Ann Math Artif Intell
DOI 10.1007/s10472-013-9346-x

The impact of transitive closure on the expressiveness of navigational query languages on unlabeled graphs

George H. L. Fletcher · Marc Gyssens · Dirk Leinders · Jan Van den Bussche · Dirk Van Gucht · Stijn Vansummeren · Yuqing Wu

Abstract Several established and novel applications motivate us to study the expressive power of navigational query languages on graphs, which represent binary relations. Our basic language has only the operators union and composition, together

Dirk Leinders carried out most of his work as a Senior Research Assistant of the Research Foundation Flanders (FWO).
Yuqing Wu carried out part of her work during a sabbatical visit to Hasselt University with a Senior Visiting Postdoctoral Fellowship of the Research Foundation Flanders (FWO).

George H. L. Fletcher
Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands
e-mail: g.h.l.fletcher@tue.nl

M. Gyssens (✉) · D. Leinders · J. Van den Bussche
School for Information Technology, Hasselt University and Transnational University of Limburg, Martelarenlaan 42, 3500 Hasselt, Belgium
e-mail: marc.gyssens@uhasselt.be

D. Leinders
e-mail: dirk.leinders@uhasselt.be

J. Van den Bussche
e-mail: jan.vandenbussche@uhasselt.be

D. Van Gucht · Y. Wu
School of Informatics and Computing, Indiana University, Lindley Hall, 150 S. Woodlawn Ave., Bloomington, IN 47405, USA

D. Van Gucht
e-mail: vgucht@cs.indiana.edu

Y. Wu
e-mail: yuqwu@cs.indiana.edu

S. Vansummeren
Department of Computer and Decision Engineering (CoDE), Université Libre de Bruxelles, CP165/15, av. F. D. Roosevelt 50, 1050 Brussels, Belgium
e-mail: stijn.vansummeren@ulb.ac.be

Published online: 02 April 2013

with the identity relation. Richer languages can be obtained by adding other features such as intersection, difference, projection and coprojection, converse, and the diversity relation. The expressive power of the languages thus obtained cannot only be evaluated at the level of path queries (queries returning binary relations), but also at the level of Boolean or yes/no queries (expressed by the nonemptiness of an expression). For the languages considered above, adding transitive closure augments the expressive power not only at the level of path queries but also at the level of Boolean queries, for the latter provided that multiple input relations are allowed. This is no longer true in the context of unlabeled graphs (i.e., in the case where there is only one input relation), however. In this paper, we prove that this is indeed not the case for the basic language to which none, one, or both of projection and the diversity relation are added, a surprising result given the limited expressive power of these languages. In combination with earlier work (Fletcher et al. 2011, 2012), this result yields a complete understanding of the impact of transitive closure on the languages under consideration.

Keywords Boolean query · Transitive closure · Relation algebra · Unlabeled graph · Expressiveness

Mathematics Subject Classifications (2010) 03C07 · 05C60 · 68P15 · 68R10

1 Introduction

In previous work [11], the present authors studied the relative expressive power of query languages on graphs (i.e., binary relations). They considered a basic language, consisting of union, composition, and the identity relation, to which one or more features can be added, such as intersection, set difference, projection, coprojection, converse, and the diversity relation. We refer to the basic language to which all the non-basic features have been added as the *relation algebra*.

A relation algebra expression can be seen as a function mapping the input binary relation to a binary relation. We call such queries *path queries* because the result can be interpreted as all the ways in which the input graph can be navigated in accordance with the expression. By identifying nonemptiness with the Boolean value *true* and emptiness with *false*, as is standard in database theory [2], we can also express yes/no queries within this framework. To distinguish them from general path queries, we shall refer to the latter as *Boolean queries*.

The present authors were able to establish the complete Hasse diagram for the relative expressive power of the various relation algebra fragments, and this both at the levels of (1) path queries and (2) Boolean queries, both for the cases where the input graph is (1) labeled (i.e., may represent multiple binary relations) and (2) unlabeled (i.e., represents a single relation).

This study was motivated by similar work on the expressive power of XPath fragments as query languages for navigating on trees, which is now well understood (e.g., [6, 9, 14, 19, 20, 26]). Motivated by data on the Web [1, 12] and new applications such as dataspaces [13], Linked Data [7, 16], and RDF [23], it is natural to look at similar navigational query languages for graphs. The languages we study are very

natural and similar to languages already considered in the fields of description logics, dynamic logics, arrow logics, and relation algebras [5, 8, 15, 17, 21, 24]. Moreover, graph query languages have a rich history in database theory, in particular in the context of object-oriented and semi-structured database systems. We refer to Angles and Gutiérrez [4] for a comprehensive review.

In addition to what has been described above, we also investigated whether adding transitive closure to a relation algebra fragment yields additional expressive power. At the level of path queries, this is obviously the case for all fragments, as the transitive closure of a binary relation is not expressible in FO [3], whereas the full relation algebra is known to be equivalent to FO^3, the three-variable fragment of first-order logic [25]. We were also able to show [11] that adding transitive closure does not result in a collapse at the level of Boolean queries, provided the input graph is labeled (i.e., there may be several input relations). The argument used to prove this could not be generalized to the Boolean queries on unlabeled graphs (i.e., on a single input relation), however. With different arguments, we were able to show [10, 11] that, for labeled graphs, there is still no collapse if the language to which transitive closure is added has one of the operators set difference, intersection, coprojection, or converse.

The purpose of the present paper is to show that in the remaining cases, i.e., if the language under consideration is the basic language augmented with none, one, or both of the features projection and diversity, adding transitive closure does *not* yield more expressive power at the level of Boolean queries on unlabeled graphs. This result completes our understanding of whether or not the relation algebra fragments with transitive closure collapse to their counterparts without transitive closure at the level of Boolean queries on unlabeled graphs.

To see the practical relevance of these results, consider the following example. Facebook is a large social network which maintains a graph of people that are connected via a friendship relationship. It is customary that people wish to communicate with their friends, navigate recursively to friends of friends, etc. This navigation can be expressed with path expressions in a suitable relation algebra fragment, either with or without using transitive closure. In addition to navigation, certain topological properties of the Facebook graph can be discovered. For example, one can discover whether there are people whose friends are all friends of each other. Again, some of these topological properties can be formulated as Boolean queries in a suitably chosen relation algebra fragment, either with or without using transitive closure. The proliferation of social networks is thus a real-world phenomenon to which our theory applies.

From this perspective, the collapse results are very meaningful. To illustrate this on our example, consider the very simple Boolean query asking if there are people who are friends of friends of ... friends of each other, which is expressed by the transitive closure of the friends relation. Obviously, the answer to this query is affirmative if and only if the answer to the query asking if there are people who are friends of each other is affirmative. The latter query, of course, is simply expressed by the friends relation, i.e., without transitive closure. The collapse results indicate precisely for which of the languages under consideration such a collapse is guaranteed.

As the collapse results were already presented in the conference version of this paper [10], the emphasis of the current paper is on the proof technique used for

establishing these collapse results, of which we give an outline here. Let the set F consist of none, one, or both of the features projection and diversity, let $\mathscr{N}(F)$ be the basic language augmented with the features in F, and let $\mathscr{N}(F \cup \{^+\})$ be this language augmented with transitive closure. Let e be an expression in $\mathscr{N}(F \cup \{^+\})$, and let G be an unlabeled graph. The proof goes in two steps.

1. Find a suitable expression $suff_{F,e}$ in $\mathscr{N}(F)$, only depending on F and e, and *not* on G, such that $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$.
2. Find an expression e' in $\mathscr{N}(F)$, only depending on e, and *not* on G, such that $suff_{F,e}(G) = \emptyset$ implies $e(G) \neq \emptyset$ if and only if $e'(G) \neq \emptyset$.

It then readily follows that, for all unlabeled graphs G, $e(G) \neq \emptyset$ if and only if $suff_{F,e}(G) \cup e'(G) \neq \emptyset$, a Boolean query expressed in $\mathscr{N}(F)$.

To establish the second step, we identify a subgraph G' of G such that $e(G) \neq \emptyset$ if and only if $e(G') \neq \emptyset$. This subgraph G' depends both on G and e, but its number of nodes is bounded by a number depending only on e, say D. Hence, we can obtain the desired expression e' from e by exhaustively replacing subexpressions of the form f^+ by $\bigcup_{i=1}^{D} f^i$.

It is interesting to note that the same proof technique works for all the relation algebra fragments obtained from adding none, one, or both of projection and diversity to the basic language. Actually, only the first step of the proof, establishing a suitable expression $suff_{F,e}(G)$ is language-dependent. One might therefore hope that this proof technique could also be applicabe to other languages.

Moreover, it turns out that $suff_{F,e}(G)$ does not fully depend on e, but only on $|e|$, the number of occurrences of "R" in e. By "R," we mean the relation symbol in the basic language that is evaluated as the relation represented by the input graph. Even stronger, $suff_{F,e}(G)$ is a fixed expression in $\mathscr{N}(F \cup \{f\})$, where f stands for $R^{|e|}$, R composed with itself $|e|$ times. One can therefore argue that our proof yields a normal form for an expression in $\mathscr{N}(F)$ equivalent to a given expression in $\mathscr{N}(F \cup \{^+\})$.

The remainder of this paper is organized as follows. In Section 2, we introduce the basic concepts of this paper, including syntax and semantics of the languages under consideration. Continuing in this vein, we define in Section 3 trace expressions as a tool to link expressions in languages with transitive closure to expressions in the corresponding languages without transitive closure. Then, in Section 4, we describe some results of earlier work [10, 11] necessary to understand the context of the main result of the present paper, which is that adding transitive closure to languages containing no other nonbasic features than projection and diversity yields no additional power at the level of Boolean queries, when evaluated on unlabeled graphs. In Section 5, we state this result formally, and present in general terms a two-step strategy to prove it. In Section 6, we deal with the first step of this proof strategy. In Section 7, we describe in more detail than in Section 5 the proof strategy for the much more elaborate second step. The ingredients of this strategy are then discussed in subsequent sections: expressions with conditionals (Section 8), line patterns and graph patterns (Section 9), normalizing trace expressions (Section 10) and canonical subgraphs (Section 11). In Section 12, these ingredients are put to work to show a partial result for the case where all projection subexpressions are replaced by conditionals. This partial result is then bootstrapped to the main result in Section 13. We conclude in Section 14 by summarizing our understanding of the impact of adding

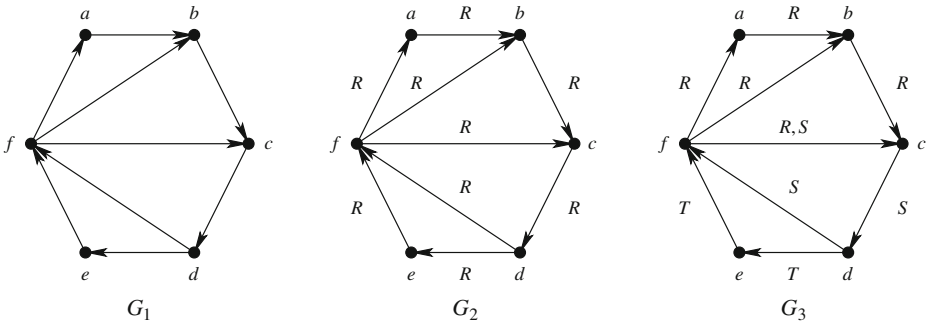


Fig. 1 Three unlabeled and labeled graphs used in Example 1

transitive closure to relation algebra fragments, which has now been completed, and discussing some directions for future work.

2 Graphs and languages

In this paper, we are interested in navigating over graphs. For our purposes, a graph is a relational structure G, consisting of a set of nodes V and a binary relation $R \subseteq V \times V$, the set of edges of G.

In what follows, both V and R may be either finite or infinite. Indeed, while we have finite graphs in mind from the point of view of the applications, we never rely on finiteness to prove the results of this paper. Therefore, the results do not only hold for finite graphs, but also for arbitrary graphs, which may be finite or infinite.

An extension of this model consists of allowing multiple binary relations, by labeling the edges.[1] For comparison, we shall sometimes refer to labeled graphs, though the emphasis of this paper is on unlabeled graphs.

Example 1 First, consider the unlabeled graph G_1 of Fig. 1. Clearly, this graph represents the relation $R(G_1) = \{(a, b), (b, c), (c, d), (d, e), (e, f), (f, a), (f, b), (f, c), (d, f)\}$. Clearly, the labeled graph G_2, in which all edges are labeled "R," represents the same relation, i.e., $R(G_2) = R(G_1)$. Finally, consider the labeled graph G_3, in which edges are labeled with "R," "S," or "T." (Notice that there is both an R- and an S-labeled edge (f, c).) This graph represents three relations, namely

$$\begin{aligned} R(G_3) &= \{(a, b), (b, c), (f, a), (f, b), (f, c)\}; \\ S(G_3) &= \{(c, d), (d, f), (f, c)\}; \text{ and} \\ T(G_3) &= \{(d, e), (e, f)\}. \end{aligned}$$

Clearly, every relational structure can be represented as a single graph, by appropriately labeling its edges. In this work, however, multiple-relation relational structures are only considered for purposes of comparison.

[1] In this case, the number of relation names is always finite.

To navigate over graphs, we consider queries at two different levels.

Definition 1

- A *path query* takes as input a graph G and returns a binary relation $e(G) \subseteq \text{adom}(G) \times \text{adom}(G)$, where $\text{adom}(G)$ denotes the *active domain* of G, which is the set of all vertices occurring in some edge of G.
- A *Boolean query* takes as input a graph G and returns *true* or *false*.

Example 2 Finding all pairs of nodes in a graph that are connected by a simple path of length 3 is an example of a path query; finding out whether a graph contains two nodes connected by a simple path of length 3 is an example of a Boolean query.

The most basic language for navigating over graphs we consider is the algebra $\mathscr{N}$ whose expressions are built recursively from the edge set symbol R, the primitive $\emptyset$, and the primitive id, using composition $(e_1 \circ e_2)$ and union $(e_1 \cup e_2)$.[2]

Semantically, each expression e in $\mathscr{N}$ represents a path query, inductively defined as follows:

$$\begin{aligned}
R(G) &= R\,;\\
\emptyset(G) &= \emptyset\,;\\
id(G) &= \{(v,v) \mid v \in \text{adom}(G)\}\,;\\
e_1 \circ e_2(G) &= \{(v,w) \mid \exists z : (v,z) \in e_1(G)\ \&\ (z,w) \in e_2(G)\}\,;\\
e_1 \cup e_2(G) &= e_1(G) \cup e_2(G)\,.
\end{aligned}$$

The basic algebra $\mathscr{N}$ can be extended by adding some of the following features: diversity (di), converse (e^{-1}), intersection ($e_1 \cap e_2$), difference ($e_1 \setminus e_2$), projections ($\pi_1(e)$ and $\pi_2(e)$), coprojections ($\overline{\pi}_1(e)$ and $\overline{\pi}_2(e)$), and transitive closure (e^+). We refer to the operators in the basic algebra $\mathscr{N}$ as *basic features*; we refer to the extensions as *nonbasic features*. The semantics of the extensions is as follows:

$$\begin{aligned}
di(G) &= \{(v,w) \mid v,w \in \text{adom}(G)\ \&\ v \neq w\}\,;\\
e^{-1}(G) &= \{(v,w) \mid (w,v) \in e(G)\}\,;\\
e_1 \cap e_2(G) &= e_1(G) \cap e_2(G)\,;\\
e_1 \setminus e_2(G) &= e_1(G) \setminus e_2(G)\,;\\
\pi_1(e)(G) &= \{(v,v) \mid v \in \text{adom}(G)\ \&\ \exists w : (v,w) \in e(G)\}\,;\\
\pi_2(e)(G) &= \{(v,v) \mid v \in \text{adom}(G)\ \&\ \exists w : (w,v) \in e(G)\}\,;\\
\overline{\pi}_1(e)(G) &= \{(v,v) \mid v \in \text{adom}(G)\ \&\ \neg\exists w : (v,w) \in e(G)\}\,;\\
\overline{\pi}_2(e)(G) &= \{(v,v) \mid v \in \text{adom}(G)\ \&\ \neg\exists w : (w,v) \in e(G)\}\,;\\
e^+(G) &= \bigcup_{k \geq 1} e^k(G)\,.
\end{aligned}$$

Here, e^k denotes $e \circ \cdots \circ e$ (k times). For future use, we put $e^0 := id$.

If F is a set of nonbasic features, we denote by $\mathscr{N}(F)$ the language obtained by adding all features in F to $\mathscr{N}$. For example, $\mathscr{N}(\cap)$ denotes the extension of $\mathscr{N}$

[2]By abuse of notation, we shall use "R" both as a symbol in the algebra $\mathscr{N}$ and as the name of the corresponding edge relation in G.

with intersection, and $\mathcal{N}(\cap, \pi, ^+)$ denotes the extension of $\mathcal{N}$ with intersection, both projections,[3] and transitive closure.

We refer to the language $\mathcal{N}(\setminus, di, ^{-1})$ as the *relation algebra*. For each set F of nonbasic features considered above not containing transitive closure, all path queries expressible in $\mathcal{N}(F)$ are also expressible in the relation algebra [17].

Language expressiveness can not only be considered at the level of path queries, but also at the level of Boolean queries.

Definition 2 Let F be a set of nonbasic features. A *path query* q is expressible in a language $\mathcal{N}(F)$ if there exists an expression e in $\mathcal{N}(F)$ such that, for every graph G, we have $e(G) = q(G)$. Similarly, a *Boolean query* q is expressible in $\mathcal{N}(F)$ if there exists an expression e in $\mathcal{N}(F)$ such that, for every graph G, we have that $e(G)$ is nonempty if and only if $q(G)$ is true. In both cases, we say that q *is expressed by* e.

In this paper, we are mainly interested in Boolean queries. Compared to path queries, this means that we are not interested in the precise set of pairs returned by an expression on a given input graph, but rather in whether or not this set is empty. Hence, if we can establish that adding transitive closure to a language does not increase its expressive power at the level of path queries, this must necessarily also be the case at the level of Boolean queries. There is no equally compelling reason why the converse should be true, however. Therefore, studying expressiveness issues is considerably more difficult at the level of Boolean queries than at the level of path queries.

To conclude these preliminaries, we formally define what we mean by a *subexpression* of a given expression.

Definition 3 Let F be a set of nonbasic features, and let e be an expression in $\mathcal{N}(F)$. The set of all *subexpressions* of e, denoted $\mathrm{Sub}(e)$, is defined recursively, as follows:

1. if e is either R, $\emptyset$, id, or di, then $\mathrm{Sub}(e) = \{e\}$;
2. if "$\diamond$" is either composition or a set operation, and if, for some expressions e_1 and e_2 in $\mathcal{N}(F)$, $e = e_1 \diamond e_2$, then $\mathrm{Sub}(e) = \mathrm{Sub}(e_1) \cup \mathrm{Sub}(e_2) \cup \{e\}$; and
3. if "θ" is either a projection, a coprojection, converse, or transitive closure, and if, for some expression f in $\mathcal{N}(F)$, $e = \theta(f)$, then $\mathrm{Sub}(e) = \mathrm{Sub}(f) \cup \{e\}$.

An expression that is either "R," "$\emptyset$," "id," or "di" is called *atomic*. For an expression e in the relation algebra with or without transitive closure, we denote by $|e|$ the number of occurrences of "R" in e.

3 Trace expressions

If we evaluate an expression e in $\mathcal{N}(\pi, di, ^+)$, then, to validate that, for some nodes v and w in a graph G, $(v, w) \in e(G)$, we must in general make some choices to arrive at that result. In particular, when evaluating a subexpression $f_1 \cup f_2$, we must decide

[3] We do not consider extensions of $\mathcal{N}$ in which only one of the two projections, respectively one of the two coprojections, is present.

whether to evaluate f_1 or f_2, Similarly, if we encounter a subexpression f^+, we must decide how many times we are going to iterate over f. To formalize this, we introduce *trace expressions*.

Definition 4 Let e be an expression in $\mathcal{N}(\pi, di, ^+)$. Then, $\mathcal{T}(e)$, the set of *trace expressions* of e, is defined recursively, as follows:

1. if e is an atomic expression, then $\mathcal{T}(e) = \{e\}$;
2. for $i = 1, 2$, $\mathcal{T}(\pi_i(e)) = \{\pi_i(f) \mid f \in \mathcal{T}(e)\}$;
3. $\mathcal{T}(e_1 \cup e_2) = \mathcal{T}(e_1) \cup \mathcal{T}(e_2)$;
4. $\mathcal{T}(e_1 \circ e_2) = \{f_1 \circ f_2 \mid f_1 \in \mathcal{T}(e_1)\ \&\ f_2 \in \mathcal{T}(e_2)\}$; and
5. $\mathcal{T}(e^+) = \bigcup_{k \geq 1} \{f_1 \circ \cdots \circ f_k \mid \forall i = 1, \ldots, k : f_i \in \mathcal{T}(e)\}$.

Notice that, indeed, trace expressions do not contain union and transitive closure. The intuition expressed in the opening paragraph of this section is now captured by Proposition 1, which follows from a straightforward structural induction argument.

Proposition 1 *Let e be an expression in $\mathcal{N}(\pi, di, ^+)$. Let G be a graph and v and w nodes of G. Then, $(v, w) \in e(G)$ if and only if there exists $f \in \mathcal{T}(e)$ such that $(v, w) \in f(G)$.*

Trace expressions represent different ways to evaluate an expression containing union and/or transitive closure, hence the name "trace." Unfortunately, trace expressions do not always carry sufficient information to match them unambiguously with particular ways to evaluate the original expression. To see this, consider the following example. Let $R^+ \cup (R \circ R)^+$ be an expression in $\mathcal{N}(^+)$. Clearly, $R \circ R$ is one of its trace expressions. However, it can result from two different ways of evaluating the orginal expression:

1. one can iterate twice over R^+, the left-hand term of the union in $R^+ \cup (R \circ R)^+$; or
2. one can iterate once over $(R \circ R)^+$, the right-hand term of the union in $R^+ \cup (R \circ R)^+$.

Since we will need to match traces unambigously with particular ways of evaluating an expression, we wish to point out that the concept of trace can be "enriched" to allow an unambigous matching.

Formally, this can be achieved by marking the original expression. That is, we tag each atom in that expression with its position in that expression. For example, ${R_1}^+ \cup (R_2 \circ R_3)^+$ is a marked version of $R^+ \cup (R \circ R)^+$. We can then define marked traces in much the same way as above, starting from the marked version of the original expression, instead of the original expression itself. The tags in a marked trace then reveal in an unambiguous manner which particular way of evaluating the original expression led to that trace.

For example, there are two marked traces of ${R_1}^+ \cup (R_2 \circ R_3)^+$ corresponding to the trace $R \circ R$ of $R^+ \cup (R \circ R)^+$: $R_1 \circ R_1$ and $R_2 \circ R_3$. The former corresponds with iterating twice over R^+, and the latter with iterating once over $(R \circ R)^+$.

In this work, we shall not introduce marked traces formally, to avoid overloading the notation. Nevertheless, we shall always assume implicitly for each trace we

consider that a marking is available that matches the symbols in the trace in an unambiguous manner with symbols in the original expression.

4 Describing the context

In this section, we describe some results [10, 11] necessary to understand the context of the results of the present paper.

First, we observe that there exist the following interdependencies between the features introduced in Section 2:

$$\begin{aligned}
\pi_1(e) &= (e \circ e^{-1}) \cap id = (e \circ (id \cup di)) \cap id = \overline{\pi}_1(\overline{\pi}_1(e));\\
\pi_2(e) &= (e^{-1} \circ e) \cap id = ((id \cup di) \circ e) \cap id = \overline{\pi}_2(\overline{\pi}_2(e));\\
\overline{\pi}_1(e) &= id \setminus \pi_1(e);\\
\overline{\pi}_2(e) &= id \setminus \pi_2(e);\\
e_1 \cap e_2 &= e_1 \setminus (e_1 \setminus e_2).
\end{aligned}$$

For a set of nonbasic features F not containing transitive closure, let $\overline{F}$ be the set obtained by augmenting F with all nonbasic features that can be expressed in $\mathcal{N}(F)$ through a repeated application of the above equalities. For example, $\overline{\{\setminus, {}^{-1}\}} = \{\setminus, {}^{-1}, \cap, \pi, \overline{\pi}\}$.

The present authors have been able to show the following result.

Proposition 2 [11] *Let F_1 and F_2 be sets of nonbasic features not containing transitive closure. The language $\mathcal{N}(F_1)$ is at most as expressive as the language $\mathcal{N}(F_2)$ at the level of path queries if and only if $F_1 \subseteq \overline{F}_2$.*

For Boolean queries, the situation is slightly more complicated, because of the following result.

Proposition 3 [11] *Let F be a set of nonbasic features not containing transitive closure. If $\overline{F}$ does not contain intersection, then $\mathcal{N}(F \cup \{^{-1}\})$ is at most as expressive as $\mathcal{N}(F \cup \{\pi\})$ at the level of Boolean queries.*

So, in the presence of projection and in the absence of intersection, converse does not add expressive power at the Boolean level. To accommodate this additional result, we define the following notion, given a set of nonbasic features F not containing transitive closure:

$$\widetilde{F} = \begin{cases} \overline{F} \cup \{^{-1}\} & \text{if } \pi \in \overline{F} \text{ and } \cap \notin \overline{F};\\ \overline{F} & \text{otherwise.} \end{cases}$$

For example, $\widetilde{\{\overline{\pi}, di\}} = \{^{-1}, \pi, \overline{\pi}, di\}$.

The present authors were able to establish the following analogue of Proposition 2 for Boolean queries.

Proposition 4 [11] *Let F_1 and F_2 be sets of nonbasic features not containing transitive closure. The language $\mathcal{N}(F_1)$ is at most as expressive as the language $\mathcal{N}(F_2)$ at the level of Boolean queries if and only if $F_1 \subseteq \widetilde{F}_2$.*

In particular, Proposition 4 establishes which relation algebra fragments not containing transitive closure are equivalent in expressive power at the level of Boolean queries.

What happens if we add transitive closure to these fragments?

At the level of path queries, the answer is straightforward, as it is well-known that the expression R^+ represents a query not expressible in FO (see, e.g., [2]). Hence, adding transitive closure always strictly increases the expressive power at the level of path queries. At the level of Boolean queries, the situation is more subtle. Indeed, the argument above is no longer applicable, as $R^+ \neq \emptyset$ if and only if $R \neq \emptyset$. Using a straightforward Ehrenfeucht-Fraïssé argument (see, e.g., [2]), it is nevertheless still possible to show that the Boolean query represented by the expression $R \circ S^+ \circ R$ is not expressible in FO. However, this expression contains *two* relation names. Hence, also at the level of Boolean queries, adding transitive closure always strictly increases the expressive power, but only if labeled input graphs with at least two relation names are allowed. This begs the question whether this result still holds for *unlabeled* input graphs. Using ad-hoc arguments, the present authors were able to establish the following.

Proposition 5 [10, 11] *Let F be a set of nonbasic features such that $\overline{F}$ contains at least one of intersection, converse, or coprojection. Then, adding transitive closure to $\mathcal{N}(F)$ strictly increases the expressive power at the level of Boolean queries, even if only unlabeled input graphs are considered.*

Taking into account Proposition 4, four relation algebra fragments are not covered by Proposition 5: $\mathcal{N}$, $\mathcal{N}(\pi)$, $\mathcal{N}(di)$, and $\mathcal{N}(\pi, di)$. It is the purpose of the present paper to prove that adding transitive closure to these fragments does *not* increase their expressive power at the level of Boolean queries if only unlabeled graphs are considered.

From now on, we assume a graph is unlabeled unless stated otherwise.

5 Main result and general proof strategy

The rest of the paper is devoted to proving that, at the level of Boolean queries, $\mathcal{N}(F \cup \{^+\})$ collapses to $\mathcal{N}(F)$ for all sets of nonbasic features F for which $F \subseteq \{\pi, di\}$. We first state this result formally.

Theorem 1 *Let $F \subseteq \{\pi, di\}$ be a set of nonbasic features. Then, for each expression e in $\mathcal{N}(F \cup \{^+\})$, there exists an expression $\tilde{e}$ in $\mathcal{N}(F)$ such that, for each unlabeled graph G, $\tilde{e}(G) \neq \emptyset$ if and only if $e(G) \neq \emptyset$.*

We next give an outline of our strategy for proving Theorem 1. We start with an introductory example.

Example 3 Consider the expression $e := \pi_1(R^3) \circ R^+ \circ di \circ \pi_2(R) \circ R^2$ in $\mathcal{N}(\pi, di, ^+)$. Let G be a graph. For $e(G)$ to be nonempty, the subexpressions to the right of "di" must return nonempty. Hence, there must exist a chain $w_0 \to w_1 \to w_2 \to w_3$ in G. Unless, for each such chain, $w_1 = w_2 = w_3$, it is readily seen that this condition is also sufficient for $e(G) \neq \emptyset$. In the other case, there must also exist an edge $v_0 \to v_1$ with a self-loop in v_1 for which $v_1 \neq w_1$ in order for $e(G)$ to be nonempty. It can now be readily verified that, in both cases, $e'(G) \neq \emptyset$, with $e' := \pi_1(R^3) \circ (R \cup R^2) \circ di \circ \pi_2(R) \circ R^2$ in $\mathcal{N}(\pi, di)$. As always $e'(G) \subseteq e(G)$, the converse implication also holds, so $e' \in \mathcal{N}(\pi, di)$ is equivalent to $e \in \mathcal{N}(\pi, di, ^+)$ at the level of Boolean queries.

The argument used to show that transitive closure can be eliminated from the expression in Example 3 is very ad-hoc. Moreover, the considered expression is very simple. We therefore need a general technique to show that, for $F \subseteq \{\pi, di\}$, $\mathcal{N}(F \cup \{^+\})$ collapses to $\mathcal{N}(F)$ at the level of Boolean queries. In this section, we outline this technique, and, in subsequent sections, we work it out in further detail. It consists of two steps. Given an expression e in $\mathcal{N}(F \cup \{^+\})$,

1. find an expression $suff_{F,e}$ in $\mathcal{N}(F)$ such that, for every graph G, $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$; and
2. find an expression e' in $\mathcal{N}(F)$ that is equivalent to e at the level of Boolean queries on all graphs G for which $suff_{F,e}(G) = \emptyset$.

It then follows immediately that, on all graphs, e is equivalent to $\tilde{e} := suff_{F,e} \cup e'$ at the level of Boolean queries, i.e., for every graph G, $\tilde{e}(G) \neq \emptyset$ if and only if $e(G) \neq \emptyset$. Intuitively, $suff_{F,e}(G) \neq \emptyset$ is a sufficient condition for $e(G)$ to be nonempty. It therefore suffices to show the collapse on graphs that do not satisfy this condition, i.e., for which $suff_{F,e}(G) = \emptyset$. If $suff_{F,e}$ is well-chosen, then the latter condition will turn out to be sufficiently restrictive for our purposes.

6 The first step

The first step of the proof strategy described in Section 5 is, given $F \subseteq \{\pi, di\}$ and an expression e in $\mathcal{N}(F \cup \{^+\})$, finding an expression $suff_{F,e}$ in $\mathcal{N}(F)$ for which $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$ for every input graph G. This first step is secured by a series of lemmas, summarized in Theorem 2.

We start with the following straightforward observations.

Lemma 1 *Let G_1 and G_2 be graphs, and let h be a homomorphism from G_1 to G_2.*

1. *Let e be an expression in $\mathcal{N}(\pi, ^+)$. Then $(v, w) \in e(G_1)$ implies $(h(v), h(w)) \in e(G_2)$.*
2. *Let e be an expression in $\mathcal{N}(\pi, di, ^+)$. If h is* injective*, then $(v, w) \in e(G_1)$ implies $(h(v), h(w)) \in e(G_2)$.*

Arguably, the simplest graphs we can consider in this contexts are *chains*. A chain of length m, denoted C_m, is a graph consisting of mutually distinct nodes $v_0, \ldots, v_m$

and edges between subsequent nodes. Lemma 1 can then help us to link the behavior of an expression on such a chain to the behavior of that expression on the given input graph.

Lemma 2 *Let e be an expression in $\mathcal{N}(\pi, ^+)$. Then, for $m \geq |e|$, $e(C_m) \neq \emptyset$.*

Proof The proof is a structural induction argument. The only non-straightforward case to consider is the induction step for composition. Thus, suppose that $e = e_1 \circ e_2$, and that e_1 and e_2 satisfy Lemma 2. In particular, $e_1(C_{|e_1|}) \neq \emptyset$ and $e_2(C_{|e_2|}) \neq \emptyset$. Let the chain $C_{|e_1|}$ consist of the nodes $v_0, \ldots, v_{|e_1|}$ and $C_{|e_2|}$ consist of the nodes $w_0, \ldots, w_{|e_2|}$. Let $(v_i, v_j) \in e_1(C_{|e_1|})$ and $(w_k, w_l) \in e_2(C_{|e_2|})$. Finally, for $m \geq |e| = |e_1| + |e_2|$, let C_m consist of the nodes $z_0, \ldots, z_m$. We now distinguish two cases.

1. $j \geq k$. Consider the homomorphism from $C_{|e_1|}$ to C_m mapping v_0 to z_0, and hence v_i to z_i and v_j to z_j. By Lemma 1, (1), $(z_i, z_j) \in e_1(C_m)$. Since $j \geq k$, there exists a homomorphism from $C_{|e_2|}$ to C_m mapping w_k to z_j, and hence w_l to z_{j+l-k}. By Lemma 1, (1), $(z_j, z_{j+l-k}) \in e_2(C_m)$. Hence, $(z_i, z_{j+l-k}) \in e(C_m)$.
2. $j < k$. Consider the homomorphism from $C_{|e_2|}$ to C_m mapping $w_{|e_2|}$ to z_m, and hence w_k to $z_{k+m-|e_2|}$ and w_l to $z_{l+m-|e_2|}$. By Lemma 1, (1), $(z_{k+m-|e_2|}, z_{l+m-|e_2|}) \in e_2(C_m)$. Since $j < k$, there exists a homomorphism from $C_{|e_1|}$ to C_m mapping v_j to $z_{k+m-|e_2|}$, and hence v_i to $z_{k+i-j+m-|e_2|}$. By Lemma 1, (1), $(z_{k+i-j+m-|e_2|}, z_{k+m-|e_2|}) \in e_1(C_m)$. Hence, $(z_{k+i-j+m-|e_2|}, z_{l+m-|e_2|}) \in e(C_m)$.

In both cases, we find that $e(C_m) \neq \emptyset$. □

Using the above lemmas, we can easily find an expression $suff_{F,e}$ if $F \subseteq \{\pi\}$.

Lemma 3 *Let e be an expression in $\mathcal{N}(\pi, ^+)$, and let G be a graph. If $R^{|e|}(G) \neq \emptyset$, then $e(G) \neq \emptyset$.*

Proof The condition $R^{|e|}(G) \neq \emptyset$ is equivalent to the existence of a homomorphism from $C_{|e|}$ to G. By Lemma 2, $e(C_{|e|}) \neq \emptyset$. Hence, by Lemma 1, (1), $e(G) \neq \emptyset$. □

We now consider the case where $F = \{di\}$.

Lemma 4 *Let e be an expression in $\mathcal{N}(di, ^+)$, and let G be a graph. If $R^{|e|} \circ di \circ R^{|e|}(G) \neq \emptyset$, then $e(G) \neq \emptyset$.*

Proof We first consider an expression f in $\mathcal{N}(di)$ of the form $f := R^{m_1} \circ di \circ R^{m_2} \circ di \circ \cdots \circ di \circ R^{m_n}$, $n \geq 1$, $1 \leq m_1, \ldots, m_n \leq |e|$, and prove that it returns a nonempty result on graphs satisfying $R^{|e|} \circ di \circ R^{|e|}(G) \neq \emptyset$.[4] Thereto, we distinguish two cases.

1. *There exist nodes v_1, w_1, v_2, and w_2 in G such that $(v_1, w_1) \in R^{|e|}(G)$, $(v_2, w_2) \in R^{|e|}(G)$, and $v_1 \neq v_2$.* For $i = 1, \ldots, n$, let $f_i = R^{m_1} \circ di \circ \cdots \circ di \circ R^{m_i}$. We prove

[4] Notice that this statement is voidly true if $|e| = 0$.

by induction that there exist nodes $z_1, \ldots, z_n$ in G such that $(v_1, z_i) \in f_i(G)$. For the base case, $i = 1$, this follows from $m_1 \leq |e|$. Assume that we have already established that, for some node z_i of G, $(v_1, z_i) \in f_i(G)$. Since $v_1 \neq v_2$, $z_i \neq v_1$ or $z_i \neq v_2$. Without loss of generality, assume the latter. Since $(v_2, w_2) \in R^{|e|}(G)$ and $m_{i+1} \leq |e|$, it follows that there exists a node z_{i+1} in G such that $(v_2, z_{i+1}) \in R^{m_{i+1}}(G)$. Hence, $(v_1, z_{i+1}) \in f_{i+1}(G)$, as was to be shown. We find in particular that $f(G) = f_n(G) \neq \emptyset$.

2. *There is only one node v in G such that, for some node w in G,* $(v, w) \in R^{|e|}(G)$. From $R^{|e|} \circ di \circ R^{|e|}(G) \neq \emptyset$, it follows that, for some node w in G with $(v, w) \in R^{|e|}(G)$, $v \neq w$.
   We distinguish two subcases.

   (a) $(v, v) \in R$. Notice that there must also exist a node z in G with $(v, z) \in R$ and $v \neq z$. Otherwise, it would be impossible that, for some node w in G with $(v, w) \in R^{|e|}(G)$, $v \neq w$. From $(v, v) \in R$ and $(v, z) \in R$, it follows that, for all $m \geq 1$, $(v, z) \in R^m(G)$. Since $v \neq z$, we may conclude that also $(v, z) \in f(G)$. In particular, $f(G) \neq \emptyset$.

   (b) $(v, v) \notin R$. By assumption, there exist nodes $v = v_0, v_1, \ldots, v_{|e|} = w$, such that, for $m = 0, \ldots, |e| - 1$, $(v_m, v_{m+1}) \in R$. From the assumption in this subcase, it immediately follows that $v = v_0 \neq v_1$. Next, consider node v_m, $2 \leq m \leq |e|$. If $v = v_0 = v_m$, then there exists k, $0 \leq k \leq m$, such that $(v_1, v_k) \in R^{|e|}(G)$, contradicting the assumption that $v = v_0$ is the unique node for which there exists a node w such that $(v, w) \in R^{|e|}(G)$. Hence, for $m = 1, \ldots, |e|$, $(v, v_m) \in R^m(G)$, and $v \neq v_m$. Following a similar reasoning as in Subcase 2a, we find that $(v, v_{m_n}) \in f(G)$. In particular, $f(G) \neq \emptyset$.

Notice that it also follows from our reasoning that expressions of the form

$$
\begin{aligned}
& di \circ R^{m_1} \circ di \circ R^{m_2} \circ di \circ \cdots \circ di \circ R^{m_n}; \\
& \quad R^{m_1} \circ di \circ R^{m_2} \circ di \circ \cdots \circ di \circ R^{m_n} \circ di; \text{ and} \\
& di \circ R^{m_1} \circ di \circ R^{m_2} \circ di \circ \cdots \circ di \circ R^{m_n} \circ di,
\end{aligned}
$$

$n \geq 1$, $1 \leq m_1, \ldots, m_n \leq |e|$, return a nonempty result on graphs satisfying $R^{|e|} \circ di \circ R^{|e|}(G) \neq \emptyset$, since this condition implies that G contains at least two nodes.

Now, consider a trace expression $f \in \mathscr{T}(e)$ for which $|f| \leq |e|$. (All trace expressions obtained by iterating only once over all transitive closure subexpressions satisfy this condition.) First, superfluous occurrences of "*id*" can be eliminated from f. Next, remember that a graph G satisfying $R^{|e|} \circ di \circ R^{|e|}(G) \neq \emptyset$ contains at least two nodes. If G contains exactly two nodes, then di^k is equivalent to *id* if k is even and to *di* if k is odd. Otherwise, di^k is equivalent to $id \cup di$. It follows that, if $|e| = 0$, then, on G, f is equivalent to either *id*, *di*, or $id \cup di$. In each case $f(G) \neq \emptyset$. If $|e| > 0$, then, on G, f is equivalent to a union of expressions of the types considered above. Hence, also in this case, we may conclude that $f(G) \neq \emptyset$.

From Proposition 1, it now immediately follows that, in all cases, $e(G) \neq \emptyset$. □

Finally, we deal with the case where $F = \{\pi, di\}$.

Lemma 5 *Let e be an expression in $\mathscr{N}(\pi, di, ^+)$, and let G be a graph. If $\pi_1(R^{|e|}) \circ \pi_2(R^{|e|}) \circ di \circ \pi_1(R^{|e|}) \circ \pi_2(R^{|e|}) \neq \emptyset$, then $e(G) \neq \emptyset$.*

Proof We first observe that the condition $\pi_1(R^{|e|}) \circ \pi_2(R^{|e|}) \circ di \circ \pi_1(R^{|e|}) \circ \pi_2(R^{|e|})(G) \neq \emptyset$ is equivalent to the existence of two sequences of not necessarily all different nodes $v_{-|e|}, \ldots, v_{-1}, v_0, v_1, \ldots, v_{|e|}$ and $w_{-|e|}, \ldots, w_{-1}, w_0, w_1, \ldots, w_{|e|}$ in G such that, (1) for $i = -|e|, \ldots, |e| - 1$, $(v_i, v_{i+1}) \in R$ and $(w_i, w_{i+1}) \in R$, and (2) $v_0 \neq w_0$.

Let H be the subgraph of G consisting of the nodes and edges singled out above. Let f be a union-free expression in $\mathscr{N}(\pi, di)$ with $|f| \leq |e|$. We show by a nested inductive argument that,

1. for all $i = 0, \ldots, |e| - |f|$, there exists j with $0 \leq j \leq i + |f|$ such that $(v_i, v_j) \in f(H)$ or $(v_i, w_j) \in f(H)$;
2. for all $i = -|e|, \ldots, 0$, there exists j with $i \leq j \leq |f|$ such that $(v_i, v_j) \in f(H)$ or $(v_i, w_j) \in f(H)$;
3. for all $i = 0, \ldots, |e| - |f|$, there exists j with $0 \leq j \leq i + |f|$ such that $(w_i, v_j) \in f(H)$ or $(w_i, w_j) \in f(H)$;
4. for all $i = -|e|, \ldots, 0$, there exists j with $i \leq j \leq |f|$ such that $(w_i, v_j) \in f(H)$ or $(w_i, w_j) \in f(H)$;
5. for all $i = -|e| + |f|, \ldots, 0$, there exists j with $i - |f| \leq j \leq 0$ such that $(v_j, v_i) \in f(H)$ or $(w_j, v_i) \in f(H)$;
6. for all $i = 0, \ldots, |e|$, there exists j with $-|f| \leq j \leq i$ such that $(v_j, v_i) \in f(H)$ or $(w_j, v_i) \in f(H)$;
7. for all $i = -|e| + |f|, \ldots, 0$, there exists j with $i - |f| \leq j \leq 0$ such that $(v_j, w_i) \in f(H)$ or $(w_j, w_i) \in f(H)$; and
8. for all $i = 0, \ldots, |e|$, there exists j with $-|f| \leq j \leq i$ such that $(v_j, w_i) \in f(H)$ or $(w_j, w_i) \in f(H)$.

We first show the first statement for the case that f is projection-free, i.e., that f is in $\mathscr{N}(di)$. First, observe that, always, $(v_i, v_i) \in id(H)$. We can view each union-free expression in $\mathscr{N}(di)$ as a composition of "id" with none, one, or more factors "R" or "di." Thus, assume that f is a union-free expression in $\mathscr{N}(di)$ with $|f| \leq |e|$, and let $f = g \circ R$, where g satisfies the first statement above. Let $0 \leq i \leq |e| - |f|$. Since $|f| = |g| + 1$, $0 \leq i \leq |e| - |g|$. By the first statement of the induction hypothesis, there exists j with $0 \leq j \leq i + |g|$ such that $(v_i, v_j) \in f(H)$ or $(v_i, w_j) \in f(H)$. Observe that $j \leq i + |g| \leq |e| - |f| + |g| = |e| - 1$. Hence, $(v_j, v_{j+1}) \in R$ and $(w_j, w_{j+1}) \in R$, as a consequence of which $(v_i, v_{j+1}) \in f(H)$ or $(v_i, w_{j+1}) \in f(H)$. Finally, notice that $j + 1 \leq i + |g| + 1 = i + |f|$. Alternatively, assume that $f = g \circ di$, where g satisfies the first statement above. Let $0 \leq i \leq |e| - |f|$. Since $|f| = |g|$, $0 \leq i \leq |e| - |g|$. By the induction hypothesis, there exists j with $0 \leq j \leq i + |g|$ such that $(v_i, v_j) \in f(H)$ or $(v_i, w_j) \in f(H)$. Without loss of generality, assume the latter. Since $v_0 \neq w_0$, $w_j \neq v_0$ or $w_j \neq w_0$. Again without loss of generality, assume the latter. Then $(v_i, w_0) \in f(H)$. We have thus shown that the first statement holds for all union-free expressions f in $\mathscr{N}(di)$ with $|f| \leq |e|$. The other statements for this case are shown analogously.

We now consider the general case, and use the case above as the basis for an induction on the number of projection subexpressions in the expression under consideration. We focus again of the first statement. Thus, assume that f is in $\mathscr{N}(\pi, di)$ with $|f| \leq |e|$, and that $0 \leq i \leq |e| - |f|$. If f is not projection-free, we can write $f = f_1 \circ \pi_1(f_2) \circ f_3$ or $f = f_1 \circ \pi_2(f_2) \circ f_3$, with f_1 projection-free, and f_2 and f_3 containing fewer projection subexpressions than f. By the first statement of the basis

Table 1 Expressions $suff_{F,e}$ in $\mathcal{N}(F)$ for which $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$, $F \subseteq \{\pi, di\}$

F	$suff_{F,e}$
$\emptyset$	$R^{\lvert e\rvert}$
$\{\pi\}$	$R^{\lvert e\rvert}$
$\{di\}$	$R^{\lvert e\rvert} \circ di \circ R^{\lvert e\rvert}$
$\{\pi, di\}$	$\pi_1(R^{\lvert e\rvert}) \circ \pi_2(R^{\lvert e\rvert}) \circ di \circ \pi_1(R^{\lvert e\rvert}) \circ \pi_2(R^{\lvert e\rvert})$

of this induction, there exists $j, 0 \leq j \leq i + |f_1|$, such that $(v_i, v_j) \in f_1(H)$ or $(v_i, w_j) \in f_1(H)$. Without loss of generality, assume the latter. Clearly, $j \leq |e| - (|f_2| + |f_3|)$, in particular, $j \leq |e| - |f_2|$ and $j \leq |e| - |f_3|$. By the latter condition and the third statement of the induction hypothesis, there exists k, $0 \leq k \leq j + |f_3| \leq i + |f_1| = |f_3| \leq i + |f|$ such that $(w_j, v_k) \in f_3(H)$ or $(w_j, w_k) \in f_3(H)$. We now distinguish the two cases.

1. $f = f_1 \circ \pi_1(f_2) \circ f_3$. As above, we can derive from the the third statement of the induction hypothesis that there exists $l, 0 \leq l \leq j + |f_2|$ such that $(w_j, v_l) \in f_2(H)$ or $(w_j, w_l) \in f_2(H)$. In particular, $(w_j, w_j) \in \pi_1(f_2)(H)$. Combining everything together, we find that $(v_i, v_k) \in f(H)$ or $(v_i, w_k) \in f(H)$, with k in the desired range.
2. $f = f_1 \circ \pi_2(f_2) \circ f_3$. By the last statement of the induction hypothesis, it follows that there exists l, $-|f_2| \leq l \leq j$, such that $(v_l, w_j) \in f_2(H)$ or $(w_l, w_j) \in f_2(H)$. In particular, $(w_j, w_j) \in \pi_2(f_2)(H)$. Combining everything together, we find that, also in this case, $(v_i, v_k) \in f(H)$ or $(v_i, w_k) \in f(H)$, with k in the desired range.

The induction step for the other seven statements is analogous.

We have thus shown, for every union-free expression f in $\mathcal{N}(\pi, di)$ with $|f| \leq |e|$, that $f(H) \neq \emptyset$, and, hence, by Lemma 1, (2), that $f(G) \neq \emptyset$. Since all trace expressions $f \in \mathcal{T}(e)$ obtained by iterating only once over transitive closure subexpressions satisfy $|f| \leq |e|$, it follows from Proposition 1 that also $e(G) \neq \emptyset$. □

Theorem 2 below summarizes Lemmas 3, 4, and 5.

Theorem 2 *Let $F \subseteq \{\pi, di\}$ be a set of nonbasic features. Let e be an expression in $\mathcal{N}(F \cup \{^+\})$. Let $suff_{F,e}$ in $\mathcal{N}(F)$ be as tabulated in Table 1. Then, for every graph G, $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$.*

7 Proof strategy for the second step

In Section 6, we established, for $F \subseteq \{\pi, di\}$ and e an expression in $\mathcal{N}(F \cup \{^+\})$, the existence of an expression $suff_{F,e}$ in $\mathcal{N}(F)$ such that, for every graph G, $suff_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$.

The second step in our general proof strategy requires finding an expression e' in $\mathcal{N}(F)$ such that, for every graph G satisfying $suff_{F,e}(G) = \emptyset$, $e'(G) \neq \emptyset$ if and only if $e(G) \neq \emptyset$. (As explained before, we may then conclude that e is equivalent to $suff_{F,e}(G) \cup e'$ at the level of Boolean queries.)

For that purpose, we need to know some information on how a graph G satisfying $suff_{F,e}(G) = \emptyset$ looks like.

For our purpose, we extend the notion of directed acyclic graph (DAG).

Definition 5 An *extended directed acyclic graph* (EDAG) is a (not necessarily connected) DAG to which self-loops may be added provided each path in the DAG contains at most one node with a self-loop. The DAG obtained from an EDAG by removing all self-loops (but not the nodes in which these self-loops occur) is called the *underlying* DAG. The *depth* of an EDAG is the depth of the underlying DAG, i.e., the maximal length of a path in that DAG.

We now have the following.

Lemma 6 *Let m be a nonzero natural number, and let G be a graph such that $\pi_1(R^m) \circ \pi_2(R^m) \circ di \circ \pi_1(R^m) \circ \pi_2(R^m)(G) = \emptyset$. Then G is an EDAG of depth at most $2m$.*

Proof If $\pi_1(R^m) \circ \pi_2(R^m) \circ di \circ \pi_1(R^m) \circ \pi_2(R^m)(G) = \emptyset$, then it is the case that, for any two sequences of nodes $v_{-m}, \ldots, v_{-1}, v_0, v_1, \ldots, v_m$ and $w_{-m}, \ldots, w_{-1}, w_0, w_1, \ldots, w_m$ in G such that, for $i = -m, \ldots, m-1$, $(v_i, v_{i+1}) \in R$ and $(w_i, w_{i+1}) \in R$, we have that $v_0 = w_0$ (cf. the proof of Lemma 5). Clearly, this is not the case if G contains either one loop of length at least two; or two self-loops; or a non-selfintersecting path of length at least $2m+1$. Hence, G is an EDAG of depth at most $2m$. □

Notice that G being an EDAG of depth at most $2m$ is not a sufficient condition for the expression in Lemma 6 to evaluate to the empty set. For instance, an EDAG may contain more than one self-loop in total (at most one on each path in the underlying DAG). Also, a DAG (which is a special case of an EDAG) of depth $2m$ may contain two paths of length $2m$ of which the middle nodes do not coincide. Hence, G being an EDAG of depth at most $2m$ is only a necessary condition for $\pi_1(R^m) \circ \pi_2(R^m) \circ di \circ \pi_1(R^m) \circ \pi_2(R^m)(G) = \emptyset$. For our purposes, however, this is all we need.

We are now ready to bootstrap Lemma 6, as follows.

Proposition 6 *Let $F \subseteq \{\pi, di\}$ be a set of nonbasic features, and let e be an expression in $\mathcal{N}(F \cup \{^+\})$. Let G be a graph such that $suff_{F,e}(G) = \emptyset$. Then G is an EDAG of depth at most $2|e|$.*

Proof Obviously, $R^{|e|}(G) = \emptyset$ implies that $\pi_1(R^{|e|}) \circ \pi_2(R^{|e|}) \circ di \circ \pi_1(R^{|e|}) \circ \pi_2(R^{|e|})(G) = \emptyset$. Furthermore, $R^{|e|}(G) \circ di \circ R^{|e|}(G) = \emptyset$ implies that $\pi_2(R^{|e|}) \circ di \circ \pi_1(R^{|e|})(G) = \emptyset$, which in turn also implies that $\pi_1(R^{|e|}) \circ \pi_2(R^{|e|}) \circ di \circ \pi_1(R^{|e|}) \circ \pi_2(R^{|e|})(G) = \emptyset$. Proposition 6 now follows from Lemma 6. □

Now assume that we are given an expression e in $\mathcal{N}(\pi, di, ^+)$ and an EDAG G of depth at most $2|e|$. The remainder of this paper is concerned with proving that there exists a nonzero natural number D depending only on e such that $e(G) = \emptyset$ if and only if $e'(G) = \emptyset$, where e' is obtained from e by exhaustively replacing subexpressions of the form f^+ by $\bigcup_{i=1}^{D} f^i$.

Notice that this expression is in $\mathcal{N}(F)$, $F \subseteq \{\pi, di\}$, whenever e is in $\mathcal{N}(F \cup \{^+\})$. Hence, there is no need to treat the cases $F = \emptyset$, $F = \{\pi\}$ and $F = \{di\}$ separately.

To achieve our goal, we intend to show (Proposition 15) that there exists a nonzero natural number D such that, for every EDAG G of depth at most $2|e|$, and for every node v of G, there exists a subgraph G_v of G containing v which has at most D nodes

and satisfies the following property: there exists a node w for which $(v, w) \in e(G)$ if and only if there exists a node w' in G_v for which $(v, w') \in e(G_v)$. To see that this property is sufficient for our purposes, assume first that $e(G) = \emptyset$. Then $e'(G) = \emptyset$, since, by construction, $e'(G) \subseteq e(G)$. Therefore, assume next that $e(G) \neq \emptyset$. Then, for some nodes v and w of G, $(v, w) \in e(G)$. Hence, there exists a node w' in G_v such that $(v, w') \in e(G_v)$. Since G_v has at most D nodes, $e(G_v) = e'(G_v)$. It follows that $e'(G_v) \neq \emptyset$. By Lemma 1, (2), $e'(G_v) \subseteq e'(G)$, and, hence, we also have that $e'(G) \neq \emptyset$.

In the remaining sections, we shall establish that such subgraphs G_v exist.

8 Expressions with conditionals

In the previous section, we have set ourselves the goal of finding a nonzero natural number D, only depending on the given expression e in $\mathscr{N}(\pi, di, ^+)$, such that, for every EDAG G of depth at most $2|e|$, and for every node v of G, there exists a subgraph G_v of G containing v which has at most D nodes and satisfies the following property: there exists a node w for which $(v, w) \in e(G)$ if and only if there exists a node w' in G_v for which $(v, w') \in e(G_v)$.

To simplify our task, we shall first of all take advantage of Proposition 1, and work with traces of the original expression e rather than e itself. In this we way, we can work with union-free expressions not containing transitive closure.

These expressions, however, can still contain projection subexpressions, which makes them still too complicated for the inductive arguments we envisage to prove the results that will lead to our goal. Therefore, we shall "abbreviate" the projection subexpressions at the outermost level in the given expression by symbols which we consider as additional, ad-hoc, features of the language. In this way, these subexpressions become "black boxes" of which we can ignore the precise syntax and, to some extent, also the semantics.

More formally, we introduce *conditionals* as additional, nonbasic atomic features. At the syntactic level, a conditional is an expression denoted by some symbol, say c. The semantics of c is given by a mapping (often left implicit) that associates to each directed graph G a set $c(G)$ of pairs of identical elements of G. Hence, $c(G) \subseteq id(G)$. This also explains the name: $(v, v) \in c(G)$ means that node v "satisfies" c in G. In this paper, we shall use conditionals to abbreviate projection subexpressions. (Hence, the mapping defining the semantics of such a conditional is described by a projection subexpression.)

Notice that the notions of *subexpression* (Definition 3) and *trace expression* (Definition 4) can be extended naturally to languages $\mathscr{N}(F)$ where F is a set of nonbasic features including conditionals. It suffices to treat the conditionals as in Case 1 of the respective definitions.

Example 4 Consider the expression $(R \circ \pi_1((R^3 \circ di \circ \pi_2(R^2) \circ R)^+))^+ \circ R^2$. If we associate a conditional c_1 to the projection subexpression $\pi_1((R^3 \circ di \circ \pi_2(R^2) \circ R)^+)$, the expression can be rewritten as $(R \circ c_1)^+ \circ R^2$, i.e., the projection has formally been eliminated. While the original expression is in $\mathscr{N}(\pi, di, ^+)$, the resulting expression is in $\mathscr{N}(c_1, di, ^+)$. Hence, every trace of $(R \circ c_1)^+ \circ R^2$ is a union-free expression in $\mathscr{N}(c_1, di)$, i.e., an expression which is also projection-free and does not contain transitive closure.

Of course, at some point the projection subexpressions will have to be reintroduced to bootstrap the results for projection-free expressions with conditionals to the desired result about the original expression. We explain here in very general terms how this can be done. Therefore, notice that, if c is a conditional the semantics of which is described by a projection subexpression of the form $\pi_1(f)$ or $\pi_2(f)$ at the outermost level of the original expression e, then the maximal nesting depth of projections in f is at least one less than the maximal nesting depth of projections in e. This observation is at the core of an inductive argument that will allow the reintroduction of projection to obtain the targeted results.

Example 5 Consider again the expression $(R \circ \pi_1((R^3 \circ di \circ \pi_2(R^2) \circ R)^+))^+ \circ R^2$. Notice that the maximal nesting depth of projection is 2 in this expression. In Example 4, we have associated to this expression in $\mathscr{N}(\pi, di, ^+)$ the expression $(R \circ c_1)^+ \circ R^2$ in $\mathscr{N}(c_1, di, ^+)$, where c_1 is a conditional the semantics of which is described by $\pi_1((R^3 \circ di \circ \pi_2(R^2) \circ R)^+)$. The operand of this projection, $(R^3 \circ di \circ \pi_2(R^2) \circ R)^+$, is still an expression in $\mathscr{N}(\pi, di, ^+)$, but the maximal nesting depth of projection in this expression is 1 instead of 2. In this way, we can build an inductive argument based on the maximal nesting depth of projection.

Alternatively, if we look at this process from an iterative point of view, we can replace in a second iteration the projection subexpression $\pi_2(R^2)$ in $(R^3 \circ di \circ \pi_2(R^2) \circ R)^+$ by a conditional c_2, yielding the expression $(R^3 \circ di \circ c_2 \circ R)^+$ in $\mathscr{N}(c_2, di, ^+)$. The operand R^2 of the projection subexpression $\pi_2(R)$ describing the semantics of c_2, finally, is projection-free.

So, for Sections 9–12, we introduce a finite set of conditionals $\Gamma = \{c_1, \ldots, c_p\}$, and consider the language $\mathscr{N}(\Gamma, di, ^+)$, as well as some of its sublanguages. In Section 13, we will link the conditionals in Γ to the projection subexpressions in the expression under consideration to obtain the results that yield the goal set out in the beginning of this Section.

9 Line patterns and graph patterns

Let $\Gamma = \{c_1, \ldots, c_p\}$ be a finite set of conditionals. A useful property of union-free expressions in $\mathscr{N}(\Gamma, di)$ is that the presence of a particular pair of nodes of a graph in the output of the expression applied to the graph can be rephrased as the existence of a particular homomorphism from a chain-like directed graph, representing the expression, into the graph.

More concretely, let f be a union-free expression in $\mathscr{N}(\Gamma, di)$. We shall associate a *line pattern* $\mathbf{L}(f)$ with f. This line pattern is a chain-like directed graph in which each edge is labeled with either "R" or "di" and each node is labeled by a (possibly empty) set of conditionals. In addition, each line pattern has one *source node*, labeled **s**, and one *target node*, labeled **t**, which may coincide. The precise, inductive, definition is given in Fig. 2.

From a straightforward inductive argument, we can derive the following result.

Proposition 7 *Let $\Gamma = \{c_1, \ldots, c_p\}$ be a finite set of conditionals and let f be a union-free expression in $\mathscr{N}(\Gamma, di)$, and let G be a graph. There exist nodes v and w in G such*

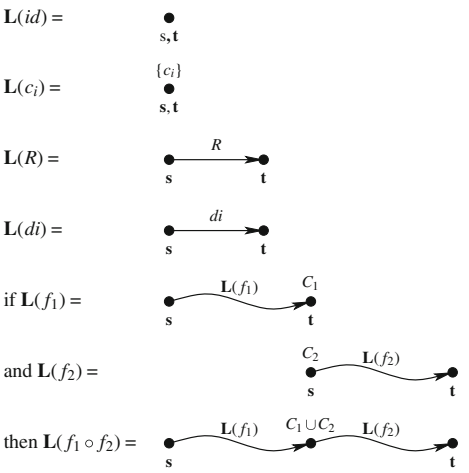


Fig. 2 Definition of the line pattern $\mathbf{L}(f)$ of a union-free expression in $\mathcal{N}(\Gamma, di)$. For the atomic expressions, we only show the node label explicitly if it is a nonempty set of conditionals. For composition, we only show the node label for the nodes at which the line patterns of the subexpressions are joined, as the labels of the other nodes remain unchanged

that $(v, w) \in f(G)$ *if and only if there exists a homomorphism* h *from* $\boldsymbol{L}(f)$ *to* G *such that* $h(\boldsymbol{s}) = v$ *and* $h(\boldsymbol{t}) = w$, *with* $\boldsymbol{s}$ *and* $\boldsymbol{t}$ *the source and target nodes of* $\boldsymbol{L}(f)$.

Line patterns are special cases of *graph patterns*. A graph pattern is a directed graph in which each edge is labeled with either "R" or "di" and each node is labeled by a (possibly empty) set of conditionals. At least one node is marked as source, and at least one node is marked as target.

Let $\mathbf{P}$ be a graph pattern, and let G be a directed graph. A mapping h from the nodes of $\mathbf{P}$ to the nodes of G is called a *homomorphism* from $\mathbf{P}$ to G if

1. for each node $\mathbf{v}$ of $\mathbf{P}$, all the conditionals by which $\mathbf{v}$ is labeled are satisfied by $h(\mathbf{v})$ in G;
2. for each edge $(\mathbf{v}, \mathbf{w})$ of $\mathbf{P}$ labeled by "R," $(h(\mathbf{v}), h(\mathbf{w}))$ is an edge of G; and
3. for each edge $(\mathbf{v}, \mathbf{w})$ of $\mathbf{P}$ labeled by "di," $h(\mathbf{v}) \neq h(\mathbf{w})$.

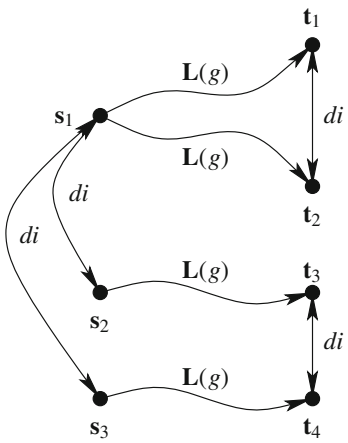


Fig. 3 Example of a graph pattern that is not a line pattern. In this graph pattern, $\mathbf{L}(g)$ represents the line pattern of some expression g

An example of a graph pattern that is not a line pattern can be seen in Fig. 3.

Notice that we use boldface characters for the nodes of line and graph patterns to distinguish them clearly from the nodes of the input graph.

General graph patterns will be put to use in Section 11 to construct, given an expression e in $\mathscr{N}(\Gamma, di, ^+)$, a natural number m, an EDAG G of depth at most m, and a node v of G, a sequence of subgraphs of G. The number of nodes of these subgraphs can be bounded by natural numbers depending only on p, m, and e. One of these subgraphs will turn out to be the subgraph G_v mentioned at the end of Section 7, for appropriate choices of the conditionals and their semantics, and for $m = 2|e|$.

10 Normalizing trace expressions

In Section 9, we associated line patterns with union-free expression in $\mathscr{N}(\Gamma, di)$, with $\Gamma = \{c_1, \ldots, c_p\}$ a set of conditionals.

Trivially, each union-free expression in $\mathscr{N}(\Gamma, di)$ is a trace expression of itself,[5] and each trace expression of an expression in $\mathscr{N}(\Gamma, di, ^+)$ is a union-free expression in $\mathscr{N}(\Gamma, di)$. Hence, the set of union-free expressions in $\mathscr{N}(\Gamma, di)$ is precisely the set of all trace expressions of expressions in $\mathscr{N}(\Gamma, di, ^+)$. By the same argument, the set of all union-free expressions in $\mathscr{N}(\Gamma)$ is precisely the set of all trace expressions of expressions in $\mathscr{N}(\Gamma, ^+)$.

However, not all trace expressions will be useful for our purposes, and, in addition, trace expressions may contain a lot of redundancy. Therefore, we define sublanguages of the union-free expressions in $\mathscr{N}(\Gamma)$, respectively $\mathscr{N}(\Gamma, di)$, parametrized by a parameter n. They consist of the so-called *n-normal trace expressions* of $\mathscr{N}(\Gamma, ^+)$, respectively $\mathscr{N}(\Gamma, di, ^+)$ (Definitions 6 and 7, below). The term "normal" will be justified in Proposition 8.

Definition 6 Let $\Gamma = \{c_1, \ldots, c_p\}$ be a set of conditionals, and let $n \geq 0$. An expression g in $\mathscr{N}(\Gamma)$ is n-normal if (1) g is union-free, (2) $|g| \leq n$, and (3) a subexpression of g consisting only of "*id*," conditionals, and composition is either "*id*" or does not contain "*id*" and contains at most one occurrence of every conditional.

Observe that, for all n, "*id*" is always n-normal. We denote the n-normal expressions of $\mathscr{N}(\Gamma)$ by $\mathscr{N}_n^{\text{norm}}(\Gamma)$.

Definition 7 Let $\Gamma = \{c_1, \ldots, c_p\}$ be a set of conditionals, and let $n \geq 0$. An expression f in $\mathscr{N}(\Gamma, di)$ is n-normal if it is of the form $g_1 \circ di \circ g_2 \circ di \circ \cdots \circ g_{k-1} \circ di \circ g_k$, with $g_1, \ldots, g_k \in \mathscr{N}_n^{\text{norm}}(\Gamma)$.

Hence, all n-normal expressions of $\mathscr{N}(\Gamma)$ are also n-normal expressions of $\mathscr{N}(\Gamma, di)$. We denote the n-normal expressions of $\mathscr{N}(\Gamma, di)$ by $\mathscr{N}_n^{\text{norm}}(\Gamma, di)$.

We now define the set $\mathscr{T}_n^{\text{norm}}(e)$ of the *n-normal trace expressions* of e as the set of all expressions in $\mathscr{N}_n^{\text{norm}}(\Gamma, di)$ for which there exists an equivalent expression

[5] Actually, the only one.

in $\mathscr{T}(e)$ at the level of path queries. The following result links expressions in $\mathscr{N}(\Gamma, di, {}^+)$ to normal trace expressions in $\mathscr{N}(\Gamma, di)$ in the context of an EDAG of bounded depth, and hence justifies the term "normal."

Proposition 8 *Let e be an expression in $\mathscr{N}(\Gamma, di, {}^+)$, let $m \geq 0$, and let G be an EDAG of depth at most m. Then, there exists a number M only dependent on e and m such that, for all nodes v and w of G, $(v, w) \in e(G)$ if and only if there exists an M-normal trace expression f in $\mathscr{T}_M^{\text{norm}}(e)$ for which $(v, w) \in f(G)$.*

Proof By Proposition 1, $(v, w) \in e(G)$ if and only if there exists a trace expression f in $\mathscr{T}(e)$ for which $(v, w) \in f(G)$. In particular, this settles the "if." For the "only if," assume that f is a trace expression of minimal length for which $(v, w) \in f(G)$. It remains to show that we can "normalize" f.

By Proposition 7, there exists a homomorphism h from $\mathbf{L}(f)$ to G with $h(\mathbf{s}) = v$ and $h(\mathbf{t}) = w$. Now consider a "di"-free subexpression g of f of maximal length. Consider the path in G defined by $h(\mathbf{L}(g))$. Notice that the length of this path, measured in the DAG underlying G, is at most m. Hence, if this path does not contain a node with a self-loop, then there are also at most m occurrences of the symbol "R" in g. Thus assume that on this path there is a node with a self-loop, and hence precisely one (cf. Definition 5), say, z.

Now, assume there is a subexpression f_1 of f that is a trace of k subsequent iterations of e_1, with e_1^+ a transitive-closure subexpression of e, such that the following conditions are satisfied:

1. the first "R" symbol in g mapped by h to the self-loop in z corresponds in e to an "R" symbol in the first of the k iterations under consideration of e_1;
2. the last "R" symbol in g mapped by h to the self-loop in z corresponds in e to an "R" symbol in the last of the k iterations under consideration of e_1.

Consequently, f_1 need not be a maximal subexpression of f that is a trace of consecutive iterations of e_1 in e.

Suppose, for the sake of contradiction, that $k > 2$. Let g_1 be the subexpression of g corresponding to the k iterations under consideration of e_1 in e, except for the first and the last one.[6] Obviously, h maps all nodes of the subpattern $\mathbf{L}(g_1)$ of $\mathbf{L}(f)$ to z. Hence, we can omit g_1 from f and still retain a trace expression $\hat{f}$ for which h is a homomorphism mapping $\mathbf{L}(\hat{f})$ to G such that $h(\mathbf{s}) = v$ and $h(\mathbf{t}) = w$, contradicting our assumption that f has minimal length. Hence, $k \leq 2$.

Now, let $\hat{e}$ be the expression in $\mathscr{N}(\pi, di)$ obtained from e by recursively substituting each subexpression of the form e_1^+ in e by $e_1 \cup e_1^2$. By the above argument, it follows that the minimal subexpression of f containing all "R" symbols of g mapped to the self-loop in z by h is also a subexpression of a trace of $\hat{e}$. Hence, the number of these "R" symbols is bounded by $|\hat{e}|$, the length of $\hat{e}$. Notice that this number solely depends on e. The number of "R" symbols of g *not* mapped to the self-loop in z is bounded by m, by the same argument as before. We may therefore conclude that, in all cases, the total number of "R" symbols in g is bounded by $M := m + |\hat{e}|$.

[6] In other words, g_1 is a trace of $k - 2$ iterations of e_1.

Finally, we can rewrite f as $f' = g_1 \circ di \circ g_2 \circ di \circ \cdots \circ g_{n-1} \circ di \circ g_n$, with $g_1, \ldots, g_n \in \mathscr{N}(\Gamma)$, by inserting "*id*" primitives where needed. By our previous argument, the number of "R" symbols in g_i, $1 \leq i \leq n$, is bounded by M. Without loss of generality, we may also assume that subexpressions of f' consisting solely of "*id*," conditionals, and composition are either "*id*" or do not contain "*id*," and contain each conditional at most once, by removing superfluous occurrences of "*id*" and repetitions of conditionals. Clearly, $f' \in \mathscr{T}_M^{\text{norm}}(e)$, and $(v, w) \in f'(G)$. □

Proposition 8 becomes interesting in conjunction with Proposition 9, below.

Proposition 9 *Let $\Gamma = \{c_1, \ldots, c_p\}$ be a set of conditionals, and let $n \geq 0$. Then,*

1. *the number of atomic subexpressions of an expression of $\mathscr{N}_n^{\text{norm}}(\Gamma)$ can be bounded by a number depending only on p and n; and*
2. *the number of expressions in $\mathscr{N}_n^{\text{norm}}(\Gamma)$ is finite, and can be bounded by a number depending only on p and n.*

Proof First, consider item (1). Let g be an expression of $\mathscr{N}_n^{\text{norm}}(\Gamma)$. We know that g contains "R" at most n times. Unless g is "*id*," we know that before the first "R," in between subsequent "R"s, and after the last "R," we can have a sequence of conditionals, in which each of these occurs at most once. Hence, the number of atomic subexpressions of g is at most $\max(1, n + (n+1)p)$. Item (2) now immediately follows. □

11 Canonical subgraphs

Given a set of conditionals $\Gamma = \{c_1, \ldots, c_p\}$, a natural number n, a directed graph G, and a node v of G, we shall define a sequence of so-called n-canonical subgraphs $G_0^v, G_1^v, G_2^v, \ldots$ of G of order $0, 1, 2, \ldots$ with respect to Γ. (In the notation, we shall leave Γ and n implicit.)

In doing so, we have two opposite concerns:

1. For some appropriately chosen set of conditionals Γ, some $n \geq 0$, and some order $i \geq 0$, G_i^v, the n-canonical subgraph of order i of G, will be the subgraph G_v of G mentioned at the end of Section 7 satisfying $e(G_v) \neq \emptyset$ if and only if $e(G) \neq \emptyset$. In order to work towards that goal, we must define $G_0^v, G_1^v, G_2^v, \ldots$ sufficiently large to ensure that we can simulate the behavior of e on G on these subgraphs of G.
2. For our proof strategy to work, it is at the same time important that there is a bound on the number of nodes of G_v that only depends on e, and not on G or v. Therefore, we may define $G_0^v, G_1^v, G_2^v, \ldots$ not too large either. In particular, we shall ensure that the number of nodes of each of these n-canonical subgraphs of G with respect to Γ depends only on its order and on p and n.

Balancing these two concerns is the motivation behind the definitions that follow.

We start by defining G_0^v.

Thereto, let g be an expression in $\mathscr{N}_n^{\text{norm}}(\Gamma)$. We define $\mathfrak{P}(g)$ to be the set of graph patterns that can be obtained from $\mathbf{L}(g)$ in the following way:

1. Start with one, two, three, or four pairwise disjoint copies of $\mathbf{L}(g)$.

2. Optionally, merge some of the source nodes of these copies.
3. Optionally, merge some of the target nodes of these copies.
4. Optionally, connect some of the remaining source nodes by "*di*" edges.
5. Optionally, connect some of the remaining target nodes by "*di*" edges.

Observe that the line pattern $\mathbf{L}(g)$ itself is always in $\mathfrak{P}(g)$.

Figure 3 shows a more representative example of a graph pattern that belongs to $\mathfrak{P}(g)$.

Now, let $\mathbf{P}$ be a graph pattern in $\mathfrak{P}(g)$, and let v be a node of G. With $\mathbf{P}$, we associate a minimal (in number of elements) set $\mathfrak{H}_v(\mathbf{P})$ of homomorphisms from $\mathbf{P}$ to G satisfying the following conditions:

1. if there exists a homomorphism from $\mathbf{P}$ to G, then $\mathfrak{H}_v(\mathbf{P}) \neq \emptyset$;
2. if, for an arbitrary node $\mathbf{v}$ of $\mathbf{P}$, there exist two homomorphisms from $\mathbf{P}$ to G mapping $\mathbf{v}$ to different nodes of G, then $\mathfrak{H}_v(\mathbf{P})$ contains two homomorphisms from $\mathbf{P}$ to G mapping $\mathbf{v}$ to different nodes of G;
3. if $\mathbf{P}$ has a single source node $\mathbf{s}$ and there exists a homomorphism from $\mathbf{P}$ to G mapping $\mathbf{s}$ to v, then $\mathfrak{H}_v(\mathbf{P})$ contains such a homomorphism; and
4. if $\mathbf{P}$ has a single target node $\mathbf{t}$ and there exists a homomorphism from $\mathbf{P}$ to G mapping $\mathbf{t}$ to v, then $\mathfrak{H}_v(\mathbf{P})$ contains such a homomorphism.

For a good understanding, we first observe the following.

- Given $\mathbf{P}$, G, and v, we *choose* a minimal set of homomorphisms $\mathfrak{H}_v(\mathbf{P})$ satisfying the above conditions. In other words, it is to be expected that, in general, several minimal sets of homomorphisms satisfy the above conditions. From these, we pick one arbitrarily, and denote it by $\mathfrak{H}_v(\mathbf{P})$.
- The definition of $\mathfrak{H}_v(\mathbf{P})$ refers explicitly to v only if $\mathbf{P}$ has either a single source node, or a single target node, or both. In all other cases, we may therefore choose $\mathfrak{H}_v(\mathbf{P})$ independent of v.

From the definition of the set $\mathfrak{H}_v(\mathbf{P})$, we can derive the following useful property.

Proposition 10 *Let $\mathbf{P}$ be a graph pattern, let $\mathbf{v}$ be a node of $\mathbf{P}$, and let v and z be nodes of G. If there exists a homomorphism from $\mathbf{P}$ to G that does not map $\mathbf{v}$ to z, then $\mathfrak{H}_v(\mathbf{P})$ contains such a homomorphism.*

Proof For the sake of contradiction, suppose that all homomorphisms from $\mathbf{P}$ to G in $\mathfrak{H}_v(\mathbf{P})$ map $\mathbf{v}$ to z. Since, by assumption, there is also a homomorphism from $\mathbf{P}$ to G that does not map $\mathbf{v}$ to z, there exist two homomorphisms from $\mathbf{P}$ to G mapping $\mathbf{v}$ to different nodes of G. By definition, $\mathfrak{H}_v(\mathbf{P})$ contains two homomorphisms from $\mathbf{P}$ to G mapping $\mathbf{v}$ to different nodes of G, a contradiction with our initial assumption. □

We are now ready to define G_0^v, the n-canonical subgraph of order 0:

$$G_0^v = \bigcup_{g \in \mathcal{N}_n^{\mathrm{norm}}(\Gamma)} \bigcup_{\mathbf{P} \in \mathfrak{P}(g)} \bigcup_{h \in \mathfrak{H}_v(\mathbf{P})} h(\mathbf{P}).$$

In the above formula, $h(\mathbf{P})$ is the subgraph of G with set of nodes $\{h(\mathbf{v}) \mid \mathbf{v}$ is a node of $\mathbf{P}\}$ and set of edges $\{(h(\mathbf{v}), h(\mathbf{w})) \mid (\mathbf{v}, \mathbf{w})$ is an R-labeled edge of $\mathbf{P}\}$. The n-canonical subgraph of order 0 is then defined as a union of some of these

subgraphs, where this union must be interpreted componentwise, i.e., the set of nodes and the set of edges of this union are the union of the sets of nodes and the union of the sets of edges of the subgraphs involved.

At this point, several aspects of the definition of the n-canonical subgraph of order 0 have been left unexplained, in particular,

- the definition of the set of graph patterns $\mathfrak{P}(g)$ for $g \in \mathscr{N}_n^{\text{norm}}(\Gamma)$, and, more specifically, why up to four copies of the line pattern $\mathbf{L}(g)$ are allowed in such a graph pattern; and
- the definition of the set of homomorphisms $\mathfrak{H}_v(\mathbf{P})$ for $\mathbf{P} \in \mathfrak{P}(g)$.

The only answer we can give at this point is that these definitions are tailored to make some of the key results in Section 12 work (in particular, Lemma 8), as is explained in that section. The essence is that, given an n-normal trace expression f in $\mathscr{T}_n^{\text{norm}}(e)$ and a homomorphism h from $\mathbf{L}(f)$ to G, we wish to show via an inductive process that there also exists such a homomorphism of which the image is fully contained in one of the n-canonical subgraphs of order 0. As argued before, we must ensure on the one hand that the n-canonical subgraphs of order 0 are sufficiently large for this process to work, but, on the other hand, we must also ensure that their size can be bounded by a bound not depending on the size of G (see Proposition 12, below). Obtaining this delicate balance is what led to the definition above.

However, the results in Section 12 are only a first albeit important step in proving the collapse of $\mathscr{N}(\pi, di, {}^+)$ to $\mathscr{N}(\pi, di)$ at the level of Boolean queries on unlabeled graphs. Indeed, the conditionals represent projection conditions, and the operands of these projections may in turn contain projection conditions.

To accommodate this, we next define G_1^v, G_2^v, ..., the n-canonical subgraphs of G of order $1, 2, \ldots$ with respect to Γ, with the following recursive rule. For $i > 0$,

$$G_i^v = G_0^v \cup \left(\bigcup_{w \text{ node of } G_0^v} G_{i-1}^w \right).$$

We note the following properties of these n-canonical subgraphs.

Proposition 11 *Let $\Gamma = \{c_1, \ldots, c_p\}$ be a set of conditionals, let $n \geq 0$, and let G be a directed graph. Consider the n-canonical subgraphs of G with respect to Γ. For every node v of G, and for $i = 0, 1, 2, \ldots$, we have that* (1) *G_i^v is a subgraph of G, and* (2) *G_i^v is a subgraph of G_{i+1}^v.*

Proof By construction, G_0^v is a subgraph of G for every node v of G. This is the basis for a straightforward induction argument to show that, for $i = 1, 2, \ldots$, G_i^v is also a subgraph of G. This settles the first statement.

The second statement can also be shown by induction. The base case, that G_0^v is a subgraph of G_1^v, follows immediately from the definition of G_1^v. As induction hypothesis, assume that, for some $i > 0$, we have already established, for all nodes v of G, that G_{i-1}^v is a subgraph of G_i^v. As induction step, we now show that G_i^v is a subgraph of G_{i+1}^v. We have that

$$G_i^v = G_0^v \cup \left(\bigcup_{w \text{ node of } G_0^v} G_{i-1}^w \right).$$

By the induction hypothesis, we know that, for each node w of G_0^v, G_{i-1}^w is a subgraph of G_i^w. Hence, G_i^v is a subgraph of

$$G_0^v \cup \left(\bigcup_{w \text{ node of } G_0^v} G_i^w \right),$$

which by definition is G_{i+1}^v. □

The n-canonical subgraphs of G of higher order are put to use in Section 13, more in particular in Proposition 15.

For the remainder of the exposition, it is important that we can also provide bounds on the sizes of the n-canonical subgraphs of G with respect to Γ.

Proposition 12 *Let $\Gamma = \{c_1, \ldots, c_p\}$ be a set of conditionals, let $n \geq 0$, and let G be a directed graph. For every node v of G, and for $i = 0, 1, 2, \ldots$, the number of nodes in G_i^v, the n-canonical subgraph of G of order i with respect to Γ, can be bounded by a number depending only on p, n, and i.*

Proof Let us first focus on G_0^v. From Proposition 9, it can easily be inferred that both the number of graph patterns involved in the construction of G_0^v and the number of nodes they contain are bounded by numbers depending only on p and n. Let us call these numbers P and N, respectively. Given a graph pattern $\mathbf{P}$, the number of homomorphisms from $\mathbf{P}$ to G in $\mathfrak{H}_v(\mathbf{P})$ is bounded by $2N$. To show this, we construct a set of homomorphisms from $\mathbf{P}$ to G, as follows. We start with the empty set. Then, for each node $\mathbf{v}$ of $\mathbf{P}$, we add one or two homomorphisms to this set according to the rules below.

1. If all homomorphisms from $\mathbf{P}$ to G map $\mathbf{v}$ to the same node of G, then select one such homomorphism arbitrarily. Regardless of whether or not $\mathbf{v}$ may be the unique source or target node of $\mathbf{P}$, we see that conditions 2–4 of the definition of $\mathfrak{H}_v(\mathbf{P})$ are satisfied for that particular node.
2. Otherwise, not all homomorphisms from $\mathbf{P}$ to G map $\mathbf{v}$ to the same node of G. If, in addition, no homomorphism from $\mathbf{P}$ to G maps $\mathbf{v}$ to v, then select two such homomorphisms arbitrarily provided they map $\mathbf{v}$ to different nodes of G. Regardless of whether or not $\mathbf{v}$ may be the unique source or target node of $\mathbf{P}$, we see that conditions 2–4 of the definition of $\mathfrak{H}_v(\mathbf{P})$ are satisfied for that particular node.
3. Otherwise, there is a homomorphism from $\mathbf{P}$ to G mapping $\mathbf{v}$ to v and there is a homomorphism from $\mathbf{P}$ to G not mapping $\mathbf{v}$ to v. Then select arbitrarily one homomorphism from the first category and one homomorphism from the second category. Regardless of whether or not $\mathbf{v}$ may be the unique source or target node of $\mathbf{P}$, we see that conditions 2–4 of the definition of $\mathfrak{H}_v(\mathbf{P})$ are satisfied for that particular node.

By construction, the set of homomorphisms from $\mathbf{P}$ to G obtained in this way contains at most $2N$ members. It clearly satisfies condition 1 as well as conditions 2–4 for all nodes of $\mathbf{P}$. Since $\mathfrak{H}_v(\mathbf{P})$ is such a set of minimal size, we may finally conclude that $\mathfrak{H}_v(\mathbf{P})$ contains at most $2N$ homomorphisms.

Consequently, the number of nodes of G_0^v is bounded by $2N^2P$, which depends only on p and n. Let us denote this last number as B. Then, a straightforward induction reveals that, for all $i \geq 0$, the number of nodes of G_i^v is bounded by $B(B^i + B^{i-1} + \cdots + B + 1)$. □

12 The key result

In Propositions 13 and 14 below, we tie together the notions introduced in the previous sections. They are the key results on which the main result of this paper hinges. We shall indeed bootstrap these two results in Section 13 to the desired result, i.e., the collapse of $\mathscr{N}(F \cup \{^+\})$ to $\mathscr{N}(F)$ at the level of Boolean queries for $F \subseteq \{\pi, di\}$. The present section is devoted to proving Propositions 13 and 14.

Proposition 13 *Let Γ be a set of conditionals, let $m \geq 0$, and let e be an expression in $\mathscr{N}(\Gamma, di, ^+)$. Then, there exists $M \geq 0$ depending only on m and e such that, for every EDAG G of depth at most m, and for every node v of G, if there exists a node w in G such that $(v, w) \in e(G)$, then there exist an M-normal trace expression f in $\mathscr{T}_M^{\text{norm}}(e)$ and a homomorphism h from $\mathbf{L}(f)$ to G such that $h(\boldsymbol{s}) = v$ and $h(\mathbf{L}(f))$ is contained in G_0^v, with $\boldsymbol{s}$ the source node of the line pattern $\mathbf{L}(f)$ and G_0^v the M-canonical subgraph of G of order 0 with respect to Γ.*

Proposition 14 *Let Γ be a set of conditionals, let $m \geq 0$, and let e be an expression in $\mathscr{N}(\Gamma, di, ^+)$. Let $M \geq 0$ be a number depending only on m and e for which Proposition 13 is satisfied. For every node w of G, if there exists a node v in G such that $(v, w) \in e(G)$, then there exist an M-normal trace expression f in $\mathscr{T}_M^{\text{norm}}(e)$ and a homomorphism h from $\mathbf{L}(f)$ to G such that $h(\boldsymbol{t}) = w$ and $h(\mathbf{L}(f))$ is contained in G_0^w, with $\boldsymbol{t}$ the target node of the line pattern $\mathbf{L}(f)$ and G_0^w the M-canonical subgraph of G of order 0 with respect to Γ.*

It is important to notice here that the homomorphism h in Propositions 13 and 14 need not be a homomorphism from $\mathbf{L}(f)$ to G_0^v, respectively G_0^w. If this were the case, then, by Proposition 7, $(v, w) \in e(G_0^v)$, respectively $(v, w) \in e(G_0^w)$, and we would have found the subgraphs G_v of G we set out to find at the end of Section 7 to achieve the second step of our proof strategy.

However, this is in general *not* the case. Indeed, let $\mathbf{v}$ be a node of $\mathbf{L}(f)$ which is labeled with a conditional c. It is very well possible that $(h(\mathbf{v}), h(\mathbf{v})) \in c(G)$ while $(h(\mathbf{v}), h(\mathbf{v})) \notin c(G_0^v)$ (respectively, $c(G_0^w)$), even if $h(\mathbf{v})$ is a node of G_0^v (respectively, G_0^w).

As mentioned in Section 8, we are interested in the case where the conditionals are in fact projection conditions. These have the property of being monotone. In order to ensure that the conditions are satisfied, we will therefore have to extend the subgraph G_0^v, and that is where the canonical subgraphs of higher order come into play, at the final stage of our development, in Section 13.

Because of the strong analogy between both Propositions, we shall focus here on the proof of Proposition 13. At the end of this section, we show that Proposition 13 follows from Propositions 7 and 8, provided the following lemma holds.

Lemma 7 *Let Γ be a set of conditionals, let $n \geq 0$, let f be an n-normal expression in $\mathcal{N}_n^{\text{norm}}(\Gamma, di)$, let G be a directed graph, and let v be a node of G. If there exists a homomorphism h from $\boldsymbol{L}(f)$ to G such that $h(\boldsymbol{s}) = v$, with $\boldsymbol{s}$ the source node of $\boldsymbol{L}(f)$, then there exists a homomorphism h' from $\boldsymbol{L}(f)$ to G such that $h'(\boldsymbol{s}) = v$ and $h'(\boldsymbol{L}(f))$ is contained in G_0^v, with G_0^v the n-canonical subgraph of G of order 0 with respect to Γ.*

If we write $f = g_1 \circ di \circ g_2 \circ di \circ \cdots \circ g_{k-1} \circ di \circ g_k$, with $g_1, \ldots, g_k \in \mathcal{N}_n^{\text{norm}}(\Gamma)$, a sensible way to prove Lemma 7 is to consider the expressions $f_i = g_1 \circ di \circ \cdots \circ di \circ g_i$, for $i = 1, \ldots, k$, and to prove the Lemma by induction on i. The basis of the induction, $i = 1$, is straightforward from the construction of the subgraph G_0^v. Thus suppose that, for $1 < i \leq k$, we have established the existence of a homomorphism h'_{i-1} from $\mathbf{L}(f_{i-1})$ to G such that $h'_{i-1}(\mathbf{s}) = v$ ($\mathbf{s}$ being the source node of $\mathbf{L}(f_{i-1})$) and $h'_{i-1}(\mathbf{L}(f_{i-1}))$ is contained in G_0^v. We would like to extend h'_{i-1} to a homomorphism h'_i from $\mathbf{L}(f_i)$ to G such that $h'_i(\mathbf{L}(f_i))$ is contained in G_0^v. Thus, consider $\mathbf{L}(g_i)$, which is a subpattern of $\mathbf{L}(f_i)$. The restriction of h to the nodes of $\mathbf{L}(g_i)$ is a homomorphism from $\mathbf{L}(g_i)$ to G. Hence, $\mathfrak{H}_v(\mathbf{L}(g_i))$ contains a homomorphism $h_{\mathbf{L}(g_i)}$ from $\mathbf{L}(g_i)$ to G, and, by construction of G_0^v, $h_{\mathbf{L}(g_i)}(\mathbf{L}(g_i))$ is contained in G_0^v. Now, let $\mathbf{t}_{i-1}$ be the target node of $\mathbf{L}(f_{i-1})$ and $\mathbf{s}_i$ the source node of $\mathbf{L}(g_i)$. If $h'_{i-1}(\mathbf{t}_{i-1}) \neq h_{\mathbf{L}(g_i)}(\mathbf{s}_i)$, the extension is straightforward. However, we cannot exclude that $h'_{i-1}(\mathbf{t}_{i-1}) = h_{\mathbf{L}(g_i)}(\mathbf{s}_i)$. If this is the case, it may even be so that $h_{\mathbf{L}(g_i)}$ is the *only* homomorphism mapping $\mathbf{L}(g_i)$ to G. Then, we cannot even consider an alternative homomorphism from $\mathbf{L}(g_i)$ to G to make our extension strategy work.

Luckily, we can avoid this pitfall by proving a slightly stronger statement.

Lemma 8 *Let Γ be a set of conditionals, let $n \geq 0$, let f be an n-normal expression in $\mathcal{N}_n^{\text{norm}}(\Gamma, di)$, let G be a directed graph, and let v be a node of G. Let G_0^v be the n-canonical subgraph of order 0 with respect to Γ. Then,*

1. *if there exists a homomorphism h from $\boldsymbol{L}(f)$ to G such that $h(\boldsymbol{s}) = v$, with $\boldsymbol{s}$ the source node of $\boldsymbol{L}(f)$, then there also exists a homomorphism h' from $\boldsymbol{L}(f)$ to G such that $h'(\boldsymbol{s}) = v$ and $h(\boldsymbol{L}(f))$ is contained in G_0^v;*
2. *in addition, if there exist homomorphisms h_1 and h_2 from $\boldsymbol{L}(f)$ to G such that $h_1(\boldsymbol{s}) = h_2(\boldsymbol{s}) = v$ and $h_1(\boldsymbol{t}) \neq h_2(\boldsymbol{t})$, with $\boldsymbol{s}$ and $\boldsymbol{t}$ the source and target nodes of $\boldsymbol{L}(f)$, then there also exist homomorphisms h'_1 and h'_2 from $\boldsymbol{L}(f)$ to G such that $h'_1(\boldsymbol{s}) = h'_2(\boldsymbol{s}) = v$, $h'_1(\boldsymbol{t}) \neq h'_2(\boldsymbol{t})$, and $h'_1(\boldsymbol{L}(f))$ and $h'_2(\boldsymbol{L}(f))$ are both contained in G_0^v.*

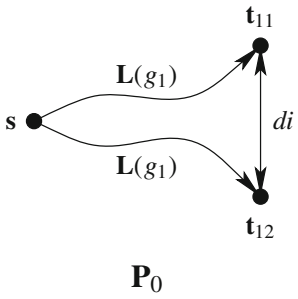
Proof By assumption, $f = g_1 \circ di \circ g_2 \circ di \circ \cdots \circ g_{n-1} \circ di \circ g_k$, with $g_1, \ldots, g_k \in \mathcal{N}_n^{\text{norm}}(\Gamma)$. We prove by induction on $i = 1, \ldots, k$ that the statement of the Lemma holds for all $f_i := g_1 \circ di \circ \cdots \circ di \circ g_i$.

Base case: $i = 1$. If there exists a homomorphism h_1 from $\mathbf{L}(g_1)$ to G with $h_1(\mathbf{s}) = v$, $\mathbf{s}$ being the source node of $\mathbf{L}(g_1)$, then $\mathfrak{H}_v(\mathbf{L}(g_1))$ contains a homomorphism $h_{\mathbf{L}(g_1)}$ from $\mathbf{L}(g_1)$ to G with $h_{\mathbf{L}(g_1)}(\mathbf{s}) = v$. Clearly, $h'_1 := h_{\mathbf{L}(g_1)}$ is the desired homomorphism the image of which is contained in G_0^v.

If, in addition, there exist homomorphisms h_{11} and h_{12} from $\mathbf{L}(g_1)$ to G with $h_{11}(\mathbf{s}) = h_{12}(\mathbf{s}) = v$ and $h_{11}(\mathbf{t}_1) \neq h_{12}(\mathbf{t}_1)$, $\mathbf{t}_1$ being the target node of $\mathbf{L}(g_1)$, then the graph pattern $\mathbf{P}_0$ (see Fig. 4) can be mapped homomorphically to G.

Hence, $\mathfrak{H}_v(\mathbf{P}_0)$ contains a homomorphism $h_{\mathbf{P}_0}$ from $\mathbf{P}_0$ to G such that $h_{\mathbf{P}_0}(\mathbf{s}) = v$ and $h_{\mathbf{P}_0}(\mathbf{t}_{11}) \neq h_{\mathbf{P}_0}(\mathbf{t}_{12})$. By construction of G_0^v, $h_{\mathbf{P}_0}(\mathbf{P}_0)$ is contained in G_0^v. The

Fig. 4 The graph pattern $\mathbf{P}_0$



restrictions of $h_{\mathbf{P}_0}$ to both isomorphic copies of $\mathbf{L}(g_1)$ in $\mathbf{P}_0$, respectively, are the desired homomorphisms h'_{11} and h'_{12} the images of which are contained in G_0^v.

Induction hypothesis. Assume that, for some i, $1 < i \leq k$, the statement of the Lemma holds for f_{i-1}.

Induction step. We show that the statement of the Lemma also holds for f_i. For this purpose, we distinguish two cases.

1. *There exist homomorphisms $h_{(i-1)1}$ and $h_{(i-1)2}$ from $\mathbf{L}(f_{i-1})$ to G such that $h_{(i-1)1}(\mathbf{s}) = h_{(i-1)2}(\mathbf{s}) = v$ and $h_{(i-1)1}(\mathbf{t}_{i-1}) \neq h_{(i-1)2}(\mathbf{t}_{i-1})$, with $\mathbf{s}$ and $\mathbf{t}_{i-1}$ the source and target nodes of $\mathbf{L}(f_{i-1})$.* From the induction hypothesis, we may deduce that there exist homomorphisms $h'_{(i-1)1}$ and $h'_{(i-1)2}$ from $\mathbf{L}(f_{i-1})$ to G such that $h'_{(i-1)1}(\mathbf{s}) = h'_{(i-1)2}(\mathbf{s}) = v$, $h'_{(i-1)1}(\mathbf{t}_{i-1}) \neq h'_{(i-1)2}(\mathbf{t}_{i-1})$, and $h'_{(i-1)1}(\mathbf{L}(f_{i-1}))$ and $h'_{(i-1)2}(\mathbf{L}(f_{i-1}))$ are both contained in G_0^v.
   We now show that f_i satisfies both statements of the Lemma.

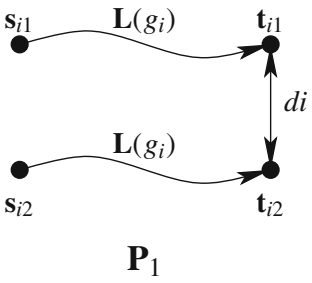
   (a) Assume there exists a homomorphism h_i from $\mathbf{L}(f_i)$ to G such that $h_i(\mathbf{s}) = v$, with $\mathbf{s}$ the source node of $\mathbf{L}(f_i)$.[7] By considering the restriction of h_i to $\mathbf{L}(g_i)$), we see that the line pattern $\mathbf{L}(g_i)$ can be mapped homomorphically to G. Hence, $\mathfrak{H}_v(\mathbf{L}(g_i)) \neq \emptyset$. Let $h_{\mathbf{L}(g_i)}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{L}(g_i))$. By construction, $h_{\mathbf{L}(g_i)}(\mathbf{L}(g_i))$ is contained in G_0^v. Since $h'_{(i-1)1}(\mathbf{t}_{i-1}) \neq h'_{(i-1)2}(\mathbf{t}_{i-1})$, at least one of them is different from $h_{\mathbf{L}(g_i)}(\mathbf{s}_i)$, with $\mathbf{s}_i$ the source node of $\mathbf{L}(g_i)$. Without loss of generality, assume that $h'_{(i-1)1}(\mathbf{t}_{i-1}) \neq h_{\mathbf{L}(g_i)}(\mathbf{s}_i)$. Then, we can extend the homomorphism $h'_{(i-1)1}$ (from $\mathbf{L}(f_{i-1})$ to G) to the desired homomorphism h'_i (from $\mathbf{L}(f_i)$ to G), as follows:

$$h'_i(\mathbf{v}) = \begin{cases} h'_{(i-1)1}(\mathbf{v}) & \text{if } \mathbf{v} \text{ is a node of } \mathbf{L}(f_{i-1}); \text{ and} \\ h_{\mathbf{L}(g_i)}(\mathbf{v}) & \text{if } \mathbf{v} \text{ is a node of } \mathbf{L}(g_i). \end{cases}$$

   It is easily verified that h'_i is a homomorphism from $\mathbf{L}(f_i)$ to G such that $h'_i(\mathbf{s}) = v$ and $h'_i(\mathbf{L}(f_i))$ is contained in G_0^v.

[7] We denote the source nodes of both $\mathbf{L}(f_{i-1})$ and $\mathbf{L}(f_i)$ by the same symbol $\mathbf{s}$, which is justified because the former line pattern is a prefix of the latter line pattern.

Fig. 5 The graph pattern $\mathbf{P}_1$

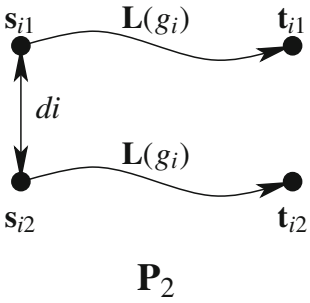


(b) Additionally, assume there exist homomorphisms h_{i1} and h_{i2} from $\mathbf{L}(f_i)$ to G such that $h_{i1}(\mathbf{s}) = h_{i2}(\mathbf{s}) = v$ and $h_{i1}(\mathbf{t}_i) \neq h_{i2}(\mathbf{t}_i)$, with $\mathbf{s}$ and $\mathbf{t}_i$ the source and target nodes of $\mathbf{L}(f_i)$.
By considering $h_{i1}(\mathbf{L}(g_i))$ and $h_{i2}(\mathbf{L}(g_i))$ together, we see that the graph pattern $\mathbf{P}_1$ shown in Fig. 5 can be mapped homomorphically to G.
Hence, $\mathfrak{H}_v(\mathbf{P}_1) \neq \emptyset$. Let $h_{\mathbf{P}_1}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{P}_1)$. By construction, $h_{\mathbf{P}_1}(\mathbf{P}_1)$ is contained in G_0^v. Since $h'_{(i-1)1}(\mathbf{t}_{i-1}) \neq h'_{(i-1)2}(\mathbf{t}_{i-1})$, at least one of them is different from $h_{\mathbf{P}_1}(\mathbf{s}_{i1})$. Without loss of generality, assume that $h'_{(i-1)1}(\mathbf{t}_{i-1}) \neq h_{\mathbf{P}_1}(\mathbf{s}_{i1})$. We now extend $h'_{(i-1)1}$ (from $\mathbf{L}(f_{i-1})$ to G) with the restriction of $h_{\mathbf{P}_1}$ to the "top" copy of $\mathbf{L}(g_i)$ in $\mathbf{P}_1$ to a homomorphism h'_{i1} (from $\mathbf{L}(f_i)$ to G), in the same way as above. By construction, $h'_{i1}(\mathbf{s}) = v$, $h'_{i1}(\mathbf{t}_i) = h_{\mathbf{P}_1}(\mathbf{t}_{i1})$, and $h'_{i1}(\mathbf{L}(f_i))$ is contained in G_0^v. Similarly, we construct a homomorphism h'_{i2} from $\mathbf{L}(f_i)$ to G such that $h'_{i2}(\mathbf{s}) = v$, $h'_{i2}(\mathbf{t}_i) = h_{\mathbf{P}_1}(\mathbf{t}_{i2})$, and $h'_{i2}(\mathbf{L}(f_i))$ is contained in G_0^v. It now suffices to observe that $h_{\mathbf{P}_1}(\mathbf{t}_{i1}) \neq h_{\mathbf{P}_1}(\mathbf{t}_{i2})$ to obtain that $h'_{i1}(\mathbf{t}_i) \neq h'_{i2}(\mathbf{t}_i)$.

2. All homomorphisms h_{i-1} from $\mathbf{L}(f_{i-1})$ to G for which $h_{i-1}(\mathbf{s}) = v$ map $\mathbf{t}_{i-1}$ to the same node of G, where $\mathbf{s}$ and $\mathbf{t}_{i-1}$ are the source and target nodes of $\mathbf{L}(f_{i-1})$.
We show that, also in this case, f_i satisfies both statements of the Lemma.

(a) Assume there exists a homomorphism h_i from $\mathbf{L}(f_i)$ to G such that $h_i(\mathbf{s}) = v$, with $\mathbf{s}$ the source node of $\mathbf{L}(f_i)$. By considering the restriction of h_i to $\mathbf{L}(f_{i-1})$, we may deduce from the induction hypothesis that there exists a homomorphism h'_{i-1} from $\mathbf{L}(f_{i-1})$ to G such that $h'_{i-1}(\mathbf{s}) = v$ and $h'_{i-1}(\mathbf{L}(f_{i-1}))$ is contained in G_0^v. Let $\mathbf{t}_{i-1}$ be the target node of $\mathbf{L}(f_{i-1})$. It follows from the initial assumption for this case that $h_i(\mathbf{t}_{i-1}) = h'_{i-1}(\mathbf{t}_{i-1})$. Let us call this node z.
By considering the restriction of h_i to $\mathbf{L}(g_i)$, we see that $\mathbf{L}(g_i)$ can be mapped homomorphically to G. Hence, $\mathfrak{H}_v(\mathbf{L}(g_i)) \neq \emptyset$. Let $h_{\mathbf{L}(g_i)}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{L}(g_i))$. By construction, $h_{\mathbf{L}(g_i)}(\mathbf{L}(g_i))$ is contained in G_0^v. If $h_{\mathbf{L}(g_i)}(\mathbf{s}_i) \neq z$, with $\mathbf{s}_i$ the source node of $\mathbf{L}(g_i)$, we can extend h'_{i-1} to a homomorphism h'_i from $\mathbf{L}(f_i)$ to G such that $h'_i(\mathbf{s}) = v$ and $h'_i(\mathbf{L}(f_i))$ is contained in G_0^v, as before.
Therefore, assume now that $h_{\mathbf{L}(g_i)}(\mathbf{s}_i) = z$. Obviously, $h_i(\mathbf{s}_i) \neq z$. By considering $h_{\mathbf{L}(g_i)}(\mathbf{L}(g_i))$ together with $h_i(\mathbf{L}(g_i))$, we see that the graph pattern $\mathbf{P}_2$ shown in Fig. 6 can be mapped homomorphically to G.
Hence, $\mathfrak{H}_v(\mathbf{P}_2) \neq \emptyset$. Let $h_{\mathbf{P}_2}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{P}_2)$. By construction, $h_{\mathbf{P}_2}(\mathbf{P}_2)$ is contained in G_0^v. Since $h_{\mathbf{P}_2}(\mathbf{s}_{i1}) \neq h_{\mathbf{P}_2}(\mathbf{s}_{i2})$, at least one of them is

Fig. 6 The graph pattern $\mathbf{P}_2$



different from z. Without loss of generality, assume that $h_{\mathbf{P}_2}(\mathbf{s}_{i1}) \neq z$. We now extend h'_{i-1} (from $\mathbf{L}(f_{i-1})$ to G) with the restriction of $h_{\mathbf{P}_2}$ to the "top" copy of $\mathbf{L}(g_i)$ in $\mathbf{P}_1$ to a homomorphism h'_i (from $\mathbf{L}(f_i)$ to G), as before. By construction, $h'_i(\mathbf{s}) = v$, and $h'_i(\mathbf{L}(f_i))$ is contained in G_0^v.

(b) Additionally, assume there exist homomorphisms h_{i1} and h_{i2} from $\mathbf{L}(f_i)$ to G such that $h_{i1}(\mathbf{s}) = h_{i2}(\mathbf{s}) = v$ and $h_{i1}(\mathbf{t}_i) \neq h_{i2}(\mathbf{t}_i)$, with $\mathbf{s}$ and $\mathbf{t}_i$ the source and target nodes of $\mathbf{L}(f_i)$. If $\mathbf{t}_{i-1}$ is the target node of $\mathbf{L}(f_{i-1})$, then it follows from our initial assumption for this case that $h_{i1}(\mathbf{t}_{i-1}) = h_{i2}(\mathbf{t}_{i-1})$. Let us call this node z. By considering the restriction of h_{i1} (or h_{i2}, for that matter) to $\mathbf{L}(f_{i-1})$, we may deduce from the induction hypothesis that there exists a homomorphism h'_{i-1} from $\mathbf{L}(f_{i-1})$ to G such that $h'_{i-1}(\mathbf{s}) = v$ and $h'_{i-1}(\mathbf{L}(f_{i-1}))$ is contained in G_0^v. By our assumption, $h'_{i-1}(\mathbf{t}_{i-1}) = z$.[8]

By considering $h_{i1}(\mathbf{L}(g_i))$ and $h_{i2}(\mathbf{L}(g_i))$ together, we see that the graph pattern $\mathbf{P}_1$ shown in Fig. 5 can be mapped homomorphically to G. Hence, $\mathfrak{H}_v(\mathbf{P}_1) \neq \emptyset$. Let $h_{\mathbf{P}_1}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{P}_1)$. By construction, $h_{\mathbf{P}_1}(\mathbf{P}_1)$ is contained in G_0^v. If both $h_{\mathbf{P}_1}(\mathbf{s}_{i1}) \neq z$ and $h_{\mathbf{P}_1}(\mathbf{s}_{i2}) \neq z$, we can extend h'_{i-1} (from $\mathbf{L}(f_{i-1})$ to G) with the restrictions of $h_{\mathbf{P}_1}$ to the "top" and "bottom" copies of $\mathbf{L}(g_i)$ in $\mathbf{P}_1$ to homomorphisms h'_{i1} and h'_{i2} (from $\mathbf{L}(f_i)$ to G), as before. By construction, $h'_{i1}(\mathbf{s}) = h'_{i2}(\mathbf{s}) = v$, $h'_{i1}(\mathbf{t}_i) \neq h'_{i2}(\mathbf{t}_i)$, and $h'_{i1}(\mathbf{L}(f_i))$ and $h'_{i2}(\mathbf{L}(f_i))$ are both contained in G_0^v.

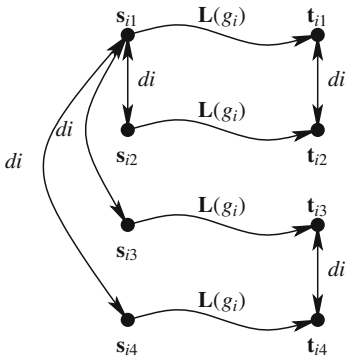
Therefore, suppose now that, e.g., $h_{\mathbf{P}_1}(\mathbf{s}_{i1}) = z$ and $h_{\mathbf{P}_1}(\mathbf{s}_{i2}) \neq z$.[9] By considering $h_{\mathbf{P}_1}(\mathbf{P}_1)$ together with $h_{i1}(\mathbf{L}(g_i))$ and $h_{i2}(\mathbf{L}(g_i))$, we see that the graph pattern $\mathbf{P}_3$ shown in Fig. 7 can be mapped homomorphically to G, with $\mathbf{s}_{i1}$ being mapped to z, and all other source nodes being mapped to other nodes of G. In particular, $\mathbf{s}_{i2}$ is *not* mapped to z.

Hence, by Proposition 10, $\mathfrak{H}_v(\mathbf{P}_3)$ contains a homomorphism $h_{\mathbf{P}_3}$ from $\mathbf{P}_3$ to G for which $h_{\mathbf{P}_3}(\mathbf{s}_{i2}) \neq z$. By construction, $h_{\mathbf{P}_3}(\mathbf{P}_3)$ is contained in G_0^v.

[8] Notice that it is not useful to consider h_{i1} and h_{i2} together in this argument, as the respective homomorphisms $h'_{(i-1)1}$ and $h'_{(i-1)2}$ that can be derived from it using the induction hypothesis may well coincide. Indeed, all we know about these homomorphisms is that $h'_{(i-1)1}(\mathbf{s}) = h'_{(i-1)2}(\mathbf{s}) = v$, $h'_{(i-1)1}(\mathbf{t}_{i-1}) = h'_{(i-1)2}(\mathbf{t}_{i-1}) = z$, and that $h'_{(i-1)1}(\mathbf{L}(f_{i-1}))$ and $h'_{(i-1)2}(\mathbf{L}(f_{i-1}))$ are both contained in G_0^v, all properties that do *not* distinguish $h'_{(i-1)1}$ and $h'_{(i-1)2}$.

[9] The case that $h_{\mathbf{P}_1}(\mathbf{s}_{i1}) \neq z$ and $h_{\mathbf{P}_1}(\mathbf{s}_{i2}) = z$ is of course completely symmetric and will therefore not be treated separately.

Fig. 7 The graph pattern $\mathbf{P}_3$



If also $h_{\mathbf{P}_3}(\mathbf{s}_{i1}) \neq z$, we can extend h'_{i-1} (from $\mathbf{L}(f_{i-1})$ to G) with the restrictions of $h_{\mathbf{P}_3}$ to the two "upper" copies of $\mathbf{L}(g_i)$ in $\mathbf{P}_3$ to homomorphisms h'_{i1} and h'_{i2} (from $\mathbf{L}(f_i)$ to G), as before. By construction, $h'_{i1}(\mathbf{s}) = h'_{i2}(\mathbf{s}) = v$, $h'_{i1}(\mathbf{t}_i) \neq h'_{i2}(\mathbf{t}_i)$, and $h'_{i1}(\mathbf{L}(f_i))$ and $h'_{i2}(\mathbf{L}(f_i))$ are both contained in G_0^v.
If, however, $h_{\mathbf{P}_3}(\mathbf{s}_{i1}) = z$, then $h_{\mathbf{P}_3}(\mathbf{s}_{i3}) \neq z$ and $h_{\mathbf{P}_3}(\mathbf{s}_{i4}) \neq z$, and we can achieve the same result as above, this time using the two "lower" copies of $\mathbf{L}(g_i)$ in $\mathbf{P}_3$.
It remains to consider the case where $h_{\mathbf{P}_1}(\mathbf{s}_{i1}) = h_{\mathbf{P}_1}(\mathbf{s}_{i2}) = z$. In this case, the homomorphism $h_{\mathbf{P}_1}$ can readily be transformed into a homomorphism $h_{\mathbf{P}_4}$ from the graph pattern $\mathbf{P}_4$, shown in Fig. 8, to G.
By considering $h_{\mathbf{P}_4}(\mathbf{P}_4)$ together with $h_{i1}(\mathbf{L}(g_i))$ and $h_{i2}(\mathbf{L}(g_i))$, we see that the graph pattern $\mathbf{P}_5$ shown in Fig. 9 can be mapped homomorphically to G.
Hence, $\mathfrak{H}_v(\mathbf{P}_5) \neq \emptyset$. Let $h_{\mathbf{P}_5}$ be a homomorphism in $\mathfrak{H}_v(\mathbf{P}_5)$. By construction, $h_{\mathbf{P}_5}(\mathbf{P}_5)$ is contained in G_0^v.
If $h_{\mathbf{P}_5}(\mathbf{s}_{i1}) \neq z$, we can extend h'_{i-1} (from $\mathbf{L}(f_{i-1})$ to G) with the restrictions of $h_{\mathbf{P}_5}$ to the two "upper" copies of $\mathbf{L}(g_i)$ in $\mathbf{P}_5$ (both originating in $\mathbf{s}_{i1}$) to homomorphisms h'_{i1} and h'_{i2} (from $\mathbf{L}(f_i)$ to G), as before. By construction, $h'_{i1}(\mathbf{s}) = h'_{i2}(\mathbf{s}) = v$, $h'_{i1}(\mathbf{t}_i) \neq h'_{i2}(\mathbf{t}_i)$, and $h'_{i1}(\mathbf{L}(f_i))$ and $h'_{i2}(\mathbf{L}(f_i))$ are both contained in G_0^v.
If, however, $h_{\mathbf{P}_5}(\mathbf{s}_{i1}) = z$, then $h_{\mathbf{P}_5}(\mathbf{s}_{i2}) \neq z$ and $h_{\mathbf{P}_5}(\mathbf{s}_{i3}) \neq z$, and we can achieve the same result as above, this time using the two "lower" copies of $\mathbf{L}(g_i)$ in $\mathbf{P}_5$.

Fig. 8 The graph pattern $\mathbf{P}_4$

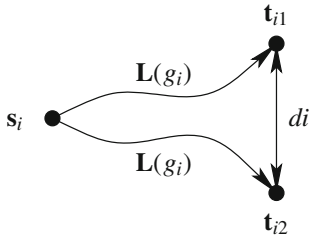
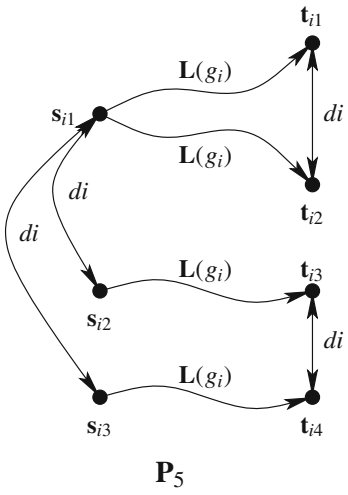


Fig. 9 The graph pattern $\mathbf{P}_5$



So, in both cases, we were able to complete the induction step successfully, which concludes the proof. □

Of course, Lemma 7 immediately follows from Lemma 8.

The proof of Proposition 13 (and hence also that of Proposition 14) is now straightforward.

Proof (Proposition 13) Let Γ be a set of conditionals, let $m \geq 0$, and let e be an expression in $\mathscr{N}(\Gamma, di, ^+)$. Let G be an EDAG of depth at most m. Let M be a number for which Proposition 8 is satisfied. Now, let v be a node of G. If there exists a node w in G such that $(v, w) \in e(G)$, then, by Proposition 8, there exists an M-normal trace expression f in $\mathscr{T}_M^{\text{norm}}(e)$ such that $(v, w) \in f(G)$. By Proposition 7, there exists a homomorphism h from $\mathbf{L}(f)$ to G with $h(\mathbf{s}) = v$ and $h(\mathbf{t}) = w$, with $\mathbf{s}$ and $\mathbf{t}$ the source and target nodes of $\mathbf{L}(f)$. Finally, by Lemma 7, there also exists a homomorphism h' from $\mathbf{L}(f)$ to G such that $h'(\mathbf{s}) = v$ and $h'(\mathbf{L}(f))$ is contained in G_0^v, with G_0^v the M-canonical subgraph of G of order 0 with respect to Γ. □

13 The collapse result

We are now ready to deal with expressions in $\mathscr{N}(\pi, di, ^+)$ and bootstrap Propositions 13 and 14 by considering that conditionals stand for projection subexpressions. We recall that the homomorphism h in the statements of these propositions is a homomorphism from $\mathbf{L}(f)$ to G such that $h(\mathbf{s}) = v$ and $h(\mathbf{L}(f))$ is contained in G_0^v, but not necessarily a homomorphism from $\mathbf{L}(f)$ to G_0^v, the reason being that a node of G_0^v satisfying a particular conditional within G does not have to satisfy the same conditional within G_0^v. Using that the conditionals stand for projection subexpressions, and using the monotonicity of the projection operator, we are able to establish that G_0^v can be extended to a higher-order canonical subgraph of G, say G_i^v, such that h is also a homomorphism from $\mathbf{L}(f)$ to G_i^v. Only then will we be able to

conclude that $(v, h(\mathbf{t})) \in e(G_i^v)$, with $\mathbf{t}$ the target node of $\mathbf{L}(f)$, and can we complete our argument.

For this purpose, we first define the *nesting depth of projection* of an expression e in $\mathscr{N}(\pi, di, ^+)$, denoted $\mathrm{depth}_\pi(e)$. Remember that we already used this notion informally in Section 8 to motivate the introduction of conditionals as abbreviations for projection subexpressions. Here, we define that notion formally:

1. if e is in $\mathscr{N}(di, ^+)$, then $\mathrm{depth}_\pi(e) = 0$;
2. $\mathrm{depth}_\pi(\pi_1(e)) = \mathrm{depth}_\pi(\pi_2(e)) = \mathrm{depth}_\pi(e) + 1$;
3. $\mathrm{depth}_\pi(e_1 \cup e_2) = \max(\mathrm{depth}_\pi(e_1), \mathrm{depth}_\pi(e_2))$;
4. $\mathrm{depth}_\pi(e_1 \circ e_2) = \max(\mathrm{depth}_\pi(e_1), \mathrm{depth}_\pi(e_2))$; and
5. $\mathrm{depth}_\pi(e^+) = \mathrm{depth}_\pi(e)$.

In the sequel, we shall refer to nesting depth of projection as "depth" for short.

We are now going to bootstrap Propositions 13 and 14 by remembering that conditionals represent projection subexpressions, and, in doing so, linking the depth of the expression to the order of the canonical subgraph required for this purpose.

Lemma 9 *Let $\delta \geq 0$, and let e be an expression of depth δ in $\mathscr{N}(\pi, di, ^+)$. Let $\Pi(e) = \{\pi_{i_1}(f_1), \ldots, \pi_{i_p}(f_p)\}$, $i_1, \ldots, i_p \in \{1, 2\}$, be the set of all projection subexpressions of e. Let $\Gamma = \{c_1, \ldots, c_p\}$, where, for $j = 1, \ldots, p$, the conditional c_j has the same semantics as the projection subexpression $\pi_{i_j}(f_j)$ of e. Let $m \geq 0$. Then, for every expression $e' \in \mathscr{N}(\Pi(e), di, ^+)$[10] with depth $\delta' \leq \delta$, there exists $M \geq 0$ depending only on m and e such that, for every EDAG G of depth at most m, we have the following:*

1. *for every node v in G, if there exists a node w in G such that $(v, w) \in e'(G)$, then there exists a node w' in $G_{\delta'}^v$ such that $(v, w') \in e'(G_{\delta'}^v)$, with $G_{\delta'}^v$ the M-canonical subgraph of G of order δ' with respect to Γ; and*
2. *for every node w in G, if there exists a node v in G such that $(v, w) \in e'(G)$, then there exists a node v' in $G_{\delta'}^w$ such that $(v', w) \in e'(G_{\delta'}^w)$, with $G_{\delta'}^v$ the M-canonical subgraph of G of order δ' with respect to Γ.*

Proof Let M be a number for which Propositions 13 and 14 are satisfied. We prove both claims of Lemma 9 by a simultaneous induction on the depth δ' of the expression e'.

Base case: $\delta' = 0$. In that case, e' is an expression of $\mathscr{N}(di, ^+)$ (and hence also of $\mathscr{N}(\Gamma, di, ^+)$). By Proposition 13, there exist an M-normal trace expression f' in $\mathscr{T}_M^{\mathrm{norm}}(e')$ and a homomorphism h from $\mathbf{L}(f')$ to G such that $h(\mathbf{s}) = v$ and $h(\mathbf{L}(f'))$ is contained in G_0^v, with $\mathbf{s}$ the source node of the line pattern $\mathbf{L}(f')$ and G_0^v the M-canonical subgraph of G of order 0 with respect to Γ. Since there are no conditionals at play here, h is also a homomorphism from $\mathbf{L}(f')$ to G_0^v. Hence, if $\mathbf{t}$ is the target node of $\mathbf{L}(f')$, then $(v, h(\mathbf{t})) \in f'(G_0^v) \subseteq e'(G_0^v)$. This settles the first claim. The second claim is proved in a completely analogous way, using Proposition 14 instead of Proposition 13.

[10] We clarify that expressions in $\mathscr{N}(\Pi(e), di, ^+)$ may contain projection subexpressions, provided they are in $\Pi(e)$.

Induction hypothesis. Assume that both claims of Lemma 9 have been established for all expressions in $\mathscr{N}(\Pi(e), di, {}^+)$ with depth strictly smaller than δ'.

Induction step. Let e' be an arbitrary expression in $\mathscr{N}(\Pi(e), di, {}^+)$ with depth δ'. We prove that both claims of Lemma 9 hold for e'.

We start with the first claim. Thus, assume that, for some node v in G, there exists a node w in G such that $(v, w) \in e'(G)$. Without loss of generality, let $\pi_{i_1}(f_1), \ldots, \pi_{i_{p'}}(f_{p'})$, $p' \leq p$, be all projection subexpressions at the outermost level in e', i.e., all projection subexpressions of e' that do *not* occur within another projection subexpression of e'. Now, let e'' be the expression in $\mathscr{N}(\Gamma, di, {}^+)$ obtained from e' by replacing each projection subexpression $\pi_{i_j}(f_j)$, $1 \leq j \leq p'$, by the conditional c_j. By construction, e' and e'' are equivalent at the level of path queries. Hence, $(v, w) \in e''(G)$. It now follows from Proposition 13 that there exist an M-normal trace expression $f'' \in \mathscr{T}_M^{\text{norm}}(e'')$ and a homomorphism h from $\mathbf{L}(f'')$ to G such that $h(\mathbf{s}) = v$ and $h(\mathbf{L}(f''))$ is contained in G_0^v, with $\mathbf{s}$ the source node of the line pattern $\mathbf{L}(f'')$ and G_0^v the M-canonical subgraph of G of order 0 with respect to Γ. By Proposition 11, (2), $h(\mathbf{L}(f''))$ is contained in $G_{\delta'}^v$, the M-canonical subgraph of G of order δ'. We next show that h is actually a homomorphism from $\mathbf{L}(f'')$ to $G_{\delta'}^v$.

Thereto, let $\mathbf{v}$ be a node of $\mathbf{L}(f'')$ labeled with some conditional c_j, $1 \leq j \leq p'$, and let $h(\mathbf{v}) = z$. Since h is a homomorphism from $\mathbf{L}(f'')$ to G, $(z, z) \in c_j(G) = \pi_{i_j}(f_j)(G)$. First consider the case that $i_j = 1$. Then, there is a node u in G such that $(z, u) \in f_j(G)$. Notice that f_j is in $\mathscr{N}(\Pi(e), di, {}^+)$ and that $\delta_j := \text{depth}_\pi(f_j) < \delta'$. By the first claim of the induction hypothesis, there exists a node u' in $G_{\delta_j}^z$, the M-canonical subgraph of G of order δ_j with respect to Γ—which, by Proposition 11, is contained in $G_{\delta'-1}^z$, the M-canonical subgraph of G of order $\delta' - 1$ with respect to Γ—such that $(z, u') \in f_j(G_{\delta_j}^z) \subseteq f_j(G_{\delta'-1}^z)$, by the monotonicity of f_j. Since, by definition, $G_{\delta'-1}^z$ is a subgraph of $G_{\delta'}^v$, and again by the monotonicity of f_j, $(z, u') \in f_j(G_{\delta'}^v)$, and hence $(z, z) \in \pi_1(f_j)(G_{\delta'}^v) = c_j(G_{\delta'}^v)$. In the case that $i_j = 2$, we make a similar reasoning with the second claim of the induction hypothesis to arrive at the same conclusion. In summary, $h(\mathbf{v})$ is a node that does not only satisfy c_j in G, but also in $G_{\delta'}^v$, and hence h is indeed a homomorphism from $\mathbf{L}(f'')$ to $G_{\delta'}^v$. If $\mathbf{t}$ is the target node of $\mathbf{L}(f'')$, then, by Proposition 7, $(v, h(\mathbf{t})) \in f''(G_{\delta'}^v) \subseteq e''(G_{\delta'}^v) = e'(G_{\delta'}^v)$. This concludes the induction step for the first claim of Lemma 9. The induction step for the second claim is completely analogous, using Proposition 14 instead of Proposition 13. □

Since e is in $\mathscr{N}(di, \Pi(e), {}^+)$, Lemma 9 also holds for $e' := e$. Thus, its first claim can be specialized as follows.

Proposition 15 *Let $\delta \geq 0$, and let e be an expression of depth δ in $\mathscr{N}(\pi, di, {}^+)$. Let $\Pi(e) = \{\pi_{i_1}(f_1), \ldots, \pi_{i_p}(f_p)\}$, $i_1, \ldots, i_p \in \{1, 2\}$, be the set of all projection subexpressions of e. Let $\Gamma = \{c_1, \ldots, c_p\}$, where, for $j = 1, \ldots, p$, the conditional c_j has the same semantics as the projection subexpression $\pi_{i_j}(f_j)$ of e. Let $m \geq 0$. Then, there exists $M \geq 0$ depending only on m and e such that, for every EDAG G of depth at most m, and for every node v in G, if there exists a node w in G such that $(v, w) \in e'(G)$, there exists a node w' in G_δ^v such that $(v, w') \in e'(G_\delta^v)$, with G_δ^v the M-canonical subgraph of G of order δ with respect to Γ.*

We are now ready to complete the second step of our proof strategy.

Theorem 3 *Let e be in $\mathscr{N}(\pi, di, {}^+)$ and let $m \geq 0$. Then, there exists e' in $\mathscr{N}(\pi, di)$ such that, for each EDAG G with depth at most m, $e(G) \neq \emptyset$ if and only if $e'(G) \neq \emptyset$.*

Proof Let $\Pi(e) = \{\pi_{i_1}(f_1), \ldots, \pi_{i_p}(f_p)\}, i_1, \ldots, i_p \in \{1, 2\}$, be the set of all projection subexpressions of e. Let $\Gamma = \{c_1, \ldots, c_p\}$, where, for $j = 1, \ldots, p$, the conditional c_j has the same semantics as the projection subexpression $\pi_{i_j}(f_j)$ of e. Let G be an EDAG of depth at most m, and let $M \geq 0$ be a number only depending on m and e for which Proposition 15 is satisfied. For every node v of G, let G^v_δ be the M-canonical subgraph of G of order δ with respect to Γ, with $\delta := \mathrm{depth}_\pi(e)$, the depth of e. By Proposition 12, there exists $D \geq 0$, depending only on p, M, and δ, such that the number of nodes of G^v_δ is bounded by D. In particular, D neither depends on the particular graph G under consideration, nor on the particular node v of G under consideration.

Now, let e' be the expression in $\mathscr{N}(\pi, di)$ obtained from e by exhaustively replacing each subexpression (at any level) of the form f^+ by $\bigcup_{i=1}^{D} f^i$. By monotonicity, $e'(G) \subseteq e(G)$, hence the "if."

We now turn to the "only if." If $e(G) \neq \emptyset$, then there exist nodes v and w in G such that $(v, w) \in e(G)$. By Proposition 15, there exists a node w' in G^v_δ such that $(v, w') \in e(G^v_\delta)$. Since the number of nodes of G^v_δ is bounded by D, it follows that $e(G^v_\delta) = e'(G^v_\delta)$, as there is no point in making more than D iterations for computing a transitive closure. Hence, $(v, w') \in e'(G^v_\delta) \subseteq e'(G)$, by monotonicity and Proposition 11, (1). We may thus conclude that $e'(G) \neq \emptyset$. □

Let $F \subseteq \{\pi, di\}$ be a set of nonbasic features. If e is an expression in $\mathscr{N}(F \cup \{^+\})$, then the expression e' constructed in the proof of Theorem 3 is in $\mathscr{N}(F)$. Hence, we can generalize Theorem 3 as follows.

Corollary 1 *Let $F \subseteq \{\pi, di\}$ be a set of nonbasic features, let e be in $\mathscr{N}(F \cup \{^+\})$, and let $m \geq 0$. Then, there exists e' in $\mathscr{N}(F)$ such that, for each EDAG G with depth at most m, $e(G) \neq \emptyset$ if and only if $e'(G) \neq \emptyset$.*

Observe that the bound D on the size of the subgraphs G^v_δ in the proof of Theorem 3 (cf. also Proposition 12) is of very high complexity in m, as a consequence of which it may require very large graphs G before the difference between G and its subgraphs G^v_δ becomes significant.

Now that we have completed the second step of our proof strategy, we can prove Theorem 1, the main result of this paper.

Proof (Theorem 1) In Theorem 2, we established that, for every graph G, $\mathit{suff}_{F,e}(G) \neq \emptyset$ implies $e(G) \neq \emptyset$, with $\mathit{suff}_{F,e}(G)$ the expression in $\mathscr{N}(F)$ tabulated in Table 1. By Proposition 6, we know that $\mathit{suff}_{F,e}(G) = \emptyset$ implies that G is an EDAG of depth at most $2|e|$. If we now apply Corollary 1 for $m = 2|e|$, we find that there exists an expression e' in $\mathscr{N}(F)$ such that, for every graph G, $\mathit{suff}_{F,e}(G) = \emptyset$ implies that $e(G) \neq \emptyset$ if and only if $e'(G) \neq \emptyset$. Finally, let $\tilde{e} := \mathit{suff}_{F,e} \cup e'$, which is also an expression in $\mathscr{N}(F)$. Let G be an arbitrary graph. If $\mathit{suff}_{F,e}(G) \neq \emptyset$, then $e(G) \neq \emptyset$ and, by construction, also $\tilde{e}(G) \neq \emptyset$. If, on the other hand, $\mathit{suff}_{F,e}(G) = \emptyset$,

then $\tilde{e}(G) = e'(G)$, and hence $e(G) \neq \emptyset$ if and only if $\tilde{e}(G) \neq \emptyset$. Combining both cases, we see that, for every graph G, $e(G) \neq \emptyset$ if and only if $\tilde{e}(G) \neq \emptyset$. □

14 Conclusions and future work

We now have a complete understanding of the impact of adding transitive closure to the relation algebra fragments considered. While it is well-known that transitive closure adds expressive power to all fragments at the level of path queries [3], and the same was established in previous work of the present authors [11] at the level of Boolean queries on labeled graphs (multiple input relations), we have now established, in contrast, that, while adding transitive closure adds expressive power to most relation algebra fragments at the level of Boolean queries on unlabeled graphs (a single input relation), it does *not* add expressive power to $\mathscr{N}(F)$, with F a set of nonbasic features, if and only if $F \subseteq \{\pi, di\}$.

As a side product of our efforts to prove this last result, we have also established a "normal form" for an expression $\tilde{e}$ in $\mathscr{N}(F)$ equivalent to an expression e in $\mathscr{N}(F \cup \{^+\})$, namely $\tilde{e} := \mathit{suff}_{F,e} \cup e'$. The left-hand term of this union only depends on F and $|e|$,[11] and the right-hand term can be obtained from the original expression e by exhaustively replacing subexpressions of the form f^+ by $\bigcup_{i=1}^{D} f^i$, for some bound D only depending on $|e|$.

Towards future work, one may examine this bound D more closely. As observed towards the end of Section 13, D is of very high complexity in $|e|$. As the main goal of this paper was to show the collapse of $\mathscr{N}(F \cup \{^+\})$ to $\mathscr{N}(F)$, we have not attempted to search for the tightest bound possible. From a practical point of view, it would indeed be interesting to find a better bound D, or to identify a reasonably large class of expressions in $\mathscr{N}(F \cup \{^+\})$ for which the bound D could be improved significantly.

Another direction for future research is to investigate similar problems as the ones addressed in this paper, but for other algebras. An operation we did not consider here, for instance, is residuation. Residuation [22] is similar to the standard relational division operation in databases, and corresponds to the set containment join [18]. The present authors have only achieved partial results in this respect, in part also because they have not yet been able to establish the complete Hasse diagram for the relative expressive power of the various fragments of the relation algebra with residuation, even if transitive closure is not considered.

Another direction the authors are currently pursuing is establishing the complete Hasse diagram for the relative expressive power of the various relation algebra fragments, and this both at the levels of path queries and Boolean queries, for the cases where the input graph is a tree. Of course, if it has been established that two fragments are equivalent for general input graphs, they must also be equivalent on trees. However, it is possible that two fragments can be separated for general graphs but not for trees as input. Some results on the expressive power of XPath fragments (e.g., [6, 9, 14, 19, 20, 26]) can readily be used for this purpose, but some relation algebra features do not straightforwardly correspond with XPath operations. Consequently, the results of this study will contribute to a better understanding of

[11] Even stronger, it is a fixed expression in $\mathscr{N}(F \cup \{f\})$, where f is short for $R^{|e|}$.

querying tree-shaped documents, in particular in the case where several documents are involved.

References

1. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann (1999)
2. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison Wesley, Reading, MA (1995)
3. Aho, A.V., Ullman, J.D.: The universality of data retrieval languages. In: POPL, pp. 110–120. ACM (1979)
4. Angles, R., Gutiérrez, C.: Survey of graph database models. ACM Comput. Surv. **40**(1), 1–39 (2008)
5. Baader, F., Calvanese, D., McGuiness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press (2003)
6. Benedikt, M., Fan, W., Kuper, G.M.: Structural properties of XPath fragments. In: Calvanese, D., Lenzerini, M., Motwani, R. (eds.) ICDT. Lecture Notes in Computer Science, vol. 2572, pp. 79–95. Springer (2003)
7. Bizer, C., Heath, T., Berners-Lee, T.: Linked data—the story so far. Int. J. Semantic Web Inf. Syst. **5**(3), 1–22 (2009)
8. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
9. ten Cate, B., Marx, M.: Navigational XPath: calculus and algebra. SIGMOD Rec. **36**(2), 19–26 (2007)
10. Fletcher, G.H.L., Gyssens, M., Leinders, D., Van den Bussche, J., Van Gucht, D., Vansummeren, S., Wu, Y.: The impact of transitive closure on the boolean expressiveness of navigational query languages on graphs. In: Lukasiewicz, T., Sali, A. (eds.) FoIKS. Lecture Notes in Computer Science, vol. 7153, pp. 124–143. Springer (2012)
11. Fletcher, G.H.L., Gyssens, M., Leinders, D., Van den Bussche, J., Van Gucht, D., Vansummeren, S., Wu, Y.: Relative expressive power of navigational querying on graphs. In: Milo, T. (ed.) ICDT, pp. 197–207. ACM (2011)
12. Florescu, D., Levy, A., Mendelzon, A.: Database techniques for the World-Wide Web: a survey. SIGMOD Rec. **27**(3), 59–74 (1998)
13. Franklin, M.J., Halevy, A.Y., Maier, D.: From databases to dataspaces: a new abstraction for information management. SIGMOD Rec. **34**(4), 27–33 (2005)
14. Gyssens, M., Paredaens, J., Van Gucht, D., Fletcher, G.H.L.: Structural characterizations of the semantics of XPath as navigation tool on a document. In: Vansummeren, S. (ed.) PODS, pp. 318–327. ACM (2006)
15. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
16. Heath, T., Bizer, C.: Linked data: evolving the web into a global data space. In: Synthesis Lectures on the Semantic Web: Theory and Technology, vol. 1, 1st edn. Morgan & Claypool Publishers (2011)
17. Maddux, R.D.: Relation Algebras. Elsevier, Amsterdam (2006)
18. Mamoulis, N.: Efficient processing of joins on set-valued attributes. In: Halevy, A.Y., Ives, Z.G., Doan, A. (eds.) SIGMOD Conference, pp. 157–168. ACM (2003)
19. Marx, M.: Conditional XPath. ACM Trans. Database Syst. **30**(4), 929–959 (2005)
20. Marx, M., de Rijke, M.: Semantic characterizations of navigational XPath. SIGMOD Rec. **34**(2), 41–46 (2005)
21. Marx, M., Venema, Y.: Multi-Dimensional Modal Logic. Springer (1997)
22. Pratt, V.R.: Origins of the calculus of binary relations. In: LICS, pp. 248–254. IEEE Computer Society (1992)
23. RDF primer (2004). http://www.w3.org/TR/rdf-primer
24. Tarski, A.: On the calculus of relations. J. Symb. Log. **6**(3), 73–89 (1941)
25. Tarski, A., Givant, S.: Formalization of Set Theory without Variables. American Mathematical Society (1987)
26. Wu, Y., Van Gucht, D., Gyssens, M., Paredaens, J.: A study of a positive fragment of Path queries: Expressiveness, normal form and minimization. Comput. J. **54**(7), 1091–1118 (2011)