# Simple, Strict, Proper, and Directed: Comparing Reachability in Directed and Undirected Temporal Graphs

**Michelle Döring**
Hasso Plattner Institute, University of Potsdam, Germany

## Abstract

Temporal graphs model networks whose connections are available only at specific points in time. Several definitional subtleties – whether paths must follow strictly increasing time labels (*strict* vs. *non-strict*), whether adjacent edges cannot appear simultaneously (*proper*), and whether edges are forbidden to appear multiple times (*simple*) – give rise to different temporal graph settings. These distinctions directly impact the definition of temporal reachability, a core concept in temporal graph theory. Casteigts, Corsini, and Sarkar [TCS24] introduced a framework of equivalence notions to compare the expressive power of these settings focusing solely on undirected temporal graphs. In this work, we extend their framework to include the fundamental dimension of *directed* vs. *undirected*.

Our contribution is three-fold. We (1) complete the undirected hierarchy by resolving the two open questions from [TCS24], (2) fully characterize the hierarchy of the directed settings, and (3) compare the directed and undirected settings, showing that directed temporal graphs are strictly more expressive than undirected temporal graphs in terms of reachability.

Our structural results highlight both the limitations and strengths of various temporal graph settings – for example, directed + strict + simple graphs can realize every possible reachability graph, while directed + proper graphs necessarily induce at least one transitive reachability on each directed cycle. We also provide transformation procedures between temporal settings offering practical tools for transferring algorithms and hardness results across models.

**2012 ACM Subject Classification** Mathematics of computing → Graph theory; Mathematics of computing → Paths and connectivity problems; Networks → Network dynamics

**Keywords and phrases** temporal graphs, directed graphs, temporal reachability, dynamic networks

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2025.27

**Related Version** *Full Version*: `https://arxiv.org/abs/2501.11697`

**Funding** *Michelle Döring*: German Federal Ministry for Education and Research (BMBF) through the project "KI Servicezentrum Berlin Brandenburg" (01IS22092).

**Acknowledgements** I warmly thank Arnaud Casteigts for his advice and inspiring discussions, and George Skretas for his steady support, encouragement, and valuable feedback on this work.

Proof of statements marked with ($\star$) can be found in the full version of the paper.

## 1 Introduction

A graph is called *temporal* if its connections change over time. Across disciplines, this concept has been given a myriad of names, including highly dynamic, edge-scheduled, multistage, time-labeled, or time-varying graph (or network). Unlike classical *static* graphs, where every edge is available at all times, temporal graphs restrict the availability of each edge to specific points in time, making them ideal for modeling dynamic interactions and processes. In the context of this paper, a temporal graph is defined as $\mathcal{G} = (V, E, \lambda)$, with a finite set of vertices $V$, a set of directed or undirected edges $E \subseteq V \times V$, and a labeling function $\lambda\colon E \to 2^{\mathbb{N}}$ assigning a set of time labels to every edge, indicating when the edge is present.

Reachability is a central concept in temporal graph theory and has been the focus of much of recent research. A vertex $u$ is said to *reach* a vertex $v$ in a temporal graph if there exists a temporal path – a sequence of edges traversed in temporal order – from $u$ to $v$. Reachability in static graphs is transitive (if $x$ reaches $y$ and $y$ reaches $z$, then $x$ reaches $z$), which forms the basis of many classical algorithms and structural results. In temporal graphs transitivity is not guaranteed, which breaks standard algorithmic techniques and leads to the hardness of many computational problems. Reachability plays a role in various problems, including optimal journeys [17, 20, 23], exploration [5, 15], infection spreading [19, 12], connected components [4, 11], flows and separators [3, 16, 22], and temporal spanners [7, 9]. It is therefore critical to understand the structure and limitations of temporal reachability.

The temporal reachability of a graph can be represented by the reachability graph $\mathcal{R}(\mathcal{G})$: a static graph which contains an edge from $u$ to $v$ if and only if a temporal path from $u$ to $v$ exists in $\mathcal{G}$. The study of temporal graphs realizing specific reachability patterns dates back to gossip theory [18, 6], which seeks fully connected temporal graphs using few temporal edges. This was later extended to temporal graph realization problems [21, 1], where one aims to construct a temporal graph satisfying prescribed constraints, such as bounded distances. More recent work has shifted towards understanding the space of **all** reachability graphs that can be realized by a given type of temporal graphs, either from a structural perspective, as in [8], or from a computational complexity perspective [14].

Our work builds on the former line of research, extending the structural analysis of Casteigts, Corsini, and Sarkar (CCS) [8] to directed graphs. Their framework of temporal graph *settings* identifies 3 dimensions within the definition of an undirected temporal graph:

**(1)** *strict* vs *non-strict* (whether times along a path are strictly increasing or non-decreasing);

**(2)** *proper* vs *arbitrary* (whether adjacent edges are forbidden to appear at the same time);

**(3)** *simple* vs *multi-labeled* (whether edges can appear at most once or multiple times).

We extend this framework by introducing a fourth, essential dimension:

**(0)** *directed* vs *undirected*.

This distinction is fundamental, as it directly shapes the definitions of graph properties and the formulation of computational problems, both in temporal and static graph theory.

Every combination of the four dimensions defines a *temporal graph setting* which we denote by its defining properties; for instance, the setting of directed, strict and proper graphs is denoted by directed + strict + proper. These temporal graph classes can be compared under four equivalence relations introduced by CCS, which capture progressively weaker notions of similarity between two graphs $\mathcal{G}$ and $\mathcal{H}$. Such a comparison is established by either providing a transformation that maps every graph in one class to an equivalent graph in the other, or by providing a *separating structure* – a graph that cannot be transformed under the given notion. The four equivalence notions are:

- *Bijective equivalence*: there is a one-to-one matching between all temporal paths in $\mathcal{G}$ and those in $\mathcal{H}$ such that the matched paths visit the same vertices in the same order.
- *Support equivalence*: every path in $\mathcal{G}$ corresponds to a path in $\mathcal{H}$ that visits the same vertices in the same order, and vice versa. Thus, $\mathcal{G}$ and $\mathcal{H}$ must have the same footprint.
- *Reachability equivalence*: both graphs induce the same *reachability graph* $\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$
- *Induced-reachability equivalence*: $\mathcal{H}$ may have additional vertices, but the subgraph of $\mathcal{R}(\mathcal{H})$ induced by $V(\mathcal{G})$ must be the same as $\mathcal{R}(\mathcal{G})$.

Refer to Figure 1 for an illustration of the first three equivalence notions. In this work, we extend the analysis of temporal settings with respect to support, reachability, and induced-reachability equivalence. Our focus lies primarily on reachability equivalence as it offers a practical middle ground: support equivalence is overly restrictive – transformations can

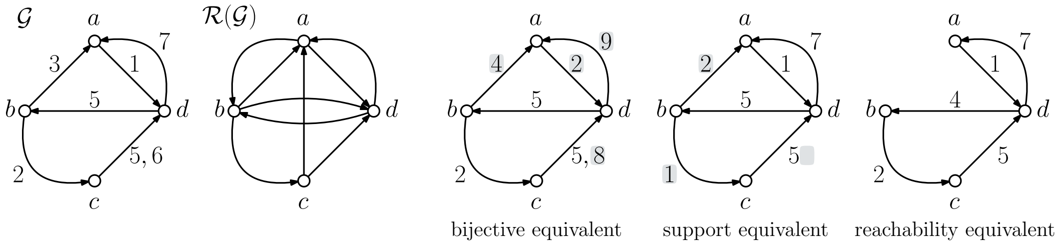



**Figure 1** A temporal graph $\mathcal{G}$ and its reachability graph $\mathcal{R}(\mathcal{G})$ under strict reachability (note that $c$ cannot reach $b$ although there exists a non-strict temporal path). On the right are three graphs which are (1) bijective equivalent, (2) support but not bijective equivalent, and (3) reachability but not support equivalent to $\mathcal{G}$. For the former two, differences in the labeling from $\mathcal{G}$ are highlighted in gray. Observe that the rightmost graph is proper and simple.

only relabel edges without modifying the underlying graph; reachability equivalence is more flexible, allowing the addition of edges; induced-reachability allows both edge and vertex additions which is so permissive that all settings collapse into a single class. Since bijective equivalence is even more restrictive than support equivalence, we do not consider this notion.

**Results and Technical Overview.** Our contribution is three-fold: (1) we resolve the two open questions from [8] in Section 3, thereby completing the hierarchy for undirected temporal settings under reachability equivalence; (2) we fully resolve the directed hierarchy in Section 4 using new separating structures and transformations; (3) we compare the directed and undirected settings in Section 5 to obtain a unified hierarchy. See Figure 2 for an overview of the reachability hierarchies.

To resolve the open questions from CCS, we show that there exists an undirected + non-strict + simple graph whose reachability graph cannot be achieved by an undirected + strict + simple graph. Consequently, these settings are incomparable under reachability equivalence and the undirected reachability hierarchy admits two strands.

We compare each pair of directed temporal graph classes and either provide a transformation or a separating structure proving that no transformation exists. The main difference to the undirected graph classes is that directed + strict + simple is equivalent to directed + strict, both of which can express every reachability graph.

Our directed separating structures are quite different and partially more intricate than the undirected separations provided by CCS. This can be explained as follows. In undirected graphs, adding an edge $ab$ to the footprint implies bidirectional reachability between $a$ and $b$, which has strong implications on the reachability graph. Therefore, already on short undirected paths the possible footprint for an equivalent graph can be quite restricted. In directed graphs, on the other hand, an edge insertions can introduce specific asymmetric reachabilities. Thus, to construct a graph whose reachability graph cannot be achieved in a specific directed graph class more intricate constructions become necessary.

On the constructive side, we extend the known transformations from [8] to transformations on directed graphs, where the application to undirected graphs emerges as special instances. Beyond this, we introduce a novel transformation called *reachability-dilation*, which replaces each connected component within a snapshot by a bidirected spanning tree. These trees are then labeled using a *pivot-labeling* inspired by [2, 19, 10] and adjusted to ensure properness. This procedure transforms each directed + non-strict graph into a reachability equivalent directed + proper graphs with at most twice the lifetime. Crucially, it also enables a transformation from an undirected to a directed setting, namely from undirected + non-strict + simple to directed + proper + simple.

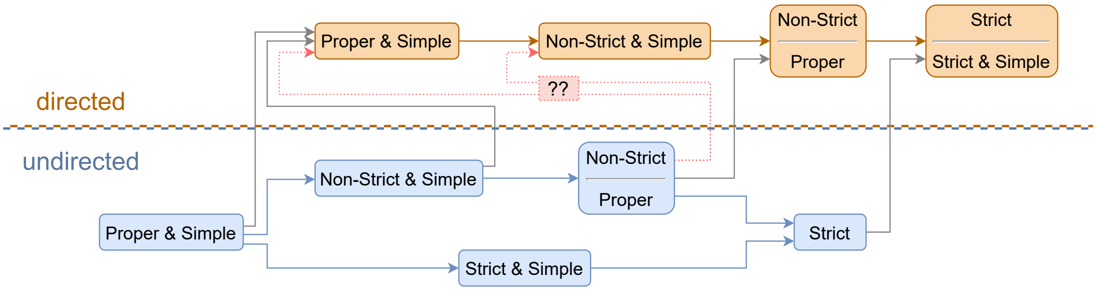



**Figure 2** Illustration of the reachability hierarchy among the temporal graph classes. An arrow from (1) to (2) indicates that every graph in (1) can be transformed into a reachability equivalent graph in (2). These transformations are transitive and implied arrows are omitted. The absence of an arrow (unless implied transitively) indicates that no such transformation can exist. If two classes appear in the same box, they induce the same reachability graph. The two red dotted arrows represent the open questions. The top half (orange) shows the directed hierarchy (Section 4) and the bottom half (blue) the undirected hierarchy (Section 3). Their comparison is discussed in Section 5.

## 2 Preliminaries

A *temporal graph* $\mathcal{G} = (V, E, \lambda)$ consists of a static graph $G = (V, E)$, called the *footprint*, and a labeling function $\lambda$. A pair $(e, t)$, where $e \in E$ and $t \in \lambda(e)$, represents a *temporal edge* with *label* $t$. The set of temporal edges is denoted by $\mathcal{E}$. The *lifetime* $\tau$ of $\mathcal{G}$ is the maximum label assigned by $\lambda$. The static graph $G_t = (V, E_t)$, where $E_t = \{e \in E \mid t \in \lambda(e)\}$, is called the *snapshot* of $\mathcal{G}$ at time $t$. A *temporal path* is a sequence of temporal edges $P = \langle(e_i, t_i)\rangle$ where $\langle e_i \rangle$ forms a path in the footprint and the time labels $\langle t_i \rangle$ are non-decreasing. If the time labels $\langle t_i \rangle$ are strictly increasing, $P$ is a *strict* temporal path; otherwise, it is *non-strict*. We denote a temporal path from $u$ to $v$ by $u \rightsquigarrow v$ and say $u$ *can reach* $v$.

### 2.1 Reachability and Equivalence Relations

The reachability relation between vertices of a temporal graph $\mathcal{G}$ can be captured by the *reachability graph* $\mathcal{R}(\mathcal{G})$ which is given by the static directed graph $(V, E_c)$ with $(u, v) \in E_c$ if and only if there exists a temporal path from $u$ to $v$. A graph is *temporally connected* if all vertices can reach each other by at least one path, i. e., $\mathcal{R}(\mathcal{G})$ is a complete directed graph. A *temporally connected component* of $\mathcal{G}$ is a subset of vertices which are temporally connected.

We now define the equivalence relations between temporal graphs in regards to reachability. The definitions are adopted from [8, Section 3]. Let $\mathcal{G}_1 = (V_1, E_1, \lambda_1)$ and $\mathcal{G}_2 = (V_2, E_2, \lambda_2)$ be temporal graphs and denote by $\mathbb{P}(\mathcal{G}_1)$ and $\mathbb{P}(\mathcal{G}_2)$ the sets of all temporal paths in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. We say two temporal paths $P$ and $P'$ share the same *support* if they visit the same vertices in the same order, i. e., their underlying static paths are the same.

▶ **Definition 2.1** (Support equivalence). *Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are* support equivalent *if $V_1 = V_2$ and for every temporal path $P$ in either graph, there exists a temporal path $P'$ in the other graph, such that $P$ and $P'$ share the same support.*

▶ **Definition 2.2** (Reachability equivalence). *Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are* reachability equivalent *if $V_1 = V_2$ and $\mathcal{R}(\mathcal{G}_1) \simeq \mathcal{R}(\mathcal{G}_2)$, i. e., the reachability graphs are isomorphic.*

Abusing terminology, we say $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same reachability graph if $\mathcal{R}(\mathcal{G}_1) \simeq \mathcal{R}(\mathcal{G}_2)$.

▶ **Definition 2.3** (induced-Reachability equivalence). *A temporal graph* $\mathcal{G}_2$ *is* induced-reachability equivalent *to* $\mathcal{G}_1$ *if* $V_1 \subseteq V_2$ *and* $\mathcal{R}(\mathcal{G}_1) \simeq \mathcal{R}(\mathcal{G}_2)[V_1]$.

The equivalence notions are sorted by decreasing requirement strength: support equivalence implies reachability equivalence, which in turn implies induced-reachability equivalence.

In this paper, we compare the expressive power of temporal settings with respect to support, reachability, and induced-reachability equivalence. Let $\mathbb{S}$ and $\mathbb{T}$ be two temporal settings, and let $X \in \{S, R, iR\}$ denote one of the three equivalence notions. We say that $\mathbb{S}$ can be *transformed into* $\mathbb{T}$ under $X$, denoted $\mathbb{S} \rightsquigarrow^X \mathbb{T}$, if for every temporal graph $\mathcal{G} \in \mathbb{S}$ there exists a graph $\mathcal{H} \in \mathbb{T}$ such that $\mathcal{G}$ and $\mathcal{H}$ are equivalent under $X$. If no such transformation exists, then there is a *separating structure* (or *separation*) with respect to $X$: a graph $\mathcal{G} \in \mathbb{S}$ for which no graph in $\mathbb{T}$ is $X$-equivalent to $\mathcal{G}$. We say that $\mathbb{T}$ is *strictly more expressive* than $\mathbb{S}$ under $X$ if $\mathbb{S} \rightsquigarrow^X \mathbb{T}$ but $\mathbb{T} \not\rightsquigarrow^X \mathbb{S}$; the settings are *incomparable* if $\mathbb{S} \not\rightsquigarrow^X \mathbb{T}$ and $\mathbb{T} \not\rightsquigarrow^X \mathbb{S}$, and they are *equivalent* if $\mathbb{S} \rightsquigarrow^X \mathbb{T}$ and $\mathbb{T} \rightsquigarrow^X \mathbb{S}$.

## 2.2 Settings: (Un)Directed, (Non-)Strict, Simple, and Proper

We identify four key dimensions for defining the settings of a temporal graph $\mathcal{G}$, following [8]:

**Directed vs. Undirected** A temporal graph is directed (resp. undirected) if its footprint is a directed (resp. undirected) graph.

**Strict vs. Non-strict** A temporal graph is strict (resp. non-strict) if the reachability graph considers only strict (resp. non-strict) temporal paths.

**Proper vs. Arbitrary** A temporal graph is *proper* if no two edges incident to the same vertex share a time label.

**Simple vs. Multi-labeled** A temporal graph is *simple* if every edge is assigned exactly one time label.

We denote the class of temporal graphs belonging to, for instance, the directed and strict setting, as directed + strict, and if $\mathcal{G}$ belongs to the class of graphs defined by this setting we write $\mathcal{G} \in$ directed + strict. For brevity, we will use the shorthand D + ... and UD + ... to denote directed and undirected settings, respectively.

Observe that strict + proper, non-strict + proper, and proper are equivalent: in a proper temporal graph, adjacent edges do not share a time label, so all temporal paths are inherently strict. Thus, we omit the "(non-)strict" when referring to a proper setting. As a result, we consider 12 settings in total: 6 directed and 6 undirected.

Moreover, by definition, any setting $\mathbb{S}$ admits a support-preserving transformation from its restriction to *proper* or to *simple*; that is, $\mathbb{S}$ + proper $\rightsquigarrow^S \mathbb{S}$ and $\mathbb{S}$ + simple $\rightsquigarrow^S \mathbb{S}$ via the identity mapping.

# 3 Undirected Hierarchy

In this section, we give a brief overview of the results of CCS for undirected temporal graphs and address the two open questions posed in [8]. CCS established three key transformations: *Dilation*, which transforms every UD + non-strict graph into a support equivalent UD + proper graph; *Saturation*, which transforms every UD + non-strict graph into a reachability equivalent UD + strict graph; and *Semaphore*, which transforms every UD + strict graph into an induced-reachability equivalent UD + proper + simple graph. For the remaining setting pairs, CCS proved the existence of separating structures, built from paths or triangles, containing 3 to 5 vertices.

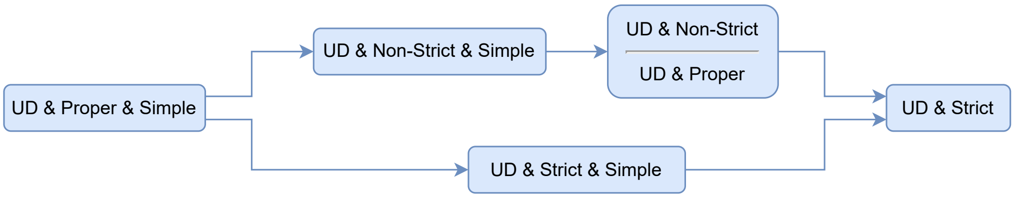
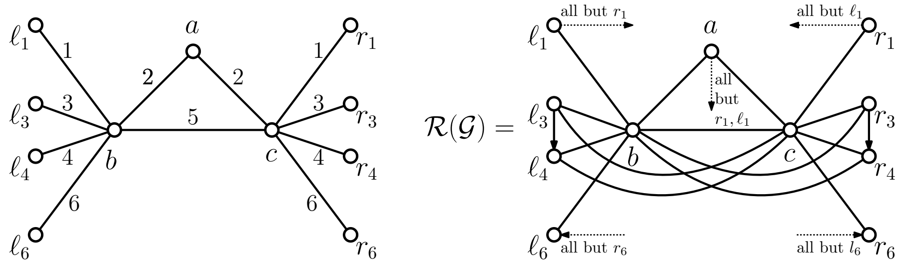



**Figure 3** The reachability hierarchy of the undirected temporal graph classes. The undirected hierarchy under support equivalence is identical. Under induced-reachability equivalence, all undirected graph classes are equivalent.

Two questions were left open: whether UD + non-strict can be transformed into UD + strict + simple, and whether UD + non-strict + simple can be transformed into UD + strict + simple. We provide a separating structure for the latter, thereby answering both questions.

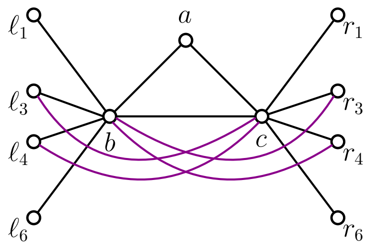
▶ **Theorem 3.1** (UD + non-strict + simple $\not\rightsquigarrow^R$ UD + strict + simple)**.** *There exists a graph $\mathcal{G} \in$ UD + non-strict + simple such that there is no $\mathcal{H} \in$ UD + strict + simple with $\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$.*

**Proof.** Consider the following temporal graph $\mathcal{G}$ in the UD + non-strict + simple setting (left) and the corresponding reachability graph (right). For readabilities sake, the dotted

$\mathcal{G} =$  $\mathcal{R}(\mathcal{G}) =$


edges adjacent to $\ell_1, r_1, a, \ell_6, r_6$ indicate incoming or outgoing edges that either connect to all vertices, or connect to all vertices except those explicitly excluded (specified next to the edge). For the sake of contradiction, let $\mathcal{H}$ be a temporal graph in the UD + strict + simple setting whose reachability graph is isomorphic to that of $\mathcal{G}$.

First, observe that there are only four undirected edges apart from the original edges of $\mathcal{G}$ in the reachability graph. That means, that the footprint of $\mathcal{H}$ has to be isomorphic to a subgraph of the following graph $G'$, with the four edges in $G' \setminus G$ highlighted in purple. We

$G' =$ 


first prove an intermediate statement regarding the purple edges.

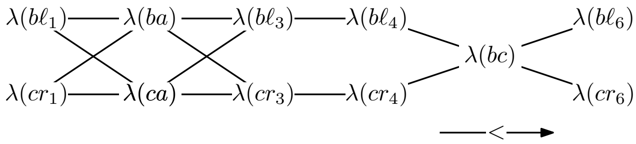
▷ Claim 1. The purple edge $c\ell_3$ is in the footprint of $\mathcal{H}$ if and only if no black edge from $\{cr_3, cr_4\}$ is in the footprint. The same holds for $c\ell_4$ and $\{cr_3, cr_4\}$, for $br_3$ and $\{b\ell_3, b\ell_4\}$, and for $br_4$ and $\{b\ell_3, b\ell_4\}$.

Proof of Claim 1. We proof the claim for $c\ell_3$ and $\{cr_3, cr_4\}$; the other three cases follow by a symmetric argument. Assume $c\ell_3$ and $cr_3$ are in the footprint of $\mathcal{H}$. Then $\ell_3$ and $r_3$ have distance two in the footprint and by [8, Lemma 1], either of them must reach the other. But since $\ell_3 r_3 \notin E(\mathcal{R}(\mathcal{G}))$ and $r_3 \ell_3 \notin E(\mathcal{R}(\mathcal{G}))$, this yields a contradiction. The same argument holds for $cr_4$ in the footprint of $\mathcal{H}$, by [8, Lemma 1] and $\ell_3 r_4, r_4 \ell_3 \notin E(\mathcal{R}(\mathcal{G}))$. ◁

This leaves two cases for the footprint of $\mathcal{H}$: either we add no purple edge and $H$ is isomorphic to $G$, or we add at least one purple edge and delete the corresponding black edges. For the former, we show that there is no simple labeling yielding the desired reachabilities for $\mathcal{H}$.

▷ Claim 2. If the footprint of $\mathcal{H}$ is isomorphic to the footprint of $\mathcal{G}$, then there can be no simple labeling achieving the same reachabilities in the strict setting.

Proof of Claim 2. We show the claim by proving that the chronological order of the edges, i. e., corresponding time labels, cannot change without changing the reachability in the strict setting. Since $\mathcal{H}$ is simple, we also cannot add labels. The chronological order of the edges in $\mathcal{G}$ can be illustrated by the following partially ordered set.




Now, since only $\ell_1$, $r_1$, $b$, $c$ can reach $a$, we infer that $\lambda(ba)$ and $\lambda(ca)$ must be strictly smaller than $\lambda(b\ell_3), \lambda(cr_3), \lambda(b\ell_4), \lambda(cr_4), \lambda(b\ell_6), \lambda(cr_6)$, but do not have to be the same. Furthermore, since $\ell_3$ can reach $\ell_4$ but not vice versa, we have $\lambda(b\ell_3) < \lambda(b\ell_4)$. Analogously, we have $\lambda(cr_3) < \lambda(cr_4)$. Now, for $r_3, r_4$ to reach $b$, and $\ell_3, \ell_4$ to reach $c$, we need $\lambda(b\ell_4) < \lambda(bc)$ and $\lambda(cr_4) < \lambda(bc)$. Lastly, since everyone can reach $\ell_6$ and $r_6$, we know that $\lambda(b\ell_6)$ and $\lambda(cr_6)$ must be greater than all other labels. Given all this, the only way for $\ell_1$ to reach everyone but $r_1$, and vice versa, is by making $\lambda(b\ell_1)$ and $\lambda(cr_1)$ strictly smaller than all other labels. Without the analysis about the ordering of $bc$ before, there would also be other ways for achieving the reachabilities of $\ell_1$ and $r_1$.

This yields exactly the same chronological order for $\mathcal{H}$ as it was the case for $\mathcal{G}$. Now, since we cannot change the relative order of the time labels, $\lambda(ba)$ or $\lambda(ca)$ has to change to enable the reachabilities of $\ell_1$ and $r_1$ in the strict setting. However, if $\lambda(ba) < \lambda(ca)$ then $r_1$ cannot reach $\ell_3$ and $\ell_4$, and if $\lambda(ba) > \lambda(ca)$ then $\ell_1$ cannot reach $r_3$ and $r_4$. Therefore, such a labeling is not possible in the UD + strict + simple setting. ◁

Therefore, $\mathcal{R}(\mathcal{G})$ is not achievable for $\mathcal{H}$ if its footprint is isomorphic to $G$. So, assume that at least one purple edge is added, without loss of generality, $c\ell_3$, and the corresponding black edges $cr_3$ and $cr_4$ are removed. To preserve the reachabilities of $r_3$ and $r_4$, they must remain connected to the graph, which can only be done by adding $br_3$ and $br_4$. This, however, implies that we need to remove the corresponding black edges $b\ell_3$ and $c\ell_4$. As a result, $\ell_4$ becomes disconnected, and we are forced to add $c\ell_4$. Thus, adding even one purple edge to the footprint requires adding all four purple edges. However, adding all purple edges and deleting the corresponding black edges results in a footprint that is isomorphic to $G$, with only $\ell_3$ and $\ell_4$ swapping places with $r_3$ and $r_4$, respectively. Thus, by Claim 2, there is no simple labeling for $\mathcal{H}$ yielding the desired reachability graph. ◀

Combining the results of [8] with Theorem 3.1, we obtain a two-strand hierarchy for undirected reachability, which is illustrated in Figure 3.

# 4 Directed Reachability

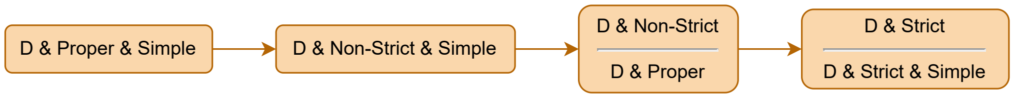



**Figure 4** The directed hierarchy under reachability equivalence. Under support equivalence, the directed hierarchy is analogous to the undirected hierarchy (Figure 3). Under induced-reachability equivalence, all directed graph classes are equivalent.

In this section, we resolve the equivalence hierarchy for directed temporal settings. We begin with a key structural observation and then present our results in three parts: reachability separations (Section 4.1), support-only separations (Section 4.2), and transformations between settings (Section 4.3) that preserve support, reachability, or induced-reachability. These results collectively resolve the directed hierarchy, as illustrated in Figure 4.

We begin with a structural limitation which is basis of several of our separations: In non-strict or proper graphs, directed cycles inevitably introduce additional reachabilities.

▶ **Lemma 4.1.** *There exists no labeling of a directed cycle of length at least 3, such that reachability graph of the corresponding* D + non-strict *temporal graph contains only the cycle.*

**Proof.** Assume towards contradiction that there exists a temporal graph $\mathcal{G} = (G, \lambda)$ such that the footprint $G$ is a directed cycle on the vertices $v_1, \dots, v_n$, and $\mathcal{R}(\mathcal{G}) = G$. It must hold that $\lambda(v_i, v_{i+1}) > \lambda(v_{i+1}, v_{i+2})$ for all subsequent edges around the cycle, as otherwise $v_i$ can reach $v_{i+2}$ via $v_{i+1}$. Following this argument, we get $\lambda(v_1, v_2) > \dots > \lambda(v_{n-1}, v_n) > \lambda(v_n, v_1) > \lambda(v_1, v_2)$, which is a contradiction. ◀

Note that such a labeling can be easily achieved in the D + strict setting: assigning the same label (e.g., 1) to every edge in the cycle avoids transitive paths, as illustrated in Figure 5.

A second structural observation reveals an inherent limitation of specifically the D + proper setting when trying to achieve complete pairwise reachability among $n$ vertices. The claim basically follows from arguments in gossip theory [19] and a short proof can be found in the long version of this paper.

▶ **Lemma 4.2.** *Any* D + proper *temporal graph on $n > 2$ vertices whose reachability graph is a clique (all vertices are pairwise connected) has to contain at least $n + 1$ temporal edges.*

## 4.1 Reachability separations

The primary separating structure is the directed triangle with the same label on every edge. For this graph in the D + strict + simple setting, there is no reachability equivalent graph in the D + non-strict setting (see Figure 5). This follows directly from Lemma 4.1.

▶ **Lemma 4.3** (D + strict + simple $\not\rightarrow^R$ D + non-strict)**.** *There exists a graph $\mathcal{G} \in$* D + strict + simple *such that there is no $\mathcal{H} \in$* D + non-strict *with $\mathcal{R}(\mathcal{H}) = \mathcal{R}(\mathcal{G})$.*

Recall that the restriction of a setting $\mathbb{S}$ to simple or to proper implies a transformation under support equivalence to $\mathbb{S}$. Hence, Lemma 4.3 also establishes D + strict (+ simple) $\not\rightarrow^R$ D + proper (+ simple) and D + strict (+ simple) $\not\rightarrow^R$ D + non-strict (+ simple).

Next, we separate D + proper (and thus D + non-strict) from D + non-strict + simple (see Figure 6), demonstrating that more than one label per edge is required to capture the full range of directed non-strict reachability. As before, we utilize the structure of a directed triangle by constructing a cycle with two overlapping triangles in the reachability graph. The proof can be found in the extended paper.

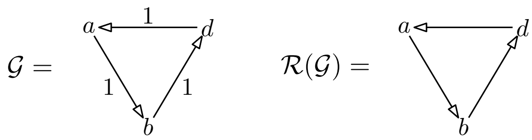



**Figure 5** Temporal graph $\mathcal{G} \in$ D + strict + simple (left) and its reachability graph (right). There exists no reachability equivalent graph in the D + non-strict setting.

▶ **Lemma 4.4** (($\star$) D + proper $\not\rightsquigarrow^R$ D + non-strict + simple)**.** *There exists a graph* $\mathcal{G} \in$ D + proper *such that there is no* $\mathcal{H} \in$ D + non-strict + simple *with* $\mathcal{R}(\mathcal{H}) = \mathcal{R}(\mathcal{G})$.

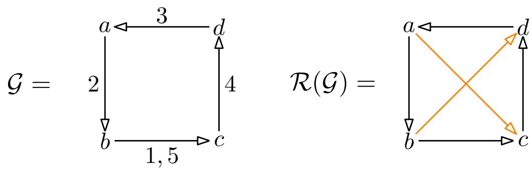


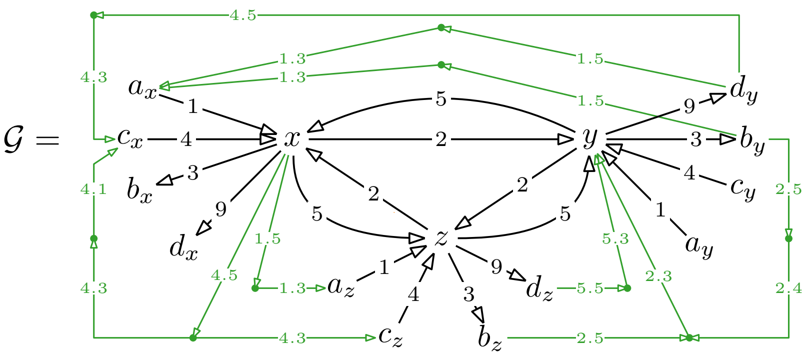

**Figure 6** Temporal graph $\mathcal{G} \in$ D + proper (left) and the corresponding reachability graph (right). There exists no reachability equivalent graph in the D + non-strict + simple setting.

Note that this also implies (i) D + non-strict $\not\rightsquigarrow^R$ D + non-strict + simple, (ii) D + proper $\not\rightsquigarrow^R$ D + proper + simple, and (iii) D + non-strict $\not\rightsquigarrow^R$ D + proper + simple.

We now present the final reachability separation, which proves that D + proper + simple is a proper subset of D + non-strict + simple. This separation uses a more intricate construction: three vertices $x, y, z$ are fully connected at two distinct time steps, forming two cliques. Meanwhile, each vertex $a_i$ and $c_i$ (for $i \in \{x, y, z\}$) must traverse these cliques to reach every $b_j$ and $d_j$ with $j \in \{x, y, z\} - i$. Crucially, this traversal has to happen at different time steps so that $a_i$ reaches both $b_j$ and $d_j$, while $c_i$ reaches only $d_j$. To enforce this double traversal via $x, y, z$, additional directed paths are used to forbid certain edges (shortcuts) in the footprint. These paths are constructed symmetrically for each combination of $x, y, z$ and block cycles in the footprint relying on Lemma 4.1.

▶ **Lemma 4.5** (D + non-strict + simple $\not\rightsquigarrow^R$ D + proper + simple)**.** *There exists a graph* $\mathcal{G} \in$ D + non-strict + simple *such that there is no* $\mathcal{H} \in$ D + proper + simple *with* $\mathcal{R}(\mathcal{H}) = \mathcal{R}(\mathcal{G})$.

**Proof.** Consider the following temporal graph $\mathcal{G}$ in the D + non-strict + simple setting.




For sake of readability, the following edges have been omitted in the illustration, with each set having one representative shown in the illustration in green. Let $C = \{x, y, z\}$.

- $E_1 = \{(i, h_i^{a_j}, 1.5), (h_i^{a_j}, a_j, 1.3), (i, h_i^{c_j}, 4.5), (h_i^{c_j}, c_j, 4.3) \colon i \in C \text{ and } j \in C \setminus \{i\}\}$;
- $E_2 = \{(b_j, h_{b_j}^i, 2.5), (h_{b_j}^i, i, 2.3), (d_j, h_{d_j}^i, 5.5), (h_{d_j}^i, i, 5.3) \colon i \in C \text{ and } j \in C \setminus \{i\}\}$;
- $E_3 = \{(b_j, h_{b_j}^{a_i}, 1.5), (h_{b_j}^{a_i}, a_i, 1.3), (d_j, h_{d_j}^{a_i}, 1.5), (h_{d_j}^{a_i}, a_i, 1.3), (d_j, h_{d_j}^{c_i}, 4.5), (h_{d_j}^{c_i}, c_i, 4.3) \colon$ $i \in C$ and $j \in C \setminus \{i\}\}$.
- $E_4 = \{(b_i, s, 2.7), (s, h_{b_j}^i, 2.5), (h_i^{c_j}, s, 4.3), (s, c_j, 4.1) \colon i \in C \text{ and } j \in C \setminus \{i\}\}$;

We refer to the vertices $h_\alpha^\beta$ as *helper* vertices from $\alpha$ to $\beta$ and to $\{x, y, z\}$ as the *center* vertices. The edges in $E_1$, $E_2$, $E_3$, and $E_4$ form paths of length two in the footprint and the time labels are chosen such that they do not form temporal paths. Specifically, for $i \in \{x, y, z\}$ and $j \in \{x, y, z\} \setminus \{i\}$, $E_1$ forms paths from $i$ to $a_j$ and $c_j$; $E_2$ forms paths from $b_i$ and $d_i$ to $j$; $E_3$ forms paths from $b_i$ and $d_i$ to $a_j$ and $c_j$; and $E_4$ forms paths from $b_i$ to the helper vertices of $b_j$, and from the helper vertices of $c_j$ to $c_i$.

For the sake of contradiction, let $\mathcal{H}$ be a temporal graph in the D + proper + simple setting whose reachability graph is isomorphic to that of $\mathcal{G}$. Observe that the center vertices form a strongly connected component in $G_2$ and in $G_5$, and recall from Lemma 4.2 that in the D + proper setting, we need at least four temporal edges to fully connect three vertices. Further, observe the following reachabilities in $\mathcal{G}$:

- $x, y, z$ reach every $b_i$ and $d_i$ with $i \in \{x, y, z\}$;
- for $i \in \{x, y, z\}$, every $a_i$ reaches $x, y, z$, $b_j$ and $d_j$ with $j \in \{x, y, z\}$ and
- every $c_i$ reaches $x, y, z$ and $d_j$ for $j \in \{x, y, z\}$;
- every helper vertex $h_\alpha^\beta$ can reach the vertex $\beta$, but not $\alpha$. Conversely, $\alpha$ can reach $h_\alpha^\beta$, but not $\beta$. However, $\beta$ can reach $\alpha$.

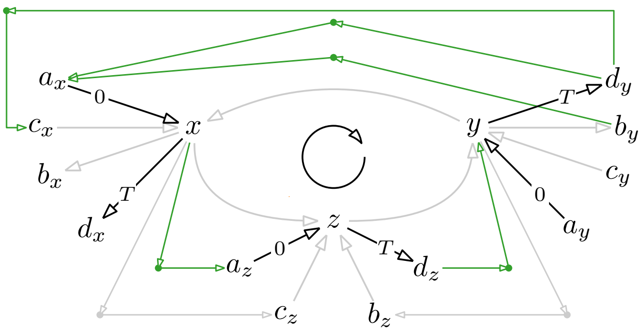
First, we show that the construction of the helper vertices $h_\beta^\alpha$ forbids the direct edges $(\beta, \alpha)$ in $\mathcal{H}$ for all but $(c_j, i)$ and $(i, b_j)$.

▷ Claim 1. In $\mathcal{H}$, there can be no edge from $F = \{(a_j, i), (i, d_j), (a_i, b_j), (a_i, d_j), (c_i, d_j),$ $(h_{b_j}^i, b_i), (c_j, h_i^{c_j}) \colon i, j \in \{x, y, z\},\ j \neq i\}$.

Proof of Claim 1. Recall that any labeling of a directed triangle in a non-strict or proper setting leads to at least one transitive reachability (Lemma 4.1), and observe that every direct edge $(\beta, \alpha)$ would form a directed triangle with the corresponding green edges $(\alpha, h_\alpha^\beta), (h_\alpha^\beta, \beta)$. Let $(\beta, \alpha) \in F$. Observe that $h_\alpha^\beta$ reaches $\beta$, but reaches no other vertex that also reaches $\beta$. Thus, the direct edge $(h_\alpha^\beta, \beta)$ must be included in the footprint $H$ of $\mathcal{H}$. Furthermore, $h_\alpha^\beta$ is reached by $\alpha$ but by no other vertex that $\alpha$ can reach. Hence, the direct edge $(\alpha, h_\alpha^\beta)$ must also be in $E(H)$. As a result, $(\beta, \alpha) \notin E(H)$, since $\alpha, \beta, h_\alpha^\beta$ form a directed triangle in $\mathcal{R}(\mathcal{G})$, which is impossible if $H$ contains the edges of the triangle. ◁

From Claim 1 follows that for $a_i$ to reach $d_j$ in $\mathcal{H}$, no shortcut edge $(a_i, j)$, $(i, d_j)$, or $(a_i, d_j)$ can be used. Additionally, $a$ reaches no vertex that also reaches $d_j$ in $\mathcal{G}$. Thus, $a_i$ must use a temporal path with support $(a_i, i, \ldots, j, d_j)$ in $\mathcal{H}$, and $(a_i, i), (j, d_j) \in E(H)$. As this holds for all $i$ and $j$, there must exist paths between all center vertices, ensuring that each $a_i$ reaches $x$, $y$ and $z$ via its respective vertex $i$. After these paths, there must also be an edge from $x$, $y$ and $z$ to their respective $d_j$. The following graph $\mathcal{H}'$ summarizes the current information on $\mathcal{H}$, with some large time step $T$. Black and green edges must be present in $\mathcal{H}$, while for gray edges no information is yet available. Next, we consider the $b$ and $c$ vertices. Note that the construction does not forbid the direct edges $(i, b_j)$ and $(c_j, i)$ for $i \neq j$ in $\mathcal{H}$:

- $h_{b_j}^i$ is reached only by $b_j$, requiring $(b_j, h_{b_j}^i) \in E(H)$. However, $h_{b_j}^i$ reaches $x, y, z$, $d_x, d_y, d_z$, and $b_i$ via $i$. Consequently, we could add $(h_{b_j}^i, k)$ for any $k \in \{x, y, z\}$.
- $h_i^{c_j}$ only reaches $c_j$, requiring $(h_i^{c_j}, c_j) \in E(H)$. Moreover, $h_i^{c_j}$ is reached by $x, y, z$, $a_x, a_y, a_z$, and $c_i$ via $i$. Consequently, we could add $(k, h_i^{c_j})$ for any $k \in \{x, y, z\}$.

$\mathcal{H}' =$ 


While we cannot establish Claim 1 for $(i, b_j)$ or $(c_j, i)$, we can prove the following slightly weaker result. For each $b_i$, there is exactly one center vertex (out of $x, y, z$) that has an edge to $b_i$, while the other two center vertices do not. Similarly, for each $c_i$, there is exactly one center vertex that $c_i$ has an edge to, while it has no edges to the remaining center vertices. We begin by proving this for the $b$ vertices.

▷ Claim 2. There exists a bijective mapping $f\colon \{x, y, z\} \to \{x, y, z\}$ so that $(f(i), b_i) \in E(H)$ and $(k, b_i) \notin E(H)$ for all $k \in \{x, y, z\} \setminus \{f(i)\}$.

Proof. By Claim 1, $a_i$ cannot reach $b_j$ in $\mathcal{H}$ via the direct edge $(a_i, b_j)$. Therefore, $a_i$ must traverse a temporal path starting at $i$. Since $b_j$ can only be reached by the center vertices $x, y, z$ and $a_x, a_y, a_z$, it follows that $(x, b_j)$, $(y, b_j)$ or $(z, b_j)$ must be in $\mathcal{H}$.

If $(i, b_i) \in E(H)$ and the corresponding helper edges are included in $\mathcal{H}$, the claim holds with $f$ defined as the identity mapping. Thus, it remains to prove the claim for the case where $(i, b_i) \notin E(H)$ for at least one $i \in \{x, y, z\}$. Without loss of generality, assume $(z, b_z) \notin E(H)$; the remaining cases follow by symmetry of the construction.

From $(z, b_z) \notin E(H)$ follows $(y, b_z) \in E(H)$ or $(x, b_z) \in E(H)$ to ensure that $b_z$ is reachable. If $(y, b_z) \in E(H)$, then $(h^y_{b_z}, y) \notin E(H)$, as otherwise, $y, b_z, h^y_{b_z}$ would form a directed triangle in both $H$ and $\mathcal{R}(\mathcal{G})$, which is impossible in a simple + proper graph. Consequently, $h^y_{b_z}$ must have an edge to $x$ or to $z$. The same reasoning applies if $(x, b_z) \in E(H)$, implying $(h^x_{b_z}, x) \notin E(H)$ and requiring $h^x_{b_z}$ to have an edge to $y$ or to $z$.

Now, observe that the helper vertices cannot have edges to arbitrary center vertices. Let the center vertex $i \in \{x, y, z\}$ have incoming edges from two helper vertices with different out-neighbors in $\mathcal{G}$, for example, $h^y_{\_}$ with an edge to $y$ in $\mathcal{G}$, and $h^z_{\_}$ with an edge to $z$ in $\mathcal{G}$. These two vertices have incompatible reachabilities: $h^y_{\_}$ must reach $b_y$ (but not $b_z$), and $h^z_{\_}$ must reach $b_z$ (but not $b_y$). Thus, at most one of the two can reach their respective $b$ vertex via $i$. If both $h^y_{\_}$ and $h^z_{\_}$ had temporal paths to their $b$ vertex starting in $i$, they could not depart $i$ at the same time, as this would result in both reaching $b_y$ and $b_z$. Therefore, one temporal path must occur earlier than the other. However, the helper vertex taking the earlier path could also take the later path and would reach both $b_y$ and $b_z$. Consequently, one of the helper vertices must reach its respective $b$ vertex via a different center vertex.

This implies that, to satisfy the reachabilities of the helper vertices, all $h^i_{b_j}$ for a fixed $i$ must reach $b_i$ via the same center vertex. Consequently, there exists a mapping $f\colon \{x, y, z\} \to \{x, y, z\}$ such that $(h^i_{b_j}, f(i)) \in E(H)$.

Consider the center vertex $y$ and assume $f^{-1}(y) = x$. According to this mapping, $y$ has incoming edges from $h^x_{b_y}$ and $h^x_{b_z}$. If $(y, b_y) \in E(H)$, then $h^x_{b_y}, y, b_y$ would form a triangle in both $H$ and $\mathcal{R}(\mathcal{H})$ – a contradiction. Similarly, if $(y, b_z) \in E(H)$, then $h^x_{b_z}, y, b_z$ would form a triangle in both $H$ and $\mathcal{R}(\mathcal{H})$. Thus, we conclude that only $(y, b_x) \in E(H)$. The same holds for any other center vertex and any possible mapping $f$.

Therefore, there exists a bijective mapping $f$ on the center vertices such that $(f(i), b_i) \in E(H)$ for all $i \in \{x, y, z\}$ and $(k, b_i) \notin E(H)$ for all $k \in \{x, y, z\} \setminus \{f(i)\}$. $\triangleleft$

The proof for the $c$ vertices follows analogously to that of the $b$ vertices. Here, every occurrence of $b_i$ is replaced with $c_i$, and the direction of each mentioned edge is reversed. Instead of two helper edges pointing toward the same center vertex leading to a contradiction, the contradiction arises from two helper edges pointing away from the same center vertex. This reversal of direction results in the same type of mapping, ensuring that each $c_i$ has an edge to exactly one center vertex, with no edges to the others. The reasoning and conclusions remain identical, making a separate proof unnecessary.

$\triangleright$ Claim 3. There exists a bijective mapping $g\colon \{x, y, z\} \to \{x, y, z\}$ so that $(c_i, g(i)) \in E(H)$ and $(c_i, k) \notin E(H)$ for all $k \in \{x, y, z\} \setminus \{g(i)\}$.

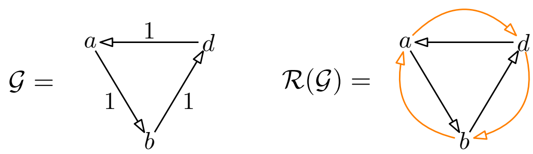
We will now draw the final conclusion, proving that $\mathcal{H}$ can have no proper + simple labeling. Let $i \in \{x, y, z\}$ and $f$ and $g$ be bijections as in Claim 2 and Claim 3. The labeling of $(a_i, i), (c_{g(i)}, i), (i, b_{f(i)}), (i, d_i)$ must ensure that $a_i$ reaches both $b_{f(i)}$ and $d_i$, while $c_{f(i)}$ reaches only $d_i$. Consequently, $\lambda(a_i, i) < \lambda(i, b_{f(i)}) < \lambda(i, c_{g(i)}) < \lambda(i, d_i)$. Now, $c_{g(i)}$ has to reach every $d_j, j \in \{x, y, z\}$. Since no shortcut edge can be used, $c_{g(i)}$ must traverse a temporal path with support $(c_{g(i)}, i, \ldots, j, d_j)$. This traversal must happen after $\lambda(c_{g(i)}, i)$, and, consequently, after $\lambda(i, b_{f(i)})$. Thus, the $a$ and the $c$ vertices must travel at different times, requiring an $x, y, z$-clique to exist at two different points in time. Each of these requires four temporal edges between $x$, $y$, and $z$ in a proper graph; eight in total. However, at most six temporal edges can exist between $x, y, z$ in a simple graph, a contradiction. ◀

## 4.2 Support separations

We present the remaining separations that hold only under support equivalence, namely that there exists a D + non-strict and a D + proper temporal graph for which there is no support equivalent D + strict + simple graph. Note that these separations do not hold under reachability equivalence as we give a transformation in Section 4.3.

▶ **Lemma 4.6** (D + non-strict + simple $\not\leftrightarrow^S$ D + strict + simple)**.** *There exists a graph* $\mathcal{G} \in$ D + non-strict + simple *such that there is no support equivalent graph* $\mathcal{H} \in$ D + strict + simple*.*

**Proof.** Consider the following simple temporal graph $\mathcal{G}$ (left) in the non-strict setting and the corresponding reachability graph (right). For the sake of contradiction, let $\mathcal{H}$ be a support equivalent temporal graph in the strict + simple setting.

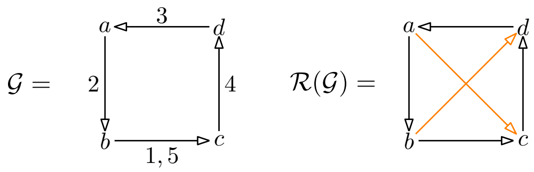



To preserve the support of $a$ to $c$ via $b$, $\mathcal{H}$ needs $\lambda(a, b) < \lambda(b, c)$. Similarly, for $b$ to reach $a$ via $c$ and $c$ to reach $b$ via $a$, we get $\lambda(a, b) < \lambda(b, c) < \lambda(c, a) < \lambda(a, b)$, a contradiction. ◀

The separation of D + proper from D + strict + simple, is achieved by the same structure separating D + proper from D + non-strict + simple (Lemma 4.4) and requires only a slightly adjusted argument to fit the support-preserving requirement.

▶ **Lemma 4.7** (D + proper $\not\rightsquigarrow^S$ D + strict + simple)**.** *There exists a graph $\mathcal{G} \in$* D + proper *such that there is no support equivalent graph $\mathcal{H} \in$* D + strict + simple*.*

**Proof.** Consider the following temporal graph $\mathcal{G}$ in the D +proper setting (left) and the corresponding reachability graph (right). For the sake of contradiction, let $\mathcal{H}$ be a support equivalent temporal graph in the D +strict+simple setting.




Observe that we require all black edges in $\mathcal{H}$ to preserve the support of the direct reachabilities. Furthermore, there are two transitive reachabilities, namely $a$ reaches $c$ via $b$ and $b$ reaches $d$ via $c$. Now observe that $c$ not reaching $a$ and $d$ not reaching $b$ imply $\lambda(c, d) > \lambda(d, a) > \lambda(a, b)$. There are two options for the label of $(b, c)$. Either $\lambda(b, c) > \lambda(a, b)$ and $\lambda(b, c) > \lambda(c, d)$, which enables $a$ to reach $c$ via $b$ and avoids $a$ reaching $d$, or $\lambda(b, c) < \lambda(a, b) < \lambda(c, d)$ which enables $b$ to reach $d$ via $c$. However, in the first case, $a$ cannot reach $c$ via $b$ and in the second, $b$ cannot reach $d$ via $c$.

Adding some of the orange edges to $\mathcal{H}$ will not achieve the transitive reachabilities, so there can be no support equivalent labeling. ◀

Since D + proper is a subset of D + strict, this also proves D + strict $\not\rightsquigarrow^S$ D + strict + simple.

## 4.3 Transformations

In this section, we present four transformations between directed temporal settings. Two of our transformations generalize techniques from [8]: the *dilation* process, extended here as the support-preserving *support-dilation*, and the *semaphore* construction, adapted for directed induced-reachability equivalence. Additionally, we introduce two new transformations: a more efficient reachability-preserving variant called *reachability-dilation*, and the following transformation that maps any directed graph to the strict + simple setting.

▶ **Observation 4.8** (D + * $\rightsquigarrow^R$ D + strict + simple)**.** *Given a directed temporal graph $\mathcal{G}$ in an arbitrary setting, let $\mathcal{H} = (\mathcal{R}(\mathcal{G}), \lambda)$ with $\lambda(e) = 1$ for all $e \in E(\mathcal{R}(\mathcal{G}))$. Then $\mathcal{H} \in$* D + strict + simple *with $\mathcal{R}(\mathcal{H}) = \mathcal{R}(\mathcal{G})$.*

Note that this transformation is specific to directed graphs and has no analogue in the undirected temporal graph classes, since reachabilities are not symmetric and directed edges are essential for preserving these dependencies.

### 4.3.1 Support-Dilation (non-strict $\rightsquigarrow^S$ proper) and Reachability-Dilation (non-strict $\rightsquigarrow^R$ proper)

Before analyzing directed dilation, we compare the structure of the snapshots of non-strict temporal graphs in an undirected and directed setting. In undirected graphs each snapshot consists of 1 to $n$ connected components, each forming a clique in the reachability graph. In contrast, snapshots of directed temporal graphs consist of 1 to $n$ weakly connected components. Each of those can be interpreted as a directed acyclic graph (DAG) with as vertices the strongly connected subgraphs (possibly of size 1) of the weakly connected component connected by directed edges in an acyclic manner. Notably, undirected snapshots

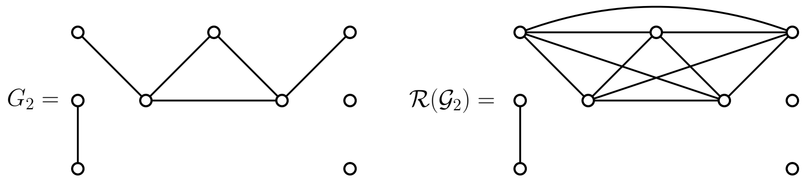



**Figure 7** A cutout of some undirected temporal graph at time 2 with four connected components (left) and the corresponding reachability graph with four maximal cliques (right).

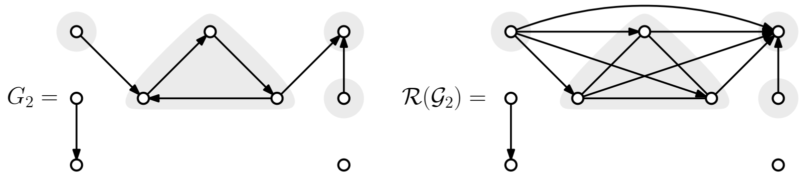



**Figure 8** A cutout of some directed temporal graph at time 2 with three weakly connected components (left) and the corresponding reachability graph (right). The strongly connected components of the size-6 weakly connected component are marked in gray.

are a special case of directed ones – an observation that enables us to generalize the undirected dilation technique from [8]. *Dilation* transforms non-strict graphs into proper graphs while preserving support. The process operates on each snapshot individually, duplicating each non-strict path within a connected component of the snapshot, thereby stretching the time step to construct equivalent strict paths. Subsequent snapshots are shifted accordingly. Note that the new lifetime may be as large as $(n-1)\cdot(\Delta+2)\cdot\tau$, where $\Delta$ is the maximum degree. For directed graphs, we generalize the dilation process to also handle weakly connected components and call it *support-dilation*.

▶ **Definition 4.9** (Support-Dilation). *Let $\mathcal{G}$ be a temporal graph and consider each snapshot $G_t$ individually. Without loss of generality, we assume $t = 1$ (otherwise, we shift the labels of the earlier snapshots in the construction).*

*Every snapshot consists of weakly connected components $\mathbb{W}$ that each can be interpreted as a DAG. We consider each $W \in \mathbb{W}$ separately, so let $D = (\mathbb{S}, E)$ be the DAG representing $W$. This DAG contains strongly connected components $\mathbb{S}$ (possibly of size 1) as vertices connected by directed edges. Any DAG can be topologically ordered, i. e., the vertices can be arranged as a linear ordering that is consistent with the edge directions. Let $\ell$ be such an ordering.*

*First, we order the edges of $E$ lexicographically by the positions of their endpoints in $\ell$, i. e., $(u, v)$ precedes $(u', v')$ if $\ell(u) < \ell(u')$, or if $\ell(u) = \ell(u')$ and $\ell(v) < \ell(v')$. Then, in this order, we label the edges in $E$* between *the strongly connected components with distinct time labels starting from 1. Next, in order of $\ell$, "dilate" the edges* inside *each strongly connected component as follows (subsequent labels on $E$ are shifted accordingly): Let $S \in \mathbb{S}$ be a strongly connected component and $\alpha = \ell(S)$. First, assign each edge the set of labels $\{\alpha+1, \alpha+2, \ldots, \alpha+k\}$, where $k$ is the longest directed path in $S$. Now, color the edges of $S$ (ignoring their directions) such that no adjacent edges share the same color. By Vizing's theorem, this requires at most $\Delta + 2$[1] colors, where $\Delta$ is the maximum degree of $S$. Let $c\colon E_S \to [0, \Delta+1]$ be such a coloring and let $0 < \varepsilon < \frac{1}{\Delta+2}$. Now, for every edge $e \in E_S$, add $c(e)\cdot\varepsilon$ to every label of $e$.*

*After processing every snapshot, support-dilation returns the adjusted temporal graph.*

[1] We thank the anonymous reviewer of ISAAC25 for pointing out that the usual bound of $\Delta + 1$ from Vizing's theorem for simple graphs does not apply and for kindly pointing to the generalized form for multigraphs which gives the $\Delta + 2$ bound.

Note that the dilation of strongly connected components is a direct analogue of the dilation process of [8], while the main addition is the ordering and temporal shifting of the strongly connected components **within** the DAG representing a weakly connected component.

▶ **Theorem 4.10** (D + non-strict $\rightsquigarrow^S$ D + proper). *Given a* D + non-strict *temporal graph* $\mathcal{G}$*,* support-dilation *transforms* $\mathcal{G}$ *into a* D + proper *graph* $\mathcal{H}$ *such that there is a non-strict temporal path in* $\mathcal{G}$ *if and only if there is a strict temporal path in* $\mathcal{H}$ *with the same support.*

**Proof.** ($\mathcal{H}$ is proper.) Every snapshot is processed independently and shifted according to changes in previous snapshots. Within each snapshot, the weakly connected components are handled separately, which is valid as they are independent with respect to a proper labeling. Within each weakly connected component, the strongly connected components are processed independently, following the ordering determined by the corresponding DAG. Each strongly connected component is shifted according to changes occurring earlier in the DAG ordering. As argued by [8], the addition of $c(e) \cdot \varepsilon$ to the labels of every edge $e$ according to the coloring $c$ in the strongly connected components ensures that components are labeled "proper"ly.
($\mathcal{H}$ preserves the temporal paths.) Given a weakly connected component in a snapshot $G_t$ of $\mathcal{G}$ considered independently, let $D = (\mathbb{S}, E)$ be the DAG representing it. Without loss of generality, assume $t = 1$. Now, consider some strongly connected component $S \in \mathbb{S}$ at $\alpha = \ell(S)$ in the ordering of $D$. The longest path in $S$ has length $k$ and in $\mathcal{H}$, every edge of $S$ is assigned all the labels from $\alpha + 1$ to $\alpha + k$. Thus, by the same argument as in [8], for every (non-strict) path of length $k'$ in $S$, there is a *strict* temporal path in the adjustment of $S$ in $\mathcal{H}$ along the same sequence of edges, going over the same edges but with labels $1, 2, \dots, k'$ (up to addition of $c(e) \cdot \varepsilon$). Now, since the order of the strongly connected components in a DAG is maintained, temporal paths over multiple strongly connected components in one snapshot are preserved. The same holds for temporal paths spreading over multiple snapshots. ◀

For a reachability-preserving transformation, we propose a more efficient dilation process prioritizing sparsity. Unlike support-dilation, which reconstructs every path within each strongly connected component, *reachability-dilation* preserves the reachabilities in each strongly connected component by replacing it with a bidirected spanning tree. The components are then connected according to the DAG ordering as in the support-dilation process. This approach replaces an edge between $u$ and $v$ at time step $t$ with at most two temporal edges, which results in a lifetime bounded by $2 \cdot \tau$. Refer to Figure 9 for an illustration.

▶ **Definition 4.11** (Reachability-Dilation). *Let* $\mathcal{G}$ *be a temporal graph and consider each snapshot* $G_t$ *individually. Without loss of generality,* $t = 1$*. Let* $D = (\mathbb{S}, E)$ *be the DAG representing a weakly connected component of* $G_1$*, and* $\ell$ *an ordering of* $D$*.*

*First, we order the edges of* $E$ *lexicographically by the positions of their endpoints in* $\ell$ *(see Definition 4.9 and label the edges in* $E$ between *the strongly connected components in this order with distinct time labels starting from 1. Second, in the order of* $\ell$*, we replace the edges* inside *each strongly connected component as follows (subsequent labels on* $E$ *are shifted accordingly): Let* $S \in \mathbb{S}$*. As* $S$ *is strongly connected, the underlying undirected graph contains a spanning tree* $T_S$*. It was shown by [10] that one can temporally connect any undirected tree using at most two labels per edge. Slightly adjusting their construction and argument, we temporally connect* $T_S$ *by turning it into a bidirected tree and placing one distinct time label one each directed edge: Choose some vertex of* $S$ *as the arbitrary root. Starting at the leafs, assign distinct, increasing labels to the upwards edge until reaching the root. Now, starting at the root, assign further increasing labels to the downwards edges until reaching every leaf.*

*After processing every snapshot, reachability-dilation returns the adjusted temporal graph.*

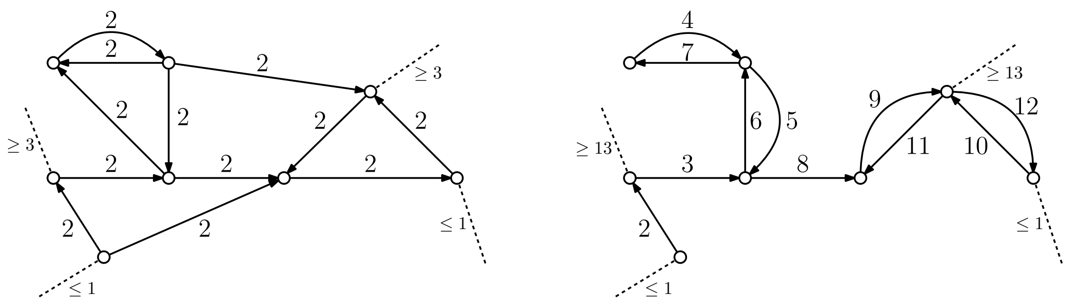



**Figure 9** Illustration of the *reachability-dilation* described in Definition 4.11 for some $\mathcal{G}$ with snapshot $G_2$ (left) and the proper labeling with the bidirected subtrees in the DAG structure (right). The dotted edges represent adjacent earlier and later (shifted in the proper labeling) edges.

In the following, we prove that reachability-dilation is reachability-preserving. But first, we show that the modified spanning tree construction yields a temporally connected graph.

▶ **Lemma 4.12.** *Any bidirected tree with root $r$, whose labels are assigned as in Definition 4.11 (starting at the leafs, assign distinct, increasing labels to the upwards edge until reaching the root – then starting at the root, assign further increasing labels to the downwards edges until reaching every leaf) is temporally connected.*

**Proof.** By construction, the upwards directed edges form temporal paths from the leafs to the root $r$, and the downwards directed edges form temporal paths from $r$ to the leafs. Since all upwards paths are assigned time labels strictly smaller than all labels of the downwards paths, $r$ is a pivot vertex, i. e., there is a time step $t$ such that every vertex of the tree reaches $r$ before $t$ and can be reached by $r$ after $t$. Thus, the tree is temporally connected. ◀

▶ **Theorem 4.13.** *Given a* D + non-strict *temporal graph $\mathcal{G}$,* reachability-dilation *transforms $\mathcal{G}$ into a* D + proper *temporal graph $\mathcal{H}$ with $\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$.*

**Proof.** ($\mathcal{H}$ is proper.) This follows by the same arguments as in Theorem 4.10 and the fact that the spanning tree labeling is proper.
($\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$.) Given a weakly connected component in a snapshot $G_t$ of $\mathcal{G}$ considered independently, let $D = (\mathbb{S}, E)$ be the DAG representing it. Let $S \in \mathbb{S}$ be a strongly connected component in $D$. By Lemma 4.12, the edges in $S$ are replaced by a proper, temporally connected bidirected tree spanning all vertices of $S$. Thus, all vertices in $S$ can reach one another in $\mathcal{H}$ within the extended time interval of $S$, so the reachability inside $S$ in $\mathcal{H}$ is the same as in $\mathcal{G}$. Now, since the chronological order of the strongly connected components in any DAG is maintained and the footprint of the DAG is the same, temporal reachability over multiple strongly connected components in one snapshot is the same in $\mathcal{H}$ and $\mathcal{G}$. Lastly, since the chronological order of the snapshots is also maintained, the temporal reachability via multiple snapshots is also equivalent. ◀

Reachability-dilation can also be executed on undirected graphs, where it replaces each connected component of every snapshot with an undirected spanning tree. This process will be discussed in more detail in Section 5, Lemma 5.3.

### 4.3.2 Directed Semaphore: D + strict $\rightsquigarrow^{\boldsymbol{iR}}$ D + simple + proper

We adapt the semaphore technique of [8] to directed graphs. This transformation preserves *induced-reachability* while producing a simple + proper graph by replacing each edge with a path of length 2 to ensure a simple labeling and shifting labels to ensure a proper labeling.

▶ **Definition 4.14** (Semaphore (directed)). *Let $\mathcal{G}$ be a directed temporal graph with temporal edges $\mathcal{E}$, let $\Delta$ be the maximum degree of the footprint $G$, and let $0 < \varepsilon < \frac{1}{2(\Delta+1)}$. Consider an edge-coloring $c$ of $G$ using $\Delta + 1$ colors (which exists by Vizing's theorem).*

*Replace every directed temporal edge $e = (u, v, t) \in \mathcal{E}$ with an auxiliary vertex $\overrightarrow{uv}$ and two temporal edges $(u, \overrightarrow{uv}, t - c(e) \cdot \varepsilon)$ and $(\overrightarrow{uv}, v, t + c(e) \cdot \varepsilon)$.*

▶ **Theorem 4.15** (D + strict $\rightsquigarrow^{iR}$ D + simple + proper). *Given a temporal graph $\mathcal{G} \in$ D + strict, semaphore transforms $\mathcal{G}$ into a* D + simple + proper *temporal graph $\mathcal{H}$ such that there is $\sigma\colon V(\mathcal{G}) \to V(\mathcal{H})$ with $(u, v) \in \mathcal{R}(\mathcal{G})$ if and only if $(\sigma(u), \sigma(v)) \in \mathcal{R}(\mathcal{H})$.*

The proof of Theorem 4.15 is almost identical to that of [8, Theorem 2], so we give only a brief intuition: $\mathcal{H}$ is obviously simple, and proper by the same reasoning as for Theorem 4.10. The reachability equivalence for $V(\mathcal{G})$ follows from the construction: Each directed edge $(u, v, t)$ is subdivided into two parts. The first half is assigned a label less than $t$, and the second half is assigned a label greater than $t$ (shifted according to the edge coloring). As a result, after traversing from $u$ to $v$ via the subdivided edge, any temporal path reaches $v$ at a time later than any first-half label of any original edge $(v, x, t)$. This ensures that a temporal path can take at most one of the original edges at each time step, and thus is strict.

As for undirected graphs, one can apply dilation and semaphore, to transform any directed setting into D + simple + proper with preserved induced-reachability. Since D + simple + proper is subset of every setting, we conclude the following.

▶ **Corollary 4.16** (D + * $\leftrightsquigarrow^{iR}$ D + simple + proper). *All directed temporal graph classes are induced-reachability equivalent.*

## 5 Comparison of Directed and Undirected Reachability

Finally, we compare directed and undirected temporal graph classes with respect to reachability equivalence. Refer back to Figure 2 for an overview.

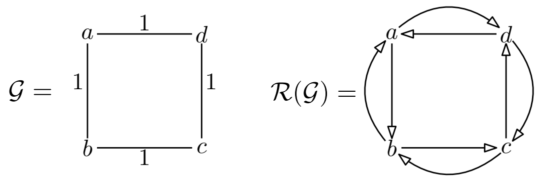
First, observe that directed settings are inherently more expressive than their undirected counterparts: the reachability graph of a single directed edge cannot be realized in any undirected setting under equivalence notions stronger than induced-reachability, whereas each undirected edge can be replaced by two opposing directed edges. This suggests that any graph in an undirected setting UD + x can be transformed into a support equivalent graph in the corresponding directed setting D + x using this simple replacement. For both strict and non-strict settings, this transformation works seamlessly, and the resulting directed graph is simple if and only if the original undirected graph was simple. For proper graphs, however, the opposing edges $(u, v)$ and $(v, u)$ derived from $(uv, t)$ inherit the same label $t$. But since such edges never appear in the same temporal path, their labels can be shifted-without affecting reachability-to restore a proper labeling. With this, we make the following observation:

▶ **Observation 5.1.** *It holds that* UD + x $\rightsquigarrow^{R}$ D + x, *but* D + x $\not\rightsquigarrow^{R}$ UD + x.

The remaining question is whether we can transform an undirected setting into a directed setting at "a level lower". For UD + strict and UD + strict + simple this is not the case.

▶ **Lemma 5.2** (UD + strict + simple $\not\rightsquigarrow^{R}$ D + non-strict). *There exists a graph $\mathcal{G} \in$* UD + strict + simple *such that there is no $\mathcal{H} \in$* D + non-strict *with $\mathcal{R}(\mathcal{H}) = \mathcal{R}(\mathcal{G})$.*

**Proof.** Consider the following temporal graph $\mathcal{G}$ in the UD + strict + simple setting (left) and the corresponding reachability graph (right). For the sake of contradiction, let $\mathcal{H}$ be a temporal graph in the D + non-strict setting whose reachability graph is isomorphic to that




of $\mathcal{G}$. Note that in $\mathcal{G}$, $a$ reaches both $b$ and $d$, while $d$ does not reach $b$. Therefore, the direct edge $(a, b)$ must be included in $\mathcal{H}$ to ensure that $a$ can reach $b$. By the same reasoning, every edge in $\mathcal{R}(\mathcal{G})$ has to be included in $\mathcal{H}$.

Consider the cycle $a, b, c, d, a$. Regardless of how the edges on this cycle are labeled, Lemma 4.1 implies that this will create at least one transitive reachability. For example, the labeling $(a, b, 4)$, $(b, c, 3)$, $(c, d, 2)$, $(d, a, 1)$, results in $d$ reaching $b$. However, no such transitive reachability appears in $\mathcal{R}(\mathcal{G})$. Therefore, there exists no valid labeling for $\mathcal{H}$. ◀

Interestingly, among the non-strict and proper settings, there exists indeed one such "level-breaking" transformation. Applying the reachability-dilation (Definition 4.11) to UD + non-strict + simple graphs, produces a reachability equivalent D + proper + simple graph. Reachability dilation replaces each connected component in a snapshot with a bidirected spanning tree, preserving reachability while ensuring properness and a simple labeling. The proof follows from the proof for directed reachability dilation (Theorem 4.13). Refer to Figure 10 for an illustration of this transformation.

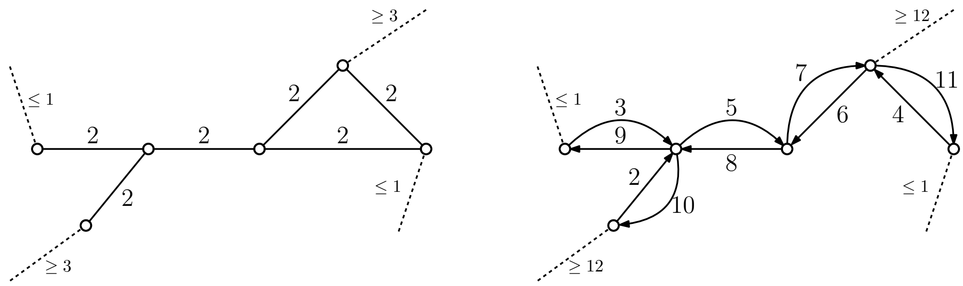



**Figure 10** Illustration of the *reachability-dilation* (Definition 4.11) applied to an UD + non-strict + simple graph $\mathcal{G}$ with snapshot $G_2$ (left) and the proper labeling with the bidirected subtrees (right). The dotted edges represent adjacent earlier and later edges (shifted in the proper labeling).

▶ **Lemma 5.3** (UD + non-strict + simple $\rightsquigarrow^R$ D + proper + simple)**.** *Given a temporal graph* $\mathcal{G} \in$ UD + non-strict + simple*,* reachability dilation *transforms* $\mathcal{G}$ *into a a* D + proper + simple *temporal graph* $\mathcal{H}$ *with* $\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$*.*

**Proof.** ($\mathcal{H}$ is proper.) This follows from the same arguments as in Theorem 4.13.
($\mathcal{H}$ is simple.) Reachability-dilation replaces each connected component of a snapshot of $\mathcal{G}$ with a spanning tree of the underlying footprint. Since the component is connected, such a tree always exists. By this process, each edge $uv$ within the component is either removed or replaced by two opposing directed temporal edges, each assigned one time label. As $\mathcal{G}$ is simple, no pair of vertices will be considered more than once, and consequently, in $\mathcal{H}$, each pair of vertices is connected by either zero or two opposing directed, single labeled edges.
($\mathcal{R}(\mathcal{G}) = \mathcal{R}(\mathcal{H})$.) This follows from the same arguments as in Theorem 4.13. ◀

## 6 Conclusion and Future Work

We extended the framework of Casteigts, Corsini, and Sarkar [8] from undirected to directed temporal graphs, analyzing how definitional choices of temporal graphs affect their reachability expressivity. Using the equivalence notions of support, reachability, and induced-reachability, we fully resolved the directed hierarchy, completed the undirected one, and nearly resolved their merged comparison.

Two natural questions remain: whether every undirected + non-strict graph can be transformed into a reachability-equivalent directed + proper + simple graph, or even into a directed + non-strict + simple graph.

**Future work.** A key structural direction is to characterize which static graphs can occur as reachability graphs in a given temporal setting. Our separating structures – such as the directed triangle – offer a first step toward such characterizations.

Another important direction is to study the efficiency of transformations between temporal settings. While this work focused on existence, future research could investigate how to minimize the number of time labels, preserve structural parameters such as treewidth or lifetime, or ensure compatibility with specific computational problems. Many (folklore) transformations already exist in the literature, and it could be valuable to collect, formalize, and analyze them within a unified framework.

Perhaps most promising is the potential impact on algorithmic research. Many hardness results and algorithms for temporal problems – such as shortest paths [23], connected components [4], exact edge covers [13], or temporal versions of Menger's Theorem [3] – are currently first shown in one setting and then reproven via ad hoc adaptations. Our framework opens the door to formalizing such generalizations by identifying which problems are invariant under equivalence notions or specific transformations. For example, maximum flow is preserved under the semaphore construction, and open connected components remain invariant under reachability equivalence. Establishing which problems admit such invariance would streamline future work, allowing results to transfer across settings without reproving them from scratch.

## References


1. Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Leszek Gasieniec. The Complexity of Optimal Design of Temporally Connected Graphs. *Theory of Computing Systems / Mathematical Systems Theory*, 61(3):907–944, April 2017. Publisher: Springer US. doi:10.1007/S00224-017-9757-X.
2. Brenda Baker and Robert Shostak. Gossips and Telephones. *Discrete Mathematics*, 2(3):191–193, June 1972. doi:10.1016/0012-365X(72)90001-5.
3. Kenneth A. Berman. Vulnerability of scheduled networks and a generalization of Menger's Theorem. *Networks: An International Journal*, 28(3):125–134, 1996. doi:10.1002/(SICI)1097-0037(199610)28:3<125::AID-NET1>3.0.CO;2-P.
4. Sandeep Bhadra and Afonso Ferreira. Complexity of Connected Components in Evolving Graphs and the Computation of Multicast Trees in Dynamic Networks. In Samuel Pierre, Michel Barbeau, and Evangelos Kranakis, editors, *Ad-Hoc, Mobile, and Wireless Networks*, pages 259–270, Berlin, Heidelberg, 2003. Springer. doi:10.1007/978-3-540-39611-6_23.
5. Hans L. Bodlaender and Tom C. van der Zanden. On exploring always-connected temporal graphs of small pathwidth. *Information Processing Letters*, 142:68–71, February 2019. doi:10.1016/j.ipl.2018.10.016.

6 Richard T. Bumby. A Problem with Telephones. *SIAM Journal on Algebraic Discrete Methods*, 2(1):13–18, March 1981. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/0602002.

7 Arnaud Casteigts and Timothée Corsini. In search of the lost tree: Hardness and relaxation of spanning trees in temporal graphs, December 2023. arXiv:2312.06260 [cs]. doi:10.48550/arXiv.2312.06260.

8 Arnaud Casteigts, Timothée Corsini, and Writika Sarkar. Simple, Strict, Proper, Happy: A Study of Reachability in Temporal Graphs. *Theoretical Computer Science*, 991:114434, April 2024. doi:10.1016/j.tcs.2024.114434.

9 Arnaud Casteigts, Michael Raskin, Malte Renken, and Viktor Zamaraev. Sharp Thresholds in Random Simple Temporal Graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–326, Denver, CO, USA, February 2022. IEEE. doi:10.1109/FOCS52979.2021.00040.

10 Esteban Christiann, Eric Sanlaville, and Jason Schoeters. On Inefficiently Connecting Temporal Networks. In Arnaud Casteigts and Fabian Kuhn, editors, *3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2024)*, volume 292 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISSN: 1868-8969. doi:10.4230/LIPIcs.SAND.2024.8.

11 Isnard Lopes Costa, Raul Lopes, Andrea Marino, and Ana Silva. On Computing Large Temporal (Unilateral) Connected Components. In *Hsieh, SY., Hung, LJ., Lee, CW. (eds) Combinatorial Algorithms*, volume 13889 of *Lecture Notes in Computer Science*. Springer, February 2023. arXiv:2302.12068 [math]. doi:10.1007/978-3-031-34347-6_24.

12 Argyrios Deligkas, Michelle Döring, Eduard Eiben, Tiger-Lily Goldsmith, and George Skretas. Being an influencer is hard: The complexity of influence maximization in temporal graphs with a fixed source. *Information and Computation*, 299:105171, August 2024. doi:10.1016/j.ic.2024.105171.

13 Argyrios Deligkas, Michelle Döring, Eduard Eiben, Tiger-Lily Goldsmith, George Skretas, and Georg Tennigkeit. How Many Lines to Paint the City: Exact Edge-Cover in Temporal Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):26498–26506, April 2025. Number: 25. doi:10.1609/aaai.v39i25.34850.

14 Thomas Erlebach, Othon Michail, and Nils Morawietz. Recognizing and Realizing Temporal Reachability Graphs. In *33rd Annual European Symposium on Algorithms, ESA 2025*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.ESA.2025.93.

15 Thomas Erlebach and Jakob T. Spooner. Exploration of k-edge-deficient temporal graphs. *Acta Informatica*, 59(4):387–407, August 2022. doi:10.1007/s00236-022-00421-5.

16 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal Graph Classes: A View Through Temporal Separators. *Theoretical Computer Science*, 806:197–218, 2020. Publisher: Elsevier. doi:10.1016/J.TCS.2019.03.031.

17 Eugen Füchsle, Hendrik Molter, Rolf Niedermeier, and Malte Renken. Delay-Robust Routes in Temporal Graphs. In *39th International Symposium on Theoretical Aspects of Computer Science*, volume 219, pages 30:1–30:15, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2022), 2022. Leibniz International Proceedings in Informatics (LIPIcs). arXiv:2201.05390 [cs]. doi:10.4230/LIPIcs.STACS.2022.30.

18 F. Göbel, J. Orestes Cerdeira, and H. J. Veldman. Label-Connected Graphs and the Gossip Problem. *Discrete Mathematics*, 87(1):29–40, January 1991. doi:10.1016/0012-365X(91)90068-D.

19 A. Hajnal, E. C. Milner, and E. Szemerédi. A Cure for the Telephone Disease. *Canadian Mathematical Bulletin*, 15(3):447–450, 1972. doi:10.4153/CMB-1972-081-0.

20 David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and Inference Problems for Temporal Networks. *Journal of Computer and System Sciences*, 64(4):820–842, June 2002. doi:10.1006/jcss.2002.1829.

21 Nina Klobas, George B. Mertzios, Hendrik Molter, and Paul G. Spirakis. Temporal Graph Realization from Fastest Paths. *LIPIcs, Volume 292, SAND 2024*, 292:16:1–16:18, 2024. Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.SAND.2024.16.

22 Mathilde Vernet, Maciej Drozdowski, Yoann Pigné, and Eric Sanlaville. A Theoretical and Experimental Study of a New Algorithm for Minimum Cost Flow in Dynamic Graphs. *Discrete Applied Mathematics*, 296:203–216, June 2021. doi:10.1016/j.dam.2019.12.012.

23 B. Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks, October 2002.