returning ordered results, with LIMIT

Query specification

```
MATCH (foo)
RETURN foo.bar AS x
  ORDER BY x DESC
  LIMIT 4
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.69 Counting an empty graph

Query specification

```
MATCH (a)
RETURN count(a) > 0
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.70 Matching variable length pattern with property predicate

Query specification

```
MATCH (a:Artist)-[:WORKED_WITH* {year: 1988}]->(b:Artist)
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.71 Variable length pattern checking labels on endnodes

Query specification

```
MATCH (a), (b)
WHERE a.id = 0
  AND (a)-[:T]->(b:Label)
  OR (a)-[:T*]->(b:MissingLabel)
RETURN DISTINCT b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.72 Variable length pattern with label predicate on both sides

Query specification

```
MATCH (a:Blue)-[r*]->(b:Green)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.73 Undirected named path

Query specification

```
MATCH p = (n:Movie)--(m)
RETURN p
  LIMIT 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.74 Named path with WITH

Query specification

```
MATCH p = (a)
WITH p
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.75 Named path with alternating directed/undirected relationships

Query specification

```
MATCH p = (n)-->(m)--(o)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.76 Named path with multiple alternating directed/undirected relationships

Query specification

```
MATCH path = (n)-->(m)--(o)--(p)
RETURN path
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.77 Named path with undirected fixed variable length pattern

Query specification

```
MATCH topRoute = (:Start)<-[:CONNECTED_TO]-()-[:CONNECTED_TO*3..3]-(:End)
RETURN topRoute
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.78 Returning a node property value

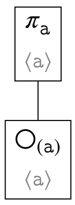
Query specification

```
MATCH (a)
RETURN a.prop
```

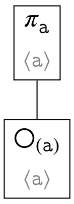
Relational algebra expression

$\pi_{\mathsf{a}}\left(\not\fallingdotseq\left(\bigcirc_{(\mathsf{a})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.79 Returning a relationship property value

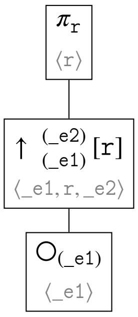
Query specification

```
MATCH ()-[r]->()
RETURN r.prop
```

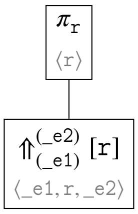
Relational algebra expression

$\pi_{\mathtt{r}}\left(\dot{\neq}\left(\uparrow\,{}^{(\_e2)}_{(\_e1)}[\mathtt{r}]\left(\bigcirc_{(\_e1)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.80 Projecting nodes and relationships

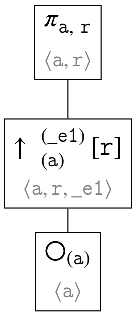
Query specification

```
MATCH (a)-[r]->()
RETURN a AS foo, r AS bar
```

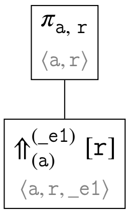
Relational algebra expression

$\pi_{\mathtt{a},\,\mathtt{r}}\left(\dot{\neq}\left(\uparrow\,{}^{(\_e1)}_{(a)}[\mathtt{r}]\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{a},\,\mathtt{r}}$
$\langle \mathtt{a}, \mathtt{r} \rangle$
$\uparrow^{(\_e1)}_{(a)} [\mathtt{r}]$
$\langle \mathtt{a}, \mathtt{r}, \_e1 \rangle$
$\bigcirc_{(a)}$
$\langle \mathtt{a} \rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{a},\,\mathtt{r}}$
$\langle \mathtt{a}, \mathtt{r} \rangle$
$\Uparrow^{(\_e1)}_{(a)} [\mathtt{r}]$
$\langle \mathtt{a}, \mathtt{r}, \_e1 \rangle$

2.18.81 Missing node property should become null

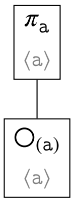
Query specification

```
MATCH (a)
RETURN a.bar
```

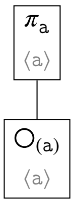
Relational algebra expression

$\pi_{\mathtt{a}}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.82 Missing relationship property should become null

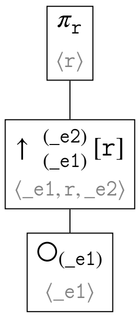
Query specification

```
MATCH ()-[r]->()
RETURN r.bar
```

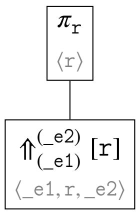
Relational algebra expression

$\pi_{\mathtt{r}}\left(\not\equiv\left(\uparrow^{(\_e2)}_{(\_e1)}[\mathtt{r}]\left(\bigcirc_{(\_e1)}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{r}}$
$\langle \mathtt{r} \rangle$
$\uparrow^{(\_e2)}_{(\_e1)} [\mathtt{r}]$
$\langle \_e1, r, \_e2 \rangle$
$\bigcirc_{(\_e1)}$
$\langle \_e1 \rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{r}}$
$\langle \mathtt{r} \rangle$
$\Uparrow^{(\_e2)}_{(\_e1)} [\mathtt{r}]$
$\langle \_e1, r, \_e2 \rangle$

2.18.83 Returning multiple node property values

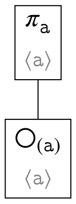
Query specification

```
MATCH (a)
RETURN a.name, a.age, a.seasons
```

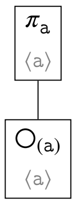
Relational algebra expression

$\pi_{\mathtt{a}} \left(\not\equiv \left(\bigcirc_{(\mathrm{a})} \right) \right)$

Relational algebra tree


$\pi_{\mathtt{a}}$
$\langle \mathtt{a} \rangle$
$\bigcirc_{(\mathrm{a})}$
$\langle \mathtt{a} \rangle$

Relational algebra tree for incremental queries


$\pi_{\mathtt{a}}$
$\langle \mathtt{a} \rangle$
$\bigcirc_{(\mathrm{a})}$
$\langle \mathtt{a} \rangle$

2.18.84 Adding a property and a literal in projection

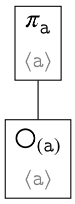
Query specification

```
MATCH (a)
RETURN a.prop + 1 AS foo
```

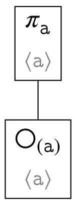
Relational algebra expression

$\pi_{\mathtt{a}} \left(\not\equiv \left(\bigcirc_{(\mathrm{a})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.85 Adding list properties in projection

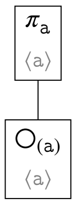
Query specification

```
MATCH (a)
RETURN a.prop2 + a.prop1 AS foo
```

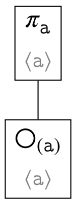
Relational algebra expression

$\pi_{a}\left(\not\fallingdotseq\left(\bigcirc_{(a)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.18.86 Variable length relationship variables are lists of relationships

Query specification

```
MATCH ()-[r*0..1]-()
RETURN last(r) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.87 Variable length patterns and nulls

Query specification

```
MATCH (a:A)
OPTIONAL MATCH (a)-[:FOO]->(b:B)
OPTIONAL MATCH (b)<-[:BAR*]-(c:B)
RETURN a, b, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.88 Projecting a list of nodes and relationships

Query specification

```
MATCH (n)-[r]->(m)
RETURN [n, r, m] AS r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.89 Projecting a map of nodes and relationships

Query specification

```
MATCH (n)-[r]->(m)
RETURN {node1: n, rel: r, node2: m} AS m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.90 Respecting direction when matching existing path

Query specification

```
MATCH p = ({prop: 'a'})-->({prop: 'b'})
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.91 Respecting direction when matching non-existent path

Query specification

```
MATCH p = ({prop: 'a'})<--({prop: 'b'})
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.92 Respecting direction when matching non-existent path with multiple directions

Query specification

```
MATCH p = (n)-->(k)<--(n)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.93 Matching path with both directions should respect other directions

Query specification

```
MATCH p = (n)<-->(k)<--(n)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.94 Matching path with multiple bidirectional relationships

Query specification

```
MATCH p=(n)<-->(k)<-->(n)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.95 Matching nodes with many labels

Query specification

```
MATCH (n:A:B:C:D:E:F:G:H:I:J:K:L:M)-[:T]->(m:Z:Y:X:W:V:U)
RETURN n, m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.96 Matching longer variable length paths

Query specification

```
MATCH (n {prop: 'start'})-[:T*]->(m {prop: 'end'})
RETURN m
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.97 Counting rows after MATCH, MERGE, OPTIONAL MATCH

Query specification

```
MATCH (a)
MERGE (b)
WITH *
OPTIONAL MATCH (a)--(b)
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.18.98 Matching a self-loop

Query specification

```
MATCH ()-[r]-()
RETURN type(r) AS r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19 MergeIntoAcceptance

2.19.1 Updating one property with ON CREATE

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r.name = 'foo'MATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19.2 Null-setting one property with ON CREATE

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r.name = nullMATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19.3 Copying properties from node with ON CREATE

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r = aMATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19.4 Copying properties from node with ON MATCH

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON MATCH SET r = aMATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19.5 Copying properties from literal map with ON CREATE

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r += {foo: 'bar', bar: 'baz'}MATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.19.6 Copying properties from literal map with ON MATCH

Query specification

```
MATCH (a {name: 'A'}), (b {name: 'B'})
MERGE (a)-[r:TYPE]->(b)
  ON MATCH SET r += {foo: 'baz', bar: 'baz'}MATCH ()-[r:TYPE]->()
RETURN [key IN keys(r) | key + '->' + r[key]] AS keyValue
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20 MergeNodeAcceptance

2.20.1 Merge node when no nodes exist

Query specification

```
MERGE (a)
RETURN count(*) AS n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.2 Merge node with label

Query specification

```
MERGE (a:Label)
RETURN labels(a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.3 Merge node with label add label on create

Query specification

```
MERGE (a:Label)
  ON CREATE SET a:Foo
RETURN labels(a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.4 Merge node with label add property on create

Query specification

```
MERGE (a:Label)
  ON CREATE SET a.prop = 42
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.5 Merge node with label when it exists

Query specification

```
MERGE (a:Label)
RETURN a.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.6 Merge node should create when it doesn't match, properties

Query specification

```
MERGE (a {prop: 43})
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.7 Merge node should create when it doesn't match, properties and label

Query specification

```
MERGE (a:Label {prop: 43})
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.8 Merge node with prop and label

Query specification

```
MERGE (a:Label {prop: 42})
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.9 Merge node with label add label on match when it exists

Query specification

```
MERGE (a:Label)
  ON MATCH SET a:Foo
RETURN labels(a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.10 Merge node with label add property on update when it exists

Query specification

```
MERGE (a:Label)
  ON CREATE SET a.prop = 42
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.11 Merge node and set property on match

Query specification

```
MERGE (a:Label)
  ON MATCH SET a.prop = 42
RETURN a.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.12 Should work when finding multiple elements

Query specification

```
CREATE (:X)
CREATE (:X)
MERGE (:X)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.13 Should handle argument properly

Query specification

```
WITH 42 AS x
MERGE (c:N {x: x})
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.14 Should handle arguments properly with only write clauses

Query specification

```
CREATE (a {p: 1})
MERGE ({v: a.p})
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.15 Should be able to merge using property from match

Query specification

```
MATCH (person:Person)
MERGE (city:City {name: person.bornIn})
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.16 Should be able to use properties from match in ON CREATE

Query specification

```
MATCH (person:Person)
MERGE (city:City)
  ON CREATE SET city.name = person.bornIn
RETURN person.bornIn
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.17 Should be able to use properties from match in ON MATCH

Query specification

```
MATCH (person:Person)
MERGE (city:City)
  ON MATCH SET city.name = person.bornIn
RETURN person.bornIn
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.18 Should be able to use properties from match in ON MATCH and ON CREATE

Query specification

```
  MATCH (person:Person)
  MERGE (city:City)
    ON MATCH SET city.name = person.bornIn
    ON CREATE SET city.name = person.bornIn
  RETURN person.bornIn
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.19 Should be able to set labels on match

Query specification

```
MERGE (a)
  ON MATCH SET a:L
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.20 Should be able to set labels on match and on create

Query specification

```
MATCH ()
MERGE (a:L)
  ON MATCH SET a:M1
  ON CREATE SET a:M2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.21 Should support updates while merging

Query specification

```
MATCH (foo)
WITH foo.x AS x, foo.y AS y
MERGE (:N {x: x, y: y + 1})
MERGE (:N {x: x, y: y})
MERGE (:N {x: x + 1, y: y})
RETURN x, y
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.22 Merge must properly handle multiple labels

Query specification

```
MERGE (test:L:B {prop: 42})
RETURN labels(test) AS labels
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.23 Merge followed by multiple creates

Query specification

```
MERGE (t:T {id: 42})
CREATE (f:R)
CREATE (t)-[:REL]->(f)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.24 Unwind combined with merge

Query specification

```
UNWIND [1, 2, 3, 4] AS int
MERGE (n {id: int})
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.25 Merges should not be able to match on deleted nodes

Query specification

```
MATCH (a:A)
DELETE a
MERGE (a2:A)
RETURN a2.value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.20.26 ON CREATE on created nodes

Query specification

```
MERGE (b)
  ON CREATE SET b.created = 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21 MergeRelationshipAcceptance

2.21.1 Creating a relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.2 Matching a relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.3 Matching two relationships

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.4 Filtering relationships

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE {name: ’r2’}]->(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.5 Creating relationship when all matches filtered out

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE {name: ’r2’}]->(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.6 Matching incoming relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)<-[r:TYPE]-(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.7 Creating relationship with property

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE {name: 'Lola'}]->(b)
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.8 Using ON CREATE on a node

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[:KNOWS]->(b)
  ON CREATE SET b.created = 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.9 Using ON CREATE on a relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r.name = 'Lola'
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.10 Using ON MATCH on created node

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[:KNOWS]->(b)
  ON MATCH SET b.created = 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.11 Using ON MATCH on created relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:KNOWS]->(b)
  ON MATCH SET r.created = 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.12 Using ON MATCH on a relationship

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
  ON MATCH SET r.name = 'Lola'
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.13 Using ON CREATE and ON MATCH

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:TYPE]->(b)
  ON CREATE SET r.name = 'Lola'
  ON MATCH SET r.name = 'RUN'
RETURN count(r)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.14 Creating relationship using merged nodes

Query specification

```
MERGE (a:A)
MERGE (b:B)
MERGE (a)-[:FOO]->(b)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.15 Mixing MERGE with CREATE

Query specification

```
CREATE (a:A), (b:B)
MERGE (a)-[:KNOWS]->(b)
CREATE (b)-[:KNOWS]->(c:C)
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.16 Introduce named paths 1

Query specification

```
MERGE (a {x: 1})
MERGE (b {x: 2})
MERGE p = (a)-[:R]->(b)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.17 Introduce named paths 2

Query specification

```
MERGE p = (a {x: 1})
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.18 Use outgoing direction when unspecified

Query specification

```
CREATE (a {id: 2}), (b {id: 1})
MERGE (a)-[r:KNOWS]-(b)
RETURN startNode(r).id AS s, endNode(r).id AS e
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.19 Match outgoing relationship when direction unspecified

Query specification

```
MATCH (a {id: 2}), (b {id: 1})
MERGE (a)-[r:KNOWS]-(b)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.20 Match both incoming and outgoing relationships when direction unspecified

Query specification

```
MATCH (a {id: 2})--(b {id: 1})
MERGE (a)-[r:KNOWS]-(b)
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.21 Using list properties via variable

Query specification

```
CREATE (a:Foo), (b:Bar)
WITH a, b
UNWIND ['a,b', 'a,b'] AS str
WITH a, b, split(str, ',') AS roles
MERGE (a)-[r:FB {foobar: roles}]->(b)
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.22 Matching using list property

Query specification

```
MATCH (a:A), (b:B)
MERGE (a)-[r:T {prop: [42, 43]}]->(b)
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.23 Using bound variables from other updating clause

Query specification

```
CREATE (a), (b)
MERGE (a)-[:X]->(b)
RETURN count(a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.24 UNWIND with multiple merges

Query specification

```
UNWIND ['Keanu Reeves', 'Hugo Weaving', 'Carrie-Anne Moss', 'Laurence Fishburne'] AS actor
MERGE (m:Movie {name: 'The Matrix'})
MERGE (p:Person {name: actor})
MERGE (p)-[:ACTED_IN]->(m)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.25 Do not match on deleted entities

Query specification

```
MATCH (a:A)-[ab]->(b:B)-[bc]->(c:C)
DELETE ab, bc, b, c
MERGE (newB:B {value: 1})
MERGE (a)-[:REL]->(newB)
MERGE (newC:C)
MERGE (newB)-[:REL]->(newC)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.26 Do not match on deleted relationships

Query specification

```
MATCH (a)-[t:T]->(b)
DELETE t
MERGE (a)-[t2:T {name: 'rel3'}]->(b)
RETURN t2.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.27 Aliasing of existing nodes 1

Query specification

```
MATCH (n)
MATCH (m)
WITH n AS a, m AS b
MERGE (a)-[r:T]->(b)
RETURN a.id AS a, b.id AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.28 Aliasing of existing nodes 2

Query specification

```
MATCH (n)
WITH n AS a, n AS b
MERGE (a)-[r:T]->(b)
RETURN a.id AS a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.29 Double aliasing of existing nodes 1

Query specification

```
MATCH (n)
MATCH (m)
WITH n AS a, m AS b
MERGE (a)-[:T]->(b)
WITH a AS x, b AS y
MERGE (a)
MERGE (b)
MERGE (a)-[:T]->(b)
RETURN x.id AS x, y.id AS y
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.21.30 Double aliasing of existing nodes 2

Query specification

```
MATCH (n)
WITH n AS a
MERGE (c)
MERGE (a)-[:T]->(c)
WITH a AS x
MERGE (c)
MERGE (x)-[:T]->(c)
RETURN x.id AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.22 MiscellaneousErrorAcceptance

2.23 NullAcceptance

2.23.1 Ignore null when setting property

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
SET a.prop = 42
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.2 Ignore null when removing property

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
REMOVE a.prop
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.3 Ignore null when setting properties using an appending map

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
SET a += {prop: 42}
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.4 Ignore null when setting properties using an overriding map

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
SET a = {prop: 42}
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.5 Ignore null when setting label

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
SET a:L
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.6 Ignore null when removing label

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
REMOVE a:L
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.7 Ignore null when deleting node

Query specification

```
OPTIONAL MATCH (a:DoesNotExist)
DELETE a
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.23.8 Ignore null when deleting relationship

Query specification

```
OPTIONAL MATCH ()-[r:DoesNotExist]-()
DELETE r
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.24 OptionalMatch

2.24.1 Satisfies the open world assumption, relationships between same nodes

Query specification

```
MATCH (p:Player)-[:PLAYS_FOR]->(team:Team)
OPTIONAL MATCH (p)-[s:SUPPORTS]->(team)
RETURN count(*) AS matches, s IS NULL AS optMatch
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.24.2 Satisfies the open world assumption, single relationship

Query specification

```
MATCH (p:Player)-[:PLAYS_FOR]->(team:Team)
OPTIONAL MATCH (p)-[s:SUPPORTS]->(team)
RETURN count(*) AS matches, s IS NULL AS optMatch
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.24.3 Satisfies the open world assumption, relationships between different nodes

Query specification

```
MATCH (p:Player)-[:PLAYS_FOR]->(team:Team)
OPTIONAL MATCH (p)-[s:SUPPORTS]->(team)
RETURN count(*) AS matches, s IS NULL AS optMatch
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25 OptionalMatchAcceptance

2.25.1 Return null when no matches due to inline label predicate

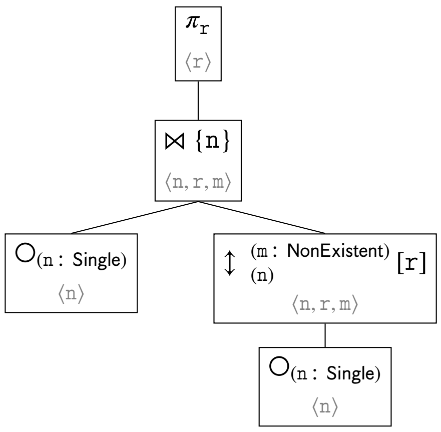
Query specification

```
MATCH (n:Single)
OPTIONAL MATCH (n)-[r]-(m:NonExistent)
RETURN r
```

Relational algebra expression

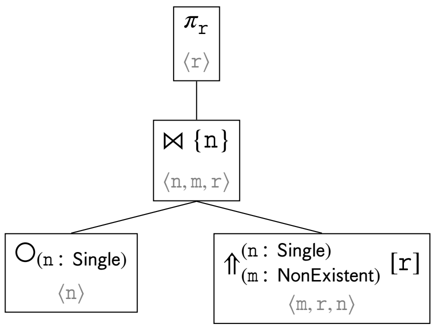
$$\pi_{\mathtt{r}}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right) \bowtie \{\mathtt{n}\} \not\doteqdot \left(\updownarrow^{(\mathtt{m}:\ \mathsf{NonExistent})}_{(\mathtt{n})} [\mathtt{r}] \left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{r}}$
⟨r⟩
$\bowtie \{\mathtt{n}\}$
⟨n, r, m⟩
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
⟨n⟩
$\updownarrow^{(\mathtt{m}:\ \mathsf{NonExistent})}_{(\mathtt{n})} [\mathtt{r}]$
⟨n, r, m⟩
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
⟨n⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{r}}$
⟨r⟩
$\bowtie \{\mathtt{n}\}$
⟨n, m, r⟩
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
⟨n⟩
$\Uparrow^{(\mathtt{n}:\ \mathsf{Single})}_{(\mathtt{m}:\ \mathsf{NonExistent})} [\mathtt{r}]$
⟨m, r, n⟩

2.25.2 Return null when no matches due to label predicate in WHERE

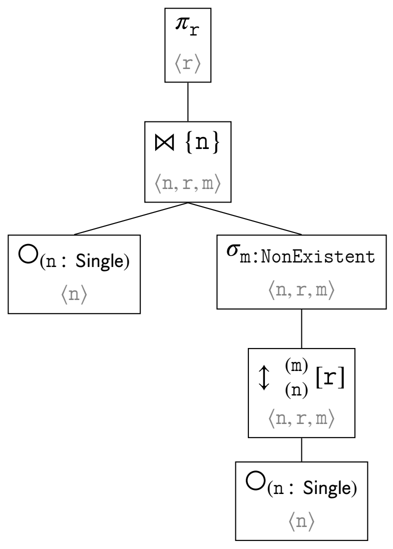
Query specification

```
MATCH (n:Single)
OPTIONAL MATCH (n)-[r]-(m)
WHERE m:NonExistent
RETURN r
```

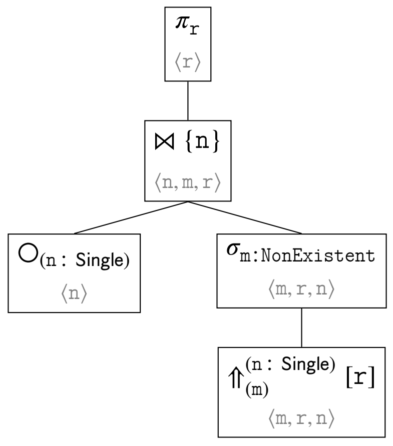
Relational algebra expression

$$\pi_{\mathtt{r}}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right) \bowtie \{\mathtt{n}\} \sigma_{\mathtt{m:NonExistent}}\left(\not\doteqdot\left(\updownarrow^{(\mathtt{m})}_{(\mathtt{n})} [\mathtt{r}] \left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.25.3 Respect predicates on the OPTIONAL MATCH

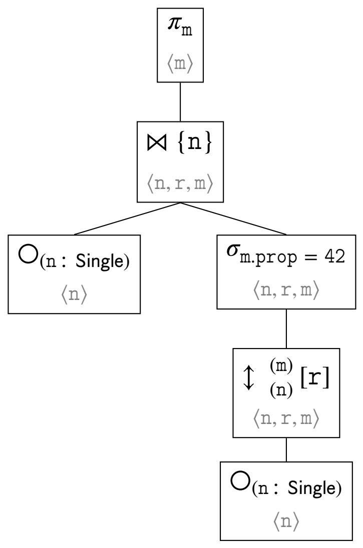
Query specification

```
MATCH (n:Single)
OPTIONAL MATCH (n)-[r]-(m)
WHERE m.prop = 42
RETURN m
```

Relational algebra expression

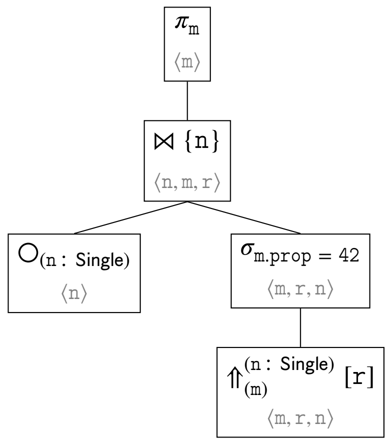
$$\pi_{\mathtt{m}}\left(\not\doteq\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right) \bowtie \{\mathtt{n}\}\sigma_{\mathtt{m.prop}\ =\ 42}\left(\not\doteq\left(\updownarrow\,{}^{(\mathtt{m})}_{(\mathtt{n})}\,[\mathtt{r}]\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right)\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{m}}$
$\langle \mathtt{m} \rangle$
$\bowtie \{\mathtt{n}\}$
$\langle \mathtt{n}, \mathtt{r}, \mathtt{m} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$
$\sigma_{\mathtt{m.prop}\ =\ 42}$
$\langle \mathtt{n}, \mathtt{r}, \mathtt{m} \rangle$
$\updownarrow {}^{(\mathtt{m})}_{(\mathtt{n})} [\mathtt{r}]$
$\langle \mathtt{n}, \mathtt{r}, \mathtt{m} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$

Relational algebra tree for incremental queries



$\pi_{\mathtt{m}}$
$\langle \mathtt{m} \rangle$
$\bowtie \{\mathtt{n}\}$
$\langle \mathtt{n}, \mathtt{m}, \mathtt{r} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$
$\sigma_{\mathtt{m.prop}\ =\ 42}$
$\langle \mathtt{m}, \mathtt{r}, \mathtt{n} \rangle$
$\Uparrow^{(\mathtt{n}:\ \mathsf{Single})}_{(\mathtt{m})} [\mathtt{r}]$
$\langle \mathtt{m}, \mathtt{r}, \mathtt{n} \rangle$

2.25.4 Returning label predicate on null node

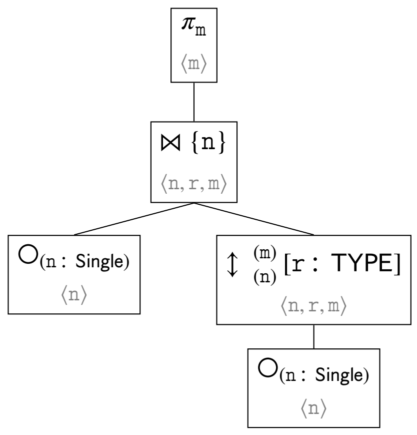
Query specification

```
MATCH (n:Single)
OPTIONAL MATCH (n)-[r:TYPE]-(m)
RETURN m:TYPE
```

Relational algebra expression

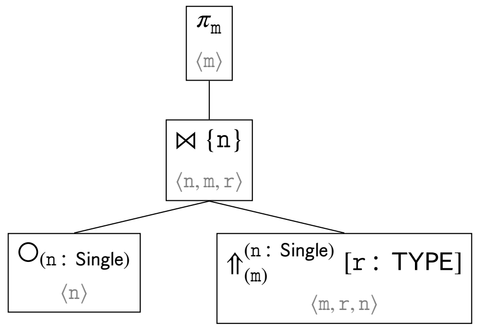
$$\pi_{\mathtt{m}}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right) \bowtie \{\mathtt{n}\} \not\doteqdot \left(\updownarrow {}^{(\mathtt{m})}_{(\mathtt{n})} [\mathtt{r}:\ \mathsf{TYPE}] \left(\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{m}}$
$\langle \mathtt{m} \rangle$
$\bowtie \{\mathtt{n}\}$
$\langle \mathtt{n}, \mathtt{r}, \mathtt{m} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$
$\updownarrow^{(\mathtt{m})}_{(\mathtt{n})} [\mathtt{r}:\ \mathsf{TYPE}]$
$\langle \mathtt{n}, \mathtt{r}, \mathtt{m} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$

Relational algebra tree for incremental queries



$\pi_{\mathtt{m}}$
$\langle \mathtt{m} \rangle$
$\bowtie \{\mathtt{n}\}$
$\langle \mathtt{n}, \mathtt{m}, \mathtt{r} \rangle$
$\bigcirc_{(\mathtt{n}:\ \mathsf{Single})}$
$\langle \mathtt{n} \rangle$
$\Uparrow^{(\mathtt{n}:\ \mathsf{Single})}_{(\mathtt{m})} [\mathtt{r}:\ \mathsf{TYPE}]$
$\langle \mathtt{m}, \mathtt{r}, \mathtt{n} \rangle$

2.25.5 MATCH after OPTIONAL MATCH

Query specification

```
MATCH (a:Single)
OPTIONAL MATCH (a)-->(b:NonExistent)
OPTIONAL MATCH (a)-->(c:NonExistent)
WITH coalesce(b, c) AS x
MATCH (x)-->(d)
RETURN d
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.6 WITH after OPTIONAL MATCH

Query specification

```
OPTIONAL MATCH (a:A)
WITH a AS a
MATCH (b:B)
RETURN a, b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.7 Named paths in optional matches

Query specification

```
MATCH (a:A)
OPTIONAL MATCH p = (a)-[:X]->(b)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.8 OPTIONAL MATCH and bound nodes

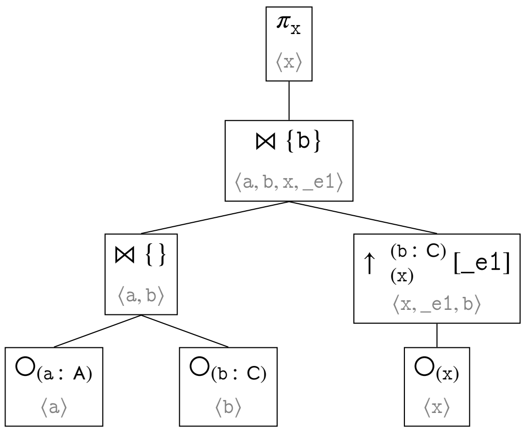
Query specification

```
MATCH (a:A), (b:C)
OPTIONAL MATCH (x)-->(b)
RETURN x
```

Relational algebra expression

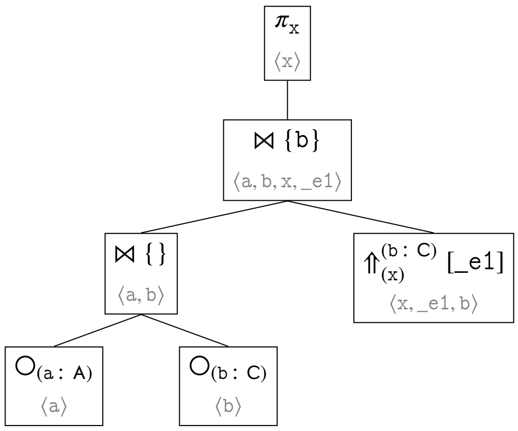
$$\pi_{\mathtt{x}} \left(\not\doteqdot \left(\bigcirc_{(\mathtt{a}:\mathtt{A})} \bowtie \{\} \bigcirc_{(\mathtt{b}:\mathtt{C})} \right) \bowtie \{\mathtt{b}\} \not\doteqdot \left(\uparrow {}^{(\mathtt{b}:\mathtt{C})}_{(\mathtt{x})} [\_\mathtt{e1}] \left(\bigcirc_{(\mathtt{x})} \right) \right) \right)$$

Relational algebra tree



$\pi_{\mathtt{x}}$
⟨x⟩
⋈ {b}
⟨a, b, x, _e1⟩
⋈ {}
⟨a, b⟩
$\uparrow {}^{(\mathtt{b}:\mathtt{C})}_{(\mathtt{x})} [\_\mathtt{e1}]$
⟨x, _e1, b⟩
$\bigcirc_{(\mathtt{a}:\mathtt{A})}$
⟨a⟩
$\bigcirc_{(\mathtt{b}:\mathtt{C})}$
⟨b⟩
$\bigcirc_{(\mathtt{x})}$
⟨x⟩

Relational algebra tree for incremental queries



2.25.9 OPTIONAL MATCH with labels on the optional end node

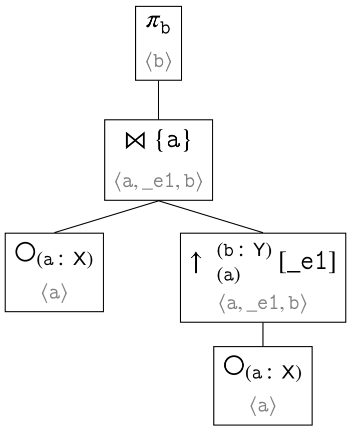
Query specification

```
MATCH (a:X)
OPTIONAL MATCH (a)-->(b:Y)
RETURN b
```

Relational algebra expression

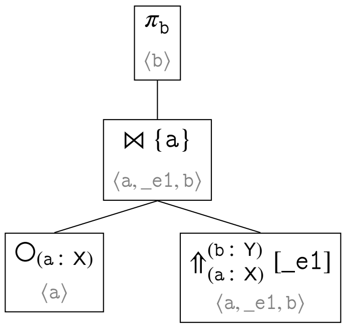
$$\pi_b\left(\doteqdot\left(\bigcirc_{(a:X)}\right) \bowtie\{a\} \doteqdot\left(\uparrow^{(b:Y)}_{(a)}[\_e1]\left(\bigcirc_{(a:X)}\right)\right)\right)$$

Relational algebra tree



π_b ⟨b⟩
$\bowtie\{a\}$ ⟨a, _e1, b⟩
$\bigcirc_{(a:X)}$ ⟨a⟩
$\uparrow^{(b:Y)}_{(a)}$ [_e1] ⟨a, _e1, b⟩
$\bigcirc_{(a:X)}$ ⟨a⟩

Relational algebra tree for incremental queries



π_b ⟨b⟩
$\bowtie\{a\}$ ⟨a, _e1, b⟩
$\bigcirc_{(a:X)}$ ⟨a⟩
$\Uparrow^{(b:Y)}_{(a:X)}$ [_e1] ⟨a, _e1, b⟩

2.25.10 Named paths inside optional matches with node predicates

Query specification

```
MATCH (a:A), (b:B)
OPTIONAL MATCH p = (a)-[:X]->(b)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.11 Variable length optional relationships

Query specification

```
MATCH (a:Single)
OPTIONAL MATCH (a)-[*]->(b)
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.12 Variable length optional relationships with length predicates

Query specification

```
MATCH (a:Single)
OPTIONAL MATCH (a)-[*3..]-(b)
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.13 Optionally matching self-loops

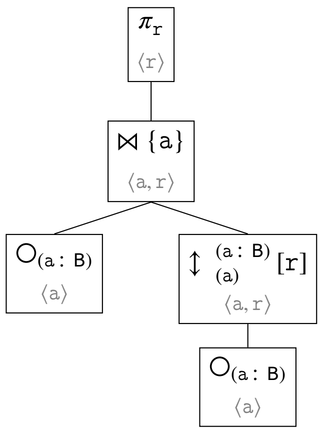
Query specification

```
MATCH (a:B)
OPTIONAL MATCH (a)-[r]-(a)
RETURN r
```

Relational algebra expression

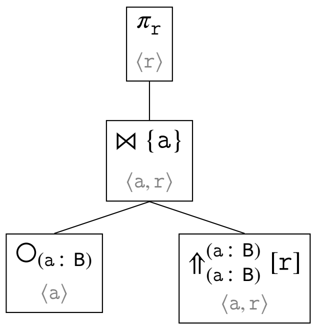
$$\pi_{\mathtt{r}}\left(\not\doteq\left(\bigcirc_{(\mathtt{a}:\,\mathtt{B})}\right)\bowtie\{\mathtt{a}\}\not\doteq\left(\updownarrow^{(\mathtt{a}:\,\mathtt{B})}_{(\mathtt{a})}[\mathtt{r}]\left(\bigcirc_{(\mathtt{a}:\,\mathtt{B})}\right)\right)\right)$$

Relational algebra tree



$\pi_{\mathtt{r}}$
⟨r⟩
$\bowtie\{\mathtt{a}\}$
⟨a, r⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{B})}$
⟨a⟩
$\updownarrow^{(\mathtt{a}:\,\mathtt{B})}_{(\mathtt{a})}[\mathtt{r}]$
⟨a, r⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{B})}$
⟨a⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{r}}$
⟨r⟩
$\bowtie\{\mathtt{a}\}$
⟨a, r⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{B})}$
⟨a⟩
$\Uparrow^{(\mathtt{a}:\,\mathtt{B})}_{(\mathtt{a}:\,\mathtt{B})}[\mathtt{r}]$
⟨a, r⟩

2.25.14 Optionally matching self-loops without matches

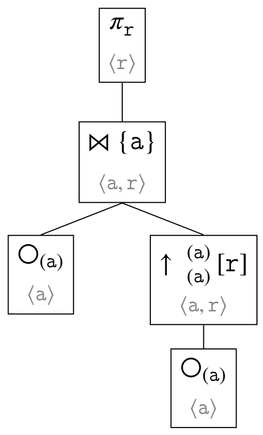
Query specification

```
MATCH (a)
WHERE NOT (a:B)
OPTIONAL MATCH (a)-[r]->(a)
RETURN r
```

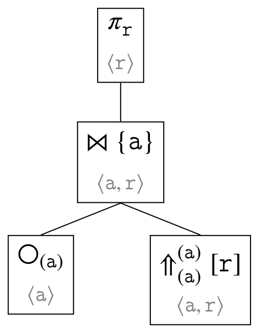
Relational algebra expression

$$\pi_{\mathtt{r}}\left(\not\doteq\left(\bigcirc_{(\mathtt{a})}\right)\bowtie\{\mathtt{a}\}\not\doteq\left(\uparrow^{(\mathtt{a})}_{(\mathtt{a})}[\mathtt{r}]\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.25.15 Variable length optional relationships with bound nodes

Query specification

```
MATCH (a:Single), (x:C)
OPTIONAL MATCH (a)-[*]->(x)
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.16 Variable length optional relationships with bound nodes, no matches

Query specification

```
MATCH (a:A), (b:B)
OPTIONAL MATCH p = (a)-[*]->(b)
RETURN p
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.17 Longer pattern with bound nodes

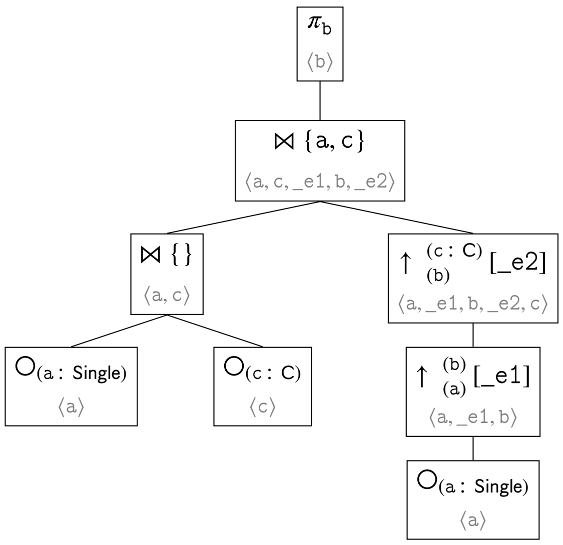
Query specification

```
MATCH (a:Single), (c:C)
OPTIONAL MATCH (a)-->(b)-->(c)
RETURN b
```

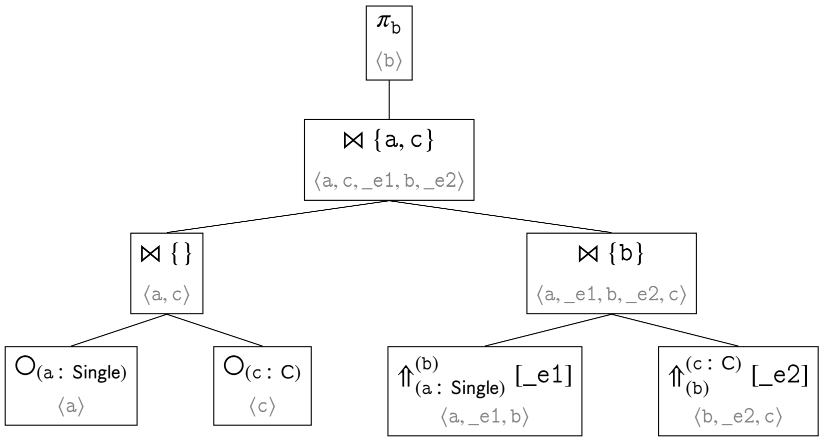
Relational algebra expression

$$\pi_{b}\left(\not\doteq\left(\bigcirc_{(a:\,\text{Single})}\bowtie\{\}\bigcirc_{(c:\,C)}\right)\bowtie\{a,c\}\not\doteq\left(\uparrow\,{}^{(c:\,C)}_{(b)}\,[\_e2]\left(\uparrow\,{}^{(b)}_{(a)}\,[\_e1]\left(\bigcirc_{(a:\,\text{Single})}\right)\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.25.18 Longer pattern with bound nodes without matches

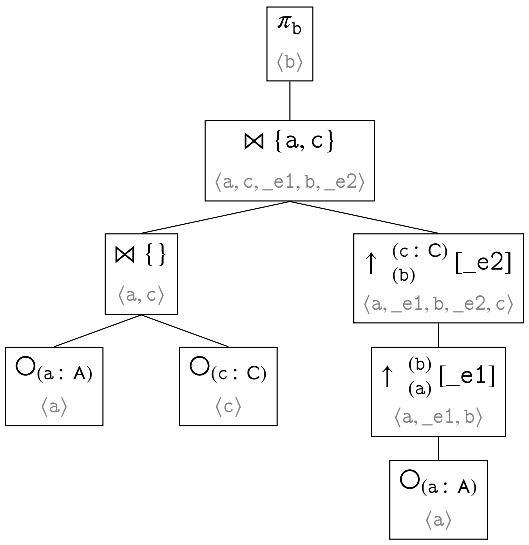
Query specification

```
MATCH (a:A), (c:C)
OPTIONAL MATCH (a)-->(b)-->(c)
RETURN b
```

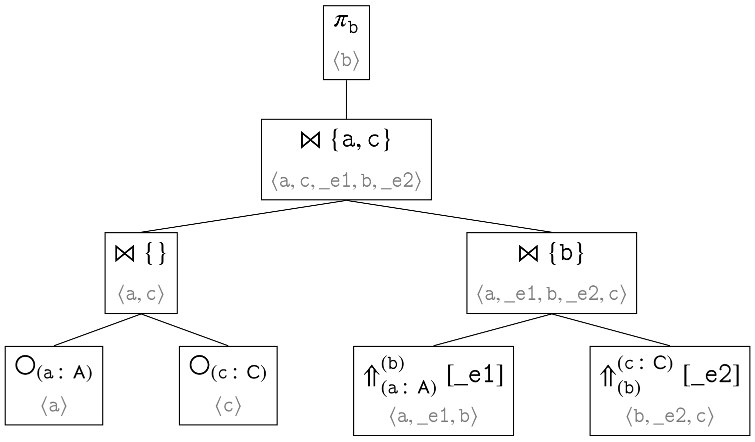
Relational algebra expression

$$\pi_{\mathtt{b}}\left(\not\doteq\left(\bigcirc_{(\mathtt{a}:\,\mathtt{A})}\bowtie\{\}\bigcirc_{(\mathtt{c}:\,\mathtt{C})}\right)\bowtie\{\mathtt{a},\mathtt{c}\}\not\doteq\left(\uparrow\,^{(\mathtt{c}:\,\mathtt{C})}_{(\mathtt{b})}[\mathtt{\_e2}]\left(\uparrow\,^{(\mathtt{b})}_{(\mathtt{a})}[\mathtt{\_e1}]\left(\bigcirc_{(\mathtt{a}:\,\mathtt{A})}\right)\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{b}}$
⟨b⟩
$\bowtie\{\mathtt{a},\mathtt{c}\}$
⟨a, c, _e1, b, _e2⟩
$\bowtie\{\}$
⟨a, c⟩
$\uparrow\,^{(\mathtt{c}:\,\mathtt{C})}_{(\mathtt{b})}[\mathtt{\_e2}]$
⟨a, _e1, b, _e2, c⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{A})}$
⟨a⟩
$\bigcirc_{(\mathtt{c}:\,\mathtt{C})}$
⟨c⟩
$\uparrow\,^{(\mathtt{b})}_{(\mathtt{a})}[\mathtt{\_e1}]$
⟨a, _e1, b⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{A})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{b}}$
⟨b⟩
$\bowtie\{\mathtt{a},\mathtt{c}\}$
⟨a, c, _e1, b, _e2⟩
$\bowtie\{\}$
⟨a, c⟩
$\bowtie\{\mathtt{b}\}$
⟨a, _e1, b, _e2, c⟩
$\bigcirc_{(\mathtt{a}:\,\mathtt{A})}$
⟨a⟩
$\bigcirc_{(\mathtt{c}:\,\mathtt{C})}$
⟨c⟩
$\Uparrow^{(\mathtt{b})}_{(\mathtt{a}:\,\mathtt{A})}[\mathtt{\_e1}]$
⟨a, _e1, b⟩
$\Uparrow^{(\mathtt{c}:\,\mathtt{C})}_{(\mathtt{b})}[\mathtt{\_e2}]$
⟨b, _e2, c⟩

2.25.19 Handling correlated optional matches; first does not match implies second does not match

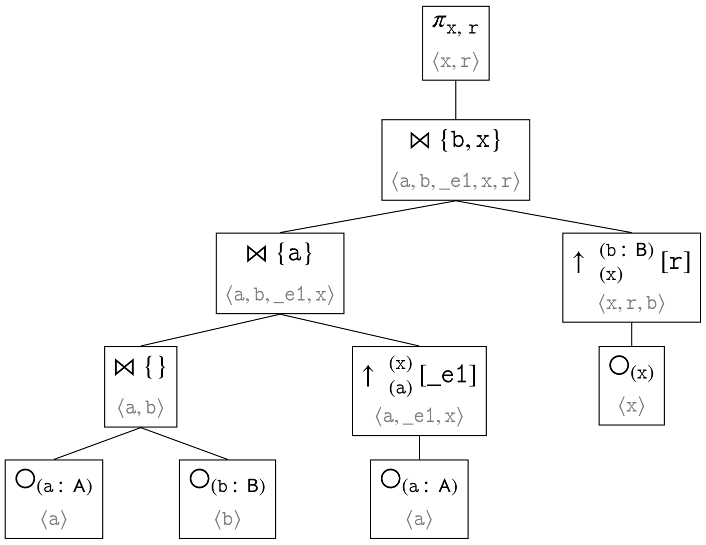
Query specification

```
MATCH (a:A), (b:B)
OPTIONAL MATCH (a)-->(x)
OPTIONAL MATCH (x)-[r]->(b)
RETURN x, r
```

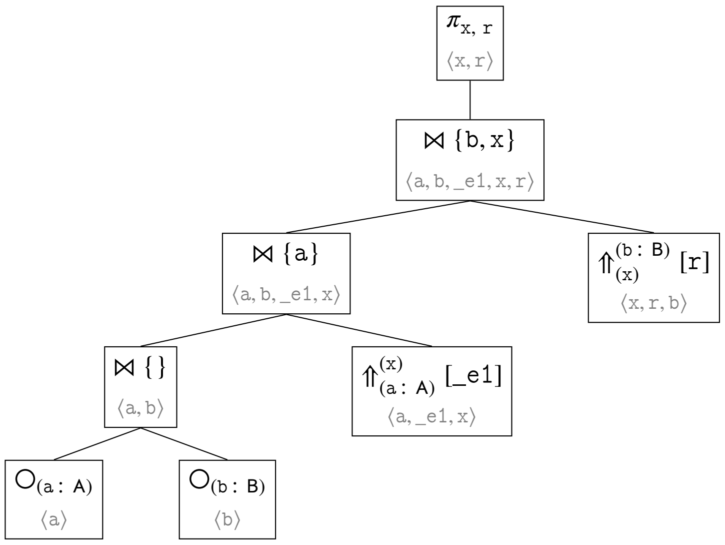
Relational algebra expression

$$\pi_{\mathtt{x,\,r}}\left(\not\doteq\left(\bigcirc_{(\mathtt{a}:\,\mathtt{A})}\bowtie\{\}\bigcirc_{(\mathtt{b}:\,\mathtt{B})}\right)\bowtie\{\mathtt{a}\}\not\doteq\left(\uparrow\,^{(\mathtt{x})}_{(\mathtt{a})}[\mathtt{\_e1}]\left(\bigcirc_{(\mathtt{a}:\,\mathtt{A})}\right)\right)\bowtie\{\mathtt{b},\mathtt{x}\}\not\doteq\left(\uparrow\,^{(\mathtt{b}:\,\mathtt{B})}_{(\mathtt{x})}[\mathtt{r}]\left(\bigcirc_{(\mathtt{x})}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



$\pi_{x, r}$
$\langle x, r \rangle$
$\bowtie \{b, x\}$
$\langle a, b, \_e1, x, r \rangle$
$\bowtie \{a\}$
$\langle a, b, \_e1, x \rangle$
$\Uparrow^{(b:B)}_{(x)} [r]$
$\langle x, r, b \rangle$
$\bowtie \{\}$
$\langle a, b \rangle$
$\Uparrow^{(x)}_{(a:A)} [\_e1]$
$\langle a, \_e1, x \rangle$
$\bigcirc_{(a:A)}$
$\langle a \rangle$
$\bigcirc_{(b:B)}$
$\langle b \rangle$

2.25.20 Handling optional matches between optionally matched entities

Query specification

```
OPTIONAL MATCH (a:NotThere)
WITH a
MATCH (b:B)
WITH a, b
OPTIONAL MATCH (b)-[r:NOR_THIS]->(a)
RETURN a, b, r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.21 Handling optional matches between nulls

Query specification

```
OPTIONAL MATCH (a:NotThere)
OPTIONAL MATCH (b:NotThere)
WITH a, b
OPTIONAL MATCH (b)-[r:NOR_THIS]->(a)
RETURN a, b, r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.25.22 OPTIONAL MATCH and ‘collect()‘

Query specification

```
OPTIONAL MATCH (f:DoesExist)
OPTIONAL MATCH (n:DoesNotExist)
RETURN collect(DISTINCT n.property) AS a, collect(DISTINCT f.property) AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26 OrderByAcceptance

2.26.1 ORDER BY should return results in ascending order

Query specification

```
MATCH (n)
RETURN n.prop AS prop
ORDER BY n.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.2 ORDER BY DESC should return results in descending order

Query specification

```
MATCH (n)
RETURN n.prop AS prop
ORDER BY n.prop DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.3 ORDER BY of a column introduced in RETURN should return salient results in ascending order

Query specification

```
WITH [0, 1] AS prows, [[2], [3, 4]] AS qrows
UNWIND prows AS p
UNWIND qrows[p] AS q
WITH p, count(q) AS rng
RETURN p
ORDER BY rng
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.4 Renaming columns before ORDER BY should return results in ascending order

Query specification

```
MATCH (n)
RETURN n.prop AS n
ORDER BY n + 2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.5 Handle projections with ORDER BY - GH#4937

Query specification

```
MATCH (c:Crew {name: 'Neo'})
WITH c, 0 AS relevance
RETURN c.rank AS rank
ORDER BY relevance, c.rank
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.6 ORDER BY should order booleans in the expected order

Query specification

```
UNWIND [true, false] AS bools
RETURN bools
ORDER BY bools
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.7 ORDER BY DESC should order booleans in the expected order

Query specification

```
UNWIND [true, false] AS bools
RETURN bools
ORDER BY bools DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.8 ORDER BY should order strings in the expected order

Query specification

```
UNWIND ['.*', '', ' ', 'one'] AS strings
RETURN strings
ORDER BY strings
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.9 ORDER BY DESC should order strings in the expected order

Query specification

```
UNWIND ['.*', '', ' ', 'one'] AS strings
RETURN strings
ORDER BY strings DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.10 ORDER BY should order ints in the expected order

Query specification

```
UNWIND [1, 3, 2] AS ints
RETURN ints
ORDER BY ints
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.11 ORDER BY DESC should order ints in the expected order

Query specification

```
UNWIND [1, 3, 2] AS ints
RETURN ints
ORDER BY ints DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.12 ORDER BY should order floats in the expected order

Query specification

```
UNWIND [1.5, 1.3, 999.99] AS floats
RETURN floats
ORDER BY floats
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.13 ORDER BY DESC should order floats in the expected order

Query specification

```
UNWIND [1.5, 1.3, 999.99] AS floats
RETURN floats
ORDER BY floats DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.14 Handle ORDER BY with LIMIT 1

Query specification

```
MATCH (p:Person)
RETURN p.name AS name
ORDER BY p.name
LIMIT 1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.15 ORDER BY with LIMIT 0 should not generate errors

Query specification

```
MATCH (p:Person)
RETURN p.name AS name
ORDER BY p.name
LIMIT 0
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.16 ORDER BY with negative parameter for LIMIT should not generate errors

Query specification

```
MATCH (p:Person)
RETURN p.name AS name
ORDER BY p.name
LIMIT $limit
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.26.17 ORDER BY with a negative LIMIT should fail with a syntax exception

Query specification

```
MATCH (p:Person)
RETURN p.name AS name
ORDER BY p.name
LIMIT -1
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27 PatternComprehension

2.27.1 Pattern comprehension and ORDER BY

Query specification

```
MATCH (liker)
RETURN [p = (liker)--() | p] AS isNew
  ORDER BY liker.time
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.2 Returning a pattern comprehension

Query specification

```
MATCH (n)
RETURN [p = (n)-->() | p] AS ps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.3 Returning a pattern comprehension with label predicate

Query specification

```
MATCH (n:A)
RETURN [p = (n)-->(:B) | p]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.4 Returning a pattern comprehension with bound nodes

Query specification

```
MATCH (a:A), (b:B)
RETURN [p = (a)-[*]->(b) | p] AS paths
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.5 Using a pattern comprehension in a WITH

Query specification

```
MATCH (n)-->(b)
WITH [p = (n)-->() | p] AS ps, count(b) AS c
RETURN ps, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.6 Using a variable-length pattern comprehension in a WITH

Query specification

```
MATCH (a:A), (b:B)
WITH [p = (a)-[*]->(b) | p] AS paths, count(a) AS c
RETURN paths, c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.7 Using pattern comprehension in RETURN

Query specification

```
MATCH (n:A)
RETURN [p = (n)-[:HAS]->() | p] AS ps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.8 Aggregating on pattern comprehension

Query specification

```
MATCH (n:A)
RETURN count([p = (n)-[:HAS]->() | p]) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.9 Using pattern comprehension to test existence

Query specification

```
MATCH (n:X)
RETURN n, size([(n)--() | 1]) > 0 AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.10 Pattern comprehension inside list comprehension

Query specification

```
MATCH p = (n:X)-->(b)
RETURN n, [x IN nodes(p) | size([(x)-->(:Y) | 1])] AS list
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.11 Get node degree via size of pattern comprehension

Query specification

```
MATCH (a:X)
RETURN size([(a)-->() | 1]) AS length
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.12 Get node degree via size of pattern comprehension that specifies a relationship type

Query specification

```
MATCH (a:X)
RETURN size([(a)-[:T]->() | 1]) AS length
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.13 Get node degree via size of pattern comprehension that specifies multiple relationship types

Query specification

```
MATCH (a:X)
RETURN size([(a)-[:T|OTHER]->() | 1]) AS length
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.14 Introducing new node variable in pattern comprehension

Query specification

```
MATCH (n)
RETURN [(n)-[:T]->(b) | b.prop] AS list
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.27.15 Introducing new relationship variable in pattern comprehension

Query specification

```
MATCH (n)
RETURN [(n)-[r:T]->() | r.prop] AS list
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28 RemoveAcceptance

2.28.1 Should ignore nulls

Query specification

```
MATCH (n)
OPTIONAL MATCH (n)-[r]->()
REMOVE r.prop
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.2 Remove a single label

Query specification

```
MATCH (n)
REMOVE n:L
RETURN n.prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.3 Remove multiple labels

Query specification

```
MATCH (n)
REMOVE n:L1:L3
RETURN labels(n)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.4 Remove a single node property

Query specification

```
MATCH (n)
REMOVE n.prop
RETURN exists(n.prop) AS still_there
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.5 Remove multiple node properties

Query specification

```
MATCH (n)
REMOVE n.prop, n.a
RETURN size(keys(n)) AS props
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.6 Remove a single relationship property

Query specification

```
MATCH ()-[r]->()
REMOVE r.prop
RETURN exists(r.prop) AS still_there
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.7 Remove a single relationship property

Query specification

```
MATCH ()-[r]->()
REMOVE r.prop
RETURN exists(r.prop) AS still_there
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.8 Remove multiple relationship properties

Query specification

```
MATCH ()-[r]->()
REMOVE r.prop, r.a
RETURN size(keys(r)) AS props
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.28.9 Remove a missing property should be a valid operation

Query specification

```
MATCH (n)
REMOVE n.prop
RETURN sum(size(keys(n))) AS totalNumberOfProps
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29 ReturnAcceptanceTest

2.29.1 Allow addition

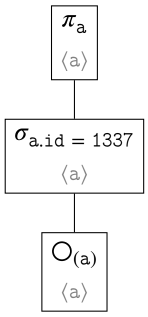
Query specification

```
MATCH (a)
WHERE a.id = 1337
RETURN a.version + 5
```

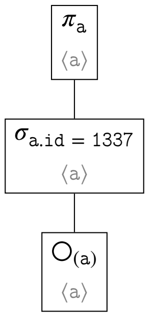
Relational algebra expression

$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.id}\,=\,1337}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.29.2 Limit to two hits

Query specification

```
MATCH (n)
RETURN n
LIMIT 2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.3 Start the result from the second row

Query specification

```
MATCH (n)
RETURN n
ORDER BY n.name ASC
SKIP 2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.4 Start the result from the second row by param

Query specification

```
MATCH (n)
RETURN n
ORDER BY n.name ASC
SKIP $skipAmount
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.5 Get rows in the middle

Query specification

```
MATCH (n)
RETURN n
ORDER BY n.name ASC
SKIP 2
LIMIT 2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.6 Get rows in the middle by param

Query specification

```
MATCH (n)
RETURN n
ORDER BY n.name ASC
SKIP $s
LIMIT $l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.7 Sort on aggregated function

Query specification

```
MATCH (n)
RETURN n.division, max(n.age)
  ORDER BY max(n.age)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.8 Support sort and distinct

Query specification

```
MATCH (a)
RETURN DISTINCT a
  ORDER BY a.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.9 Support column renaming

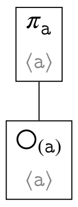
Query specification

```
MATCH (a)
RETURN a AS ColumnName
```

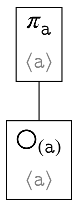
Relational algebra expression

$\pi_{\mathtt{a}} \left(\dot{\not\equiv} \left(\bigcirc_{(\mathtt{a})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.29.10 Support ordering by a property after being distinct-ified

Query specification

```
MATCH (a)-->(b)
RETURN DISTINCT b
  ORDER BY b.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.11 Arithmetic precedence test

Query specification

```
RETURN 12 / 4 * 3 - 2 * 4
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.12 Arithmetic precedence with parenthesis test

Query specification

```
RETURN 12 / 4 * (3 - 2 * 4)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.13 Count star should count everything in scope

Query specification

```
MATCH (a)
RETURN a, count(*)
ORDER BY count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.14 Absolute function

Query specification

```
RETURN abs(-1)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.29.15 Return collection size

Query specification

```
RETURN size([1, 2, 3]) AS n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30 ReturnAcceptance2

2.30.1 Accept valid Unicode literal

Query specification

```
RETURN '\u01FF' AS a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.2 LIMIT 0 should return an empty result

Query specification

```
MATCH (n)
RETURN n
  LIMIT 0
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.3 Ordering with aggregation

Query specification

```
MATCH (n)
RETURN n.name, count(*) AS foo
  ORDER BY n.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.4 DISTINCT on nullable values

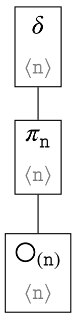
Query specification

```
MATCH (n)
RETURN DISTINCT n.name
```

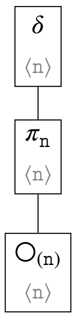
Relational algebra expression

$\delta\left(\pi_{\mathtt{n}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.30.5 Return all variables

Query specification

```
MATCH p = (a:Start)-->(b)
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.6 Setting and returning the size of a list property

Query specification

```
MATCH (n)
SET n.x = [1, 2, 3]
RETURN size(n.x)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.7 Setting and returning the size of a list property

Query specification

```
MATCH (n)
SET n.x = [1, 2, 3]
RETURN size(n.x)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.8 ‘sqrt()‘ returning float values

Query specification

```
RETURN sqrt(12.96)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.9 Arithmetic expressions inside aggregation

Query specification

```
MATCH (me)-[r1:ATE]->()<-[r2:ATE]-(you)
WHERE me.name = 'Michael'
WITH me, count(DISTINCT r1) AS H1, count(DISTINCT r2) AS H2, you
MATCH (me)-[r1:ATE]->()<-[r2:ATE]-(you)
RETURN me, you, sum((1 - abs(r1.times / H1 - r2.times / H2)) * (r1.times + r2.times) / (H1 + H2)) AS
    sum
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.10 Matching and disregarding output, then matching again

Query specification

```
MATCH ()-->()
WITH 1 AS x
MATCH ()-[r1]->()<--()
RETURN sum(r1.times)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.11 Returning a list property

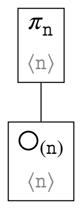
Query specification

```
MATCH (n)
RETURN n
```

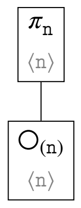
Relational algebra expression

$\pi_{\mathtt{n}}\left(\not\doteqdot\left(\bigcirc_{(\mathtt{n})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.30.12 Returning a projected map

Query specification

```
RETURN {a: 1, b: 'foo'}
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.13 Returning an expression

Query specification

```
MATCH (a)
RETURN exists(a.id), a IS NOT NULL
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.14 Concatenating and returning the size of literal lists

Query specification

```
RETURN size([[], []] + [[]]) AS l
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.15 Concatenating and returning the size of literal lists

Query specification

```
MATCH (n)
SET n.array = [1, 2, 3, 4, 5]
RETURN tail(tail(n.array))
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.16 Limiting amount of rows when there are fewer left than the LIMIT argument

Query specification

```
MATCH (a)
RETURN a.count
  ORDER BY a.count
  SKIP 10
  LIMIT 10
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.17 ‘substring()‘ with default second argument

Query specification

```
RETURN substring('0123456789', 1) AS s
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.18 Returning all variables with ordering

Query specification

```
MATCH (n)
RETURN *
  ORDER BY n.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.19 Using aliased DISTINCT expression in ORDER BY

Query specification

```
MATCH (n)
RETURN DISTINCT n.id AS id
  ORDER BY id DESC
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.20 Returned columns do not change from using ORDER BY

Query specification

```
MATCH (n)
RETURN DISTINCT n
  ORDER BY n.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.21 Arithmetic expressions should propagate null values

Query specification

```
RETURN 1 + (2 - (3 * (4 / (5 ^ (6 % null))))) AS a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.22 Indexing into nested literal lists

Query specification

```
RETURN [[1]][0][0]
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.23 Aliasing expressions

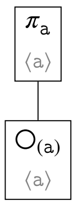
Query specification

```
MATCH (a)
RETURN a.id AS a, a.id
```

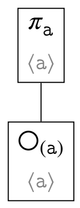
Relational algebra expression

$\pi_{\mathtt{a}}\left(\not\fallingdotseq\left(\bigcirc_{(\mathtt{a})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.30.24 Projecting an arithmetic expression with aggregation

Query specification

```
MATCH (a)
RETURN a, count(a) + 3
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.25 Multiple aliasing and backreferencing

Query specification

```
CREATE (m {id: 0})
WITH {first: m.id} AS m
WITH {second: m.first} AS m
RETURN m.second
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.26 Aggregating by a list property has a correct definition of equality

Query specification

```
MATCH (a)
WITH a.a AS a, count(*) AS count
RETURN count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.27 Reusing variable names

Query specification

```
MATCH (person:Person)<--(message)<-[like]-(:Person)
WITH like.creationDate AS likeTime, person AS person
  ORDER BY likeTime, message.id
WITH head(collect({likeTime: likeTime})) AS latestLike, person AS person
RETURN latestLike.likeTime AS likeTime
  ORDER BY likeTime
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.28 Concatenating lists of same type

Query specification

```
RETURN [1, 10, 100] + [4, 5] AS foo
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.29 Appending lists of same type

Query specification

```
RETURN [false, true] + false AS foo
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.30 DISTINCT inside aggregation should work with lists in maps

Query specification

```
MATCH (n)
RETURN count(DISTINCT {foo: n.list}) AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.31 Handling DISTINCT with lists in maps

Query specification

```
MATCH (n)
WITH DISTINCT {foo: n.list} AS map
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.32 DISTINCT inside aggregation should work with nested lists in maps

Query specification

```
MATCH (n)
RETURN count(DISTINCT {foo: [[n.list, n.list], [n.list, n.list]]}) AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.30.33 DISTINCT inside aggregation should work with nested lists of maps in maps

Query specification

```
MATCH (n)
RETURN count(DISTINCT {foo: [{bar: n.list}, {baz: {apa: n.list}}]}) AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.31 SemanticErrorAcceptance

2.31.1 Handling property access on the Any type

Query specification

```
WITH [{prop: 0}, 1] AS list
RETURN (list[0]).prop
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.31.2 Bad arguments for ‘range()‘

Query specification

```
RETURN range(2, 8, 0)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32 SetAcceptance

2.32.1 Setting a node property to null removes the existing property

Query specification

```
MATCH (n:A)
SET n.property1 = null
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.2 Setting a relationship property to null removes the existing property

Query specification

```
MATCH ()-[r]->()
SET r.property1 = null
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.3 Set a property

Query specification

```
MATCH (n:A)
WHERE n.name = 'Andres'
SET n.name = 'Michael'
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.4 Set a property to an expression

Query specification

```
MATCH (n:A)
WHERE n.name = 'Andres'
SET n.name = n.name + ' was here'
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.5 Set a property by selecting the node using a simple expression

Query specification

```
MATCH (n:A)
SET (n).name = 'neo4j'
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.6 Set a property by selecting the relationship using a simple expression

Query specification

```
MATCH ()-[r:REL]->()
SET (r).name = 'neo4j'
RETURN r
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.7 Setting a property to null removes the property

Query specification

```
MATCH (n)
WHERE n.name = 'Michael'
SET n.name = null
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.8 Add a label to a node

Query specification

```
MATCH (n:A)
SET n:Foo
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.9 Adding a list property

Query specification

```
MATCH (n:A)
SET n.x = [1, 2, 3]
RETURN [i IN n.x | i / 2.0] AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.10 Concatenate elements onto a list property

Query specification

```
CREATE (a {foo: [1, 2, 3]})
SET a.foo = a.foo + [4, 5]
RETURN a.foo
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.11 Concatenate elements in reverse onto a list property

Query specification

```
CREATE (a {foo: [3, 4, 5]})
SET a.foo = [1, 2] + a.foo
RETURN a.foo
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.12 Overwrite values when using +=

Query specification

```
MATCH (n:X {foo: ’A’})
SET n += {bar: ’C’}
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.13 Retain old values when using +=

Query specification

```
MATCH (n:X {foo: 'A'})
SET n += {bar: 'B'}
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.14 Explicit null values in a map remove old values

Query specification

```
MATCH (n:X {foo: 'A'})
SET n += {foo: null}
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.32.15 Non-existent values in a property map are removed with SET =

Query specification

```
MATCH (n:X {foo: 'A'})
SET n = {foo: 'B', baz: 'C'}
RETURN n
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.33 SkipLimitAcceptanceTest

2.33.1 SKIP with an expression that depends on variables should fail

Query specification

```
MATCH (n) RETURN n SKIP n.count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.33.2 LIMIT with an expression that depends on variables should fail

Query specification

```
MATCH (n) RETURN n LIMIT n.count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.33.3 SKIP with an expression that does not depend on variables

Query specification

```
MATCH (n)
WITH n SKIP toInteger(rand()*9)
WITH count(*) AS count
RETURN count > 0 AS nonEmpty
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.33.4 LIMIT with an expression that does not depend on variables

Query specification

```
MATCH (n)
WITH n LIMIT toInteger(ceil(1.7))
RETURN count(*) AS count
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.34 StartingPointAcceptance

2.34.1 Find all nodes

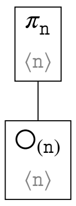
Query specification

```
MATCH (n)
RETURN n
```

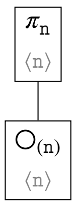
Relational algebra expression

$\pi_{\mathtt{n}} \left(\not\fallingdotseq \left(\bigcirc_{(\mathtt{n})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.34.2 Find labelled nodes

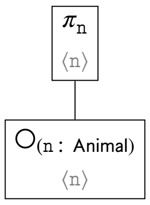
Query specification

```
MATCH (n:Animal)
RETURN n
```

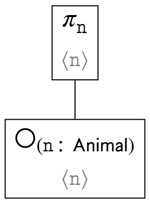
Relational algebra expression

$\pi_{\mathtt{n}} \left(\not\fallingdotseq \left(\bigcirc_{(\mathtt{n}\,:\,\mathsf{Animal})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.34.3 Find nodes by property

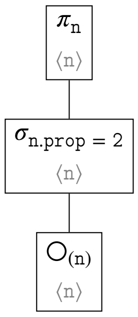
Query specification

```
MATCH (n)
WHERE n.prop = 2
RETURN n
```

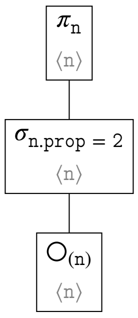
Relational algebra expression

$\pi_{\mathtt{n}}\left(\sigma_{\mathtt{n.prop}\,=\,2}\left(\not\equiv\left(\bigcirc_{(\mathtt{n})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35 StartsWithAcceptance

2.35.1 Finding exact matches

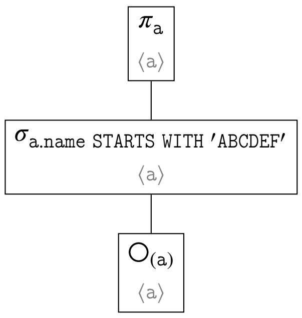
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH 'ABCDEF'
RETURN a
```

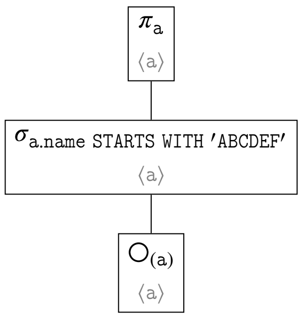
Relational algebra expression

$\pi_{\text{a}}\left(\sigma_{\text{a.name STARTS WITH 'ABCDEF'}}\left(\not\doteq\left(\bigcirc_{(\text{a})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.2 Finding beginning of string

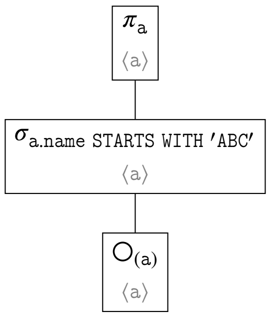
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH 'ABC'
RETURN a
```

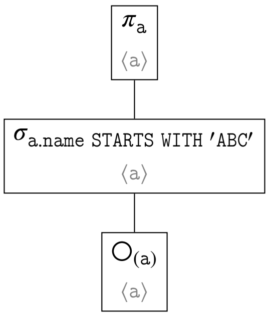
Relational algebra expression

$\pi_{\text{a}}\left(\sigma_{\text{a.name STARTS WITH 'ABC'}}\left(\not\doteq\left(\bigcirc_{(\text{a})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.3 Finding end of string 1

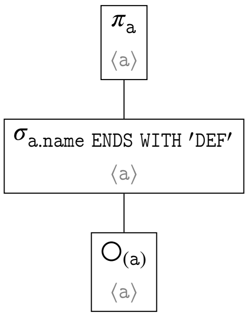
Query specification

```
MATCH (a)
WHERE a.name ENDS WITH 'DEF'
RETURN a
```

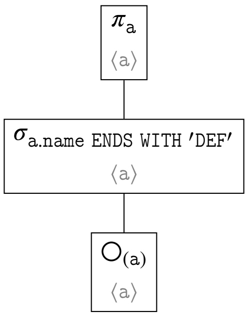
Relational algebra expression

$\pi_{a}\left(\sigma_{a.name\ \mathtt{ENDS\ WITH}\ 'DEF'}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.4 Finding end of string 2

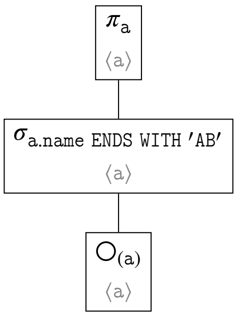
Query specification

```
MATCH (a)
WHERE a.name ENDS WITH 'AB'
RETURN a
```

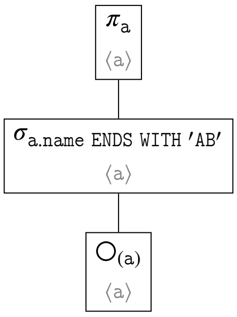
Relational algebra expression

$\pi_{a}\left(\sigma_{a.name\ \mathtt{ENDS\ WITH}\ 'AB'}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.5 Finding middle of string

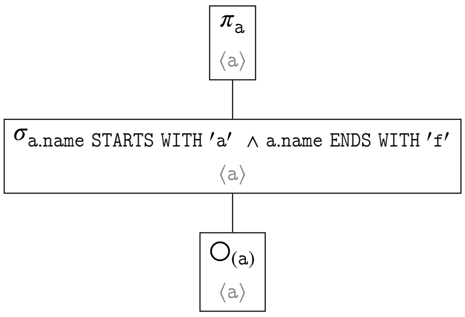
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH 'a'
  AND a.name ENDS WITH 'f'
RETURN a
```

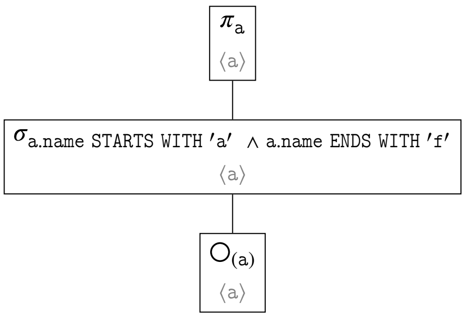
Relational algebra expression

$$\pi_{\text{a}}\left(\sigma_{\text{a.name STARTS WITH 'a'} \wedge \text{a.name ENDS WITH 'f'}}\left(\not\doteqdot\left(\bigcirc_{\text{(a)}}\right)\right)\right)$$

Relational algebra tree


π_{a}
⟨a⟩
$\sigma_{\text{a.name STARTS WITH 'a'} \wedge \text{a.name ENDS WITH 'f'}}$
⟨a⟩
$\bigcirc_{\text{(a)}}$
⟨a⟩

Relational algebra tree for incremental queries


π_{a}
⟨a⟩
$\sigma_{\text{a.name STARTS WITH 'a'} \wedge \text{a.name ENDS WITH 'f'}}$
⟨a⟩
$\bigcirc_{(\text{a})}$
⟨a⟩

2.35.6 Finding the empty string

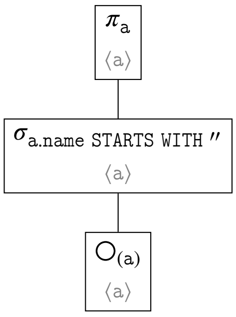
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH ''
RETURN a
```

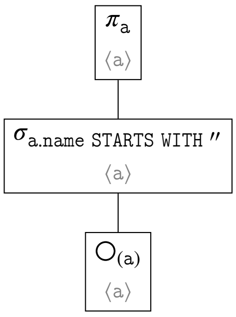
Relational algebra expression

$$\pi_{\text{a}}\left(\sigma_{\text{a.name STARTS WITH ''}}\left(\not\doteq\left(\bigcirc_{(\text{a})}\right)\right)\right)$$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.7 Finding when the middle is known

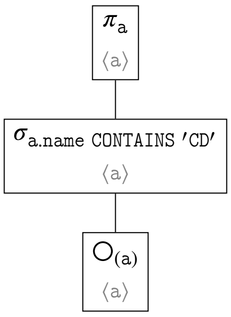
Query specification

```
MATCH (a)
WHERE a.name CONTAINS 'CD'
RETURN a
```

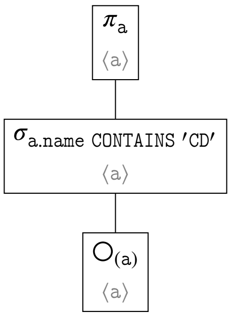
Relational algebra expression

$$\pi_{\text{a}}\left(\sigma_{\text{a.name CONTAINS 'CD'}}\left(\not\doteq\left(\bigcirc_{(\text{a})}\right)\right)\right)$$

Relational algebra tree


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ CONTAINS\ 'CD'}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ CONTAINS\ 'CD'}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

2.35.8 Finding strings starting with whitespace

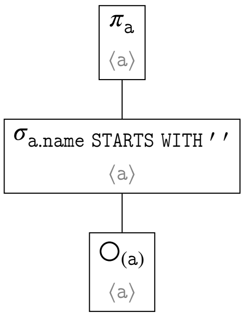
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH ' '
RETURN a.name AS name
```

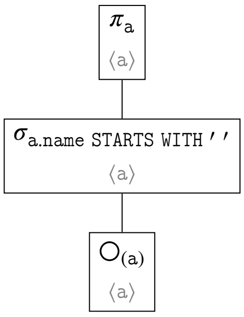
Relational algebra expression

$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.name\ STARTS\ WITH\ ''}}\left(\not\doteq\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ ''}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ ''}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

2.35.9 Finding strings starting with newline

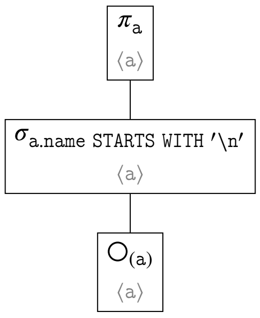
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH '\n'
RETURN a.name AS name
```

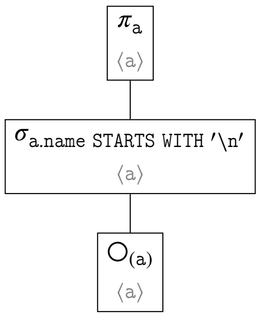
Relational algebra expression

$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.name\ STARTS\ WITH\ '\backslash n'}}\left(\not\equiv\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ '\backslash n'}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ '\backslash n'}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

2.35.10 Finding strings ending with newline

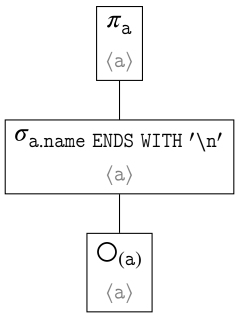
Query specification

```
MATCH (a)
WHERE a.name ENDS WITH '\n'
RETURN a.name AS name
```

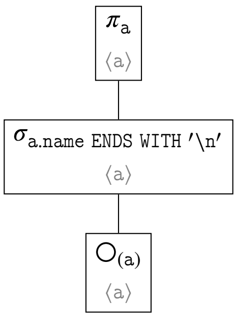
Relational algebra expression

$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.name\ ENDS\ WITH\ '\backslash n'}}\left(\not\equiv\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$

Relational algebra tree


$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ ENDS\ WITH\ '\backslash n'}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries



π_{a}
⟨a⟩
$\sigma_{\mathrm{a.name\ ENDS\ WITH\ '\backslash n'}}$
⟨a⟩
$\bigcirc_{(\mathrm{a})}$
⟨a⟩

2.35.11 Finding strings ending with whitespace

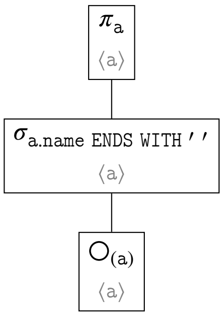
Query specification

```
MATCH (a)
WHERE a.name ENDS WITH ' '
RETURN a.name AS name
```

Relational algebra expression

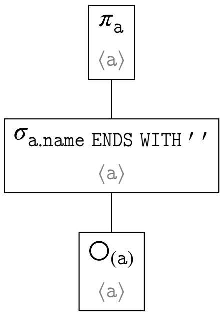
$\pi_{\mathrm{a}} \left(\sigma_{\mathrm{a.name\ ENDS\ WITH\ ''}} \left(\risingdotseq \left(\bigcirc_{(\mathrm{a})}\right)\right)\right)$

Relational algebra tree



π_{a}
⟨a⟩
$\sigma_{\mathrm{a.name\ ENDS\ WITH\ ''}}$
⟨a⟩
$\bigcirc_{(\mathrm{a})}$
⟨a⟩

Relational algebra tree for incremental queries



π_{a}
⟨a⟩
$\sigma_{\mathrm{a.name\ ENDS\ WITH\ ''}}$
⟨a⟩
$\bigcirc_{(\mathrm{a})}$
⟨a⟩

2.35.12 Finding strings containing whitespace

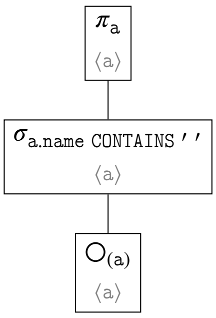
Query specification

```
MATCH (a)
WHERE a.name CONTAINS ' '
RETURN a.name AS name
```

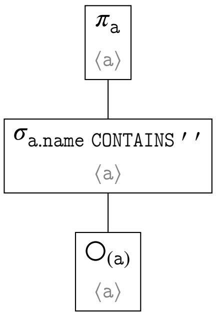
Relational algebra expression

$\pi_{\mathrm{a}} \left(\sigma_{\mathrm{a.name\ CONTAINS\ ''}} \left(\risingdotseq \left(\bigcirc_{(\mathrm{a})}\right)\right)\right)$

Relational algebra tree


$\pi_{\texttt{a}}$
⟨a⟩
$\sigma_{\texttt{a.name CONTAINS ''}}$
⟨a⟩
$\bigcirc_{(\texttt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\texttt{a}}$
⟨a⟩
$\sigma_{\texttt{a.name CONTAINS ''}}$
⟨a⟩
$\bigcirc_{(\texttt{a})}$
⟨a⟩

2.35.13 Finding strings containing newline

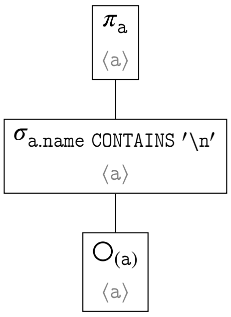
Query specification

```
MATCH (a)
WHERE a.name CONTAINS '\n'
RETURN a.name AS name
```

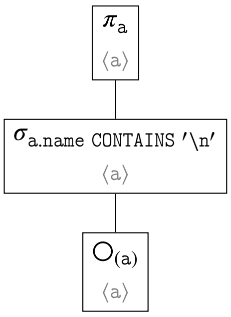
Relational algebra expression

$\pi_{\texttt{a}}\left(\sigma_{\texttt{a.name CONTAINS '\textbackslash n'}}\left(\not\doteq\left(\bigcirc_{(\texttt{a})}\right)\right)\right)$

Relational algebra tree


$\pi_{\texttt{a}}$
⟨a⟩
$\sigma_{\texttt{a.name CONTAINS '\textbackslash n'}}$
⟨a⟩
$\bigcirc_{(\texttt{a})}$
⟨a⟩

Relational algebra tree for incremental queries


$\pi_{\texttt{a}}$
⟨a⟩
$\sigma_{\texttt{a.name CONTAINS '\textbackslash n'}}$
⟨a⟩
$\bigcirc_{(\texttt{a})}$
⟨a⟩

2.35.14 No string starts with null

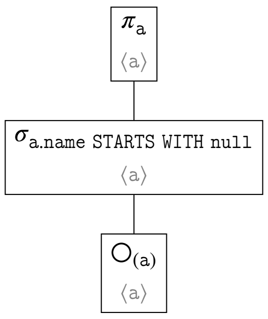
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH null
RETURN a
```

Relational algebra expression

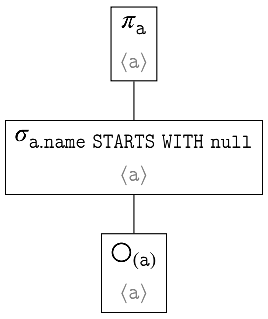
$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.name}\ \mathtt{STARTS\ WITH\ null}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathtt{a})}\right)\right)\right)$

Relational algebra tree



$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ null}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

Relational algebra tree for incremental queries



$\pi_{\mathtt{a}}$
⟨a⟩
$\sigma_{\mathtt{a.name\ STARTS\ WITH\ null}}$
⟨a⟩
$\bigcirc_{(\mathtt{a})}$
⟨a⟩

2.35.15 No string does not start with null

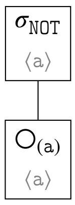
Query specification

```
MATCH (a)
WHERE NOT a.name STARTS WITH null
RETURN a
```

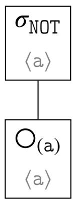
Relational algebra expression

$\sigma_{\mathtt{NOT}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathtt{a})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.16 No string ends with null

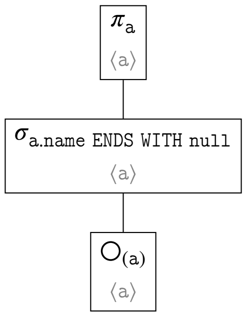
Query specification

```
MATCH (a)
WHERE a.name ENDS WITH null
RETURN a
```

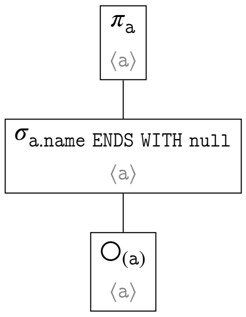
Relational algebra expression

$\pi_{\mathtt{a}}\left(\sigma_{\mathtt{a.name\ ENDS\ WITH\ null}}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.17 No string does not end with null

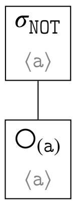
Query specification

```
MATCH (a)
WHERE NOT a.name ENDS WITH null
RETURN a
```

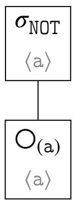
Relational algebra expression

$\sigma_{\mathtt{NOT}}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.18 No string contains null

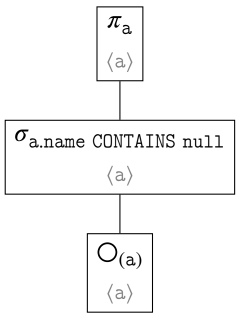
Query specification

```
MATCH (a)
WHERE a.name CONTAINS null
RETURN a
```

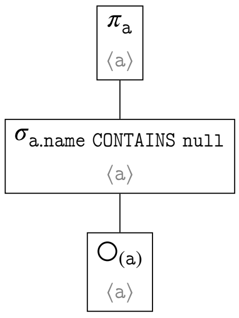
Relational algebra expression

$\pi_{\mathrm{a}}\left(\sigma_{\mathrm{a.name}\ \texttt{CONTAINS null}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathrm{a})}\right)\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.19 No string does not contain null

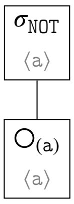
Query specification

```
MATCH (a)
WHERE NOT a.name CONTAINS null
RETURN a
```

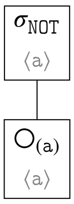
Relational algebra expression

$\sigma_{\texttt{NOT}}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.35.20 Combining string operators

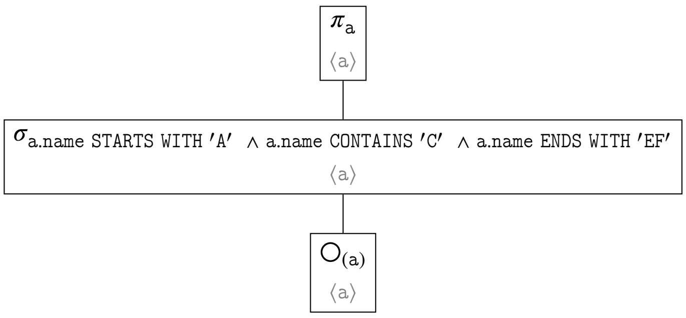
Query specification

```
MATCH (a)
WHERE a.name STARTS WITH 'A'
  AND a.name CONTAINS 'C'
  AND a.name ENDS WITH 'EF'
RETURN a
```

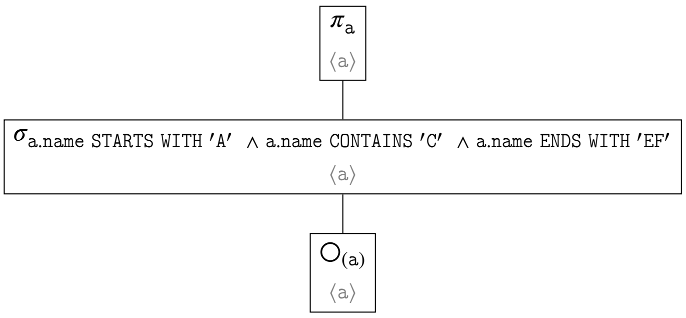
Relational algebra expression

$\pi_{a}\left(\sigma_{a.name\ \texttt{STARTS WITH}\ '\texttt{A}'\ \wedge\ a.name\ \texttt{CONTAINS}\ '\texttt{C}'\ \wedge\ a.name\ \texttt{ENDS WITH}\ '\texttt{EF}'}\left(\not\equiv\left(\bigcirc_{(a)}\right)\right)\right)$

Relational algebra tree


πa
⟨a⟩
σa.name STARTS WITH 'A' ∧ a.name CONTAINS 'C' ∧ a.name ENDS WITH 'EF'
⟨a⟩
○(a)
⟨a⟩

Relational algebra tree for incremental queries


π a
⟨a⟩
σ a.name STARTS WITH 'A' ∧ a.name CONTAINS 'C' ∧ a.name ENDS WITH 'EF'
⟨a⟩
○(a)
⟨a⟩

2.35.21 NOT with CONTAINS

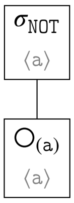
Query specification

```
MATCH (a)
WHERE NOT a.name CONTAINS 'b'
RETURN a
```

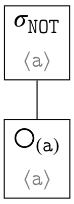
Relational algebra expression

$\sigma_{\texttt{NOT}} \left(\not\doteq \left(\bigcirc_{(a)}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.36 SyntaxErrorAcceptance

2.36.1 Using a non-existent function

Query specification

```
MATCH (a)
RETURN foo(a)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.36.2 Using ‘rand()‘ in aggregations

Query specification

```
RETURN count(rand())
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.36.3 Supplying invalid hexadecimal literal 1

Query specification

```
RETURN 0x23G34
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.36.4 Supplying invalid hexadecimal literal 2

Query specification

```
RETURN 0x23j
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.37 TernaryLogicAcceptanceTest

2.37.1 The inverse of a null is a null

Query specification

```
RETURN NOT null AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.37.2 A literal null IS null

Query specification

```
RETURN null IS NULL AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.37.3 A literal null is not IS NOT null

Query specification

```
RETURN null IS NOT NULL AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.37.4 It is unknown - i.e. null - if a null is equal to a null

Query specification

```
RETURN null = null AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.37.5 It is unknown - i.e. null - if a null is not equal to a null

Query specification

```
RETURN null <> null AS value
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38 TriadicSelection

2.38.1 Handling triadic friend of a friend

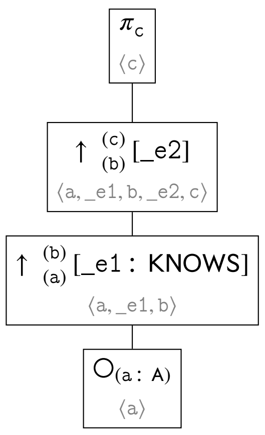
Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
RETURN c.name
```

Relational algebra expression

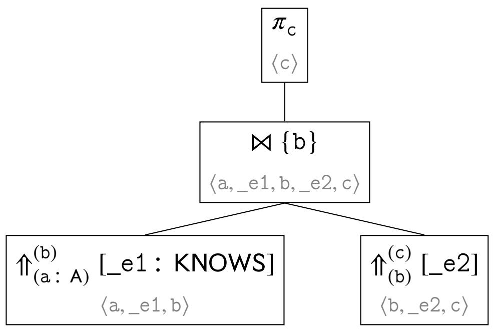
$$\pi_{\mathsf{c}}\left(\not\equiv\left(\uparrow\,{}^{(\mathsf{c})}_{(\mathsf{b})}[\_\mathsf{e2}]\left(\uparrow\,{}^{(\mathsf{b})}_{(\mathsf{a})}[\_\mathsf{e1}:\,\mathsf{KNOWS}]\left(\bigcirc_{(\mathsf{a}:\,\mathsf{A})}\right)\right)\right)\right)$$

Relational algebra tree



π_{c}
⟨c⟩
$\uparrow\,{}^{(\mathsf{c})}_{(\mathsf{b})}[\_\mathsf{e2}]$
⟨a, _e1, b, _e2, c⟩
$\uparrow\,{}^{(\mathsf{b})}_{(\mathsf{a})}[\_\mathsf{e1}:\,\mathsf{KNOWS}]$
⟨a, _e1, b⟩
$\bigcirc_{(\mathsf{a}:\,\mathsf{A})}$
⟨a⟩

Relational algebra tree for incremental queries



2.38.2 Handling triadic friend of a friend that is not a friend

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.3 Handling triadic friend of a friend that is not a friend with different relationship type

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:FOLLOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.4 Handling triadic friend of a friend that is not a friend with superset of relationship type

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.5 Handling triadic friend of a friend that is not a friend with implicit subset of relationship type

Query specification

```
MATCH (a:A)-->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.6 Handling triadic friend of a friend that is not a friend with explicit subset of relationship type

Query specification

```
MATCH (a:A)-[:KNOWS|FOLLOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.7 Handling triadic friend of a friend that is not a friend with same labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c:X)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.8 Handling triadic friend of a friend that is not a friend with different labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c:Y)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.9 Handling triadic friend of a friend that is not a friend with implicit subset of labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c:X)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.10 Handling triadic friend of a friend that is not a friend with implicit superset of labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.11 Handling triadic friend of a friend that is a friend

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.12 Handling triadic friend of a friend that is a friend with different relationship type

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:FOLLOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.13 Handling triadic friend of a friend that is a friend with superset of relationship type

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.14 Handling triadic friend of a friend that is a friend with implicit subset of relationship type

Query specification

```
MATCH (a:A)-->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.15 Handling triadic friend of a friend that is a friend with explicit subset of relationship type

Query specification

```
MATCH (a:A)-[:KNOWS|FOLLOWS]->(b)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.16 Handling triadic friend of a friend that is a friend with same labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c:X)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.17 Handling triadic friend of a friend that is a friend with different labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c:Y)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.18 Handling triadic friend of a friend that is a friend with implicit subset of labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b)-->(c:X)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.38.19 Handling triadic friend of a friend that is a friend with implicit superset of labels

Query specification

```
MATCH (a:A)-[:KNOWS]->(b:X)-->(c)
OPTIONAL MATCH (a)-[r:KNOWS]->(c)
WITH c WHERE r IS NOT NULL
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39 TypeConversionFunctions

2.39.1 ‘toBoolean()‘ on valid literal string

Query specification

```
RETURN toBoolean('true') AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.2 ‘toBoolean()‘ on booleans

Query specification

```
UNWIND [true, false] AS b
RETURN toBoolean(b) AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.3 ‘toBoolean()‘ on variables with valid string values

Query specification

```
UNWIND ['true', 'false'] AS s
RETURN toBoolean(s) AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.4 ‘toBoolean()‘ on invalid strings

Query specification

```
UNWIND [null, '', ' tru ', 'f alse'] AS things
RETURN toBoolean(things) AS b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.5 ‘toInteger()‘

Query specification

```
MATCH (p:Person { age: '42' })
WITH *
MATCH (n)
RETURN toInteger(n.age) AS age
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.6 ‘toInteger()‘ on float

Query specification

```
WITH 82.9 AS weight
RETURN toInteger(weight)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.7 ‘toInteger()‘ returning null on non-numerical string

Query specification

```
WITH 'foo' AS foo_string, '' AS empty_string
RETURN toInteger(foo_string) AS foo, toInteger(empty_string) AS empty
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.8 ‘toInteger()‘ handling mixed number types

Query specification

```
WITH [2, 2.9] AS numbers
RETURN [n IN numbers | toInteger(n)] AS int_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.9 ‘toInteger()‘ handling Any type

Query specification

```
WITH [2, 2.9, '1.7'] AS things
RETURN [n IN things | toInteger(n)] AS int_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.10 ‘toInteger()‘ on a list of strings

Query specification

```
WITH ['2', '2.9', 'foo'] AS numbers
RETURN [n IN numbers | toInteger(n)] AS int_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.11 ‘toFloat()‘

Query specification

```
MATCH (m:Movie { rating: 4 })
WITH *
MATCH (n)
RETURN toFloat(n.rating) AS float
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.12 ‘toFloat()‘ on mixed number types

Query specification

```
WITH [3.4, 3] AS numbers
RETURN [n IN numbers | toFloat(n)] AS float_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.13 ‘toFloat()‘ returning null on non-numerical string

Query specification

```
WITH 'foo' AS foo_string, '' AS empty_string
RETURN toFloat(foo_string) AS foo, toFloat(empty_string) AS empty
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.14 ‘toFloat()‘ handling Any type

Query specification

```
WITH [3.4, 3, '5'] AS numbers
RETURN [n IN numbers | toFloat(n)] AS float_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.15 ‘toFloat()‘ on a list of strings

Query specification

```
WITH ['1', '2', 'foo'] AS numbers
RETURN [n IN numbers | toFloat(n)] AS float_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.16 ‘toString()‘

Query specification

```
MATCH (m:Movie { rating: 4 })
WITH *
MATCH (n)
RETURN toString(n.rating)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.17 ‘toString()‘ handling boolean properties

Query specification

```
MATCH (m:Movie)
RETURN toString(m.watched)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.18 ‘toString()‘ handling inlined boolean

Query specification

```
RETURN toString(1 < 0) AS bool
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.19 ‘toString()‘ handling boolean literal

Query specification

```
RETURN toString(true) AS bool
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.20 ‘toString()‘ should work on Any type

Query specification

```
RETURN [x IN [1, 2.3, true, 'apa'] | toString(x) ] AS list
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.39.21 ‘toString()‘ on a list of integers

Query specification

```
WITH [1, 2, 3] AS numbers
RETURN [n IN numbers | toString(n)] AS string_numbers
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.40 UnionAcceptance

2.40.1 Should be able to create text output from union queries

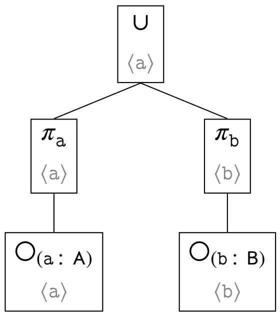
Query specification

```
MATCH (a:A)
RETURN a AS a
UNION
MATCH (b:B)
RETURN b AS a
```

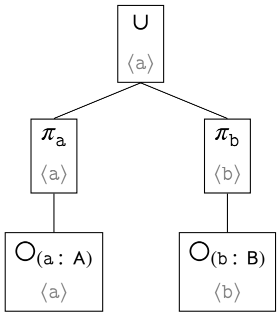
Relational algebra expression

$\pi_{\mathrm{a}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathrm{a}:\ \mathrm{A})}\right)\right) \cup \pi_{\mathrm{b}}\left(\dot{\not\equiv}\left(\bigcirc_{(\mathrm{b}:\ \mathrm{B})}\right)\right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.40.2 Two elements, both unique, not distinct

Query specification

```
RETURN 1 AS x
UNION ALL
RETURN 2 AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.40.3 Two elements, both unique, distinct

Query specification

```
RETURN 1 AS x
UNION
RETURN 2 AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.40.4 Three elements, two unique, distinct

Query specification

```
RETURN 2 AS x
UNION
RETURN 1 AS x
UNION
RETURN 2 AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.40.5 Three elements, two unique, not distinct

Query specification

```
RETURN 2 AS x
UNION ALL
RETURN 1 AS x
UNION ALL
RETURN 2 AS x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41 UnwindAcceptance

2.41.1 Unwinding a list

Query specification

```
UNWIND [1, 2, 3] AS x
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.2 Unwinding a range

Query specification

```
UNWIND range(1, 3) AS x
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.3 Unwinding a concatenation of lists

Query specification

```
WITH [1, 2, 3] AS first, [4, 5, 6] AS second
UNWIND (first + second) AS x
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.4 Unwinding a collected unwound expression

Query specification

```
UNWIND RANGE(1, 2) AS row
WITH collect(row) AS rows
UNWIND rows AS x
RETURN x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.5 Unwinding a collected expression

Query specification

```
MATCH (row)
WITH collect(row) AS rows
UNWIND rows AS node
RETURN node.id
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.6 Creating nodes from an unwound parameter list

Query specification

```
UNWIND $events AS event
MATCH (y:Year {year: event.year})
MERGE (e:Event {id: event.id})
MERGE (y)<-[:IN]-(e)
RETURN e.id AS x
ORDER BY x
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.7 Double unwinding a list of lists

Query specification

```
WITH [[1, 2, 3], [4, 5, 6]] AS lol
UNWIND lol AS x
UNWIND x AS y
RETURN y
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.8 Unwinding the empty list

Query specification

```
UNWIND [] AS empty
RETURN empty
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.9 Unwinding null

Query specification

```
UNWIND null AS nil
RETURN nil
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.10 Unwinding list with duplicates

Query specification

```
UNWIND [1, 1, 2, 2, 3, 3, 4, 4, 5, 5] AS duplicate
RETURN duplicate
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.11 Unwind does not prune context

Query specification

```
WITH [1, 2, 3] AS list
UNWIND list AS x
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.12 Unwind does not remove variables from scope

Query specification

```
MATCH (a:S)-[:X]->(b1)
WITH a, collect(b1) AS bees
UNWIND bees AS b2
MATCH (a)-[:Y]->(b2)
RETURN a, b2
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.13 Multiple unwinds after each other

Query specification

```
WITH [1, 2] AS xs, [3, 4] AS ys, [5, 6] AS zs
UNWIND xs AS x
UNWIND ys AS y
UNWIND zs AS z
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.41.14 Unwind with merge

Query specification

```
UNWIND $props AS prop
MERGE (p:Person {login: prop.login})
SET p.name = prop.name
RETURN p.name, p.login
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42 VarLengthAcceptance

2.42.1 Handling unbounded variable length match

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.2 Handling explicitly unbounded variable length match

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*..]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.3 Handling single bounded variable length match 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*0]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.4 Handling single bounded variable length match 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.5 Handling single bounded variable length match 3

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.6 Handling upper and lower bounded variable length match 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*0..2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.7 Handling upper and lower bounded variable length match 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1..2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.8 Handling symmetrically bounded variable length match, bounds are zero

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*0..0]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.9 Handling symmetrically bounded variable length match, bounds are one

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1..1]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.10 Handling symmetrically bounded variable length match, bounds are two

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*2..2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.11 Handling upper and lower bounded variable length match, empty interval 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*2..1]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.12 Handling upper and lower bounded variable length match, empty interval 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1..0]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.13 Handling upper bounded variable length match, empty interval

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*..0]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.14 Handling upper bounded variable length match 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*..1]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.15 Handling upper bounded variable length match 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*..2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.16 Handling lower bounded variable length match 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*0..]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.17 Handling lower bounded variable length match 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1..]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.18 Handling lower bounded variable length match 3

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*2..]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.19 Handling a variable length relationship and a standard relationship in chain, zero length 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*0]->()-[:LIKES]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.20 Handling a variable length relationship and a standard relationship in chain, zero length 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES]->()-[:LIKES*0]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.21 Handling a variable length relationship and a standard relationship in chain, single length 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*1]->()-[:LIKES]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.22 Handling a variable length relationship and a standard relationship in chain, single length 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES]->()-[:LIKES*1]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.23 Handling a variable length relationship and a standard relationship in chain, longer 1

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES*2]->()-[:LIKES]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.24 Handling a variable length relationship and a standard relationship in chain, longer 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES]->()-[:LIKES*2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.25 Handling a variable length relationship and a standard relationship in chain, longer 3

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES]->()-[:LIKES*3]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.26 Handling mixed relationship patterns and directions 1

Query specification

```
MATCH (a:A)
MATCH (a)<-[:LIKES]-()-[:LIKES*3]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.27 Handling mixed relationship patterns and directions 2

Query specification

```
MATCH (a:A)
MATCH (a)-[:LIKES]->()<-[:LIKES*3]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.28 Handling mixed relationship patterns 1

Query specification

```
MATCH (a:A)
MATCH (p)-[:LIKES*1]->()-[:LIKES]->()-[r:LIKES*2]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.42.29 Handling mixed relationship patterns 2

Query specification

```
MATCH (a:A)
MATCH (p)-[:LIKES]->()-[:LIKES*2]->()-[r:LIKES]->(c)
RETURN c.name
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.43 VarLengthAcceptance2

2.43.1 Handling relationships that are already bound in variable length paths

Query specification

```
MATCH ()-[r:EDGE]-()
MATCH p = (n)-[*0..1]-()-[r]-()-[*0..1]-(m)
RETURN count(p) AS c
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.44 WhereAcceptance

2.44.1 NOT and false

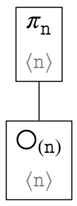
Query specification

```
MATCH (n)
WHERE NOT(n.name = 'apa' AND false)
RETURN n
```

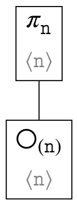
Relational algebra expression

$\pi_{\mathtt{n}} \left(\not\risingdotseq \left(\bigcirc_{(\mathtt{n})} \right) \right)$

Relational algebra tree



Relational algebra tree for incremental queries



2.45 WithAcceptance

2.45.1 Passing on pattern nodes

Query specification

```
MATCH (a:A)
WITH a
MATCH (a)-->(b)
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.2 ORDER BY and LIMIT can be used

Query specification

```
MATCH (a:A)
WITH a
ORDER BY a.name
LIMIT 1
MATCH (a)-->(b)
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.3 No dependencies between the query parts

Query specification

```
MATCH (a)
WITH a
MATCH (b)
RETURN a, b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.4 Aliasing

Query specification

```
MATCH (a:Begin)
WITH a.prop AS property
MATCH (b:End)
WHERE property = b.prop
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.5 Handle dependencies across WITH

Query specification

```
MATCH (a:Begin)
WITH a.prop AS property
  LIMIT 1
MATCH (b)
WHERE b.id = property
RETURN b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.6 Handle dependencies across WITH with SKIP

Query specification

```
MATCH (a)
WITH a.prop AS property, a.key AS idToUse
  ORDER BY property
  SKIP 1
MATCH (b)
WHERE b.id = idToUse
RETURN DISTINCT b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.7 WHERE after WITH should filter results

Query specification

```
MATCH (a)
WITH a
WHERE a.name = 'B'
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.8 WHERE after WITH can filter on top of an aggregation

Query specification

```
MATCH (a)-->()
WITH a, count(*) AS relCount
WHERE relCount > 1
RETURN a
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.9 ORDER BY on an aggregating key

Query specification

```
MATCH (a)
WITH a.bar AS bars, count(*) AS relCount
ORDER BY a.bar
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.10 ORDER BY a DISTINCT column

Query specification

```
MATCH (a)
WITH DISTINCT a.bar AS bars
ORDER BY a.bar
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.11 WHERE on a DISTINCT column

Query specification

```
MATCH (a)
WITH DISTINCT a.bar AS bars
WHERE a.bar = 'B'
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.12 A simple pattern with one bound endpoint

Query specification

```
MATCH (a:A)-[r:REL]->(b:B)
WITH a AS b, b AS tmp, r AS r
WITH b AS a, r
LIMIT 1
MATCH (a)-[r]->(b)
RETURN a, r, b
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.13 Null handling

Query specification

```
OPTIONAL MATCH (a:Start)
WITH a
MATCH (a)-->(b)
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.14 Nested maps

Query specification

```
WITH {foo: {bar: 'baz'}} AS nestedMap
RETURN nestedMap.foo.bar
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.15 Connected components succeeding WITH

Query specification

```
MATCH (n:A)
WITH n
LIMIT 1
MATCH (m:B), (n)-->(x:X)
RETURN *
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.16 Single WITH using a predicate and aggregation

Query specification

```
MATCH (n)
WITH n
WHERE n.prop = 42
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.

2.45.17 Multiple WITHs using a predicate and aggregation

Query specification

```
MATCH (david {name: 'David'})--(otherPerson)-->()
WITH otherPerson, count(*) AS foaf
WHERE foaf > 1
WITH otherPerson
WHERE otherPerson.name <> 'NotOther'
RETURN count(*)
```

Relational algebra expression

Cannot convert to expression.

Relational algebra tree

Cannot visualize tree.

Relational algebra tree for incremental queries

Cannot visualize incremental tree.