Relative Expressive Power of Navigational Querying on Graphs using Transitive Closure

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

June 11, 2015

Abstract

Motivated by both established and new applications, we study navigational query languages for graphs (binary relations). The simplest language has only the two operators union and composition, together with the identity relation. We make more powerful languages by adding any of the following operators: intersection; set difference; projection; coprojection; converse; transitive closure; and the diversity relation. All these operators map binary relations to binary relations. We compare the expressive power of all resulting languages, both for binary-relation queries as well as for boolean queries. In the absence of transitive closure, a complete Hasse diagram of relative expressiveness has already been established [8]. Moreover, it has already been shown that for boolean queries over a single edge label, transitive closure does not add any expressive power when only projection and diversity may be present [11]. In the present paper, we now complete the Hasse diagram in the presence of transitive closure, both for the case of a single edge label, as well as for the case of at least two edge labels. The main technical results are the following:

1. In contrast to the above-stated result [11] transitive closure does add expressive power when coprojection is present.
2. Transitive closure also adds expressive power as soon as converse is present.
3. Conversely, converse adds expressive power in the presence of transitive closure. In particular, the converse elimination result from [8] no longer works in the presence of transitive closure.
4. As a corollary, we show that the converse elimination result from [8] necessitates an exponential blow-up in the degree of the expressions.

1 Introduction

Graph databases, and the design and analysis of query languages appropriate for graph data, have a rich history in database systems and theory research [4]. Originally investigated from the perspective of object-oriented databases, interest in graph databases research has been continually renewed, motivated by data on the Web [1, 12] and new applications such as dataspaces [13], Linked Data [6] and RDF [22].

Typical of access to graph-structured data is its navigational nature. Indeed, in restriction to trees, there is a standard navigational query language, called XPath, whose expressive power has been intensively studied [5, 17]. XPath has been formalized in terms of a number of basic operators on binary relations [18]. Hence a natural approach [3,15,20] is to take this same set of operators but now evaluate them over graphs instead of over trees.

In a project that has been going on over the past few years, our goal has been to understand the relative importance of the different operators in this setting. The main results were already summarized in a conference paper in 2011 [10]. The present article is the final one in a series [8, 9, 11] of journal articles providing full details and proofs for that conference paper.

Concretely we consider a number of natural operators on binary relations (graphs): union; composition; intersection; set difference; projection; coprojection; converse; transitive closure and the identity and diversity relations. The largest language that we consider has all operators, while the smallest language has only union, composition and the identity relation. Expressions are built up from input relation names using these operators. Since each operator maps binary relations to binary relations, these query languages express queries from binary relations to binary relations: we call such queries *path queries*. By identifying nonemptiness with the boolean value 'true' and emptiness with 'false', 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*.

In our previous paper [8] we established a complete comparison of the expressiveness of all resulting languages not containing transitive closure, and this for both general path queries and boolean queries. The contribution of the present paper is to complete the picture by adding transitive closure. At the level of path queries, transitive closure obviously adds expressive power, since the languages without transitive closure are contained in first-order logic, whereas the transitive closure of a binary relation is not expressible in first-order logic [14]. When both languages L_1 and L_2 have transitive closure, however, we will show here that $L_1 \leq^{\text{path}} L_2$, meaning that every path query expressible in L_1 is also expressible in L_2, holds *precisely* when $L_1' \leq^{\text{path}} L_2'$, where L_i' denotes L_i with transitive closure removed again. To establish this characterization we will make use of the "strong" separations shown in our previous paper.

For boolean queries, the situation is more complicated. On the one hand, in the absence of transitive closure, we already know [8] that adding converse to a language containing projection but not containing intersection does not add boolean expressive power. We will show here, however, that this no longer holds

in the presence of transitive closure: adding converse then always adds boolean expressive power. Using the same intermediary results we establish that the aforementioned converse elimination in the absence of transitive closure has an inherent exponential character.

We then again consider the question whether adding the transitive closure operator strictly increases the expressive power of a language not yet containing transitive closure, but now for boolean queries. Over structures with at least two binary relations (equivalently, graphs with multiple edge labels), this question has an obvious affirmative answer. The question is more difficult, however, for boolean queries over structures consisting of a single binary relation (equivalently, graphs with a single edge label). Indeed, in a companion article [11], we have already shown that for boolean queries over a single edge label, transitive closure does *not* add any expressive power when only projection and diversity may be present. In the present paper, we show that these are essentially the only exceptions. Concretely, we show that, even over a single edge label, transitive closure adds expressive power for boolean queries as soon as either intersection, coprojection, or converse is present.

Let us briefly discuss some of the methods we use. The main technical result of the paper, in which we show that the collapse in expressive power involving the converse operator disappears in the presence of transitive closure, is proven using invariance under bisimulation from arrow logics [7]. The main technical challenge in such arguments is to establish bisimulations for increasing quantifier rank among pairs of graphs of increasing size. Our bisimulation argument also implies the exponential blowup inherent to converse elimination. To prove the cases where transitive closure does add expressive power at the level of boolean queries over a single edge label, we employ standard techniques from finite model theory such as Hanf-locality and first-order reductions [14].

For further motivational material on why we think our results are interesting, as well as extensive comparisons to the literature, we refer to our companion papers [8, 9, 11].

This paper is further organized as follows. In Section 2, we define the class of languages studied in the paper. In Section 3, we state the complete relative expressiveness theorems including transitive closure, this at the level of path queries as well as at the level of boolean queries. We prove these theorems in Sections 4 to 7. In Section 4 we do this for path queries and in Section 5 for boolean queries. Section 6 details the proofs of the bisimulation and exponential blowup results. Finally, Section 7 looks at the specific case of graphs with a single edge label.

2 Preliminaries

In this paper, we are interested in navigating over graphs whose edges are labeled by symbols from a finite, nonempty set of labels Λ. We can regard these edge labels as binary relation names and thus regard Λ as a relational database schema. For our purposes, then, a *graph* G is an instance of this database

schema Λ. That is, assuming an infinite universe V of data elements called *nodes*, G assigns to every $R \in \Lambda$ a relation $G(R) \subseteq V \times V$. Each pair in $G(R)$ is called an *edge* with label R. In what follows, $G(R)$ may be infinite, unless explicitly stated otherwise. All inexpressibility results in this paper already hold when restricting to finite graphs, however.

The most basic language for navigating over graphs we consider is the algebra $\mathcal{N}$ whose expressions are built recursively from the edge labels, the primitive $\emptyset$, and the primitive id, using composition $(e_1 \circ e_2)$ and union $(e_1 \cup e_2)$. Semantically, each expression $e \in \mathcal{N}$ defines a path query. A *path query* is a function q taking any graph G as input and returning a binary relation $q(G) \subseteq \mathrm{adom}(G) \times \mathrm{adom}(G)$. Here, $\mathrm{adom}(G)$ denotes the *active domain* of G, which is the set of all entries occurring in one of the relations of G. Formally,

$$\mathrm{adom}(G) = \{m \mid \exists n, \exists R \in \Lambda : (m,n) \in G(R) \vee (n,m) \in G(R)\}.$$

In detail, the semantics of $\mathcal{N}$ is inductively defined as follows:

$$\begin{aligned}
R(G) &= G(R)\,; \\
\emptyset(G) &= \emptyset\,; \\
id(G) &= \{(m,m) \mid m \in \mathrm{adom}(G)\}\,; \\
e_1 \circ e_2(G) &= \{(m,n) \mid \exists p\,((m,p) \in e_1(G)\ \&\ (p,n) \in e_2(G))\}\,; \\
e_1 \cup e_2(G) &= e_1(G) \cup e_2(G)\,.
\end{aligned}$$

The basic algebra $\mathcal{N}$ 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 $\mathcal{N}$ as *basic features*; we refer to the extensions as *nonbasic features*. The semantics of the extensions is as follows:

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

Here, e^k denotes $e \circ \cdots \circ e$ (k times).

If F is a set of nonbasic features, we denote by $\mathcal{N}(F)$ the language obtained by adding all features in F to $\mathcal{N}$.

We will actually compare language expressiveness at the level of both path queries and boolean queries. Path queries were defined above; a *boolean query* is a function from graphs to $\{\text{true}, \text{false}\}$.

Definition 2.1. 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.

For the given set of edge labels Λ, we write $\mathcal{N}(F_1) \leq_\Lambda^{\text{path}} \mathcal{N}(F_2)$ if every path query expressible in $\mathcal{N}(F_1)$ on graphs over Λ is also expressible in $\mathcal{N}(F_2)$. Similarly, we write $\mathcal{N}(F_1) \leq_\Lambda^{\text{bool}} \mathcal{N}(F_2)$ if every boolean query expressible in $\mathcal{N}(F_1)$ on graphs over Λ is also expressible in $\mathcal{N}(F_2)$. We write $\not\leq_\Lambda^{\text{path}}$ and $\not\leq_\Lambda^{\text{bool}}$ for the negation of $\leq_\Lambda^{\text{path}}$ and $\leq_\Lambda^{\text{bool}}$.

When discussing $\leq_\Lambda^{\text{path}}$ and $\leq_\Lambda^{\text{bool}}$, in a context that is valid for any Λ, however, we will omit the subscript Λ and simply write $\leq^{\text{path}}$ and $\leq^{\text{bool}}$. We already use this convention in the next definition, which introduces stronger variants of $\leq^{\text{path}}$ and $\leq^{\text{bool}}$.

Note that $\mathcal{N}(F_1) \leq^{\text{path}} \mathcal{N}(F_2)$ implies $\mathcal{N}(F_1) \leq^{\text{bool}} \mathcal{N}(F_2)$, but not necessarily the other way around.

It will also be interesting to consider stronger variants of $\not\leq^{\text{path}}$ and $\not\leq^{\text{bool}}$.

Definition 2.2. The language $\mathcal{N}(F_1)$ is *strongly separable from* the language $\mathcal{N}(F_2)$ *at the level of path queries* if there exists a path query q expressible in $\mathcal{N}(F_1)$ and a finite graph G, such that, for every expression $e \in \mathcal{N}(F_2)$, we have $q(G) \neq e(G)$. We write $\mathcal{N}(F_1) \not\leq_{\text{strong}}^{\text{path}} \mathcal{N}(F_2)$ in this case. Similarly, $\mathcal{N}(F_1)$ is *strongly separable from* $\mathcal{N}(F_2)$ *at the level of boolean queries* if there exists a boolean query q expressible in $\mathcal{N}(F_1)$ and two finite graphs G_1 and G_2, with $q(G_1)$ true and $q(G_2)$ false, such that, for every expression $e \in \mathcal{N}(F_2)$, $e(G_1)$ and $e(G_2)$ are both empty, or both nonempty. We write $\mathcal{N}(F_1) \not\leq_{\text{strong}}^{\text{bool}} \mathcal{N}(F_2)$ in this case.

Notice that strong separation indeed implies normal separation, i.e., $\not\leq_{\text{strong}}^{\text{path}}$ implies $\leq^{\text{path}}$ and $\not\leq_{\text{strong}}^{\text{bool}}$ implies $\leq^{\text{bool}}$. Intuitively, strong separation already establishes the separation by exhibiting just a single counterexample graph (for path queries), or just two counterexample graphs (for boolean queries). When strong separation can be established, it is very useful. However, this is not always possible. For a simple example, if F does not contain transitive closure then $\mathcal{N}(F \cup \{^+\}) \not\leq^{\text{path}} \mathcal{N}(F)$, but $\mathcal{N}(F \cup \{^+\}) \not\leq_{\text{strong}}^{\text{path}} \mathcal{N}(F)$ is impossible. Indeed, on a given finite graph G, every expression in $\mathcal{N}(F \cup \{^+\})$ can be "unrolled" into an expression without transitive closure that yields the same result on G (see the proof of Proposition 4.1).

3 Results

We recall the following notation [8]. If F is a set of nonbasic features, then $\overline{F}$ is the set obtained by augmenting F with all nonbasic features that can be expressed in $\mathcal{N}(F)$ through a repeated application of the following equalities:

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

The relative expressive power at the level of path queries for languages not containing transitive closure is captured by the following theorem [8]:

Theorem 3.1 ([8]). *Let Λ be an arbitrary finite nonempty set of edge labels, and let F_1 and F_2 be sets of nonbasic features such that $^+ \notin F_1$ and $^+ \notin F_2$. Then, $\mathcal{N}(F_1) \leq_\Lambda^{\text{path}} \mathcal{N}(F_2)$ if and only if $F_1 \subseteq \overline{F_2}$.*

In this paper we extend this theorem to include transitive closure:

Theorem 3.2. *Let Λ be an arbitrary finite nonempty set of edge labels. Let F_1 and F_2 be sets of nonbasic features. Then $\mathcal{N}(F_1) \leq_\Lambda^{\text{path}} \mathcal{N}(F_2)$ if and only if $F_1 \subseteq \overline{F_2}$.*

We discuss and prove Theorem 3.2 in Section 4.

The above theorem settles the relative expressive power at the level of path queries. Let us now turn our attention to boolean queries. To state what is already known we introduce the following additional notation. For a set of nonbasic features F, we define

$$\widehat{F} = \begin{cases} (F \setminus \{^{-1}\}) \cup \{\pi\}, & \text{if } ^{-1} \in \overline{F}, \cap \notin \overline{F}, ^+ \notin \overline{F} \\ F, & \text{otherwise} \end{cases}$$

This notation extends notation introduced earlier [8] to include transitive closure. Indeed, for languages without transitive closure, the following is already known [8]:

Theorem 3.3 ([8]). *Let Λ be an arbitrary finite nonempty set of edge labels, and let F_1 and F_2 be sets of nonbasic features such that $^+ \notin F_1$ and $^+ \notin F_2$. Then, $\mathcal{N}(F_1) \leq_\Lambda^{\text{bool}} \mathcal{N}(F_2)$ if and only if $F_1 \subseteq \overline{F_2}$ or $\widehat{F_1} \subseteq \overline{F_2}$.*

By the above Theorem, when $\overline{F}$ contains neither intersection nor transitive closure, boolean queries expressed in $\mathcal{N}(F)$ can be translated to expressions in $\mathcal{N}(\widehat{F})$, thus effectively eliminating converse (at the expense of adding projection). We will refer to this phenomenon as *converse elimination*.

In this paper we will extend the analysis to include transitive closure. It turns out that we will need to consider the case where there are at least two edge labels separately from the case where there is only one edge label. For the first case, we can again generalize the above theorem verbatim.

Theorem 3.4. *Assume that Λ contains at least two edge labels, and let F_1 and F_2 be sets of nonbasic features. Then, $\mathcal{N}(F_1) \leq_{\Lambda}^{\text{bool}} \mathcal{N}(F_2)$ if and only if $F_1 \subseteq \overline{F_2}$ or $\widehat{F_1} \subseteq \overline{F_2}$.*

We discuss and prove Theorem 3.4 in Sections 5 and 6.

In the case of a single edge label, all edges have the same label. Hence, the edge label carries no information which is why we refer to this case as the "unlabeled" case. We will use the notation $\leq_{\text{unl}}^{\text{bool}}$ to denote this case. The theorem is now as follows:

Theorem 3.5. *Let F_1 and F_2 be sets of nonbasic features. Then $\mathcal{N}(F_1) \leq_{\text{unl}}^{\text{bool}} \mathcal{N}(F_2)$ if and only if at least one of the following conditions hold:*

1. *$F_1 \subseteq \overline{F_2}$;*
2. *$\widehat{F_1} \subseteq \overline{F_2}$;*
3. *$^+ \in F_1$, $^+ \notin F_2$, $F_1 \subseteq \{\pi, di, ^+\}$ and $F_1 \setminus \{^+\} \subseteq \overline{F_2}$.*

We thus see that in the unlabeled case, transitive closure does not always add expressive power for boolean queries. We discuss and prove Theorem 3.5 in Section 7.

4 Path queries

The goal of this section is to show that Theorem 3.2 holds. Thereto, we need the following result: if there is a separation at the level of path queries for languages not containing transitive closure, then the separation still stands when we include transitive closure.

Proposition 4.1. *Let F_1 and F_2 be sets of nonbasic features such that $^+ \notin F_1$ and $^+ \notin F_2$. If $\mathcal{N}(F_1) \not\leq^{\text{path}} \mathcal{N}(F_2)$ then $\mathcal{N}(F_1) \not\leq^{\text{path}} \mathcal{N}(F_2 \cup \{^+\})$.*

Proof. In our previous work [8] we have proven that if $\mathcal{N}(F_1) \not\leq^{\text{path}} \mathcal{N}(F_2)$, then either $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$ or $\mathcal{N}(F_1) \not\leq_{\text{strong}}^{\text{path}} \mathcal{N}(F_2)$. If $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$, the result will follow directly from Proposition 5.3 (which clearly does not rely on the present proposition). Now suppose $\mathcal{N}(F_1) \not\leq_{\text{strong}}^{\text{path}} \mathcal{N}(F_2)$. Then there exists a path query q expressible in $\mathcal{N}(F_1)$, and a finite graph G such that $q(G) \neq e(G)$ for every expression $e \in \mathcal{N}(F_2)$. We will show that q is not expressible in $\mathcal{N}(F_2 \cup \{^+\})$. To this end let $e' \in \mathcal{N}(F_2 \cup \{^+\})$. Note that, for any graph H with at most n nodes, and any expression $f \in \mathcal{N}(F_2)$, we have $f^+(H) = (\bigcup_{i=1}^{n} f^i)(H)$. Hence if we remove transitive closure applications in e' using this equality, we obtain an expression $e'' \in \mathcal{N}(F_2)$ such that $e''(G) = e'(G)$.

Therefore, $q(G) \neq e''(G)$ since $q(G) \neq e'(G)$. Thus we can conclude that q is not expressible in $\mathcal{N}(F_2 \cup \{^+\})$, whence $\mathcal{N}(F_1) \not\leq^{\text{path}}_{\text{strong}} \mathcal{N}(F_2 \cup \{^+\})$ as desired.□

We are now ready to prove Theorem 3.2.

Proof of Theorem 3.2. We first take care of the 'if' direction. Every expression $e \in \mathcal{N}(F_1)$ can be transformed into an equivalent expression $e' \in \mathcal{N}(F_2)$ by using the appropriate interdependencies introduced at the beginning of Section 3.

For the 'only if' direction we split our proof into three cases.

- If $^+ \notin F_1$ and $^+ \notin F_2$, then our theorem coincides with Theorem 3.1.
- If $^+ \in F_1$ and $^+ \notin F_2$, then any query expressible in $\mathcal{N}(F_2)$ is also expressible in first-order logic. It is well known, however, that the transitive closure of a binary relation is not expressible in first-order logic [14].
- If $^+ \in F_2$, then $\overline{F_2} = \overline{F_2 \setminus \{^+\}} \cup \{^+\}$. Hence $F_1 \subseteq \overline{F_2}$ iff $F_1 \setminus \{^+\} \subseteq \overline{F_2 \setminus \{^+\}}$.

  We now show the contrapositive of the 'only if' direction. To this end, suppose that $F_1 \setminus \{^+\} \not\subseteq \overline{F_2 \setminus \{^+\}}$. Then Theorem 3.1 implies that $\mathcal{N}(F_1 \setminus \{^+\}) \not\leq^{\text{path}} \mathcal{N}(F_2 \setminus \{^+\})$. The result now follows directly from Proposition 4.1. □

5 Boolean queries

The goal of this section is to show that Theorem 3.4 holds. Thereto, we first need a few preliminary results. At the level of boolean queries, transitive closure adds expressive power if Λ contains at least two edge labels.

Proposition 5.1. *Assume that Λ contains at least two edge labels, and let F_1 and F_2 be sets of nonbasic features. If $^+ \in \overline{F_1}$ and $^+ \notin \overline{F_2}$, then $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\Lambda} \mathcal{N}(F_2)$.*

Proof. Let S and R be two different edge labels in Λ. If the boolean query expressed by $S \circ R^+ \circ S \in \mathcal{N}(F_1)$ would be expressible in $\mathcal{N}(F_2)$, it would also be expressible in first-order logic. It is well known, however, that such a reachability query is not expressible in first-order logic [2]. Hence it is not expressible in $\mathcal{N}(F_2)$ either. □

In Section 7 we explore whether transitive closure adds expressive power in the unlabeled case.

It has been shown that if $\pi \in \overline{F_1}$ and $F_2 \subseteq \{^{-1}, di\}$ then $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$ [8]. We extend this result to include transitive closure:

Proposition 5.2. *Let F_1 and F_2 be sets of nonbasic features. If $\pi \in \overline{F_1}$ and $F_2 \subseteq \{^{-1}, di, ^+\}$, then $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$.*

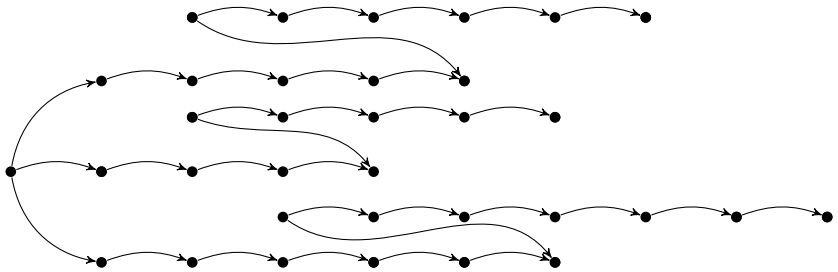

Figure 1: The graph B_{ZZZ} used to prove Proposition 5.2. All edges are assumed to have the same label R.

Proof. It is sufficient to show that $\mathcal{N}(\pi) \not\leq^{\text{bool}} \mathcal{N}(^{-1}, di, ^{+})$ since $\mathcal{N}(\pi) \leq^{\text{bool}} \mathcal{N}(F_1)$ and $\mathcal{N}(F_2) \leq^{\text{bool}} \mathcal{N}(^{-1}, di, ^{+})$. We adapt the proof of Proposition 4.1 in our previous work [8] with some minor changes so that it provides a proof for Proposition 5.2. Let B_{ZZZ} be the graph displayed in Figure 1, and let Q be the boolean query

$$\pi_1(R^4 \circ \pi_2(\pi_1(R^4) \circ R)) \circ \pi_1(R^5 \circ \pi_2(\pi_1(R^5) \circ R)) \circ \pi_1(R^6 \circ \pi_2(\pi_1(R^6) \circ R)) \neq \emptyset$$

clearly expressible in $\mathcal{N}(\pi)$. This query accepts B_{ZZZ}, i.e., $Q(B_{ZZZ}) \neq \emptyset$. Since every transitive closure application f^+ can be written as an infinite union $\bigcup_{i=1}^{\infty} f^i$ and unions in $\mathcal{N}(^{-1}, di)$ can be brought outside, every expression $e \in \mathcal{N}(^{-1}, di, ^{+})$ can be written as an infinite union of union-free expressions in $\mathcal{N}(^{-1}, di)$. Therefore, we can write the expression $Q \in \mathcal{N}(^{-1}, di, ^{+})$, which supposedly expresses Q_{ZZZ}, as $Q = \bigcup_{i=1}^{\infty} e_i$ where e_i is a union-free expression in $\mathcal{N}(^{-1}, di)$ for every natural number i. Since $Q(B_{ZZZ}) \neq \emptyset$, there exists an expression $e \in \{e_i \mid i \in \mathbb{N}\}$ such that $e(B_{ZZZ}) \neq \emptyset$. The proof now proceeds exactly as in the proof of Proposition 4.1 in our previous work [8]. □

If there is a separation at the level of boolean queries for languages not containing transitive closure, the separation still stands when we include transitive closure.

Proposition 5.3. *Let F_1 and F_2 be sets of nonbasic features such that $^{+} \notin F_1$ and $^{+} \notin F_2$. If $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$ then $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2 \cup \{^{+}\})$.*

Proof. In our previous work [8] we have proven that if $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$ then either $\pi \in \overline{F_1}$ and $F_2 \subseteq \{^{-1}, di\}$, or $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{strong}} \mathcal{N}(F_2)$. If $\pi \in \overline{F_1}$ and $F_2 \subseteq \{^{-1}, di\}$, then $F_2 \cup \{^{+}\} \subseteq \{^{-1}, di, ^{+}\}$, whence $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2 \cup \{^{+}\})$ due to Proposition 5.2.

So from here we may assume that $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{strong}} \mathcal{N}(F_2)$. In this case there exists an expression $q \in \mathcal{N}(F_1)$ and two graphs G_1 and G_2 with $q(G_1) = \mathit{true}$ and $q(G_2) = \mathit{false}$, such that for every expression $e \in \mathcal{N}(F_2)$, $e(G_1)$ and $e(G_2)$ are both empty or both nonempty. Now, let $n = \max(\text{adom}(G_1), \text{adom}(G_2))$.

We show that q is not expressible in $\mathcal{N}(F_2 \cup \{^+\})$. To this end let $e' \in \mathcal{N}(F_2 \cup \{^+\})$. Note that, for any graph H with at most n nodes, and any expression $f \in \mathcal{N}(F_2)$, we have $f^+(H) = (\bigcup_{i=1}^{n} f^i)(H)$. Hence if we remove transitive closure applications in e' using this equality, we obtain an expression $e'' \in \mathcal{N}(F_2)$ such that $e''(G_1) = e'(G_1)$ and $e''(G_2) = e'(G_2)$. Since $e''(G_1)$ and $e''(G_2)$ are both empty or both nonempty, $e'(G_1)$ and $e'(G_2)$ are both empty or both nonempty. Thus we can conclude that q is not expressible in $\mathcal{N}(F_2 \cup \{^+\})$, whence $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{strong}} \mathcal{N}(F_2 \cup \{^+\})$ as desired. □

In Section 6 we will prove the following proposition, showing that converse elimination, discussed in Section 3, no longer occurs in the presence of transitive closure. Specifically, we will show that the boolean query $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is not expressible in the largest language without converse.

Proposition 5.4. *Let F_1 and F_2 be sets of nonbasic features. If $^{-1} \in \overline{F}_1$, $^+ \in \overline{F}_1$, and $^{-1} \notin \overline{F}_2$, then $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2)$.*

For later purposes, we first examine a selection of languages for which the requirement for multiple edge labels of Theorem 3.4 can be removed.

Proposition 5.5. *Let Λ be an arbitrary finite nonempty set of edge labels, and let F_1 and F_2 be sets of nonbasic features such that $^+ \in F_1$ implies $^+ \in F_2$. Then, $\mathcal{N}(F_1) \leq^{\text{bool}} \mathcal{N}(F_2)$ if and only if $F_1 \subseteq \overline{F_2}$ or $\widehat{F_1} \subseteq \overline{F_2}$.*

Proof. We split our proof into three cases.

- If $^+ \notin F_1$ and $^+ \notin F_2$, then our theorem coincides with Theorem 3.3.
- If $^+ \notin F_1$ and $^+ \in F_2$, then $\overline{F_2} = \overline{F_2 \setminus \{^+\}} \cup \{^+\}$. Let us first consider the 'if' direction. By the argument above, $F_1 \subseteq \overline{F_2 \setminus \{^+\}}$. Therefore $\mathcal{N}(F_1) \leq^{\text{bool}} \mathcal{N}(F_2 \setminus \{^+\})$ by Theorem 3.3 since $^+ \notin F_1$, whence $\mathcal{N}(F_1) \leq^{\text{bool}} \mathcal{N}(F_2)$.

  For the 'only if' direction, we consider its contrapositive. In this case $\widehat{F_1} \not\subseteq \overline{F_2} \setminus \{^+\} = \overline{F_2 \setminus \{^+\}}$ and $F_1 \subseteq \overline{F_2} \setminus \{^+\} = \overline{F_2 \setminus \{^+\}}$. Theorem 3.3 can now be applied, which tells us that $\mathcal{N}(F_1) \not\leq^{\text{bool}} \mathcal{N}(F_2 \setminus \{^+\})$. The result now directly follows from Proposition 5.3.
- If $^+ \in F_1$ and $^+ \in F_2$, then

  $$\widehat{F_1} \subseteq \overline{F_2} \vee F_1 \subseteq \overline{F_2} \Leftrightarrow F_1 \subseteq \overline{F_2} \Leftrightarrow F_1 \setminus \{^+\} \subseteq \overline{F_2} \setminus \{^+\} = \overline{F_2 \setminus \{^+\}}.$$

  First we take care of the 'if' direction. By the first equivalence above, we may assume that $F_1 \subseteq \overline{F_2}$. By Theorem 3.2 we have $\mathcal{N}(F_1) \leq^{\text{path}} \mathcal{N}(F_2)$, whence also $\mathcal{N}(F_1) \leq^{\text{bool}} \mathcal{N}(F_2)$.

  For the 'only if' direction, we consider its contrapositive. By the second equivalence above, we may assume that $F_1 \setminus \{^+\} \not\subseteq \overline{F_2 \setminus \{^+\}}$. If $^{-1} \notin F_1$, then $\widehat{F_1 \setminus \{^+\}} = F_1 \setminus \{^+\} \not\subseteq \overline{F_2 \setminus \{^+\}}$ by definition. Hence we can apply Theorem 3.3. The result now follows from Proposition 5.3.

Conversely, suppose that $^{-1} \in F_1$. Additionally, if $^{-1} \notin F_2$, then the result follows directly from Proposition 5.4. On the other hand, if $^{-1} \in F_2$ then $F_1 \setminus \{^+\} \not\subseteq \overline{F_2 \setminus \{^+\}}$ implies that another feature $x \in F_1 \setminus \{^+\}$ not equal to $^{-1}$ is not present in $\overline{F_2 \setminus \{^+\}}$, whence $\widehat{F_1 \setminus \{^+\}} \not\subseteq \overline{F_2 \setminus \{^+\}}$ by definition. Hence, $\mathcal{N}(F_1 \setminus \{^+\}) \not\leq^{\text{bool}} \mathcal{N}(F_2 \setminus \{^+\})$ by Theorem 3.3. The result now follows directly from Proposition 5.3. □

We are now ready for the proof of Theorem 3.4.

Proof of Theorem 3.4. If $^+ \in F_1$ and $^+ \notin F_2$, then clearly $\widehat{F_1} \not\subseteq \overline{F_2}$ and $F_1 \not\subseteq \overline{F_2}$. Furthermore, $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\Lambda} \mathcal{N}(F_2)$ by Proposition 5.1, which we can apply since Λ contains at least two edge labels. This concludes our proof in this case.

In the remaining cases our theorem follows directly from Proposition 5.5. □

6 Converse cannot be eliminated in the presence of transitive closure

The goal of this section is to prove Proposition 5.4. To do so we will prove in Section 6.2 that $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is not expressible in the largest language without converse.

To show this inexpressibility result, we will employ invariance results under the notion of bisimulation below. In essence, this notion is based on the notion of bisimulation known from arrow logics [19]. Below, we adapt this notion to the current setting.

Let $\mathbf{G} = (G, a, b)$ denote a *marked graph*, i.e., a graph G with $a, b \in \text{adom}(G)$. The *degree* of an expression e is the maximum depth of nested applications of composition, projection and coprojection in e. For example, the degree of $R \circ R$ is 1, while the degree of both $R \circ (R \circ R)$ and $\pi_1(R \circ R)$ is 2. Intuitively, the degree of e corresponds to the quantifier rank of the standard translation of e into FO^3. For a set of features F, $\mathcal{N}(F)_k$ denotes the set of expressions in $\mathcal{N}(F)$ of degree at most k.

In what follows, we are only concerned with bisimulation results regarding $\mathcal{N}(\setminus, di)$. The following is an appropriate notion of bisimulation for this language.

Definition 6.1 (Bisimilarity). Let k be a natural number and let $\mathbf{G}_1 = (G_1, a_1, b_1)$ and $\mathbf{G}_2 = (G_2, a_2, b_2)$ be marked graphs. We say that $\mathbf{G}_1$ is bisimilar to $\mathbf{G}_2$ up to depth k, denoted $\mathbf{G}_1 \simeq_k \mathbf{G}_2$, if the following conditions are satisfied:

Atoms $a_1 = b_1$ if and only if $a_2 = b_2$; and $(a_1, b_1) \in G_1(R)$ if and only if $(a_2, b_2) \in G_2(R)$, for every $R \in \Lambda$;

Forth if $k > 0$, then, for every c_1 in $\text{adom}(G_1)$, there exists some c_2 in $\text{adom}(G_2)$ such that

$$(G_1, a_1, c_1) \simeq_{k-1} (G_2, a_2, c_2) \quad \text{and} \quad (G_1, c_1, b_1) \simeq_{k-1} (G_2, c_2, b_2);$$

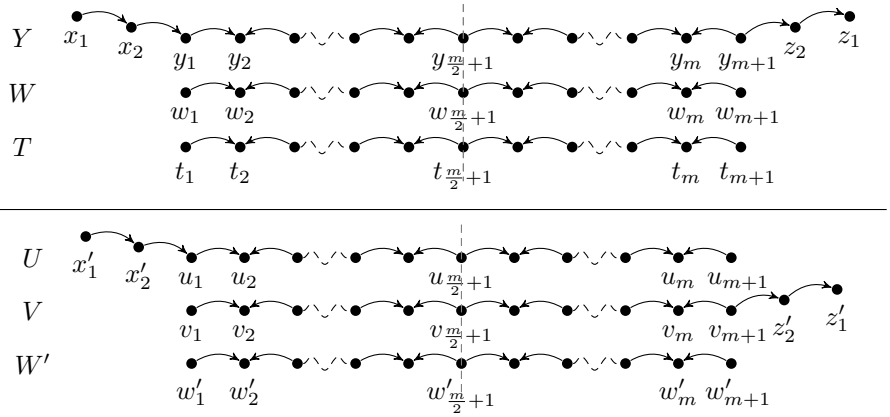

Figure 2: Graphs G_1^m (top) and G_2^m (bottom) used to establish boolean separation in the proof of Proposition 5.4.

Back if $k > 0$, then, for every c_2 in $\mathrm{adom}(G_2)$, there exists some c_1 in $\mathrm{adom}(G_1)$ such that

$$(G_1, a_1, c_1) \simeq_{k-1} (G_2, a_2, c_2) \quad \text{and} \quad (G_1, c_1, b_1) \simeq_{k-1} (G_2, c_2, b_2).$$

We also say that there is a bisimulation of depth k between $\mathbf{G}_1$ and $\mathbf{G}_2$ if $\mathbf{G}_1 \simeq_k \mathbf{G}_2$.

Recall the following adequacy theorem for bisimulations.

Theorem 6.2 ([9]). *Let k be a natural number; and let $\mathbf{G}_1 = (G_1, a_1, b_1)$ and $\mathbf{G}_2 = (G_2, a_2, b_2)$ be marked graphs. We have, $\mathbf{G}_1 \simeq_k \mathbf{G}_2$ iff $(a_1, b_1) \in e(G_1) \Leftrightarrow (a_2, b_2) \in e(G_2)$ for every $e \in \mathcal{N}(\backslash, di)_k$.*

Intuitively, this proposition tells us that marked graphs are indistinguishable by k-degree path queries iff these graphs are bisimilar up to depth k.

In Section 6.1, we will establish bisimulations between the classes of graphs G_1^m and G_2^m displayed in Figure 2. This is then used in Section 6.2 to prove Proposition 5.4. Moreover, in Section 6.3 we use the bisimulations to show an exponential blowup for converse elimination (cf. the discussion following Theorem 3.3). Specifically, we will prove:

Theorem 6.3. *Let F be a set of nonbasic features such that $^{-1} \in F$, $\cap \notin \overline{F}$ and $^{+} \notin F$. Furthermore, let h be a function that translates expressions e in $\mathcal{N}(F)$ to $e' \in \mathcal{N}(\widehat{F})$ such that e' is equivalent to e at the level of boolean queries (such a function h exists by Theorem 3.3). If $f : \mathbb{N} \to \mathbb{N}$ is a function such that for every $e \in \mathcal{N}(F)$ we have $\mathrm{degree}(h(e)) \leq f(\mathrm{degree}(e))$, then $f \neq o(2^n)$.*

6.1 A bisimulation result

In this section we will establish the required bisimulations to prove Proposition 5.4. For the remainder of this section let $m > 4$ be an integer multiple of

four, let G_1^m be the graph at the top and G_2^m be the graph at the bottom in Figure 2. It is important to note that these graphs have the displayed form only when m is a multiple of four.

The goal is to establish the following theorem:

Theorem 6.4. *For every pair* $(a_1, b_1) \in \mathrm{adom}(G_1^m)^2$ *there exists another pair* $(a_2, b_2) \in \mathrm{adom}(G_2^m)^2$ *such that* $(G_1^m, a_1, b_1) \simeq_{m/2-1} (G_2^m, a_2, b_2)$.

Before we can do so, we introduce some terminology. We say that a pair $(x, y) \in \mathrm{adom}(G_1^m) \times \mathrm{adom}(G_2^m)$ is *valid* if the following conditions hold:

- if $x \in \{y_i, w_i, t_i\}$ then $y \in \{u_i, v_i, w_i'\}$;
- if $x = x_1$ then $y = x_1'$;
- if $x = x_2$ then $y = x_2'$;
- if $x = z_1$ then $y = z_1'$;
- if $x = z_2$ then $y = z_2'$.

Intuitively, the pair (x, y) is valid if x and y are displayed in the same column in Figure 2, so formally, instead of saying that (x, y) is valid, we also say that x and y are *in the same column*. Moreover, we will extend this terminology for nodes x and y belonging to the same graph, with the obvious meaning.

Definition 6.5. A 4-tuple $(a_1, b_1, a_2, b_2) \in \mathrm{adom}(G_1^m)^2 \times \mathrm{adom}(G_2^m)^2$ is *valid* if the following conditions hold:

(a) (a_1, a_2) and (b_1, b_2) are valid;

(b) $(a_1, b_1) \in G_1^m$ if and only if $(a_2, b_2) \in G_2^m$; and $a_1 = b_1$ if and only if $a_2 = b_2$. Note that this is the Atoms condition for bisimilarity;

(c) if $a_1 = x_2$, $b_1 = y_2$ and $a_2 = x_2'$, then $b_2 = u_2$;

(d) if $a_1 = x_2$, $a_2 = x_2'$ and $b_2 = u_2$, then $b_1 = y_2$.

Intuitively, a valid quadruple is a potential starting point for a bisimulation between G_1^m and G_2^m.

For any node $x \in \mathrm{adom}(G_1^m)$ we introduce the following terminology.

- If x equals x_1 or x_2, or y_i, w_i or t_i with $0 \leq i \leq m/2 + 1$, we call x a *left* element.
- If x is not a left element, i.e., x equals z_1 or z_2, or y_i, w_i or t_i with $m/2 + 1 < i \leq m + 1$, we call x a *right* element.
- If x equals y_i for any i, we call x a Y element. Analogously, if x equals w_i, t_i, x_i, or z_i for any i, we call x a W, T, X or Z element, respectively.

Clearly we can combine these adjectives and thus speak about a Y left element, for example.

For any node $y \in \text{adom}(G_2^m)$ we can use the analogous terminology of left, right, U, V, W', X' and Z' elements with analogous meaning.

Let us now define a function f mapping valid pairs to natural numbers:

$$f(d,e) = \begin{cases} m/2 & \text{if } d = y_i \text{ left and } e = u_i \\ i-1 & \text{if } d = y_i \text{ left and } (e = v_i \text{ or } e = w'_i) \\ m+1-i & \text{if } d = y_i \text{ right and } (e = u_i \text{ or } e = w'_i) \\ m/2 & \text{if } d = y_i \text{ right and } e = v_i \\ i-1 & \text{if } (d = w_i \text{ or } d = t_i) \text{ left and } e = u_i \\ m/2 & \text{if } (d = w_i \text{ or } d = t_i) \text{ left and } (e = v_i \text{ or } e = w'_i) \\ m/2 & \text{if } (d = w_i \text{ or } d = t_i) \text{ right and } (e = u_i \text{ or } e = w'_i) \\ m+1-i & \text{if } (d = w_i \text{ or } d = t_i) \text{ right and } e = v_i \\ i-1 & \text{if } d = t_i \text{ left and } e = u_i \\ m/2 & \text{if } d = t_i \text{ left and } (e = v_i \text{ or } e = w'_i) \\ m/2 & \text{if } d = t_i \text{ right and } (e = u_i \text{ or } e = w'_i) \\ m+1-i & \text{if } d = t_i \text{ right and } e = v_i \\ m/2 & \text{if } d = x_i \text{ and } e = x'_i \\ m/2 & \text{if } d = z_i \text{ and } e = z'_i \end{cases}$$

Intuitively, $f(d,e) = m/2$ only when d and e are in the middle column or d and e are on the side of chains with similar endings in Figure 2, i.e., d is Y left iff e is U left, and d is Y right iff e is V right. In all other cases $f(d,e) < m/2$. For example, let us examine the values for the valid pairs (x,y), (w,z), (a_1,a_2) and (b_1,b_2) in the graphs G_1^8 and G_2^8 displayed in Figure 3. In this case $m = 8$, thus $f(x,y) = 2$, $f(w,z) = m+1-7 = 2$, $f(a_1,a_2) = m/2 = 4$ and $f(b_1,b_2) = 3$.

Our key idea to establish Theorem 6.4 is to show that $\min(f(a_1,a_2), f(b_1,b_2))$ is a lower bound on the bisimulation depth between (G_1^m, a_1, b_1) and (G_2^m, a_2, b_2); this will be our key Lemma 6.22. Before proving this in detail, we intuitively describe the overall strategy.

To establish a bisimulation of depth d between (G_1^m, a_1, b_1) and (G_2^m, a_2, b_2), we need that (a_1, b_1, a_2, b_2) satisfies the Atoms condition, and we need that the Forth and Back conditions hold. A first characteristic of our strategy is that we take care to maintain not just the Atoms condition, but the stronger property of *validity* from Definition 6.5. Viewing a bisimulation argument as a game, the validity property provides tighter control on the possible game situations that can arise.

For the Forth condition we need to find a node $c_2 \in \text{adom}(G_2^m)$ for every node $c_1 \in \text{adom}(G_1^m)$ such that there is a bisimulation of depth $d-1$ between (G_1^m, a_1, c_1) and (G_2^m, c_1, a_2), and (G_1^m, c_1, b_1) and (G_2^m, c_2, b_2). For the Back condition we need to do the same thing except that the roles of c_1 and c_2 are switched.

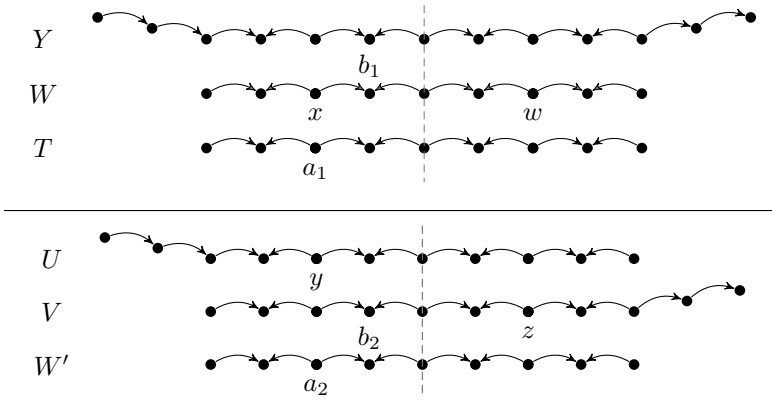

Figure 3: The graphs G_1^8 at the top, and G_2^8 at the bottom. Notice here that (x, y), (w, z), (a_1, a_2) and (b_1, b_2) are valid pairs. Since $m = 8$ in this case, we have that $f(x, y) = 2$, $f(w, z) = m + 1 - 7 = 2$, $f(a_1, b_2) = m/2 = 4$ and $f(b_1, b_2) = 3$

Actually, instead of directly working with bisimulations with a certain depth, we will show that we can pick a $c_2 \in \text{adom}(G_2^m)$ for every $c_1 \in \text{adom}(G_1^m)$ (and vice versa) that ensures validity of (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) while providing a lower bound on $f(c_1, c_2)$. This will provide enough information to prove Lemma 6.22 by induction.

So let us now have an intuitive look at the strategy used in the technical lemmas to pick such a $c_2 \in \text{adom}(G_2^m)$ for every $c_1 \in \text{adom}(G_1^m)$. First, remember that we only work with valid quadruples, so c_2 has to be in the same column as c_1. This leaves us with three candidate nodes (or just one in case c_1 is an X or Z element). We pick one of these nodes according to the following strategy:

1. First, we check whether $a_1 = c_1$, $c_1 = b_1$, (a_1, c_1) is an edge, or (c_1, b_1) is an edge. If this is indeed the case, we say that c_1 is *related* to a_1 or b_1. Here we pick c_2 so that it is related in the same way as c_1 is related to a_1 or b_1. The relation of c_1 and to a_1 of b_1 ensures that c_1 is in the column next to, or in the same column as a_1 or b_1. This implies that $f(c_1, c_2)$ is at most one lower than $f(a_1, a_2)$ or $f(b_1, b_2)$.

   For example, if (c_1, b_1) is an edge, we pick c_2 in the same column as c_1 such that (c_2, b_2) is an edge (see Figure 4).

2. If c_1 is not related to a_1 or b_1, i.e., if $a_1 \neq c_1$, $c_1 \neq b_1$, (a_1, c_1) is not an edge, and (c_1, b_1) is not an edge, we check whether it is possible to pick c_2 such that $f(c_1, c_2) = m/2$ without breaking validity. Since $m/2$ is the maximum output of f, we can be sure that $f(c_1, c_2)$ is sufficiently large. For an example of this scenario see Figure 5.

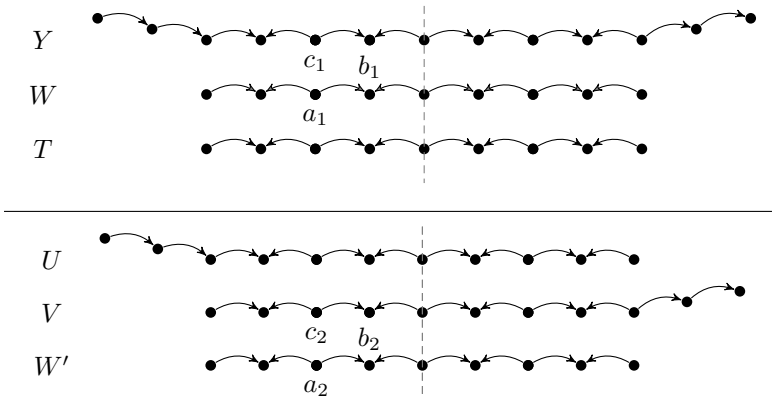

Figure 4: An example of the first step in our strategy on the graphs G_1^m and G_2^m with $m = 8$. Here c_1 is related to b_1, i.e, (c_1, b_1) is an edge. The node c_2 is thus picked such that (c_2, b_2) is an edge. The validity of (a_1, b_1, a_2, b_2) then ensures that a_2 is not related to c_2. Notice also that $f(c_1, c_2) = f(b_1, b_2) - 1$ by definition.

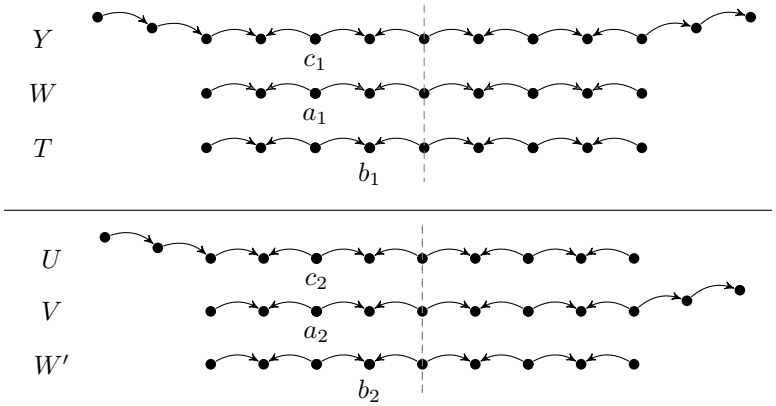

Figure 5: An example of the second step in our strategy on G_1^m and G_2^m with $m = 8$. Hence c_1 is not related a_1 and b_1 ($a_1 \neq c_1$, $c_1 \neq b_1$, (a_1, c_1) is not an edge, and (c_1, b_1) is not an edge), and it is possible to pick c_2 such that $f(c_1, c_2) = m/2$ without violating validity. Notice that c_2 has to be picked on U in this example since only then $f(c_1, c_2) = m/2$ by definition.

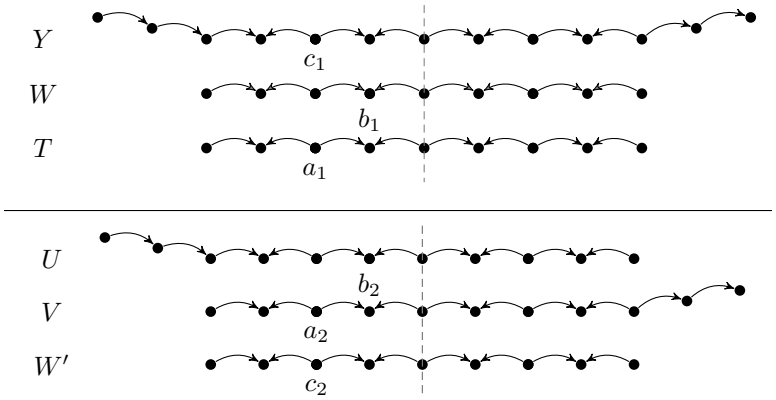

Figure 6: An example of the third step in our strategy on G_1^m and G_2^m with $m = 8$. Hence c_1 is not related to a_1 and b_2 ($a_1 \neq c_1$, $c_1 \neq b_1$, (a_1, c_1) is not an edge, and (c_1, b_1) is not an edge), and it is not possible to pick c_2 such that $f(c_1, c_2) = m/2$. Indeed, here $f(c_1, c_2) = m/2$ only if c_2 is on U. Thus c_2 has to be picked on the chain not containing a_2 and b_2. Clearly $f(c_1, c_2) = f(b_1, b_2) - 1$ by definition.

3. If we cannot pick c_2 such that $f(c_1, c_2) = m/2$ without breaking validity, we just pick c_2 such that validity is ensured. It turns out that even then $f(c_1, c_2)$ is sufficiently large, i.e., at most one lower than $f(a_1, a_2)$ or $f(b_1, b_2)$. For an example of this scenario see Figure 6.

The strategy by itself may seem quite arbitrary. Why do we not provide a single c_2 for each c_1 without the trial and error in the second step of the strategy? The reason why we introduced the trial and error step, is because a failure in that step tells us something about the location of a_1 and a_2, and b_1 and b_2. Indeed, if the validity of (a_1, c_1, a_2, c_2) is broken, for example, we know that $a_2 = c_2$, or that (a_2, c_2) is an edge, which implies that a_1 and a_2 are in the same column as, or in the column next to c_1 and c_2. Using these facts, we will be able to determine the values of $f(a_1, a_2)$ and $f(b_1, b_2)$, which will appear to be sufficiently low by itself so that we can pick c_2 without having to worry about $f(c_1, c_2)$.

We will now start the technical proof with several lemmas. Lemmas 6.6 and 6.7 take care of first step of the strategy outlined above. Lemmas 6.8 and 6.17 take care of the second and third step. To establish these last two steps, we use several sublemmas for clarity (Lemmas 6.9 to 6.16).

Lemma 6.6. *Suppose that (a_1, b_1, a_2, b_2) is valid, $f(a_1, a_2) > 0$, $f(b_1, b_2) > 1$ and $c_1 \in \mathrm{adom}(G_1^m)$ such that $a_1 = c_1$, $b_1 = c_1$, (a_1, c_1) is an edge, or (c_1, b_1) is an edge. Then there exists $c_2 \in \mathrm{adom}(G_2^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) are valid, and $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Proof. First suppose that $a_1 = c_1$. Then we pick $c_2 = a_2$. Clearly (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) are valid by construction. Furthermore, $f(c_1, c_2) = f(a_1, a_2) \geq \min((fa_1, a_2)), f(b_1, b_2)) - 1$. The case where $c_1 = b_1$ is analogous.

Now suppose that (a_1, c_1) is an edge. Then we pick c_2 in the same column as c_1 (thus (c_1, c_2) is valid) such that (a_2, c_2) is an edge. This is clearly possible if $a_1 \neq y_{m+1}$, since in that case any node in the same column of a_2 has a forward or backward outgoing edge in the same way as a_1. On the other hand, if $a_1 = y_{m+1}$, then $a_2 = v_{m+1}$ since $f(a_1, a_2) > 0$. Again y_{m+1} in G_1^m and v_{m+1} in G_2^m have similar outgoing edges. Clearly (a_1, c_1, a_2, c_2) is valid by construction. The validity of (c_1, b_1, c_2, b_2) is not so evident. Note, however, that $b_1 = y_2$ iff $b_2 = u_2$ since $f(b_1, b_2) > 1$. Thus conditions (c) and (d) for the validity of (c_1, b_1, c_2, b_2) are trivially satisfied. Thus we only have to show that (c_1, b_1, c_2, b_2) satisfies the Atoms condition.

$$\begin{aligned} b_1 = c_1 &\iff (a_1, b_1) \text{ is an edge} && \text{(since } (a_1, c_1) \text{ is an edge)} \\ &\iff (a_2, b_2) \text{ is an edge} && \text{(since } (a_1, b_1, a_2, b_2) \text{ is valid)} \\ &\iff b_2 = c_2 && \text{(since } (c_1, c_2) \text{ is valid and } (a_2, c_2) \text{ is an edge)} \end{aligned}$$

Suppose (c_1, b_1) is also an edge, then $c_1 \in \{x_2, y_1, z_2\}$ because these are the only nodes with incoming as well as outgoing edges. If $c_1 = x_2$, then $c_2 = x_2'$, and $b_1 = y_1$, whence $b_2 = u_1$ since $f(b_1, b_2) > 0$. On the other hand, if $c_1 = y_1$, then $a_1 = x_2$, $c_2 = u_1$, $b_1 = y_2$, and $a_2 = x_2'$. Now by conditions (c) and (d) from the validity of (a_1, b_1, a_2, b_2) we have that $b_2 = u_2$. Finally, if $c_1 = z_2$, then $c_2 = z_2'$ and $b_1 = z_1$, whence $b_2 = z_1'$ since (a_1, b_1, a_2, b_2) is valid. In either case, (c_2, b_2) is an edge as desired.

On the other hand suppose that (c_2, b_2) is an edge, then $c_2 \in \{x_2', u_1, z_2'\}$ because these are the only nodes with incoming as well as outgoing edges. If $c_2 = x_2'$, then $c_1 = x_2$, and $b_2 = u_1$, whence $b_1 = y_1$ since $f(b_1, b_2) > 0$. On the other hand, if $c_2 = u_1$, then $c_1 = y_1$, $a_2 = x_2'$ and $b_2 = u_2$. Now by conditions (c) and (d) from the validity of (a_1, b_1, a_2, b_2) we have that $b_2 = y_2$. Finally, if $c_2 = z_2'$, then $c_1 = z_2$ and $b_2 = z_1'$, whence $b_2 = z_1$ since (a_1, b_1, a_2, b_2) is valid. In either case, (c_1, b_1) is an edge as desired.

So it remains to be shown that $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. Since (a_1, c_1) is an edge, it is clear that c_1 is in the column to the left or right of a_1. Thus if $f(a_1, a_2) < m/2$, we must have that $f(c_1, c_2) \geq f(a_1, a_2) - 1 \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. On the other hand, suppose that $f(a_1, a_2) = m/2$. Let us list the possibilities for $f(a_1, a_2)$ to equal $m/2$: the column of a_1 is $m/2 + 1$; a_1 is Y left and a_2 is U left; a_1 is W left and a_2 is W' left; a_1 is W left and a_2 is V left; a_1 is T left and a_2 is W' left; a_1 is T left and a_2 is V left; a_1 is Y right and a_2 is V right; a_1 is W right and a_2 is W' right; a_1 is W right and a_2 is U right; a_1 is T right and a_2 is W' right; or a_1 is T right and a_2 is U right. Therefore, unless $a_1 \in \{y_{\frac{m}{2}+1}, t_{\frac{m}{2}+1}, w_{\frac{m}{2}+1}\}$, c_1 is on the same side of the chain as a_1, and c_2 is on the same side (left or right) of the chain as a_2 since (a_1, c_1) and (a_2, c_2) are edges. The definition of f implies that $f(c_1, c_2) = m/2$. If $a_1 \in \{y_{\frac{m}{2}+1}, t_{\frac{m}{2}+1}, w_{\frac{m}{2}+1}\}$, then the column of c_1 and c_2 is $m/2$ or $m/2 + 2$ since (a_1, c_1) and (a_2, c_2) are edges. Therefore

$f(c_1, c_2) \geq m + 1 - (m/2 + 2) = m/2 - 1$ as desired.

The case where (c_1, b_1) is an edge is analogous to the case where (a_1, c_1) is an edge. □

Notice that three consecutive columns in G_1^m are isomorphic to the three corresponding columns in G_2^m displayed in Figure 2. Hence we can exchange the roles of c_1 and c_2 in the proof of the previous lemma without violating the Atoms condition since the Atoms condition can only fail if there is a problem on the columns directly surrounding c_1 and c_2. Furthermore, notice that the value of $f(a_1, a_2)$ only depends on how a_1 and a_2 relate to one another on one side of the graph (the left or right-hand side). Hence, the condition on $f(c_1, c_2)$ also remains intact, since G_1^m and G_2^m look completely the same on the left-hand (right-hand) side. Thus the proof of the following lemma is analogous to the proof of Lemma 6.6.

Lemma 6.7. *Suppose that (a_1, b_1, a_2, b_2) is valid, $f(a_1, a_2) > 0$, $f(b_1, b_2) > 1$ and $c_2 \in \mathrm{adom}(G_2^m)$ such that $a_2 = c_2$, $b_2 = c_2$, (a_2, c_2) is an edge, or (c_2, b_2) is an edge. Then there exists $c_1 \in \mathrm{adom}(G_1^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) are valid, and $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Let us now take care of steps two and three in the intuitive strategy outlined before Lemma 6.6, i.e., when c_1 is not related to a_1 or b_1.

Lemma 6.8. *Suppose that (a_1, b_1, a_2, b_2) is valid, $f(a_1, a_2) > 0$, $f(b_1, b_2) > 1$ and $c_1 \in \mathrm{adom}(G_1^m)$ such that $a_1 \neq c_1$, $c_1 \neq b_1$, (a_1, c_1) and (c_1, b_1) are not edges. Then there exists $c_2 \in \mathrm{adom}(G_2^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) are valid, and $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Proof. The goal is to follow the following strategy, unless it breaks the Atoms condition for (a_1, c_1, a_2, c_2) or (c_1, b_1, a_2, c_2). Henceforth we will refer to this strategy as the *Greedy Strategy*.

$$\begin{aligned}
c_1 = z_i \wedge 1 \leq i \leq 2 &\implies c_2 = z_i' \\
c_1 = x_i \wedge 1 \leq i \leq 2 &\implies c_2 = x_i' \\
c_1 = y_i \wedge 0 \leq i \leq m/2 + 1 &\implies c_2 = u_i \\
c_1 = y_i \wedge m/2 + 1 < i \leq m + 1 &\implies c_2 = v_i \\
c_1 = w_i &\implies c_2 = w_i' \\
c_1 = t_i \wedge 0 \leq i \leq m/2 + 1 &\implies c_2 = v_i \\
c_1 = t_i \wedge m/2 + 1 < i \leq m + 1 &\implies c_2 = u_i.
\end{aligned}$$

The reason why we use this strategy is because in this case $f(c_1, c_2) = m/2$, in which case it is trivial that $f(c_1, c_2) \geq \min\{f(a_1, a_2), f(b_1, b_2)\} - 1$.

First, we establish that the Atoms conditions cannot be broken in the following situations: $c_1 = y_1$; $c_1 = y_{m+1}$; $c_1 = z_i$ with $i = 1, 2$; $c_1 = x_i$ with $i = 1, 2$; or $(a_1, c_1) = (x_2, y_2)$. To prove this, suppose first that $c_1 = y_1$; then by the strategy outlined above $c_2 = u_1$.

- If (a_2, c_2) is an edge then $a_2 = x_2'$, whence $a_1 = x_2$ since (a_1, b_1, a_2, b_2) is valid. Thus (a_1, c_1) is also an edge, which is a contradiction.
- If $a_2 = c_2$ then $a_2 = u_1$, whence $a_1 = y_1$ since $f(a_1, a_2) > 0$. Thus $a_1 = c_1$ which is a contradiction.
- $\star$ If (c_2, b_2) is an edge then $b_2 = u_2$, whence $b_1 = y_2$ since $f(b_1, b_2) > 1$. Thus (c_1, b_1) is also an edge, which is a contradiction. (This item is specially marked with $\star$ for later reference in the proof of Lemma 6.20.)
- If $b_2 = c_2$ then $b_2 = u_1$, whence $b_1 = y_1$ since $f(b_1, b_2) > 0$. Thus $c_1 = b_1$ which is a contradiction.

So, when $c_1 = y_1$ the chosen c_2 does not break the Atoms conditions.

Next suppose that $c_1 = y_{m+1}$; then by the Greedy Strategy $c_2 = v_{m+1}$.

- (a_2, c_2) cannot be an edge since v_{m+1} has no incoming edges.
- If $a_2 = c_2$ then $a_2 = v_{m+1}$, whence $a_1 = y_{m+1}$ since $f(a_1, a_2) > 0$. Thus $a_1 = c_1$ which is a contradiction.
- If (c_2, b_2) is an edge then $b_2 = z_2'$, whence $b_1 = z_2$ since (a_1, b_1, a_2, b_2) is valid. Thus (c_1, b_1) is also an edge, which is a contradiction.
- If $c_2 = b_2$ then $b_2 = v_{m+1}$, whence $b_1 = y_{m+1}$ since $f(b_1, b_2) > 0$. Thus $b_1 = c_1$ which contradicts the given.

Next suppose that $c_1 = x_2$; then by the Greedy Strategy $c_2 = x_2'$.

- If (a_2, c_2) is an edge, then $a_2 = x_1'$, whence $a_1 = x_1$ since (a_1, b_1, a_2, b_2). Thus (a_1, c_1) is also an edge, which is a contradiction.
- If $a_2 = c_2$ then $a_2 = x_2'$, whence $a_1 = x_2$. Thus $a_1 = c_1$ which is a contradiction.
- If (c_2, b_2) is an edge then $b_2 = u_1$, whence $b_1 = y_1$ since $f(b_1, b_2) > 0$. Thus (c_1, b_1) is an edge which is a contradiction.
- If $b_2 = c_2$ then $b_2 = x_2'$, whence $b_1 = x_2$ since (a_1, b_1, a_2, b_2) is valid. Thus $b_1 = c_1$ which contradicts the given.

The situations where $c_1 = x_1$ or $c_1 = z_i$ with $i = 1, 2$ are similar to the previous case.

Finally, suppose that $(a_1, c_1) = (x_2, y_2)$; then by the Greedy Strategy $(a_2, c_2) = (x_2', u_2)$. Now, for the Atoms condition to be broken, we must have that $c_2 = b_2$ since u_2 only has outgoing edges. Thus $(a_1, b_1, a_2, b_2) = (x_2, b_1, x_2', u_2)$, whence $b_1 = y_2$ by condition (d) for the validity of (a_1, b_1, a_2, b_2). But then $c_1 = b_1$ which contradicts the given.

At this point, we may assume that the Atoms condition is broken if c_2 is picked according to the Greedy Strategy. By the arguments before, then, c_1 is not y_1, y_{m+1} or z_i, x_i for $i = 1, 2$, and $(a_1, c_1) \neq (x_2, y_2)$.

Furthermore, we do not have to consider cases where c_1 is in the middle column, or the two columns directly adjacent to it, i.e., the column directly to the left and right of the middle one. Indeed, since there are three chains in G_2^m, we can always pick another node c_2^{new} on the chain that does not contain a_2 and b_2. Thus $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ are certainly valid. Since c_1 and c_2^{new} is located on either of the three middle columns, we have that $f(c_1, c_2^{new}) \geq m/2 - 1 \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$ since $f(x, y)$ is at most $m/2$ for any pair of nodes $(x, y) \in \text{adom}(G_1^m) \times \text{adom}(G_2^m)$.

From here we will write c_2^{old} for the c_2 chosen by the Greedy Strategy.

We will split the proof into several sublemmas (Lemmas 6.9 to 6.16). First, in Lemmas 6.9 to 6.14 we show, for each case where the Atoms condition is broken, that we can pick a $c_2^{new} \in \text{adom}(G_2^m)$ such that conditions (a) and (b) for the validity of both $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ are satisfied, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. Then, in Lemmas 6.15 and 6.16 we show that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ also satisfy conditions (c) and (d) for validity.

Lemma 6.9. *If $a_2 = c_2^{old}$ or (a_2, c_2^{old}) is an edge, and c_1 is on Y then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Proof. If c_1 is Y left (respectively Y right), c_2^{old} is U left (respectively V right). Since c_1 is not in the middle three columns, $c_1 \notin \{x_1, x_2, y_1\}$, and $a_2 = c_2^{old}$ or (a_2, c_2^{old}) is an edge, we have that a_2 is also U left (respectively V right), whence $f(a_1, a_2) < m/2$ by definition. We now pick c_2^{new} on the chain that does not contain a_2 or b_2, in the same column as c_1, whence $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity. This is indeed possible since there are three chains. Thus we may conclude that c_2^{new} is not U left (respectively V right), and hence $f(c_1, c_2) < m/2$. Therefore, if $a_2 = c_2^{old}$, clearly $f(c_1, c_2^{new}) = f(a_1, a_2) < m/2$ by definition, since then c_1 is in the same column as a_1 and a_2. On the other hand, if (a_2, c_2^{old}) is an edge, then $f(c_1, c_2^{new}) \geq f(a_1, a_2) - 1$ by definition, since then c_1 is in one of the columns next to a_1 and a_2. Thus we may conclude that $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. □

The proof of the following lemma is similar to the proof of Lemma 6.9 where the roles of a_1 and a_2 are replaced by b_1 and b_2, and (a_2, c_2) being an edge is replaced by (c_2, b_2) being an edge.

Lemma 6.10. *If $b_2 = c_2^{old}$ or (c_2^{old}, b_2) is an edge, and c_1 is on Y then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Lemma 6.9 and 6.10 have considered the scenarios where the Atoms condition was broken when c_1 is located on Y. The scenarios when c_1 is located on W are handled by Lemmas 6.11 and 6.12, and the scenarios when c_1 is located on

T are handled by Lemmas 6.13 and 6.14. We now have a look at the scenarios where c_1 is located on W.

Lemma 6.11. *If $a_2 = c_2^{old}$ or (a_2, c_2^{old}) is an edge, and c_1 is on W then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that conditions (a) and (b) for the validity of both $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ are satisfied, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Proof. In this case c_2^{old} is on W', whence a_2 is also on W' since $a_2 = c_2^{old}$, or (a_2, c_2^{old}) is an edge. Since $a_1 \neq c_1$ and (a_1, c_1) is not an edge, we have that a_1 is on Y or on T. If a_1 is on Y, then $f(a_1, a_2) < m/2$ since c_1 is not in the three middle columns. Hence whatever new c_2^{new} we pick such that (c_1, c_2) is valid, we have $f(c_1, c_2^{new}) \geq f(a_1, a_2) - 1$ since c_1 and c_2^{new} are either located in the same column as, or in the column next to a_1 and a_2. Thus, if we pick c_2^{new} on the chain that does not contain a_2 and b_2, in the same column as c_1, we have that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.

On the other hand, suppose that a_1 is on T then $f(a_1, a_2) = m/2$. This could be problematic if a_1 is T left (respectively T right) and if we cannot put c_2^{new} on the left side of V (respectively the right side of U), in the same column as c_1, simultaneously. That is, if putting c_2^{new} on the left side of V, in the same column as c_1, (respectively right side of U) makes $b_2 = c_2^{new}$ or (c_2^{new}, b_2) an edge. If this is not the case, then we simply put c_2^{new} on V, in the same column as c_1 (respectively U). Then by construction $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ satisfy conditions (a) and (b) for validity and $f(c_1, c_2) = m/2$.

In the problematic case we will show that $f(b_1, b_2)$ is sufficiently low. So in this case putting c_2^{new} in the same column as c_1 on the left side of V (respectively right side of U) violates the Atoms condition for $(c_1, b_1, c_2^{new}, b_2)$. Then b_2 is V left (respectively U right), in the same column as, or in the column next to c_1 and c_2^{old}. Since $c_2^{old} = a_2$ or (a_2, c_2^{old}) is an edge, a_2 must be on W' as well. This implies that $a_2 \neq b_2$ and that (a_2, b_2) is not an edge, since b_2 is on V (respectively U) as mentioned before. Therefore, by the validity of (a_1, b_1, a_2, b_2), we can also conclude that $a_1 \neq b_1$ and that (a_1, b_1) is not an edge. Thus b_1 is certainly not on T since then $a_1 = b_1$ or (a_1, b_1) would be an edge. It cannot be on W either because then $c_1 = b_1$ or (c_1, b_1) would be an edge, which contradicts the given. Thus we may conclude that in this case b_1 is on Y, whence $f(b_1, b_2) < m/2$ since b_1 is V left (respectively U right). If we now put c_2^{new} on the chain that does not contain a_2 or b_2, in the same column as c_1, then $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ certainly satisfy conditions (a) and (b) for validity, and we have that $f(c_1, c_2^{new}) \geq f(b_1, b_2) - 1 \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$ since c_1 and c_2^{new} are either in the column next to, or in the same column as b_1 or b_2. For an example of this scenario see Figure 7. □

The proof of the following lemma is similar to the proof of Lemma 6.11 where the roles of a_1 and a_2 are replaced by b_1 and b_2, and (a_2, c_2) being an edge is replaced by (c_2, b_2) being an edge.

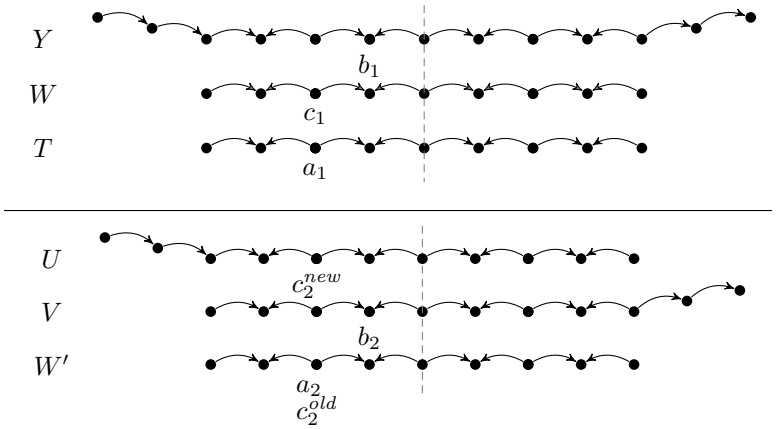

Figure 7: An example of a problem scenario in Lemma 6.11. Clearly c_2^{old} breaks the Atoms condition. Furthermore, if we would have picked c_2^{new} on V, (c_2^{new}, b_2) would have been an edge, which is not allowed. Thus we are forced to pick c_2^{new} on U. This, however, is no problem since in this scenario b_1 and b_2 are on sides of chains with different endings.

Lemma 6.12. *If $c_2^{old} = b_2$ or (c_2^{old}, b_2) is an edge, and c_1 is on W then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

As announced we now look at the scenarios when c_1 is located on T. The reasoning used to prove the following lemma is again analogous to the proof of Lemma 6.11, but since the Greedy Strategy deviates in this scenario compared to the scenario of Lemma 6.11, we need to address some detailed differences.

Lemma 6.13. *If $a_2 = c_2^{old}$ or (a_2, c_2^{old}) is an edge, and c_1 is on T then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Proof. If c_1 is T left, then c_2^{old} is V left, while if c_1 is T right, then c_2^{old} is U right. Furthermore, if c_2^{old} is V left, then a_2 is also V left, and if c_2^{old} is U right, a_2 is also U right. This is because $a_2 = c_2^{old}$ or (a_2, c_2^{old}) is an edge, and c_1 is not located in the middle three columns. Since $a_1 \neq c_1$ and (a_1, c_1) is not an edge, we have that a_1 is on Y or on W. If a_1 is on Y then $f(a_1, a_2) < m/2$ since c_1 is not in the three middle columns. Hence whatever new c_2^{new} we pick in the same column as c_1 we have $f(c_1, c_2^{new}) \geq f(a_1, a_2) - 1$. Thus, if we pick c_2^{new} on the chain that does not contain a_2 and b_2, in the same column as c_1, we have that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.

On the other hand, suppose that a_1 is on W, then $f(a_1, a_2) = m/2$. This could be problematic if a_1 is W left (respectively W right) and if we cannot

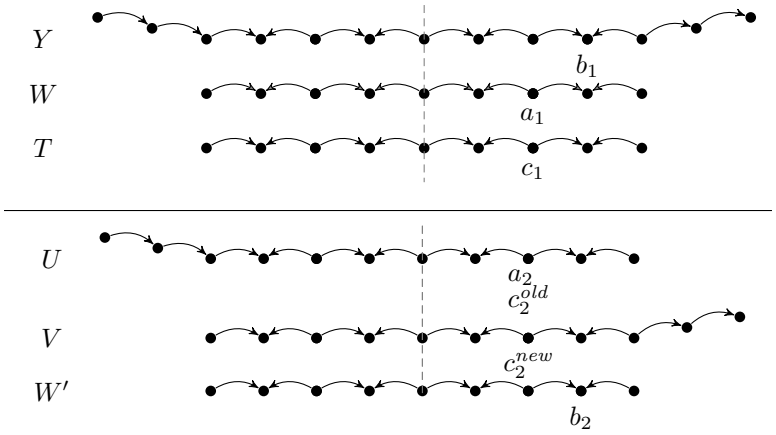

Figure 8: An example of a problem scenario in Lemma 6.13. Clearly c_2^{old} breaks the Atoms condition. Furthermore, if we would have picked c_2^{new} on W', (c_2^{new}, b_2) would have been an edge, which is not allowed. Thus we are forced to pick c_2^{new} on V. This, however, is no problem since in this scenario b_1 and b_2 are on sides of chains with different endings.

put c_2^{new} on W', in the same column as c_1, simultaneously, i.e., if putting c_2^{new} on W', in the same column as c_1, makes $b_2 = c_2^{new}$ or (c_2^{new}, b_2) an edge. If this is not the case we simply put c_2^{new} on W', in the same column as c_1. Then by construction $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2) = m/2$.

In the problematic case we will show that $f(b_1, b_2)$ is sufficiently low. So in this case putting c_2^{new} on W', in the same column as c_1, violates the Atoms condition for $(c_1, b_1, c_2^{new}, b_2)$. Then b_2 is on located on W', in the same column as, or in the column next to c_1 and c_2^{old}. Since $c_2^{old} = a_2$ or (a_2, c_2^{old}) is an edge, and c_1 is not in the middle three columns, a_2 must be on V if c_1 is T left, or on U if c_1 is U right. In either case, this implies that $a_2 \neq b_2$ and that (a_2, b_2) is not an edge, since b_2 is on W' as mentioned before. Therefore, by the validity of (a_1, b_1, a_2, b_2), we can also conclude that $a_1 \neq b_1$ and that (a_1, b_1) is not an edge. Thus b_1 is certainly not on W since then $a_1 = b_1$ or (a_1, b_1) would be an edge. It cannot be on T either because then $c_1 = b_1$ or (c_1, b_1) would be an edge, which contradicts the given. Thus we may conclude that in this case b_1 is on Y, whence $f(b_1, b_2) < m/2$ since b_1 is W'. If we now put c_2^{new} on the chain that does not contain a_2 or b_2, in the same column as c_1, then $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ certainly satisfy conditions (a) and (b) for validity, and we have that $f(c_1, c_2^{new}) \geq f(b_1, b_2) - 1 \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$ since c_1 and c_2^{new} are either in the column next to, or in the same column as b_1 or b_2. For an example of this scenario see Figure 8. □

The proof of the following lemma is similar to the proof of Lemma 6.13 where the roles of a_1 and a_2 are replaced by b_1 and b_2, and (a_2, c_2) being an edge is

replaced by (c_2, b_2) being an edge.

Lemma 6.14. *If $c_2^{old} = b_2$ or (c_2^{old}, b_2) is an edge, and c_1 is on T then there exists $c_2^{new} \in \text{adom}(G_2^m)$ such that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ both satisfy conditions (a) and (b) for validity, and $f(c_1, c_2^{new}) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$.*

Together Lemma 6.9 to 6.14 cover all scenarios for c_1 where one of the Atoms conditions was broken. Thus, all that remains to establish Lemma 6.8 is to show that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ satisfy conditions (c) and (d) for validity. Let us first take care of $(a_1, c_1, a_2, c_2^{new})$.

Lemma 6.15. *Let $c_2^{new} \in \text{adom}(G_2^m)$ be the node chosen in Lemmas 6.9 to 6.14. Then $(a_1, c_1, a_2, c_2^{new})$ also satisfies conditions (c) and (d) for validity.*

Proof. Condition (c) is only involved when $(a_1, c_1) = (x_2, y_2)$, a case we have already excluded at the start of the proof.

Condition (d) is only involved when $(a_2, c_2^{new}) = (x_2', u_2)$. Since (a_1, b_1, a_2, b_2) is valid, we must have that $a_1 = x_2$, whence $f(a_1, a_2) = m/2$. We now show that $c_1 = y_2$. Suppose for the sake of contradiction that $c_1 \neq y_2$. Then by definition $f(c_1, c_2^{new}) = 1$. Furthermore, $c_2^{old} = v_2$ or $c_2^{old} = w_2'$ by the Greedy Strategy. Since $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1 = \min(m/2, f(b_1, b_2)) - 1 = f(b_1, b_2) - 1$ by assumption, we have $f(b_1, b_2) \leq 2$. Remember that the Atoms condition for either $(a_1, c_1, a_2, c_2^{old})$ or $(c_1, b_1, c_2^{old}, b_2)$ was broken. Notice that in this case the Atoms condition for $(a_1, c_1, a_2, c_2^{old})$ was not broken, since c_1 and c_2^{old} are two columns to the right of a_1 and a_2. Thus the Atoms condition for $(c_1, b_1, c_2^{old}, b_2)$ was broken. Hence $c_2^{old} = b_2$ or (c_2^{old}, b_2) is an edge (because by assumption c_1 is not related to b_1). It is not possible for (c_2^{old}, b_2) to be an edge since v_2 and w_2' have no outgoing edges. Thus we may conclude that $c_2^{old} = b_2 = v_2$ or $c_2^{old} = b_2 = w_2'$. Hence $b_1 = y_2$ in both cases since $f(b_1, b_2) \leq 2$. Therefore $(a_1, b_1, a_2, b_2) = (x_2, y_2, x_2', b_2)$ where $b_2 = v_2$ or w_2', which contradicts condition (c) for the validity of (a_1, b_1, a_2, b_2). □

Finally, we take care of $(c_1, b_1, c_2^{new}, b_2)$.

Lemma 6.16. *Let $c_2^{new} \in \text{adom}(G_2^m)$ be the node chosen in Lemmas 6.9 to 6.14. Then $(c_1, b_1, c_2^{new}, b_2)$ also satisfies conditions (c) and (d) for validity.*

Proof. Condition (c) is only involved when $b_1 = y_2$. Then $b_2 = u_2$ since $f(b_1, b_2) > 1$, as desired.

Condition (d) is only involved when $b_2 = u_2$. Then $b_1 = y_2$ since $f(b_1, b_2) > 1$, as desired. □

Together Lemmas 6.15 and 6.16 establish that both $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ also satisfy conditions (c) and (d) for validity. Since we already established that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ satisfy conditions (a) and (b) for validity, we may conclude that $(a_1, c_1, a_2, c_2^{new})$ and $(c_1, b_1, c_2^{new}, b_2)$ are both valid, which concludes the proof of Lemma 6.8. □

The proof of the following lemma is analogous to the proof of Lemma 6.8, this is because of the same reasons why the proof of Lemma 6.7 was analogous to the proof of Lemma 6.6.

Lemma 6.17. *Suppose that* (a_1, b_1, a_2, b_2) *is valid,* $f(a_1, a_2) > 0$*,* $f(b_1, b_2) > 1$ *and* $c_2 \in \text{adom}(G_2^m)$ *such that* $a_1 \neq c_1$*,* $c_1 \neq b_1$*,* (a_1, c_1) *and* (c_1, b_1) *are not edges. Then there exists* $c_1 \in \text{adom}(G_1^m)$ *such that* (a_1, c_1, a_2, c_2) *and* (c_1, b_1, c_2, b_2) *are valid, and* $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$*.*

Combining Lemmas 6.6 and 6.8 we get the following corollary.

Corollary 6.18. *If* (a_1, b_1, a_2, b_2) *is valid,* $f(a_1, a_2) > 0$ *and* $f(b_1, b_2) > 1$*, then for every* $c_1 \in \text{adom}(G_1^m)$ *there exists* $c_2 \in \text{adom}(G_2^m)$ *such that* (a_1, c_1, a_2, c_2) *and* (c_1, b_1, c_2, b_2) *are valid, and* $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$*.*

We will see later that this corollary is crucial to show that the duplicator has a winning strategy starting in (a_1, b_1, a_2, b_2).

On the other hand, combining Lemmas 6.7 and 6.17 yields the following corollary.

Corollary 6.19. *If* (a_1, b_1, a_2, b_2) *is valid,* $f(a_1, a_2) > 0$ *and* $f(b_1, b_2) > 1$*, then for every* $c_2 \in \text{adom}(G_2^m)$ *there exists* $c_1 \in \text{adom}(G_1^m)$ *such that* (a_1, c_1, a_2, c_2) *and* (c_1, b_1, c_2, b_2) *are valid, and* $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$*.*

Before we can finally start the bisimulations needed for the proof of Proposition 5.4, notice that until now we have always required that $f(b_1, b_2) > 1$. The cases where $f(b_1, b_2) = 1$ are handled separately. Indeed, when $f(b_1, b_2) = 1$, we cannot necessarily guarantee that (c_1, b_1, c_2, b_2) is valid (see Figure 9). We can only guarantee the Atoms condition as shown Lemmas 6.20 and 6.21. This will turn out to be sufficient.

Lemma 6.20. *Suppose that* (a_1, b_1, a_2, b_2) *is valid,* $f(a_1, a_2) > 0$*,* $f(b_1, b_2) = 1$*. Then, for every* $c_1 \in \text{adom}(G_1^m)$ *there exists* $c_2 \in \text{adom}(G_2^m)$ *such that* (a_1, c_1, a_2, c_2) *and* (c_1, b_1, c_2, b_2) *satisfy the Atoms condition.*

Proof. Careful inspection of the proofs of Lemmas 6.6 and 6.8 reveals that $f(b_1, b_2) > 1$ is only used for showing conditions (c) and (d) for the validity of (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2), except in the case where $c_1 = y_1$ and $b_2 = u_2$ (item marked with $\star$ in the proof of Lemma 6.8). If we are not in this case, we can simply pick the same c_2 as in these proofs.

Now suppose we are in this exceptional case. Since $f(b_1, b_2) = 1$, b_1 is not on Y. Notice that (a_1, c_1) cannot be an edge, since then $a_1 = x_2$, and hence also $a_2 = x_2'$ since (a_1, b_1, a_2, b_2) is valid. Thus we have $(a_1, b_1, a_2, b_2) = (x_2, b_1, x_2', u_2)$. Condition (d) for the validity of (a_1, b_1, a_2, b_2) then implies that $b_1 = y_2$, which contradicts the fact that b_1 is not on Y.

If $a_1 = c_1$, then we pick $a_2 = c_2$. Notice that in this case $b_2 \neq c_2$. Indeed, if $b_2 = c_2 = a_2$, then $a_1 = b_1$ by the validity of (a_1, b_1, a_2, b_2). Thus $c_1 = b_1$ which is a contradiction.

On the other hand, if $a_1 \neq c_1$, we simply pick c_2 on the chain not containing a_2 or b_2, in the same column as c_1. This is possible since there are three chains.□

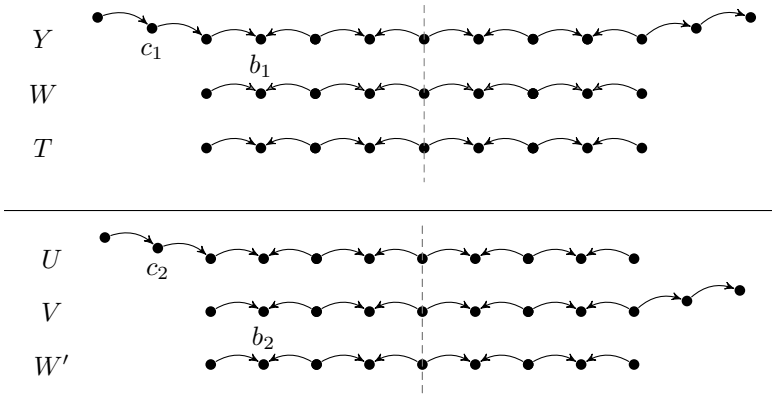

Figure 9: An example of a problem scenario where we are forced to pick a c_2 such that (c_1, b_1, c_2, b_2) does not satisfy condition (c) for validity. It turns out that it is sufficient to only satisfy the Atoms condition because this scenario only occurs when $f(b_1, b_2) = 1$.

The proof of the following lemma is analogous to the proof of the previous lemma. This is because of the same reasons why the proof of Lemma 6.7 was analogous to the proof of Lemma 6.6.

Lemma 6.21. *Suppose that (a_1, b_1, a_2, b_2) is valid, $f(a_1, a_2) > 0$, $f(b_1, b_2) = 1$. Then, for every $c_2 \in \mathrm{adom}(G_2^m)$ there exists $c_1 \in \mathrm{adom}(G_1^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) satisfy the Atoms condition.*

We are now ready to show our key Lemma.

Lemma 6.22. *Let s be a natural number and let $m > 4$ be a natural number divisible by four. If $(a_1, b_1, a_2, b_2) \in \mathrm{adom}(G_1^m)^2 \times \mathrm{adom}(G_2^m)^2$ is valid and $s \leq \min(f(a_1, a_2), f(b_1, b_2))$, then $(G_1^m, a_1, b_1) \simeq_s (G_2^m, a_2, b_2)$.*

Proof. We prove this lemma by induction on s. If $s = 0$ then, $(G_1^m, a_1, b_1) \simeq_s (G_2^m, a_2, b_2)$ since the Atoms condition is implied by the validity of (a_1, b_1, a_2, b_2).

Now let $s > 0$, so both $f(a_1, a_2) > 0$ and $f(b_1, b_2) > 0$. If $f(b_1, b_2) = 1$ then Lemma 6.20 implies that for every $c_1 \in \mathrm{adom}(G_1^m)$, there exists $c_2 \in \mathrm{adom}(G_2^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_2, c_2, b_2) satisfy the Atoms condition. This, however, is equivalent to

$$(G_1^m, a_1, c_1) \simeq_0 (G_2^m, a_2, c_2) \quad \text{and} \quad (G_1^m, c_1, b_1) \simeq_0 (G_2^m, c_2, b_2).$$

Hence the Forth condition holds. Furthermore, Lemma 6.21 implies that for every $c_2 \in \mathrm{adom}(G_2^m)$, there exists $c_1 \in \mathrm{adom}(G_1^m)$ such that (a_1, c_1, a_2, c_2) and (c_1, b_2, c_2, b_2) both satisfy the Atoms condition. Again this is equivalent to $(G_1^m, a_1, c_1) \simeq_0 (G_2^m, a_2, c_2)$ and $(G_1^m, c_1, b_1) \simeq_0 (G_2^m, c_2, b_2)$. Hence the Back condition holds. Thus $(G_1^m, a_1, b_1) \simeq_1 (G_2^m, a_2, b_2)$.

Now suppose that $f(a_1, a_2) > 0$ and $f(b_1, b_2) > 1$. We will first show that the Forth condition holds. Suppose that $c_1 \in \mathrm{adom}(G_1^m)$. Then by Corollary 6.18 there exists $c_2 \in \mathrm{adom}(G_2^m)$ such that both (a_1, c_1, a_2, c_2) and (c_1, b_1, c_2, b_2) are valid and $f(c_1, c_2) \geq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. Furthermore, $f(c_1, c_2) \geq s - 1$ since $s - 1 \leq \min(f(a_1, a_2), f(b_1, b_2)) - 1$. Hence $s - 1 \leq \min(f(c_1, c_2), f(a_1, a_2))$ and $s - 1 \leq \min(f(c_1, c_2), f(b_1, b_2))$. Therefore we can apply our induction hypothesis, which tells us that $(G_1^m, a_1, c_1) \simeq_{s-1} (G_2^m, a_2, c_2)$ and $(G_1^m, c_1, b_1) \simeq_{s-1} (G_2^m, c_2, b_2)$ as desired.

The Back condition is verified similarly using Corollary 6.19. □

Theorem 6.4 finally follows:

Proof of Theorem 6.4. First, if $(a_1, b_1) = (y_{m/2+1}, y_{m/2+2})$, then we pick the pair $(a_2, b_2) = (u_{m/2+1}, u_{m/2+2})$. In this case (a_1, b_1, a_2, b_2) is valid, $f(a_1, a_2) = m/2$ and $f(b_1, b_2) = m + 1 - (m/2 + 2) = m/2 - 1$ and thus $(G_1, a_1, b_1) \simeq_{m/2-1} (G_2, a_2, b_2)$ due to Lemma 6.22.

If $(a_1, b_1) \neq (y_{m/2+1}, y_{m/2+2})$ then we use the following strategy:

$$\begin{aligned} a_1 = y_i \wedge 0 \leq i \leq m/2 + 1 &\implies a_2 = u_i \\ a_1 = y_i \wedge m/2 + 1 < i \leq m + 1 &\implies a_2 = v_i \\ a_1 = w_i &\implies a_2 = w'_i \\ a_1 = t_i &\implies a_2 = w'_i \end{aligned}$$

We use the same strategy to determine b_2 from b_1. Clearly in this case (a_1, b_1, a_2, b_2) is valid, and $f(a_1, a_2) = f(b_1, b_2) = m/2$, whence $(G_1, a_1, b_1) \simeq_{m/2-1} (G_2, a_2, b_2)$ due to Lemma 6.22. □

The bisimulations that we use always require that (a_1, b_1, a_2, b_2) is valid. There might be a a bisimulation of a larger depth when we remove this restriction. It turns out that we can find an upper bound on the depth.

Proposition 6.23. *There is no bisimulation between* $(G_1^m, y_{\frac{m}{2}+1}, y_{\frac{m}{2}+1})$ *and* (G_2^m, a, b) *for any* $(a, b) \in \mathrm{adom}(G_1^m)^2$ *of depth* $3m/4 + 2$.

Proof. By Theorem 6.2 it suffices to show that there exists an expression $e \in \mathcal{N}(\setminus, di)$ of degree $3m/4 + 2$ such that $(y_{\frac{m}{2}+1}, y_{\frac{m}{2}+1}) \in e(G_1^m)$ and $(a, b) \notin e(G_2^m)$. To this end, define the following family of expressions:

$$\begin{aligned} e_0 &:= \pi_2(R^3) \\ e'_0 &:= \pi_1(R^2) \\ e_1 &:= \pi_1(R \circ e_0) \\ e_{n+1} &:= \pi_1(R \cap ((R \circ di) \circ (e_n \circ R))) && \text{(for } n > 1\text{)} \\ e'_{n+1} &:= \pi_1(R \cap ((R \circ di) \circ (e'_n \circ R))) && \text{(for } n > 0\text{)} \end{aligned}$$

For $n = 1, \ldots, m/2$, we have $(y_{2n+1}, y_{2n+1}) \in e_n(G_1^m)$ and $(y_{m+1-2n}, y_{m+1-2n}) \in e'_n(G_1^m)$. Thus we may also conclude that $(y_{\frac{m}{2}+1}, y_{\frac{m}{2}+1}) \in e_{m/4} \cap e'_{m/4}(G_1^m)$.

On the other hand, $e_n(G_2^m)$ only contains pairs of nodes on U, while $e'_n(G_2^m)$ only contains nodes on V for any $n = 1 \dots m/2$. Hence $e_n \cap e'_n(G_2^m)$ is empty for $n = 1, \dots, m/2$. Thus we may conclude that $e_{m/4} \cap e'_{m/4}(G_2^m)$ is empty, and thus does not contain (a, b) either.

Since e_n and e'_n have degree $3n + 2$, the degree of $e_{m/4} \cap e'_{m/4}$ is $3m/4 + 2$ as desired. □

6.2 Inexpressibility of the query $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$

Using the established bisimulation in Theorem 6.4 and the characterization in Theorem 6.2 we can finally show that $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is not expressible in $e \in \mathcal{N}(\setminus, di, ^+)$

Proposition 6.24. *The boolean query $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is not expressible in $\mathcal{N}(\setminus, di, ^+)$.*

Proof. Suppose that q denotes the boolean query $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$. Assume for the sake of contradiction that q is expressible by an expression $e \in \mathcal{N}(\setminus, di, ^+)$. Define $\mathcal{G}_n$ as the class of graphs with an active domain of size at most n and define e_n as the expression e where every subexpression of the form f^+ in e is replaced with $\cup_{i=1}^n f^i$. Remember that expressions of the form f^+ are equivalent to the expression $\cup_{i=1}^n f^i$ when we only consider graphs in $\mathcal{G}_n$. Therefore e_n is equivalent to e on $\mathcal{G}_n$. Now, notice that if we carefully arrange the compositions in f^i, we obtain that $\mathit{degree}(\cup_{i=1}^n f^i) = \mathit{degree}(f) + \lceil \log_2 n \rceil$. Furthermore, note that in the worst case scenario every operation which contributes to the degree of e is a transitive closure application. Hence if we carefully arrange the compositions in e_n, we obtain that $\mathit{degree}(e_n) \leq d\lceil \log_2 n \rceil$ where d is the degree of e.

Observe that $|\operatorname{adom}(G_1^m)| = |\operatorname{adom}(G_2^m)| = 3m+7$ for every natural number m. By the argument above, we know that e_{3m+7} has degree at most $d\lceil \log_2(3m+7) \rceil$ for any natural number m. Moreover, there has to exist a natural number m such that $d\lceil \log_2(3m+7) \rceil \leq (3m+7)/6 - 3 < m/2 - 1$. The first inequality holds since there clearly exists a natural number l' such that for every $l \geq l'$: $d\lceil \log_2 l \rceil \leq l/6 - 3$, and the second inequality holds since $(3m+7)/6 - 3 = m/2 - 11/6 < m/2 - 1$. Hence the degree of e_{3m+7} is less than $m/2 - 1$, i.e., $e_{3m+7} \in \mathcal{N}(di, \setminus)_{m/2-1}$. Since $|\operatorname{adom}(G_1^m)| = |\operatorname{adom}(G_2^m)| = 3m + 7$, we know that e_{3m+7} agrees with e on G_1^m and G_2^m. Since $e \neq \emptyset$ is supposed to express q, and since $q(G_1^m)$ is clearly true, we have $e_{3m+7}(G_1^m) \neq \emptyset$. Thus let $(a_1, b_1) \in e_{3m+7}(G_1^m)$. By Theorem 6.4 there exists (a_2, b_2) such that $(G_1^m, a_1, b_1) \simeq_{m/2-1} (G_2^m, a_2, b_2)$. Then by Theorem 6.2 also $(a_2, b_2) \in e_{3m+7}(G_2^m)$. However, since $q(G_2^m)$ is clearly false, $e_{3m+7}(G_2^m)$ should be empty. We have thus obtained a contradiction. □

We are now ready to prove Proposition 5.4.

Proof. First observe that $\mathcal{N}(^{-1}, ^+) \leq^{\text{bool}} \mathcal{N}(F_1)$. Hence it suffices to prove that $\mathcal{N}(^{-1}, ^+) \not\leq^{\text{bool}} \mathcal{N}(F_2)$. Furthermore, since $F_2 \subseteq \overline{\{\setminus, di, ^+\}}$, it follows

from Theorem 3.1 that $\mathcal{N}(F_2) \leq^{\text{path}} \mathcal{N}(\setminus, di, ^+)$ and therefore also $\mathcal{N}(F_2) \leq^{\text{bool}} \mathcal{N}(\setminus, di, ^+)$. Thus it is sufficient to show that $\mathcal{N}(^{-1}, ^+) \not\leq^{\text{bool}} \mathcal{N}(\setminus, di, ^+)$.

The boolean query $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is clearly expressible in $\mathcal{N}(^{-1}, ^+)$. It is, however, not expressible in $\mathcal{N}(\setminus, di, ^+)$ by Proposition 6.24, which concludes our proof. □

6.3 Exponential blow-up on eliminating converse

In this section we will show that Theorem 6.3 holds. We will do so by employing the bisimulation result in Section 6.1.

Let a function $f : \mathbb{N} \to \mathbb{N}$ be given as in the statement of Theorem 6.3. Now suppose for the sake of contradiction that $f(n) = o(2^n)$. Let Q be the path query $R^2 \circ (R \circ R^{-1})^+ \circ R^2$. Define $\mathcal{G}_n$ as the class of graphs with an active domain of size at most n and define Q_n as the expression Q where every subexpression of the form f^+ in Q is replaced with $\cup_{i=1}^n f^i$. Remember that expressions of the form f^+ are equivalent to the expression $\cup_{i=1}^n f^i$ when we only consider graphs in $\mathcal{G}_n$. Therefore Q_n is equivalent to Q on $\mathcal{G}_n$=. As in the proof of Proposition 6.24, by carefully arranging the compositions in f^i, we obtain that $\mathit{degree}(\cup_{i=1}^n f^i) = \mathit{degree}(f) + \lceil \log_2 n \rceil$. Hence we can conclude that $\mathit{degree}(Q_n) = \lceil \log_2 n \rceil + 3$.

We now show that $f(\mathit{degree}(Q_n)) = o(n)$. Since $f(n) = o(2^n)$, we have by definition that $\lim_{n\to\infty} f(n)/2^n = 0$. Notice that $\mathit{degree}(Q_n)$ goes to infinity as n goes to infinity. Therefore, we have that $\lim_{n\to\infty} f(\mathit{degree}(Q_n))/2^{\mathit{degree}(Q_n)} = 0$ as well. We now show that this last limit implies that $f(\mathit{degree}(Q_n)) = o(n)$:

$$0 = \lim_{n\to\infty} \frac{f(\mathit{degree}(Q_n))}{2^{\mathit{degree}(Q_n)}} = \lim_{n\to\infty} \frac{f(\lceil \log_2 n \rceil + 3)}{2^{\lceil \log_2 n \rceil + 3}} \geq \lim_{n\to\infty} \frac{f(\lceil \log_2 n \rceil + 3)}{16n} \geq 0.$$

Notice that Q_n is an expression in $\mathcal{N}(^{-1})$, whence by assumption $h(Q_n)$ is an expression in $\mathcal{N}(\pi)$. We now show that there exists a natural number k such that for every $m \geq k$, $h(Q_{3m+7})$ is an expression in $\mathcal{N}(\pi)_{m/2-1}$.

Since $f(\mathit{degree}(Q_n)) = o(n)$, also $f(\mathit{degree}(Q_{3m+7})) = o(3m + 7)$. Furthermore, since $o(3m+7) = o(m/2-1)$, we may conclude that $f(\mathit{degree}(Q_{3m+7})) = o(m/2 - 1)$. Thus by definition, $\lim_{m\to\infty} f(\mathit{degree}(Q_n))/(m/2 - 1) = 0$. Hence

$$\forall \varepsilon > 0, \exists k \in \mathbb{N}, \forall m \in \mathbb{N} : m \geq k \Rightarrow \frac{f(\mathit{degree}(Q_{3m+7}))}{m/2 - 1} < \varepsilon.$$

Hence if we set $\varepsilon = 1$, we can find a k such that for every $m \geq k$ we have $f(\mathit{degree}(Q_{3m+7}))/(m/2 - 1) < 1$, or equivalently $f(\mathit{degree}(Q_{3m+7})) < m/2 - 1$. This implies that $\mathit{degree}(h(Q_{3m+7})) < m/2 - 1$ for any $m \geq k$ since it is given that $\mathit{degree}(h(Q_n)) \leq f(\mathit{degree}(Q_n))$ for any n. Thus we may conclude that $h(Q_{3m+7})$ is an expression in $\mathcal{N}(\pi)_{m/2-1}$ for any $m \geq k$.

Now let m be a multiple of four, greater then k, and let G_1^m be the top and G_2^m be the bottom graph in Figure 2. Since $|\operatorname{adom}(G_1^m)| = |\operatorname{adom}(G_2^m)| = 3m + 7$, we know that Q_{3m+7} agrees with Q on G_1^m and G_2^m. Thus $Q_{3m+7}(G_1^m) \neq \emptyset$ since $Q(G_1^m)$ is nonempty. Furthermore, because $h(Q_{3m+7})$ is equivalent to

Q_{3m+7} at the level of boolean queries, it must also be that $h(Q_{3m+7})(G_1^m) \neq \emptyset$. Thus let $(a_1, b_1) \in h(Q_{3m+7})(G_1^m)$. By Theorem 6.4 there exists (a_2, b_2) such that $(G_1^m, a_1, b_1) \simeq_{m/2-1} (G_2^m, a_2, b_2)$. Then by Theorem 6.2 also $(a_2, b_2) \in h(Q_{3m+7})(G_2^m)$. However, since $Q(G_2^m)$ is clearly empty, $Q_{3m+7}(G_2^m)$ as well as $h(Q_{3m+7})(G_2^m)$ should be empty. We have thus obtained a contradiction. Thus we may conclude that $f \neq o(2^n)$.

7 Boolean queries in the unlabeled case

In this section, we will prove Theorem 3.5. Here, the set Λ of edge labels is a singleton. In other words, a graph G is then a relational structure consisting of a set of nodes V and a simple relation $E(G) \subseteq V \times V$, the set of edges of G. As said before, we use the notation $\leq^{\text{bool}}_{\text{unl}}$ to compare the expressiveness of languages in this *unlabeled case*. It has been shown that at the level of boolean queries, a counterpart of Proposition 5.1 does not exist in this unlabeled case. That is, transitive closure does not always add expressive power in the unlabeled case [11].

Theorem 7.1 ([11]). *At the level of boolean queries in the unlabeled case, we have:*

$$\begin{aligned}
\mathcal{N}(^+) &\leq^{\text{bool}}_{\text{unl}} \mathcal{N}, \\
\mathcal{N}(\pi, ^+) &\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(\pi), \\
\mathcal{N}(di, ^+) &\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(di), \text{ and} \\
\mathcal{N}(di, \pi, ^+) &\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(di, \pi).
\end{aligned}$$

We will now show that in all other cases, transitive closure *does* add expressive power. We start by showing that it does in the presence of intersection or converse.

Proposition 7.2. *Let F_1 and F_2 be sets of nonbasic features.*

1. *If $^+ \in \overline{F}_1$, $\cap \in \overline{F}_1$, and $^+ \notin \overline{F}_2$, then $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$.*
2. *If $^+ \in \overline{F}_1$, $^{-1} \in \overline{F}_1$, and $^+ \notin \overline{F}_2$, then $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$.*

Before we prove this proposition we will first recall some basic terminology and notions concerning Hanf-locality [14]. Let G be an unlabeled graph, $a \in \text{adom}(G)$ and r a natural number. The ball with radius r around a is the set

$$B_r^G(a) = \{x \in \text{adom}(G) \mid d_G(x, a) \leq r\}$$

where $d_G(x, a)$ is defined as the length of the shortest undirected path between x and a. (An undirected path does not need to respect the direction of edges.) If no such path exists then $d_G(x, a)$ is defined as $+\infty$. The r-neighborhood of a in $\text{adom}(G)$, denoted by $N_r^G(a)$, is the pair (G', a) where the graph G' is defined as follows:

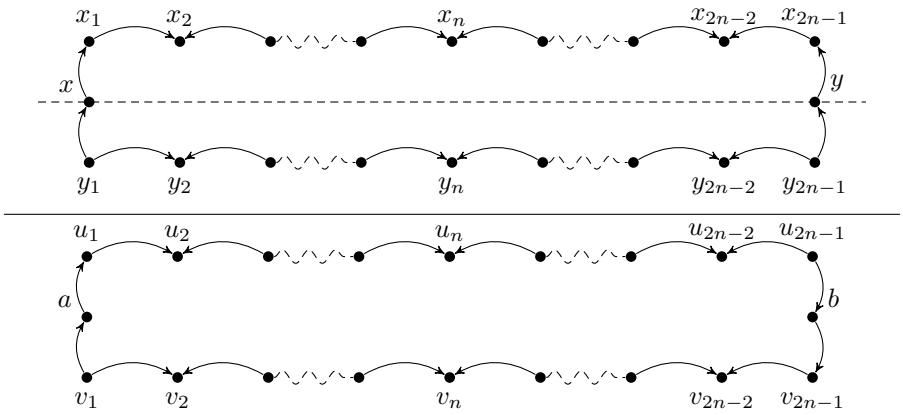

Figure 10: The only difference between the top graph and the bottom graph is that the edges incident to b in the bottom graph have the converse direction to the edges incident to y in the top graph.

- Its nodes are precisely $B_r^G(a)$;
- Its edge relation is $E(G) \cap (B_r^G(a) \times B_r^G(a))$.

For a node a in graph G_1 and a node b in graph G_2, we say that $N_r^{G_1}(a) = (G_3, a)$ is isomorphic to $N_r^{G_2}(b) = (G_4, b)$, denoted by $N_r^G(a) \cong N_r^G(b)$ if there is a graph isomorphism f from G_3 to G_4 such that $f(a) = b$.

Let G_1 and G_2 be graphs, and let d be a natural number. We write $G_1 \leftrightarrows_d G_2$ if there exists a bijection $f : \text{adom}(G_1) \to \text{adom}(G_2)$ such that for every $c \in \text{adom}(G_1)$: $N_d^{G_1}(c) \cong N_d^{G_2}(f(c))$.

A boolean query q is Hanf-local if there exists a natural number d such that $G_1 \leftrightarrows_d G_2$ implies $q(G_1) = \mathit{true} \Leftrightarrow q(G_2) = \mathit{true}$.

Theorem 7.3 ([14]). *Every boolean query expressible in first-order logic is Hanf-local.*

We are now ready to prove Proposition 7.2.

Proof of Proposition 7.2. For (1), it is well known that the query that checks whether a graph contains a cycle cannot be expressed in first order logic (see, e.g., [2]). The query, however, is expressed by $R^+ \cap id \neq \emptyset$.

For (2), we will show that $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is not expressible in $\mathcal{N}(F_2)$. Let G_1^n be the graph at the top and G_2^n be the graph at the bottom of Figure 10 for any natural number n greater than 1. Define f: $\text{adom}(G_1^n) \to \text{adom}(G_2^n)$ as

follows:

$$f(c) = \begin{cases} v_i & \text{if } c = x_i \text{ where } n < i \leq 2n-1; \\ u_i & \text{if } c = y_i \text{ where } n < i \leq 2n-1; \\ u_j & \text{if } c = x_j \text{ where } 1 \leq j \leq n; \\ v_j & \text{if } c = y_j \text{ where } 1 \leq j \leq n; \\ a & \text{if } c = x; \\ b & \text{if } c = y. \end{cases}$$

Intuitively, f mirrors the right hand nodes in G_1^n along the dotted line. We observe that $N_{n-1}^{G_1^n}(c) \cong N_{n-1}^{G_2^n}(f(c))$ for all $c \in \text{adom}(G_1^n)$ since $N_{n-1}^{G_1^n}(c) \cap \{(y, x_{2n-1}), (y_{2n-1}, y)\} \neq \emptyset$ implies $N_{n-1}^{G_2^n}(f(c)) \cap \{(a, u_1), (v_1, a)\} = \emptyset$, and $N_{n-1}^{G_1^n}(c) \cap \{(x, x_1), (y_1, x)\} \neq \emptyset$ implies $N_{n-1}^{G_2^n}(f(c)) \cap \{(u_{2n-1}, b), (b, v_{2n-1})\} = \emptyset$. Therefore $G_1^n \leftrightarrows_{n-1} G_2^n$ since f is a bijection. Now, suppose that the boolean query q expressed by $R^2 \circ (R \circ R^{-1})^+ \circ R^2 \neq \emptyset$ is expressible in $\mathcal{N}(F_2)$. Then certainly, it is also expressible in first-order logic. Hence q is Hanf-local by Theorem 7.3 and thus by definition, there has to exist a natural number d such that for every finite graphs A and B, $A \leftrightarrows_d B$ implies that q agrees on A and B. However, we established that $G_1^{d+1} \leftrightarrows_d G_2^{d+1}$, but $q(G_1^{d+1})$ is true and $q(G_2^{d+1})$ is false, which contradicts that q is Hanf-local. □

The two languages not covered by Theorem 7.1 and Proposition 7.2 are $\mathcal{N}(\overline{\pi}, ^+)$ and $\mathcal{N}(di, \overline{\pi}, ^+)$. These languages happen to be non-monotone[1]. We show that transitive closure does add expressive power for these non-monotone languages at the level of boolean queries in the unlabeled case.

Proposition 7.4. *Let F_1 and F_2 be sets of nonbasic features. If $^+ \in \overline{F_1}$, $\overline{\pi} \in \overline{F_1}$ and $^+ \notin F_2$, then $\mathcal{N}(F_1) \not\leq_{\text{unl}}^{\text{bool}} \mathcal{N}(F_2)$.*

Proof. The boolean query Q:"there is a non-sink node from which no sink node[2] can be reached" is expressible in $\mathcal{N}(\overline{\pi}, ^+)$ by the boolean query $\overline{\pi}_1((R^+ \circ \overline{\pi}_1(R)) \cup R) \neq \emptyset$. If this query would be expressible in $\mathcal{N}(F_2)$, it would also be expressible in first-order logic, which we show is impossible.

Suppose for the sake of contradiction that the first-order sentence ψ expresses the boolean query Q. We now show that this contradicts that the parity query is not expressible on linear chains in first-order logic [14]. Let $C_n = \{\{x_1, \ldots, x_n\}, \{E(x_i, x_{i+1}) \mid 1 \leq i < n\}\}$ be a linear chain with n nodes and define the graph G_n as follows:

- G_n contains n nodes, $x_1, \ldots, x_n$;
- Add an edge from x_i to x_{i+2} for every $i \in \{1, \ldots, n-2\}$;
- Add an edge from x_n to x_1.

[1] An expression e is monotone if $e(G) \not\subseteq e(G')$ for every two graphs G and G' such that $G \subseteq G'$. A language is non-monotone when it contains an expression that is not monotone.

[2] A sink node in a graph is a node in that graph with outdegree zero.

Let us now define a first-order formula $\varphi(x, y)$, such that $(a, b) \in G_n$ if and only if $C_n \models \varphi[a, b]$. Clearly

$$\varphi(x, y) := (\exists a : E(x, a) \wedge E(a, y)) \vee (\neg\exists a : E(x, a) \wedge \neg\exists a : E(a, y))$$

fulfills this property. Notice that G_n has a disjoint component in the form of a cycle if n is odd and that $G_n \cong C_n$ if n is even. Now, notice that if a graph has such a cycle component, it also has a non-sink node from which no sink node can be reached since there simply are no sink nodes on such a cycle. On the other hand, if a graph is isomorphic to C_n, every non-sink node can reach a sink node. Hence $G_n \models \neg\psi$ if and only if n is even.

Now let χ be the sentence formed by replacing each atomic sub-formula of the form $R(x, y)$ by $\varphi(x, y)$ in ψ. Then, $C_n \models \neg\chi$ if and only if n is even, which contradicts that the parity query on linear chains is not expressible in first-order logic. □

The reduction in the proof of the proposition above is based on the reduction of parity to connectivity [14].

We are now ready to prove Theorem 3.5.

Proof of Theorem 3.5. We first take care of the case where $^+ \in F_1$ and $^+ \notin F_2$. For the 'if' direction we may then suppose that the third condition holds since conditions one and two cannot hold. Since $F_1 \subseteq \{di, \pi, ^+\}$, we have $\mathcal{N}(F_1) \leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_1 \setminus \{^+\})$ by Theorem 7.1. Additionally, because we assumed that $F_1 \setminus \{^+\} \subseteq \overline{F_2}$, we have $\mathcal{N}(F_1 \setminus \{^+\}) \leq^{\text{path}} \mathcal{N}(F_2)$ due to Theorem 3.2, whence $\mathcal{N}(F_1 \setminus \{^+\}) \leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$. Therefore $\mathcal{N}(F_1) \leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$ by transitivity.

For the 'only if' direction we consider its contrapositive. Since we are in the case where $^+ \in F_1$ and $^+ \notin F_2$, we may then clearly assume that $F_1 \not\subseteq \{\pi, di, ^+\}$ or $F_1 \setminus \{^+\} \not\subseteq \overline{F_2}$ is true. First, suppose that $F_1 \not\subseteq \{\pi, di, ^+\}$. Then $F_1 \cap \{\setminus, \cap, ^{-1}, \overline{\pi}\} \neq \emptyset$. If $\setminus$ or $\cap$ is present in F_1, the result follows directly from Proposition 7.2(1). On the other hand, if $^{-1} \in F_1$, it follows from Proposition 7.2(2). In the remaining scenario where $\overline{\pi} \in F_1$, it follows from Proposition 7.4.

To finish the 'only if' direction in this case suppose that $F_1 \subseteq \{\pi, di, ^+\}$ and $F_1 \setminus \{^+\} \not\subseteq \overline{F_2}$. Then, $\widehat{F_1 \setminus \{^+\}} = F_1 \setminus \{^+\} \not\subseteq \overline{F_2}$ since $^{-1}$ is not present in F_1. Therefore $\mathcal{N}(F_1 \setminus \{^+\}) \not\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$ by Proposition 5.5, whence $\mathcal{N}(F_1) \not\leq^{\text{bool}}_{\text{unl}} \mathcal{N}(F_2)$.

In the remaining cases, our theorem coincides with Proposition 5.5. Indeed condition three is never true, and the presence of transitive closure in F_1 implies its presence in F_2. □

8 Conclusion

The main results of this paper have shown that, even in a setting where one focuses on expressing boolean queries on unlabeled graphs in rather weak fragments of the calculus of relations, the transitive closure operator can still add

expressiveness. These results provide a counterpart to the cited Theorem 7.1 where, for other fragments, transitive closure was shown *not* to add expressiveness.

In the direction of further research it would be interesting to consider the interplay between transitive closure and the residuals [21]. Residuation is a derived operator of the calculus of relations, and interesting to consider separately, as we have done for projection and coprojection. Residuation is interesting from a database perspective because it corresponds to the set containment join [16].

Acknowledgment

We thank the referees for their constructive criticism on an earlier draft of this paper. We also thank one of the referees for suggesting that Theorem 6.3 could be proven from our bisimulation result.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From relations to semistructured data and XML*. Morgan Kaufmann, 2000.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[3] R. Angles, P. Barceló, and G. Rios. A practical query language for graph dbs. In L. Bravo and M. Lenzerini, editors, *Proceedings 7th Alberto Mendelzon International Workshop on Foundations of Data Management*, volume 1087 of *CEUR Workshop Proceedings*, 2013.

[4] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Computing Surveys*, 40(1):article 1, 2008.

[5] M. Benedikt, W. Fan, and G. Kuper. Structural properties of XPath fragments. *Theoretical Comput. Sci.*, 336(1):3–31, 2005.

[6] C. Bizer, T. Heath, and T. Berners-Lee. Linked data—the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[7] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[8] G. Fletcher, M. Gyssens, D. Leinders, D. Surinx, J. Van den Bussche, D. Van Gucht, S. Vansummeren, and Y. Wu. Relative expressive power of navigational querying on graphs. *Information Sciences*, 298:390–406, 2015.

[9] G. Fletcher, M. Gyssens, D. Leinders, J. Van den Bussche, D. Van Gucht, and S. Vansummeren. Similarity and bisimilarity notions appropriate for

characterizing indistinguishability in fragments of the calculus of relations. *Journal of Logic and Computation*, 2014. Published online, 24 March.

[10] G. Fletcher, M. Gyssens, D. Leinders, J. Van den Bussche, D. Van Gucht, S. Vansummeren, and Y. Wu. Relative expressive power of navigational querying on graphs. In *Proceedings 14th International Conference on Database Theory*, 2011.

[11] G. Fletcher, M. Gyssens, D. Leinders, J. Van den Bussche, D. Van Gucht, S. Vansummeren, and Y. Wu. The impact of transitive closure on the expressiveness of navigational query languages on unlabeled graphs. *Annals of Mathematics and Artificial Intelligence*, 73(1–2):167–203, 2015.

[12] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.

[13] A. Halevy, M. Franklin, and D. Maier. Principles of dataspace systems. In *Proceedings 25th ACM Symposium on Principles of Database Systems*, pages 1–9, 2006.

[14] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[15] L. Libkin, W. Martens, and D. Vrgoč. Quering graph databases with XPath. In *Proceedings 16th International Conference on Database Theory*. ACM, 2013.

[16] N. Mamoulis. Efficient processing of joins on set-valued attributes. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 157–168, 2003.

[17] M. Marx. Conditional XPath. *ACM Trans. Database Syst.*, 30(4):929–959, 2005.

[18] M. Marx and M. de Rijke. Semantic characterizations of navigational XPath. *SIGMOD Record*, 34(2):41–46, 2005.

[19] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*. Springer, 1997.

[20] J. Pérez, M. Arenas, and C. Gutierrez. nSPARQL: A navigational language for RDF. *Journal of Web Semantics*, 8(4):255–270, 2010.

[21] V. Pratt. Origins of the calculus of binary relations. In *Proceedings 7th Annual IEEE Symposium on Logic in Computer Science*, pages 248–254, 1992.

[22] RDF primer. W3C Recommendation, Feb. 2004.