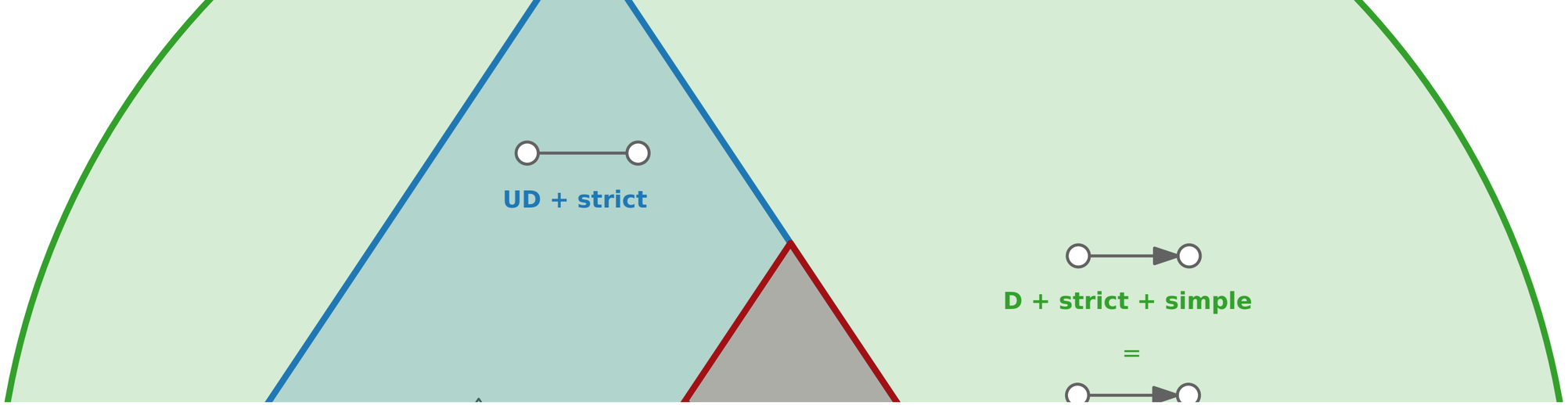
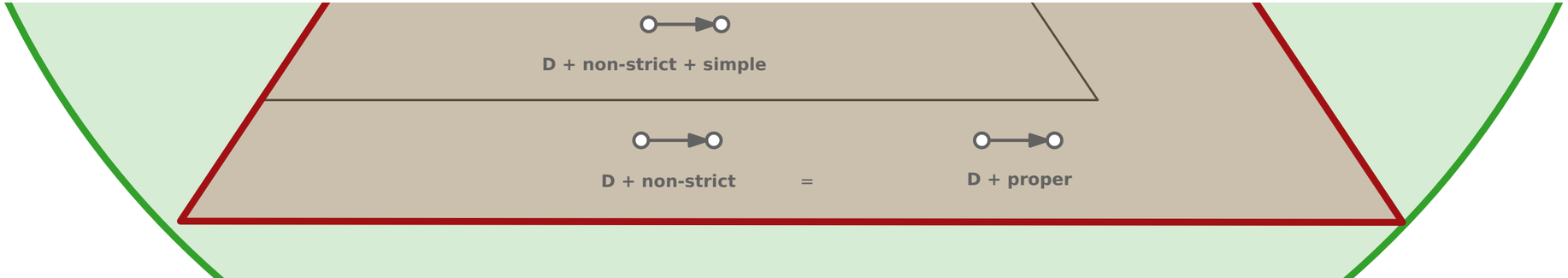



# Reachability in Temporal Graph Settings: A Structural Comparison

Michelle Döring

Algorithmic Aspects of Temporal Graphs
ICALP 2025 Satellite Workshop
Aarhus, Denmark, 07. July 2025

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

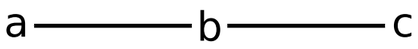


# Why Should We Read Preliminaries?

# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$

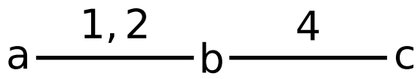
$G$ :



# Why Should We Read Preliminaries?

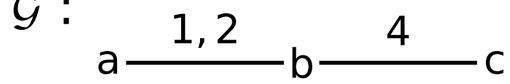
- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$

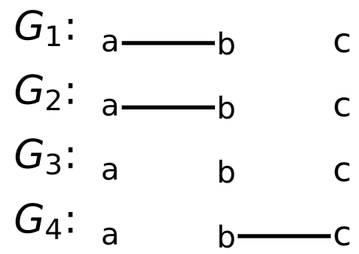
$\mathcal{G}$ :



# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$

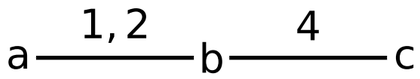


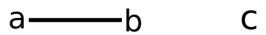
# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

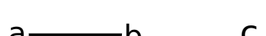
$\mathcal{G}$ :




$G_1$:



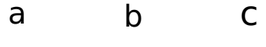

$G_2$:




$G_3$:



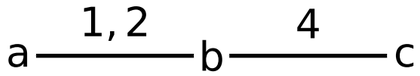

$G_4$:



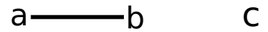
# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

$\mathcal{G}$ :




$G_1$:




$G_2$:


$G_3$:



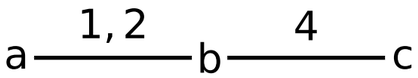

$G_4$:


directed / undirected

# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

$\mathcal{G}$ :



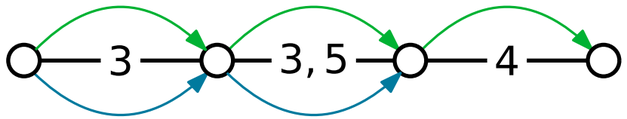

$G_1$: a b c

$G_2$: a b c

$G_3$: a b c

$G_4$: a b c

directed / undirected

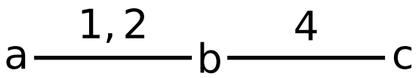



strict / non-strict

# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

$\mathcal{G}$ :



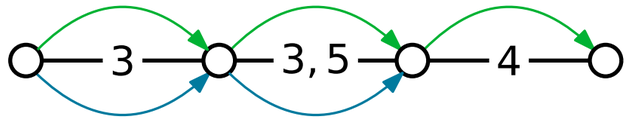

$G_1$: a b c

$G_2$: a b c

$G_3$: a b c

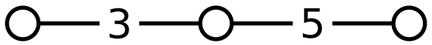
$G_4$: a b c

directed / undirected




strict / non-strict

## $\lambda$ (not) restricted to simple

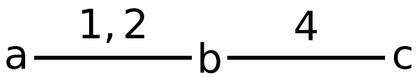
simple 


not simple 

# Why Should We Read Preliminaries?

- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

$\mathcal{G}$ :



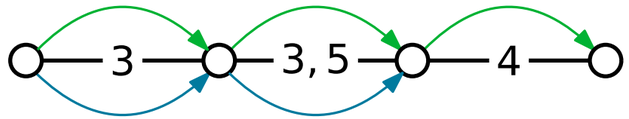

$G_1$: a b c

$G_2$: a b c

$G_3$: a b c

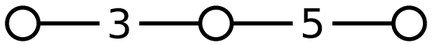
$G_4$: a b c

directed / undirected



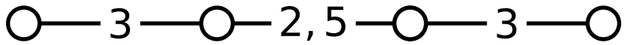

strict / non-strict

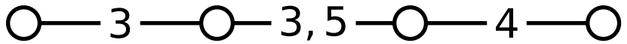
## $\lambda$ (not) restricted to simple

simple 


not simple 


## $\lambda$ (not) restricted to proper

proper 

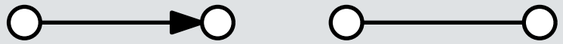

not proper 

# Why Should We Read Preliminaries?

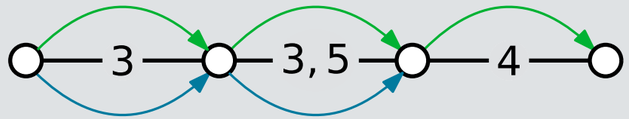
- static graph is a tuple $G = (V, E)$
- temporal graph is a tuple $\mathcal{G} = (V, E, \lambda)$ or a sequence of snapshots $G_1, \ldots, G_\tau$
- temporal path is a sequence of temporal edges $(e_1, t_1), \ldots, (e_k, t_k)$ such that the time labels $t_i$ are non-decreasing

$\mathcal{G}$ : a —1,2— b —4— c

$G_1$: a—b c
$G_2$: a—b c
$G_3$: a b c
$G_4$: a b—c



directed / undirected




strict / non-strict

## TEMPORAL SETTINGS

[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]

### $\lambda$ (not) restricted to simple

simple — 3 — 5

not simple — 3 — 3, 5

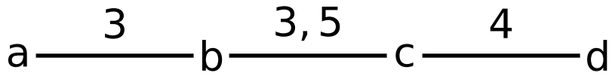
### $\lambda$ (not) restricted to proper

proper — 3 — 2, 5 — 3

not proper — 3 — 3, 5 — 4

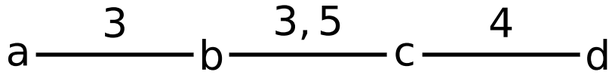
# (How) Does It Make a Difference?

$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict

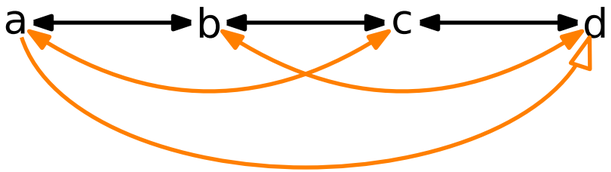


# (How) Does It Make a Difference?

$\mathcal{G}$



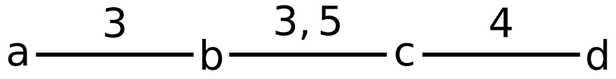

$\mathcal{R}(\mathcal{G})$ non-strict



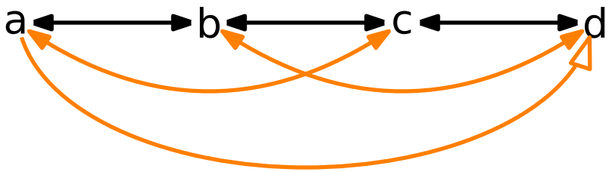

strict

# (How) Does It Make a Difference?

$\mathcal{G}$



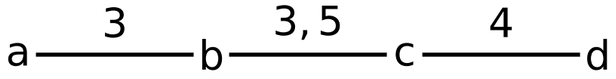

$\mathcal{R}(\mathcal{G})$ non-strict



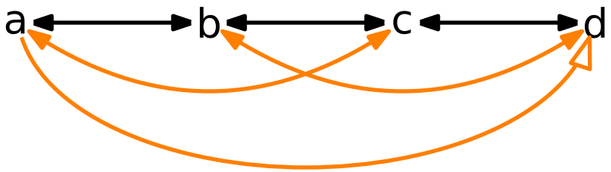

strict

# (How) Does It Make a Difference?

$\mathcal{G}$



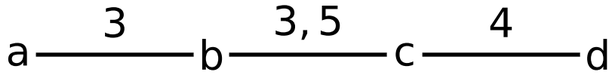

$\mathcal{R}(\mathcal{G})$ non-strict



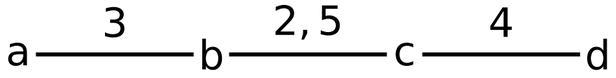

$\neq$ strict

# (How) Does It Make a Difference?

$\mathcal{G}$

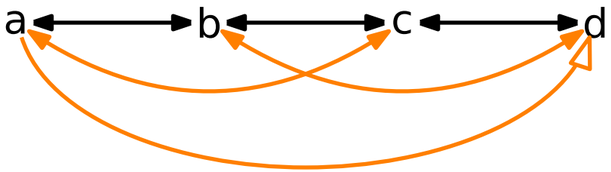



proper




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

non-strict

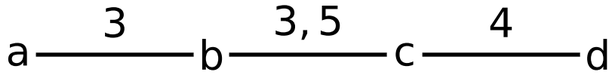



$\neq$

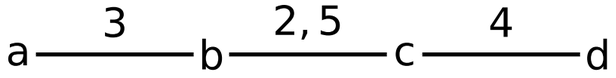
strict

# (How) Does It Make a Difference?

$\mathcal{G}$

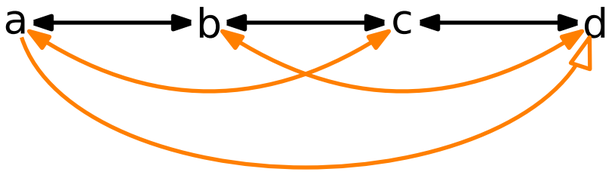



proper




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

non-strict



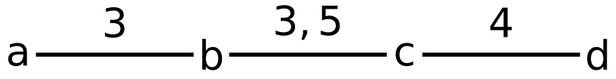


$\neq$

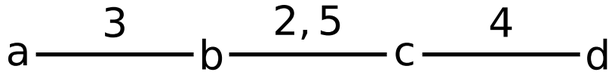
strict

# (How) Does It Make a Difference?

$\mathcal{G}$

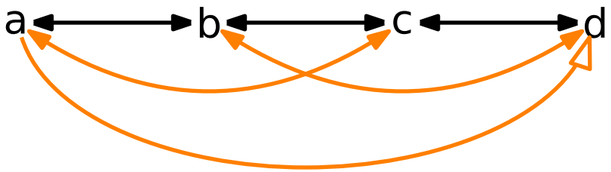



proper




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

non-strict



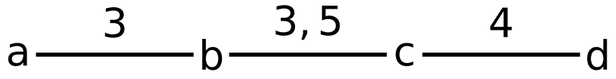

$\neq$

strict

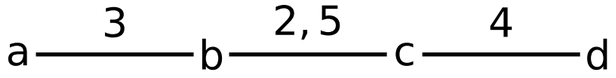


# (How) Does It Make a Difference?

$\mathcal{G}$

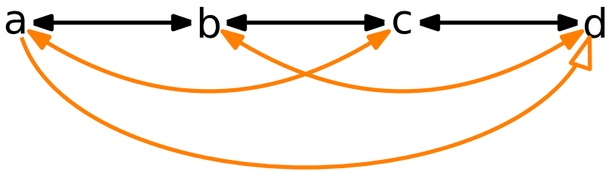



proper




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

non-strict




$\neq$

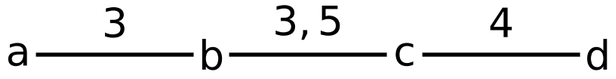
strict



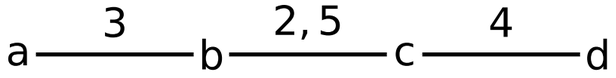
$=$

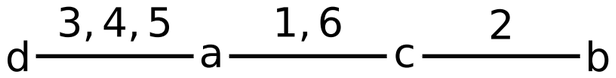
# (How) Does It Make a Difference?

$\mathcal{G}$



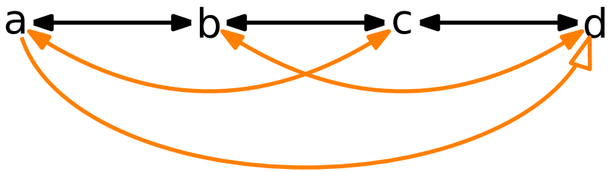

proper




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

non-strict




$\neq$

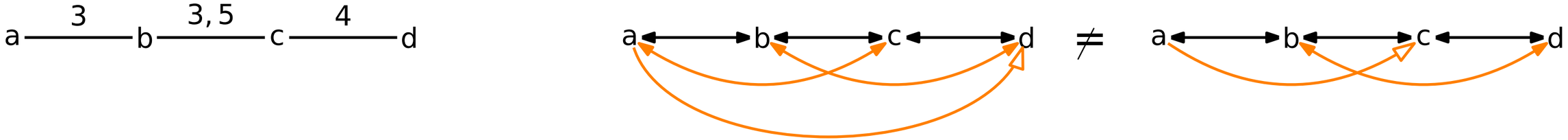
strict



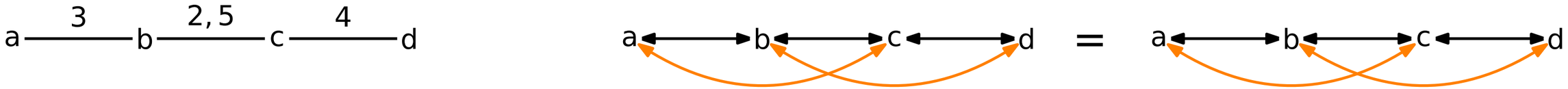
$=$

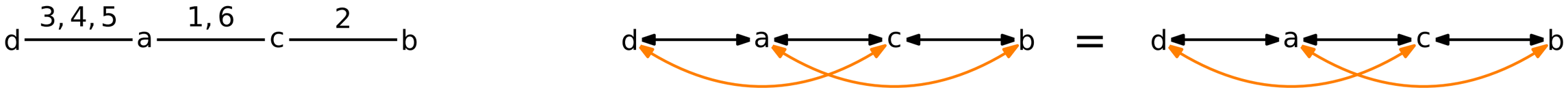
# (How) Does It Make a Difference?

$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict



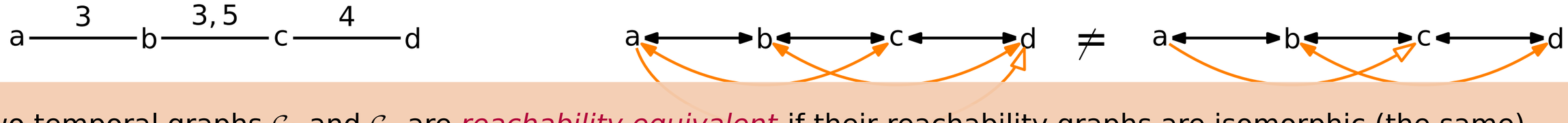

proper



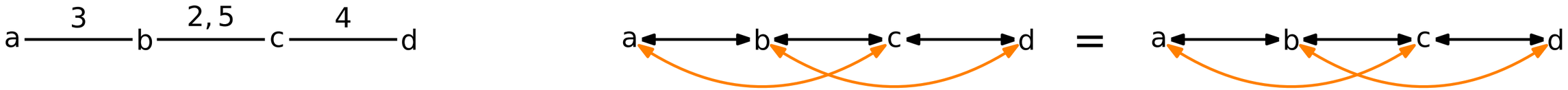
# (How) Does It Make a Difference?

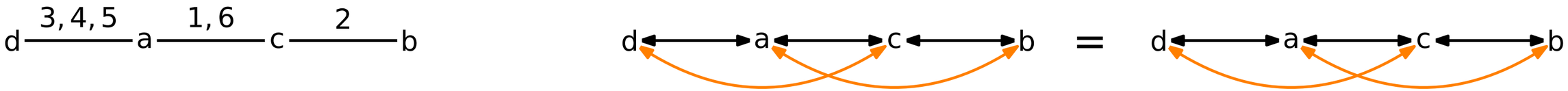
$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict




Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

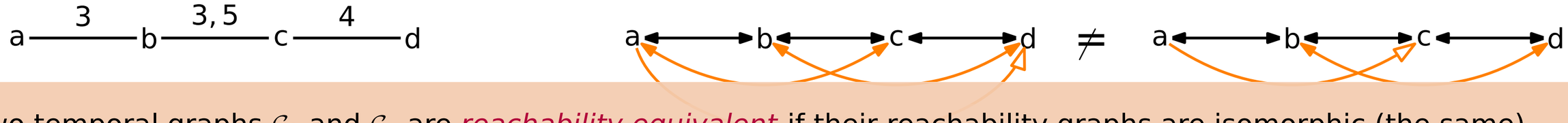
proper



# (How) Does It Make a Difference?

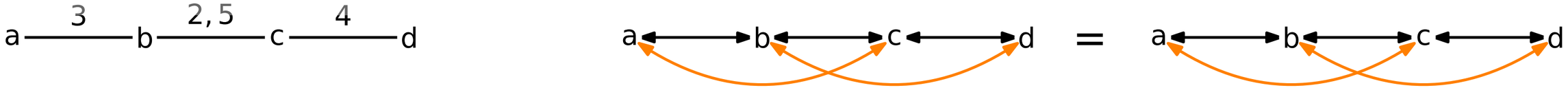
$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict

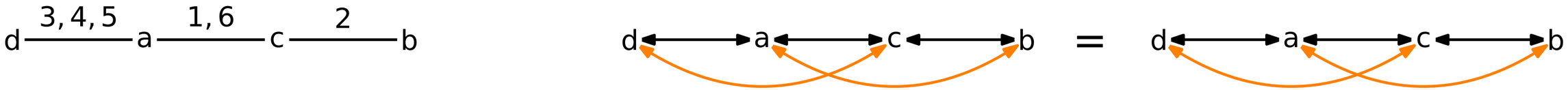



Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

A temporal graph setting $\mathbb{S}$ can *simulate* setting $\mathbb{T}$, if for every $\mathcal{G}_\mathbb{T} \in \mathbb{T}$ exists a reachability equivalent $\mathcal{G}_\mathbb{S} \in \mathbb{S}$.

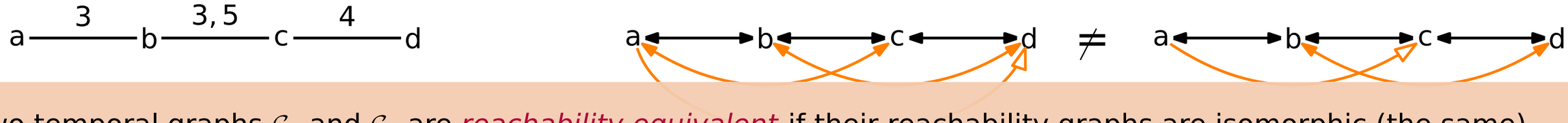
proper



# (How) Does It Make a Difference?

$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict

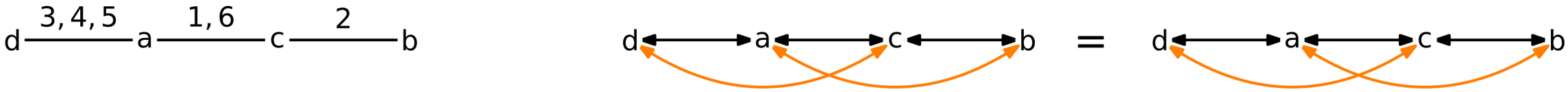



Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

A temporal graph setting $\mathbb{S}$ can *simulate* setting $\mathbb{T}$, if for every $\mathcal{G}_\mathbb{T} \in \mathbb{T}$ exists a reachability equivalent $\mathcal{G}_\mathbb{S} \in \mathbb{S}$.

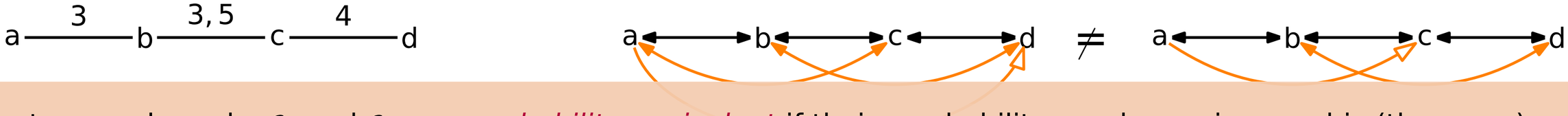
A temporal graph setting $\mathbb{S}$ is *strictly more expressive* than setting $\mathbb{T}$, if

1. $\mathbb{S}$ can simulate $\mathbb{T}$ but
2. $\mathbb{T}$ cannot simulate $\mathbb{S}$.



# (How) Does It Make a Difference?

$\mathcal{G}$ $\mathcal{R}(\mathcal{G})$ non-strict strict

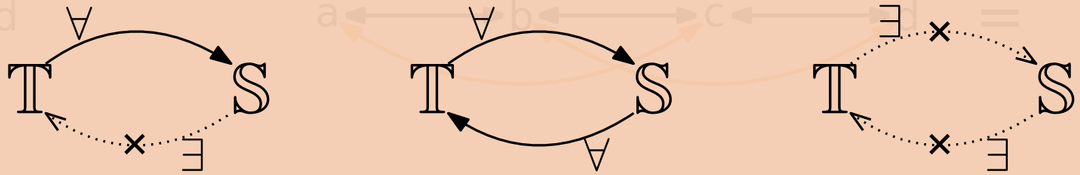


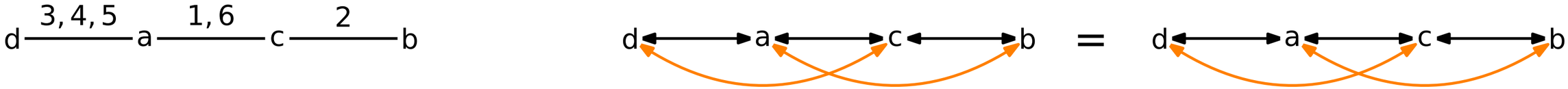

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

A temporal graph setting $\mathbb{S}$ can *simulate* setting $\mathbb{T}$, if for every $\mathcal{G}_\mathbb{T} \in \mathbb{T}$ exists a reachability equivalent $\mathcal{G}_\mathbb{S} \in \mathbb{S}$.

A temporal graph setting $\mathbb{S}$ is *strictly more expressive* than setting $\mathbb{T}$, if

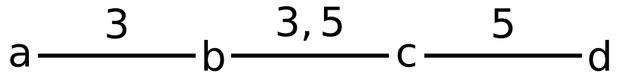
1. $\mathbb{S}$ can simulate $\mathbb{T}$ but
2. $\mathbb{T}$ cannot simulate $\mathbb{S}$.



# UD + Strict is More Expressive than UD + Strict + Simple

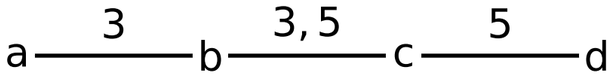
$\mathcal{G} \in$ undirected + strict + multilabel




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

# UD + Strict is More Expressive than UD + Strict + Simple

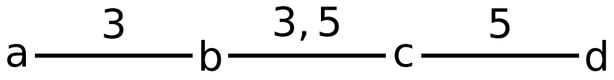
$\mathcal{G} \in$ undirected + strict + multilabel




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

# UD + Strict is More Expressive than UD + Strict + Simple

$\mathcal{G} \in$ undirected + strict + multilabel




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

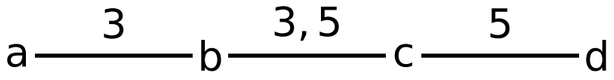

- possible footprint of transformed $\mathcal{H} \in$ undirected + strict + simple: all undirected edges in $\mathcal{R}(\mathcal{G})$

$\mathcal{H} \in$ undirected + strict + simple


$\mathcal{R}(\mathcal{H})$

# UD + Strict is More Expressive than UD + Strict + Simple

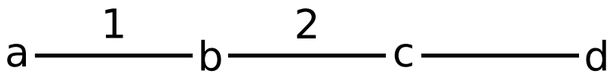
$\mathcal{G} \in$ undirected + strict + multilabel



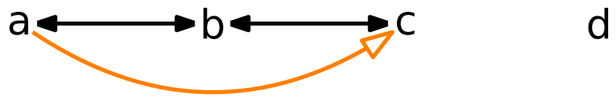

$\mathcal{R}(\mathcal{G})$


- possible footprint of transformed $\mathcal{H} \in$ undirected + strict + simple: all undirected edges in $\mathcal{R}(\mathcal{G})$
- $a \rightsquigarrow c$ requires $\lambda(ab) < \lambda(bc)$

$\mathcal{H} \in$ undirected + strict + simple

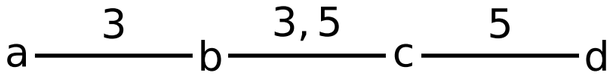



$\mathcal{R}(\mathcal{H})$



# UD + Strict is More Expressive than UD + Strict + Simple

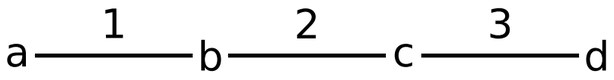
$\mathcal{G} \in$ undirected + strict + multilabel




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


- possible footprint of transformed $\mathcal{H} \in$ undirected + strict + simple: all undirected edges in $\mathcal{R}(\mathcal{G})$
- $a \rightsquigarrow c$ requires $\lambda(ab) < \lambda(bc)$
- $b \rightsquigarrow d$ requires $\lambda(bc) < \lambda(cd)$

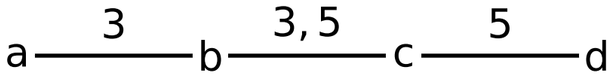
$\mathcal{H} \in$ undirected + strict + simple




$\mathcal{R}(\mathcal{H})$

# UD + Strict is More Expressive than UD + Strict + Simple

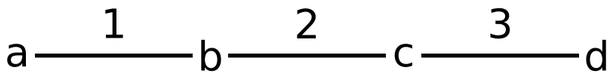
$\mathcal{G} \in$ undirected + strict + multilabel



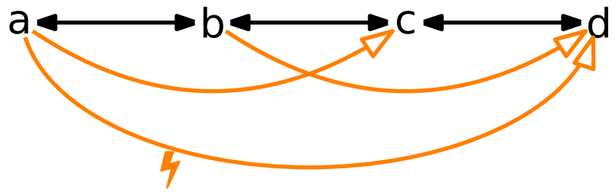

$\mathcal{R}(\mathcal{G})$


- possible footprint of transformed $\mathcal{H} \in$ undirected + strict + simple: all undirected edges in $\mathcal{R}(\mathcal{G})$
- $a \rightsquigarrow c$ requires $\lambda(ab) < \lambda(bc)$
- $b \rightsquigarrow d$ requires $\lambda(bc) < \lambda(cd)$
- but now $a \rightsquigarrow d$ ↯

$\mathcal{H} \in$ undirected + strict + simple

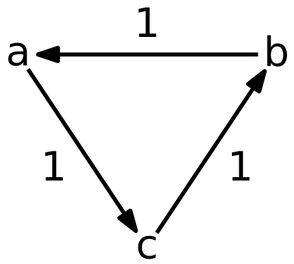



$\mathcal{R}(\mathcal{H})$

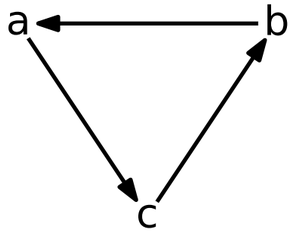


# D + Strict is More Expressive than D + Nonstrict

$\mathcal{G} \in$ directed + strict + multilabel

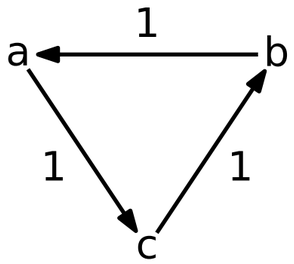



$\mathcal{R}(\mathcal{G})$

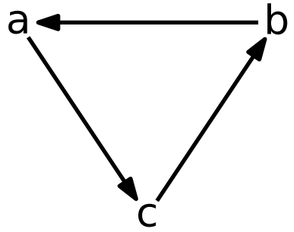


# D + Strict is More Expressive than D + Nonstrict

$\mathcal{G} \in$ directed + strict + multilabel

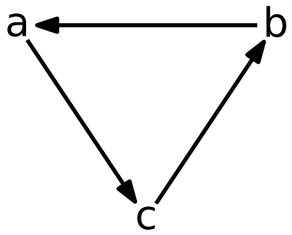



$\mathcal{R}(\mathcal{G})$

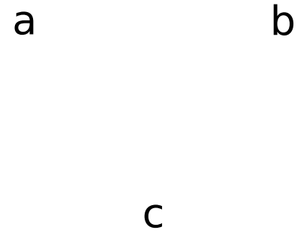



- possible footprint of transformed $\mathcal{H} \in$ directed + nonstrict + multilabel: all directed edges in $\mathcal{R}(\mathcal{G})$

$\mathcal{H} \in$ directed + nonstrict + multilabel

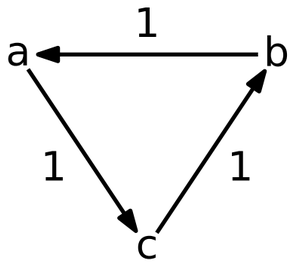



$\mathcal{R}(\mathcal{H})$

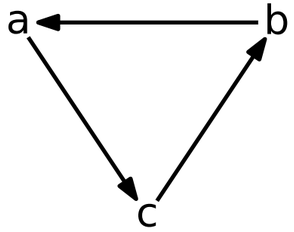


# D + Strict is More Expressive than D + Nonstrict

$\mathcal{G} \in$ directed + strict + multilabel

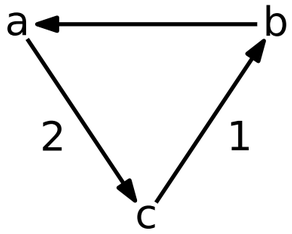



$\mathcal{R}(\mathcal{G})$

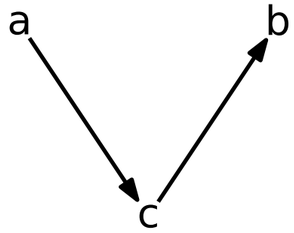



- possible footprint of transformed $\mathcal{H} \in$ directed + nonstrict + multilabel: all directed edges in $\mathcal{R}(\mathcal{G})$
- $a \not\rightsquigarrow b$ requires $\lambda(ac) > \lambda(cb)$

$\mathcal{H} \in$ directed + nonstrict + multilabel

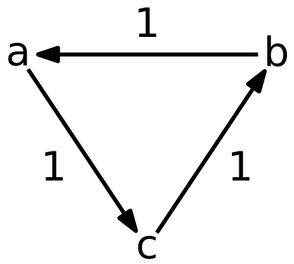



$\mathcal{R}(\mathcal{H})$

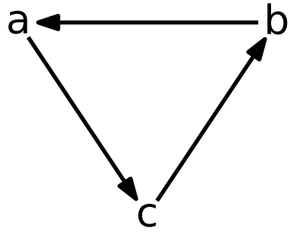


# D + Strict is More Expressive than D + Nonstrict

$\mathcal{G} \in$ directed + strict + multilabel

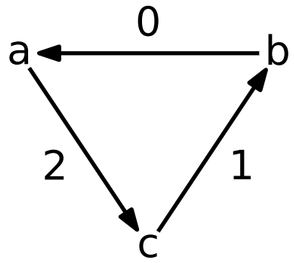



$\mathcal{R}(\mathcal{G})$

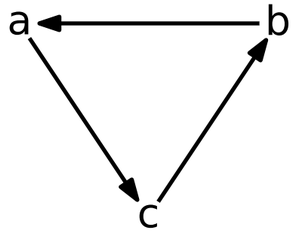



- possible footprint of transformed $\mathcal{H} \in$ directed + nonstrict + multilabel: all directed edges in $\mathcal{R}(\mathcal{G})$
- $a \not\rightsquigarrow b$ requires $\lambda(ac) > \lambda(cb)$
- $c \not\rightsquigarrow a$ requires $\lambda(cb) > \lambda(ba)$

$\mathcal{H} \in$ directed + nonstrict + multilabel

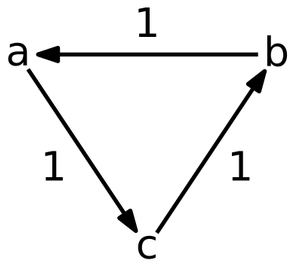



$\mathcal{R}(\mathcal{H})$

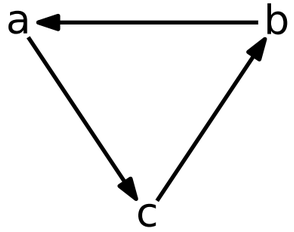


# D + Strict is More Expressive than D + Nonstrict

$\mathcal{G} \in$ directed + strict + multilabel

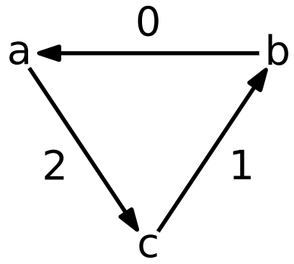



$\mathcal{R}(\mathcal{G})$

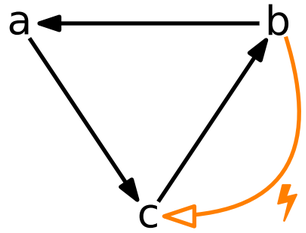



- possible footprint of transformed $\mathcal{H} \in$ directed + nonstrict + multilabel: all directed edges in $\mathcal{R}(\mathcal{G})$
- $a \not\rightsquigarrow b$ requires $\lambda(ac) > \lambda(cb)$
- $c \not\rightsquigarrow a$ requires $\lambda(cb) > \lambda(ba)$
- but now $b \rightsquigarrow c$ ↯

$\mathcal{H} \in$ directed + nonstrict + multilabel




$\mathcal{R}(\mathcal{H})$



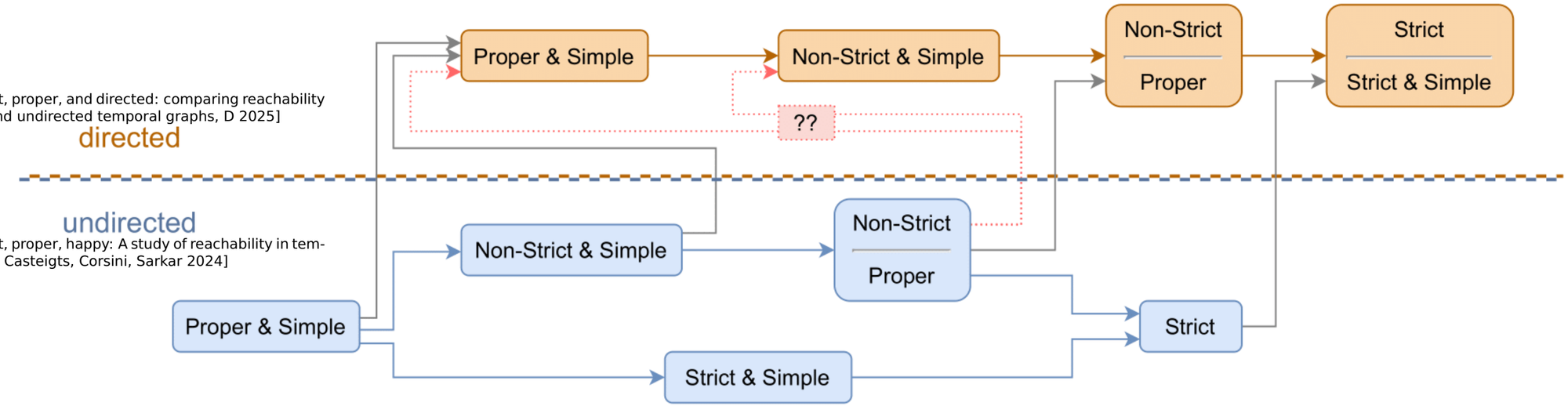
# Directed and Undirected Reachability Hierarchies

[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

directed

undirected

[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]



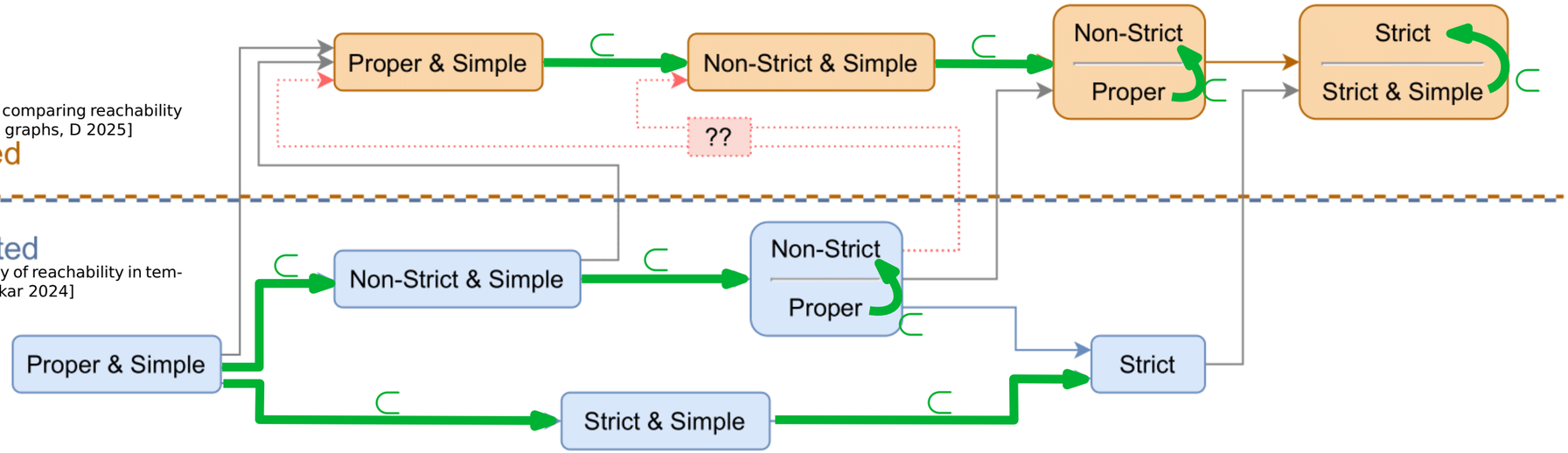
# Directed and Undirected Reachability Hierarchies

[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

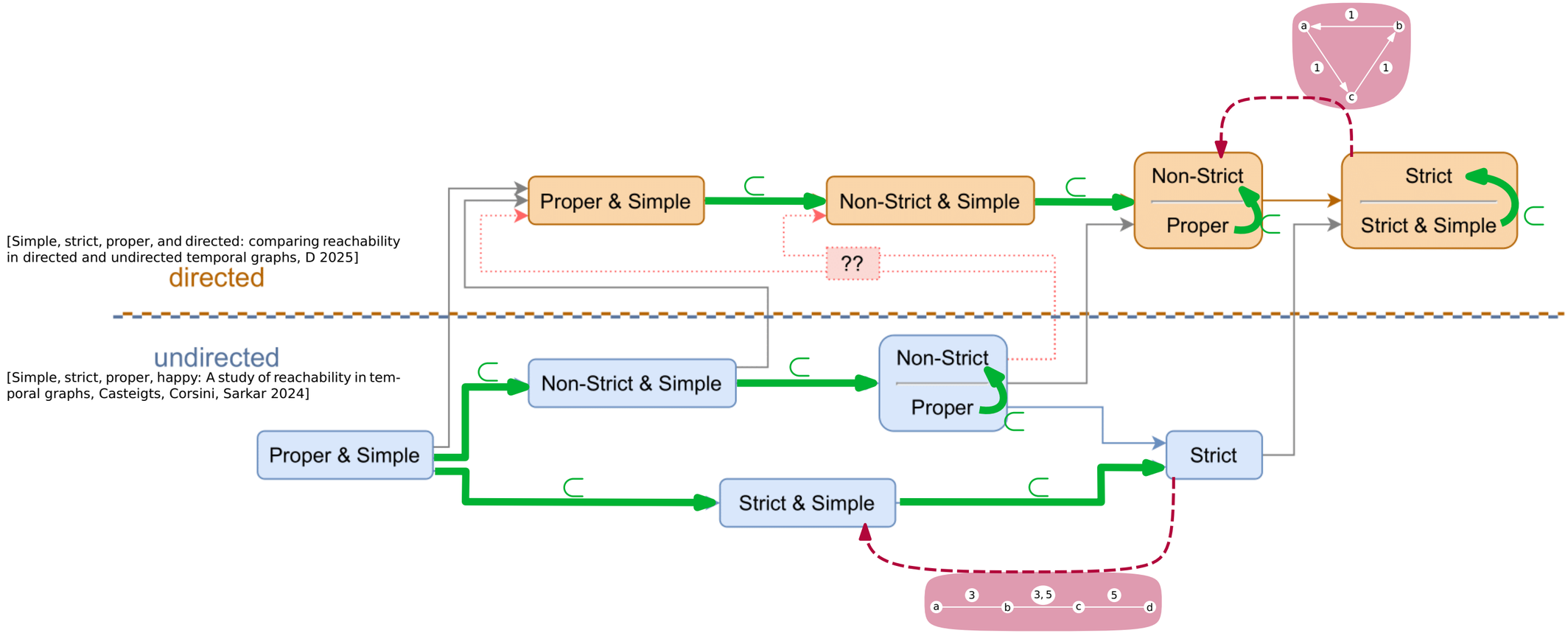
directed

undirected

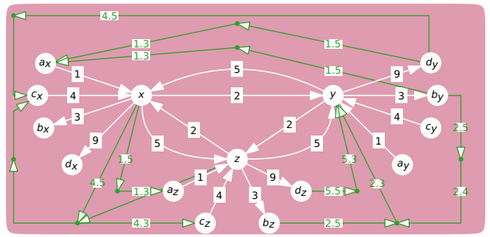
[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]

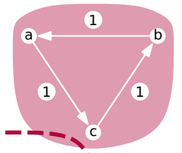


# Directed and Undirected Reachability Hierarchies



# Directed and Undirected Reachability Hierarchies





[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

directed

Proper & Simple

Non-Strict & Simple

Non-Strict

Proper

Strict

Strict & Simple

??

undirected

[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]

Non-Strict & Simple

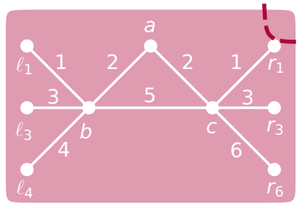
Non-Strict

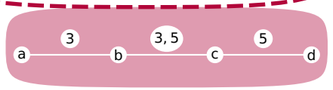
Proper

Proper & Simple

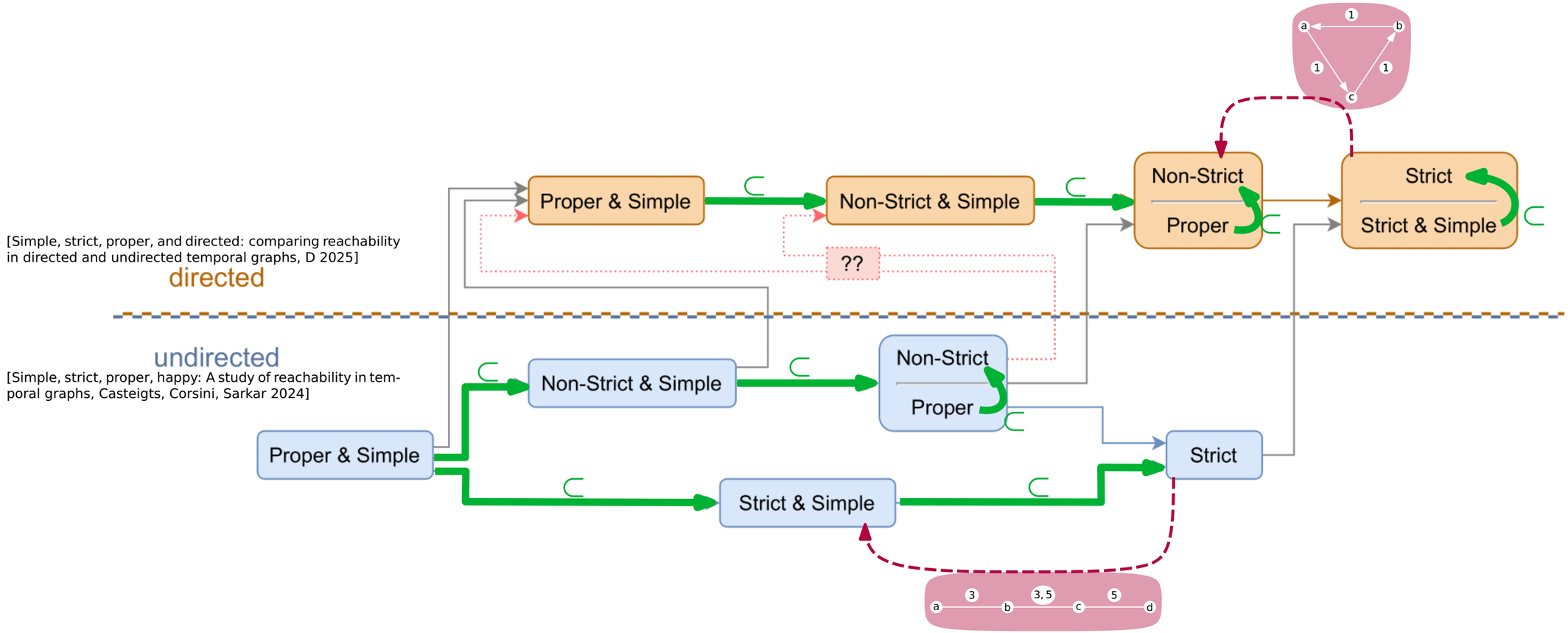
Strict

Strict & Simple





# Directed and Undirected Reachability Hierarchies



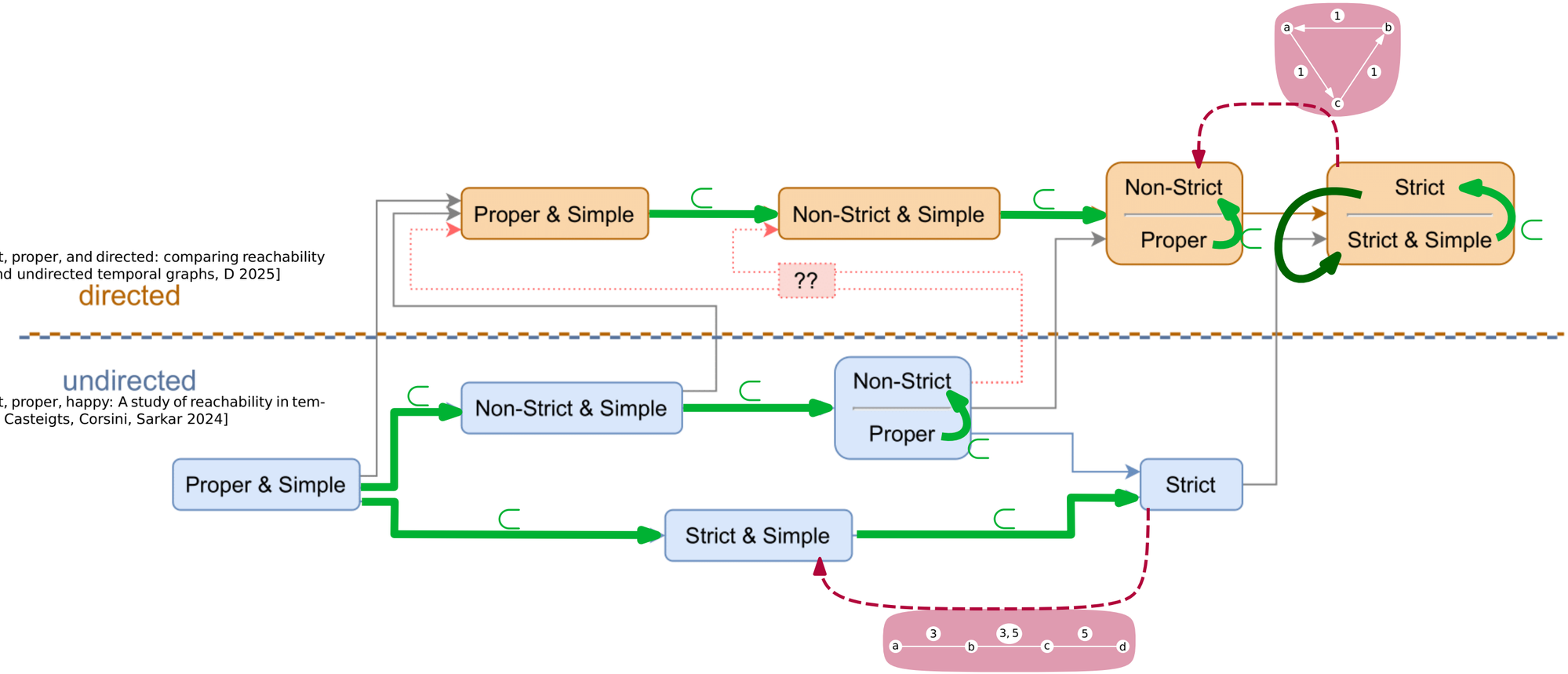
# Directed and Undirected Reachability Hierarchies

[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

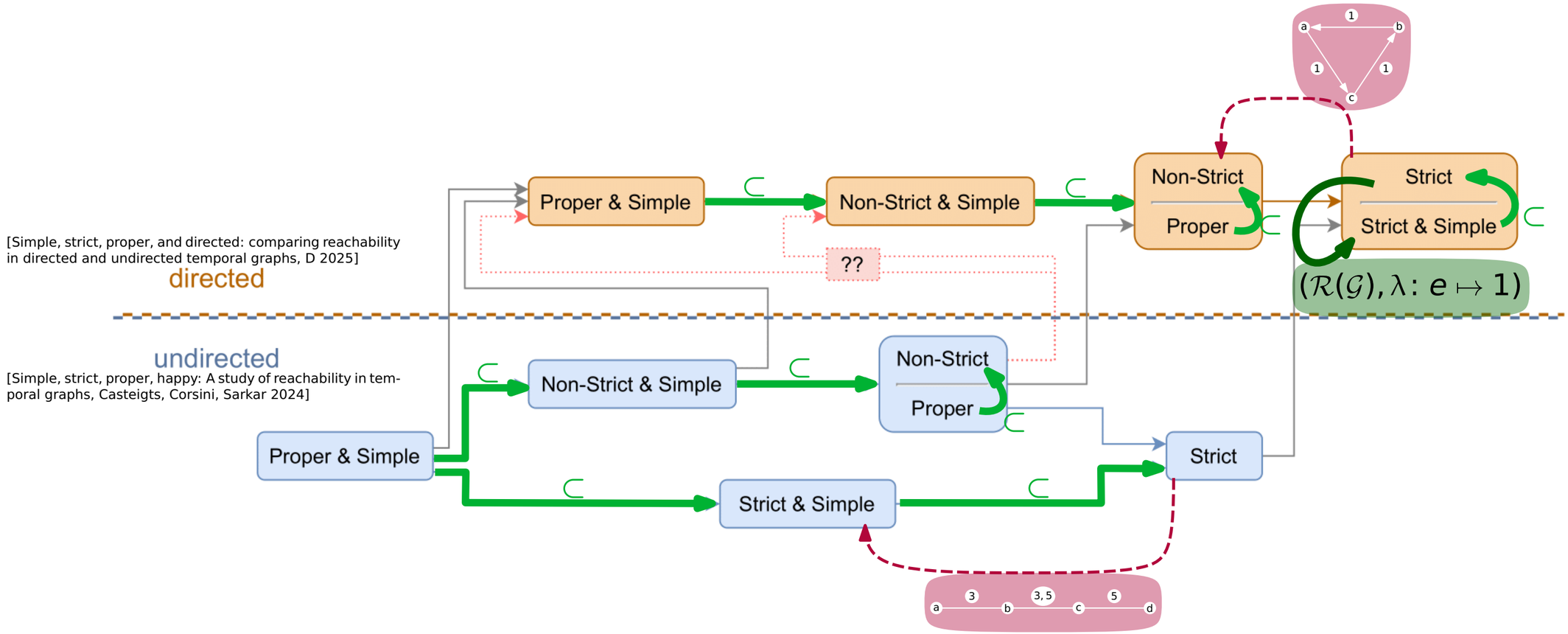
directed

undirected

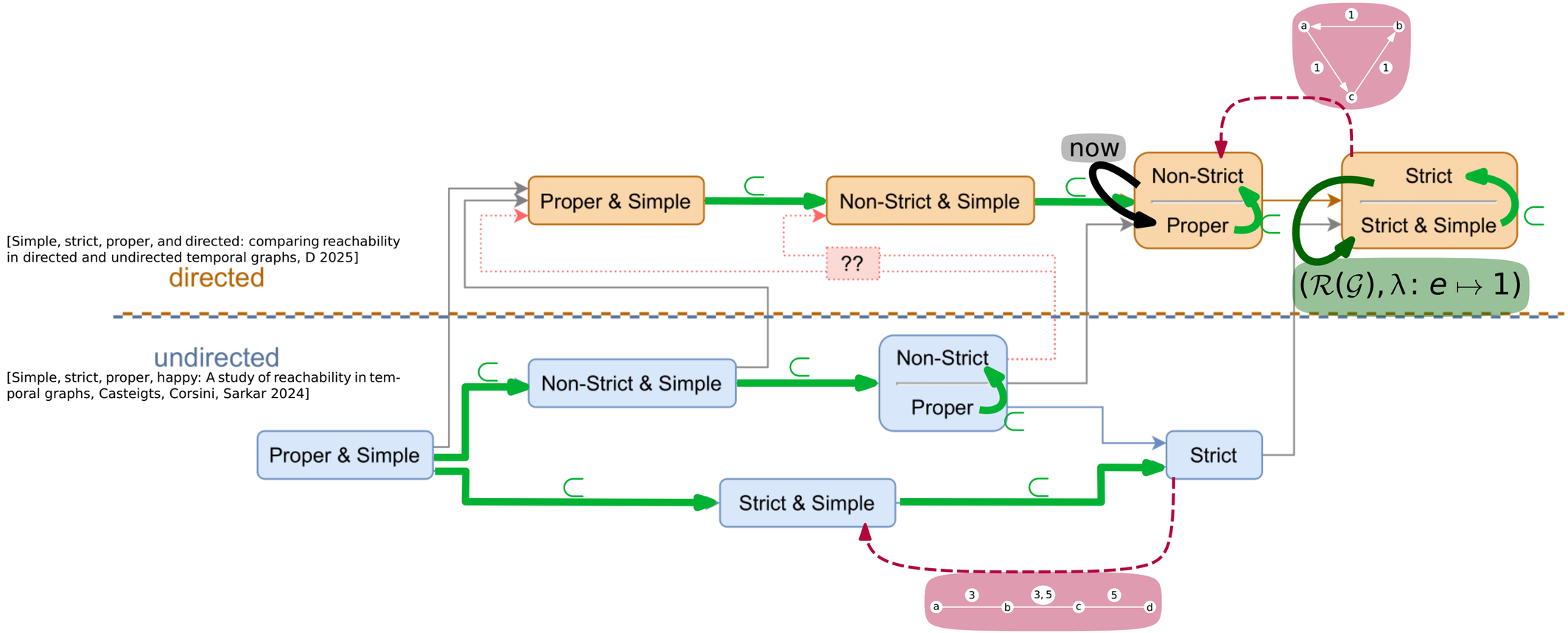
[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]



# Directed and Undirected Reachability Hierarchies



# Directed and Undirected Reachability Hierarchies



# Transforming Non-strict to Proper

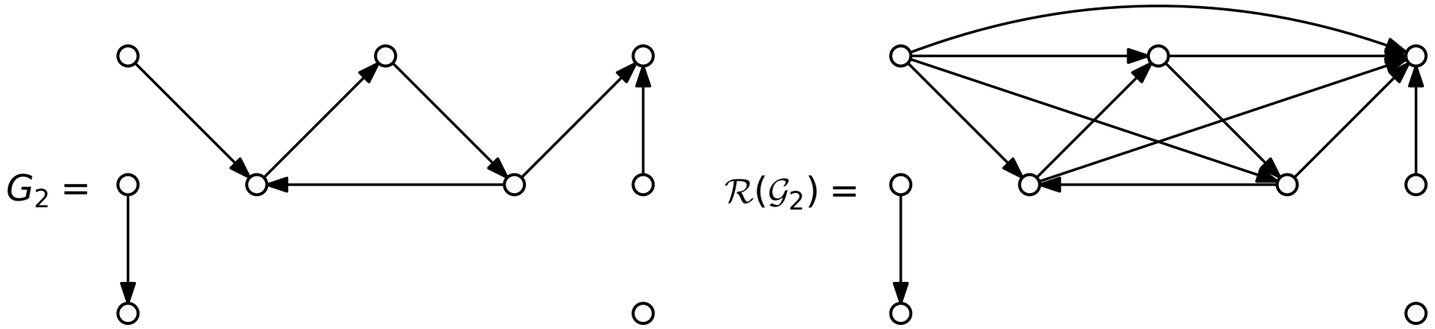
**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)

# Transforming Non-strict to Proper

**Goal**:

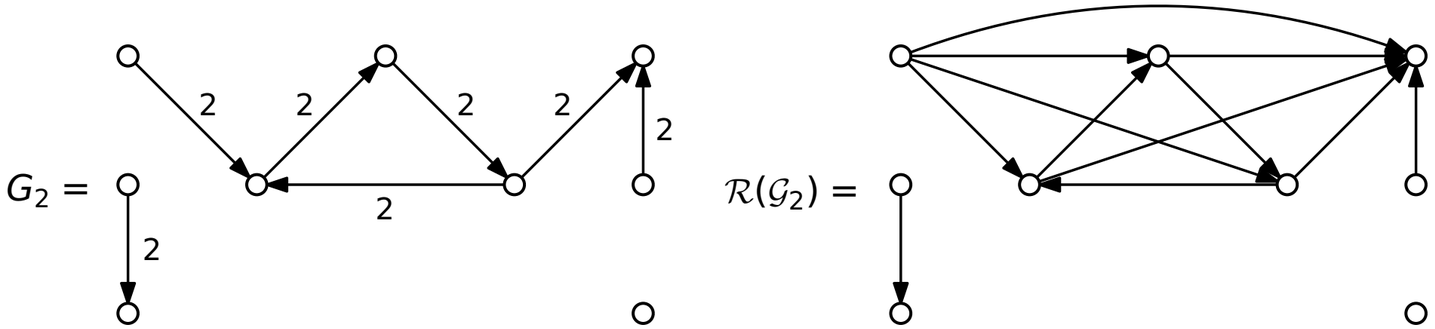
- replace edges with same time labels
- preserve reachabilities (temporal trips)



# Transforming Non-strict to Proper

**Goal**:

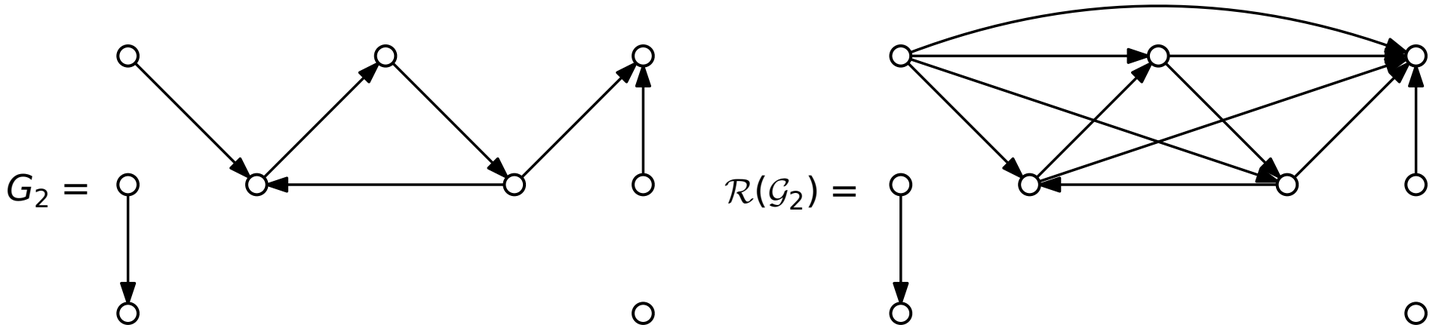
- replace edges with same time labels
- preserve reachabilities (temporal trips)



# Transforming Non-strict to Proper

**Goal**:

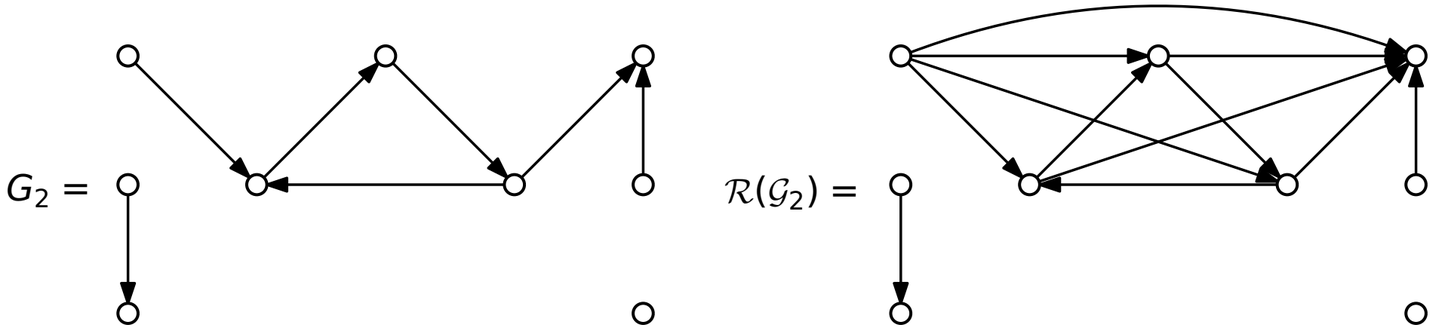
- replace edges with same time labels
- preserve reachabilities (temporal trips)



# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)

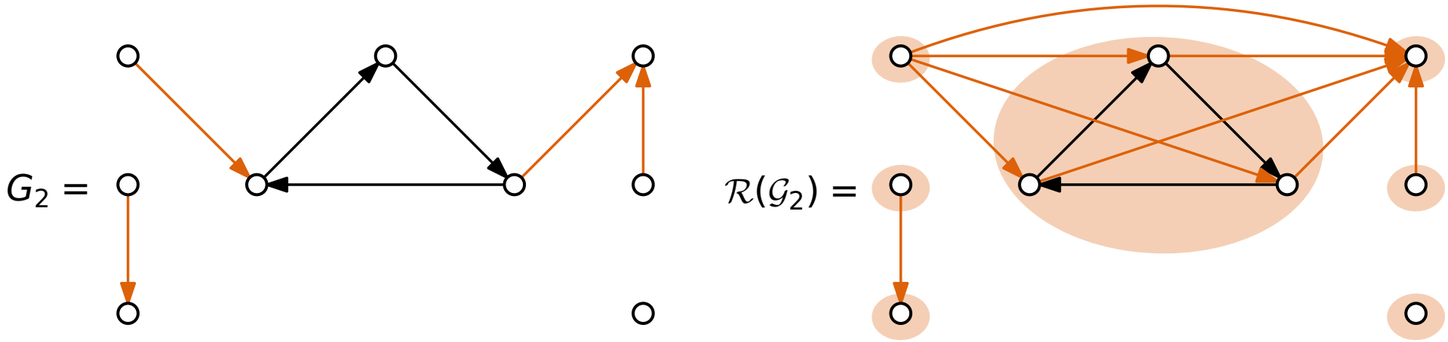



- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components

# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)

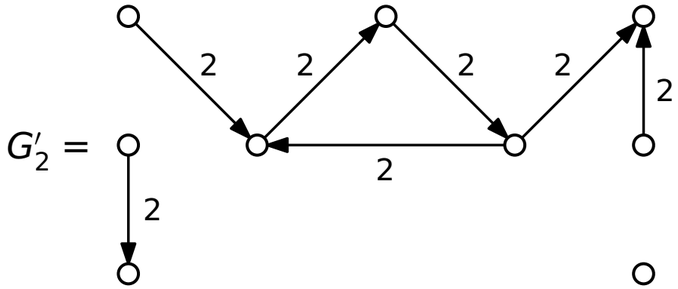


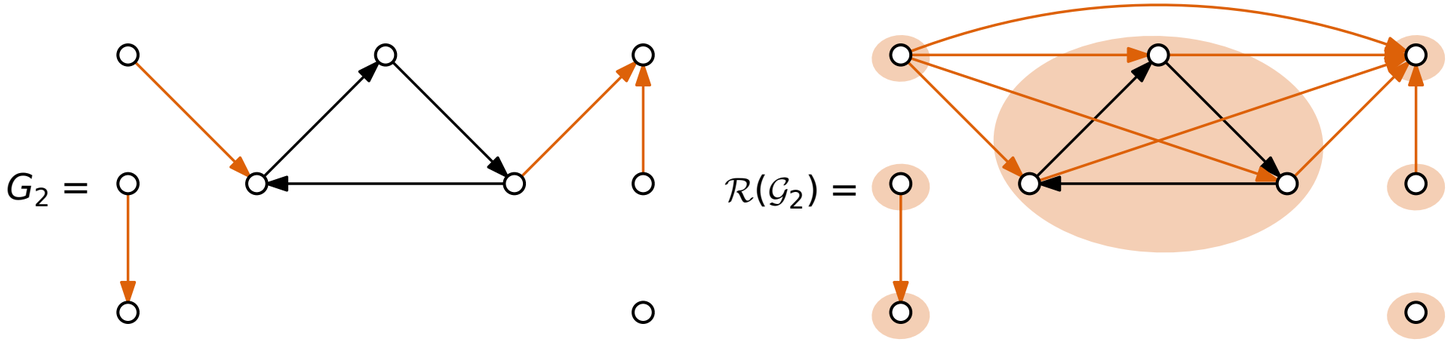

- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)




- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

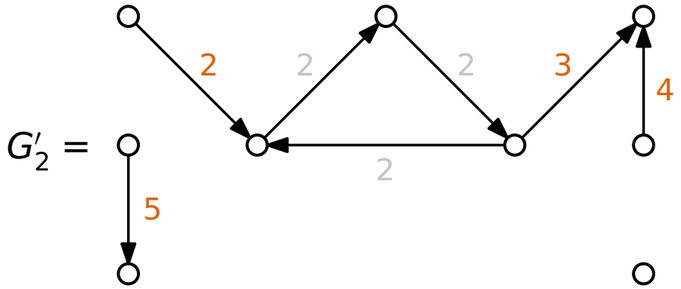
- now, for every snapshot:

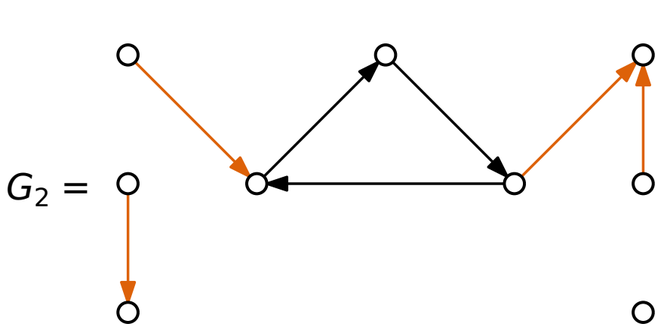
  1. replace the labels of the DAG-edges in the order of the DAG

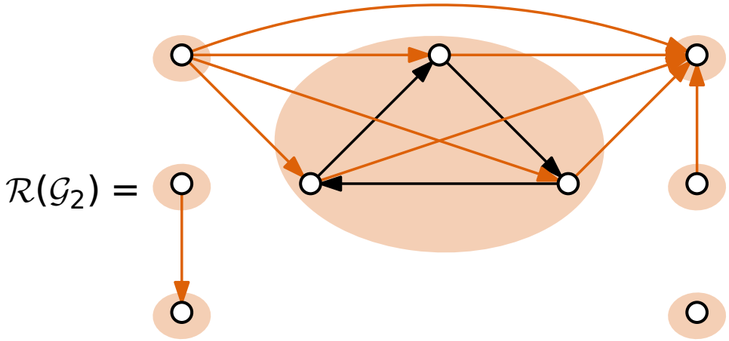
# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)




- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

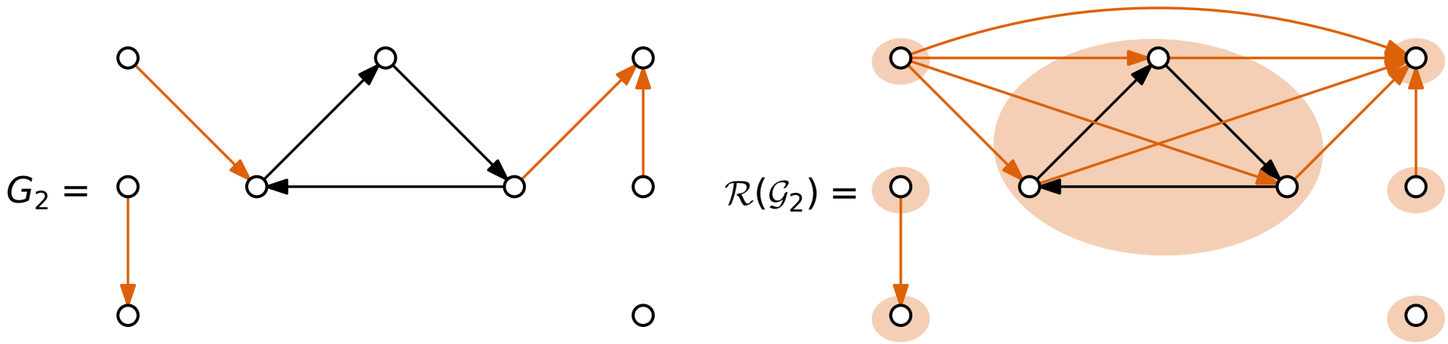
- now, for every snapshot:

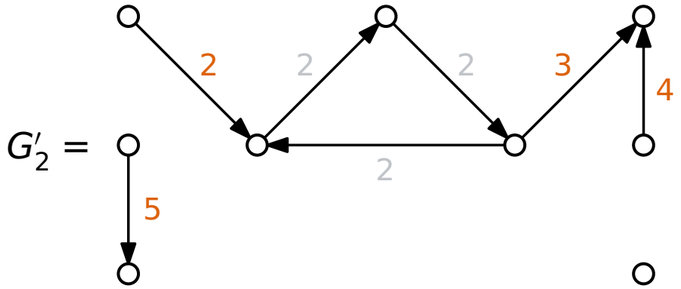
  1. replace the labels of the DAG-edges in the order of the DAG

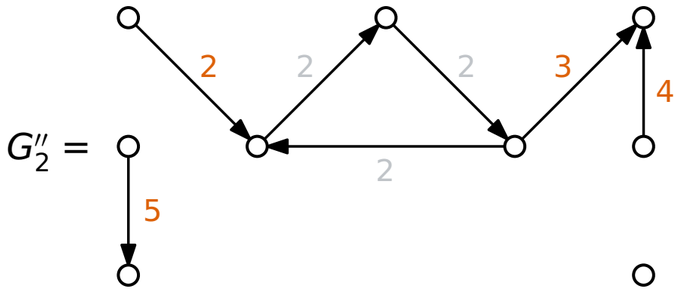
# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)



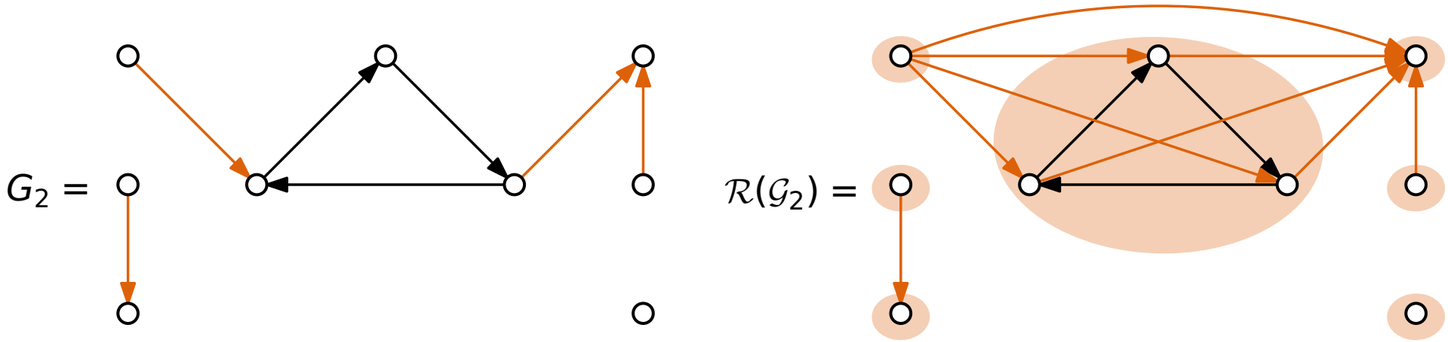

- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

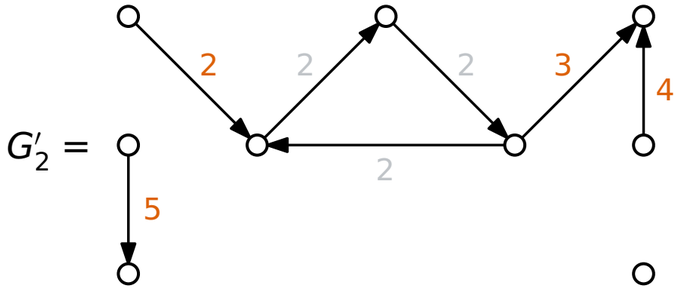
- now, for every snapshot:
  1. replace the labels of the DAG-edges in the order of the DAG
  2. replace each strongly connected component with a connected graph (e.g., simple, proper bidirected tree)

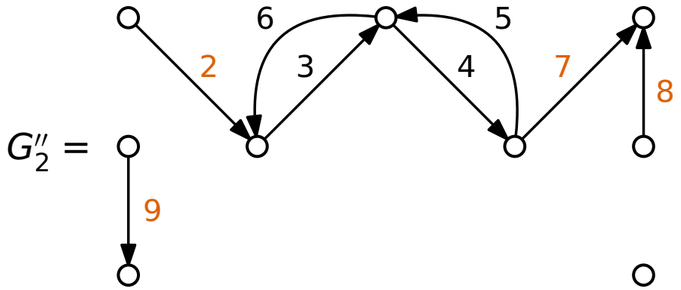
# Transforming Non-strict to Proper

**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)



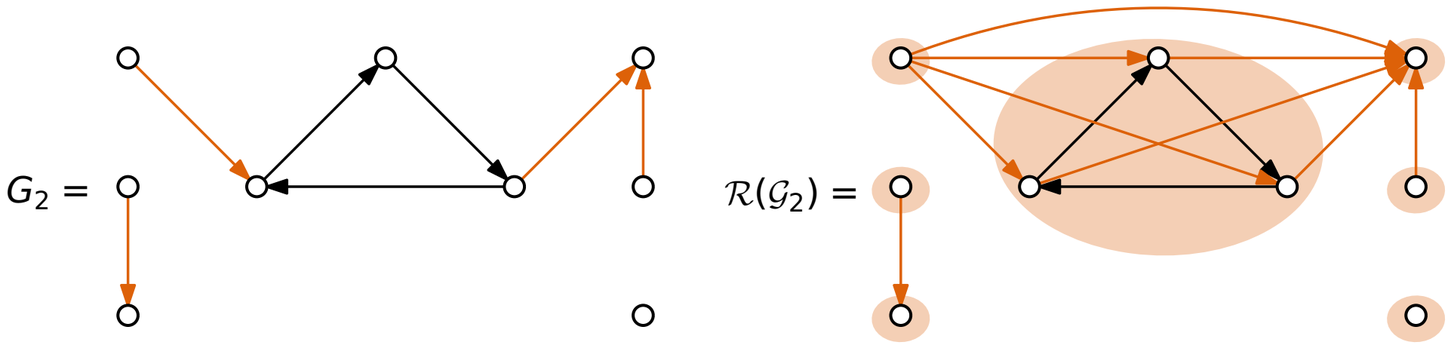

- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

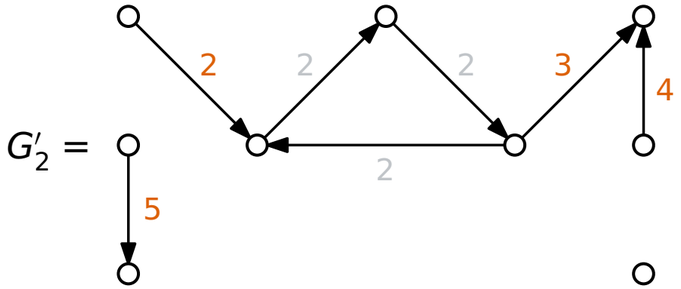
- now, for every snapshot:
  1. replace the labels of the DAG-edges in the order of the DAG
  2. replace each strongly connected component with a connected graph (e.g., simple, proper bidirected tree)

# Transforming Non-strict to Proper

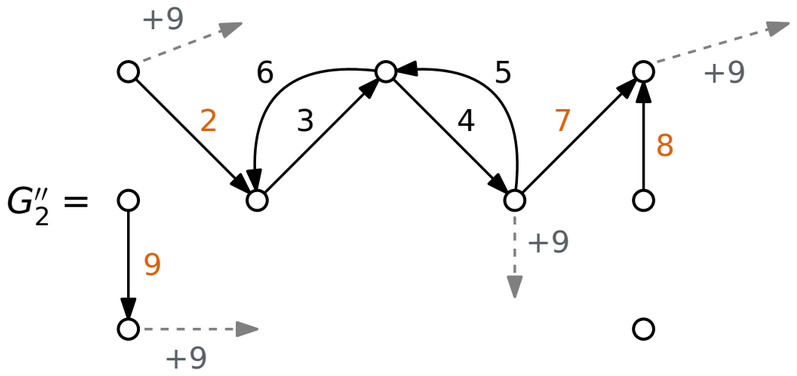
**Goal**:

- replace edges with same time labels
- preserve reachabilities (temporal trips)




- each snapshot $G_t$ of $\mathcal{G}$ consists of 1 to $n$ weakly connected components
- interpretation as DAG: each strongly connected subgraph is a vertex, directed, acyclic edges between

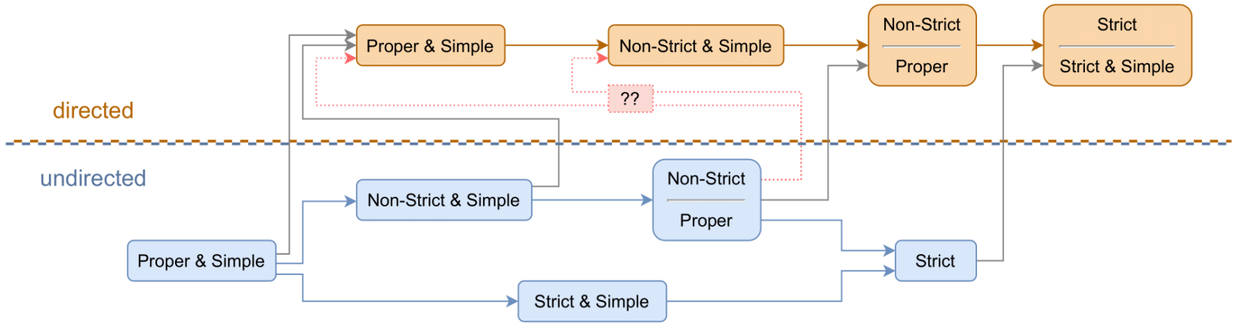



- now, for every snapshot:
  1. replace the labels of the DAG-edges in the order of the DAG
  2. replace each strongly connected component with a connected graph (e.g., simple, proper bidirected tree)
- shift labels of subsequent snapshots

# Summary, but Make it Triangular

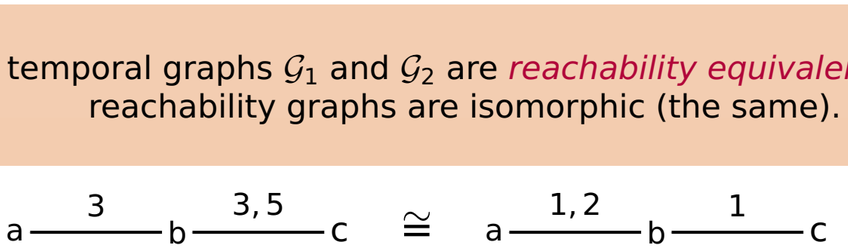
# Summary, but Make it Triangular

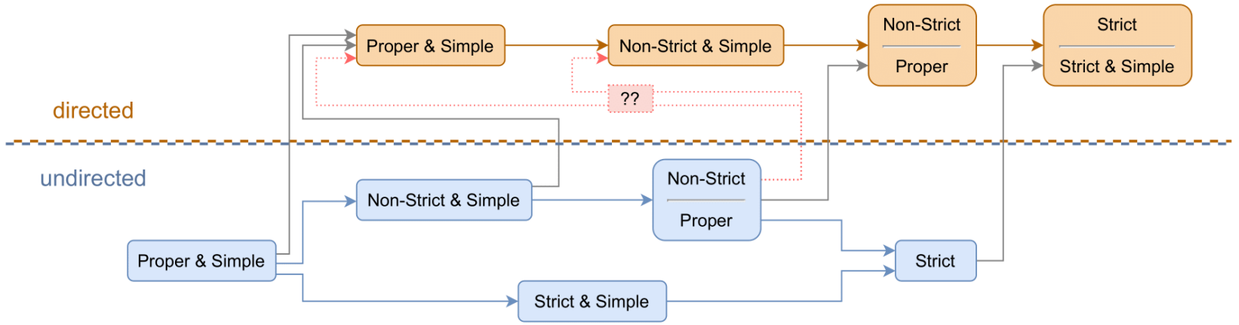
Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

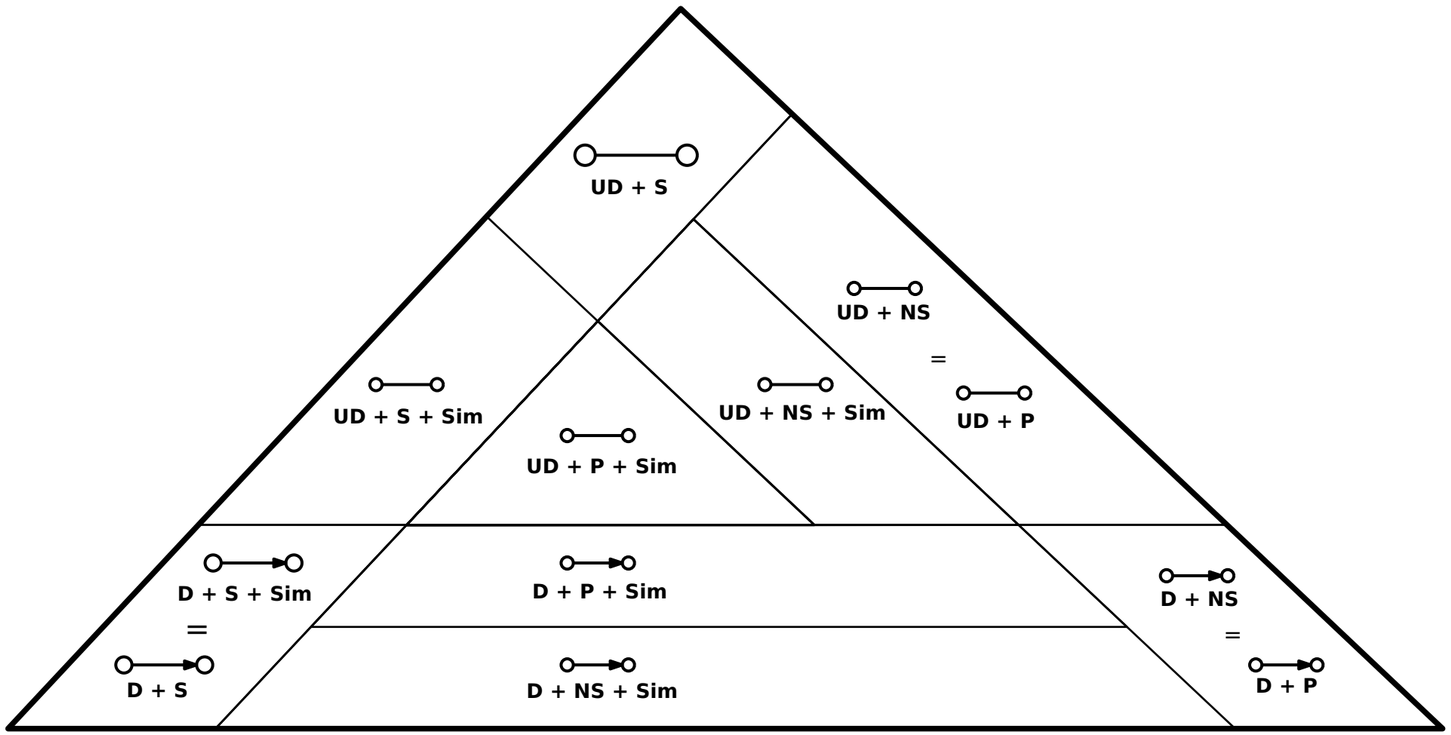



# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

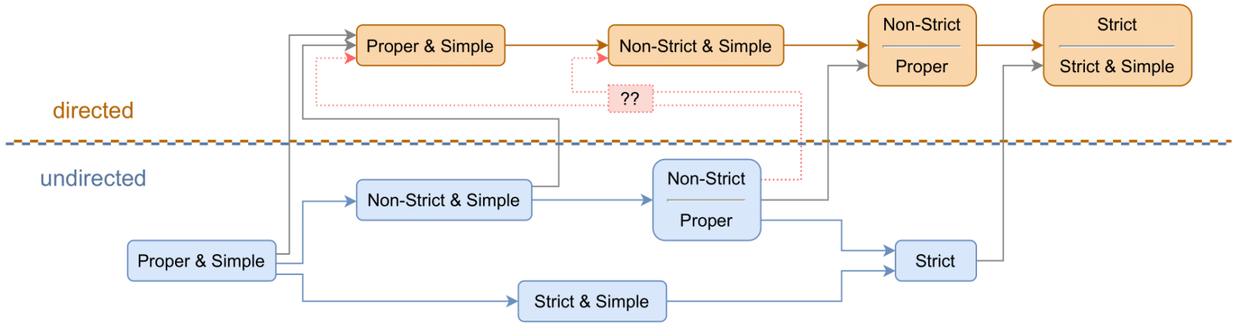


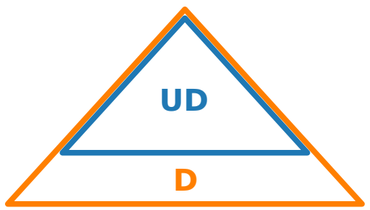



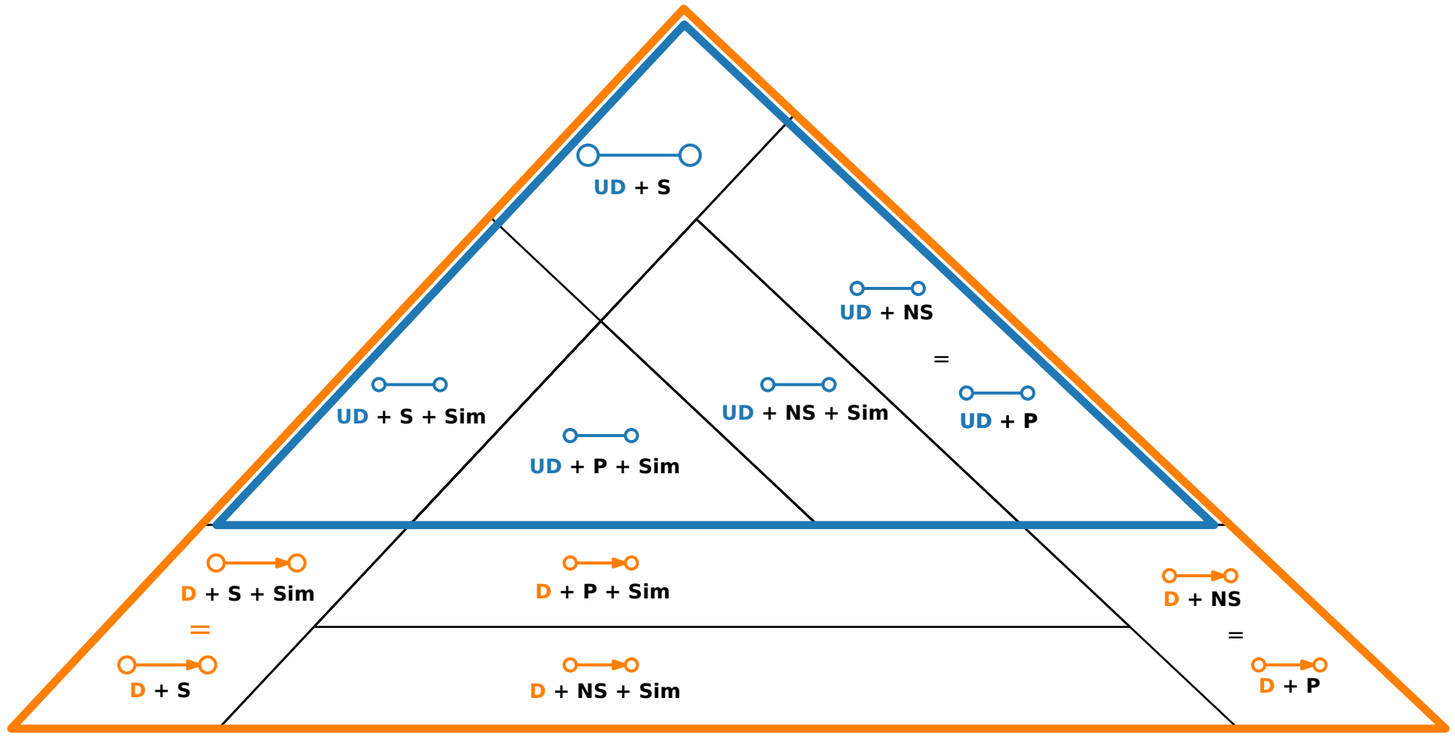



# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

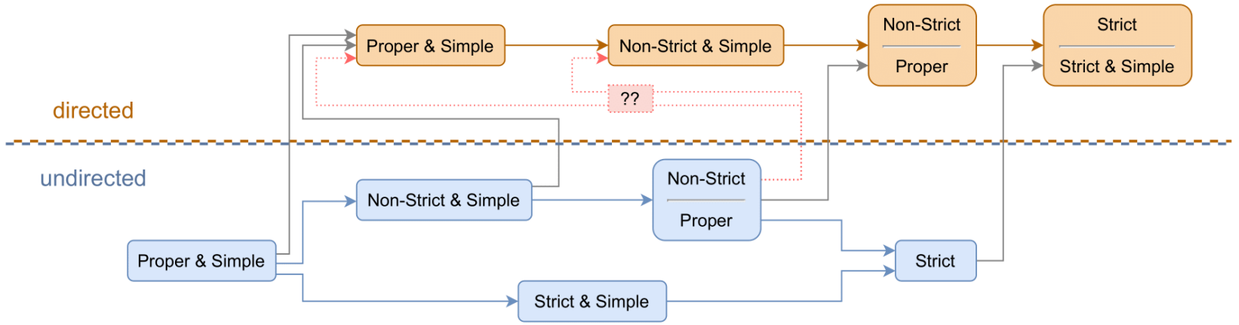



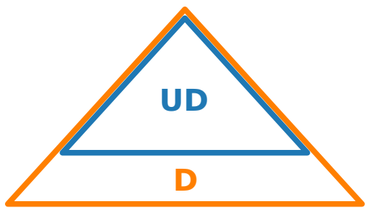



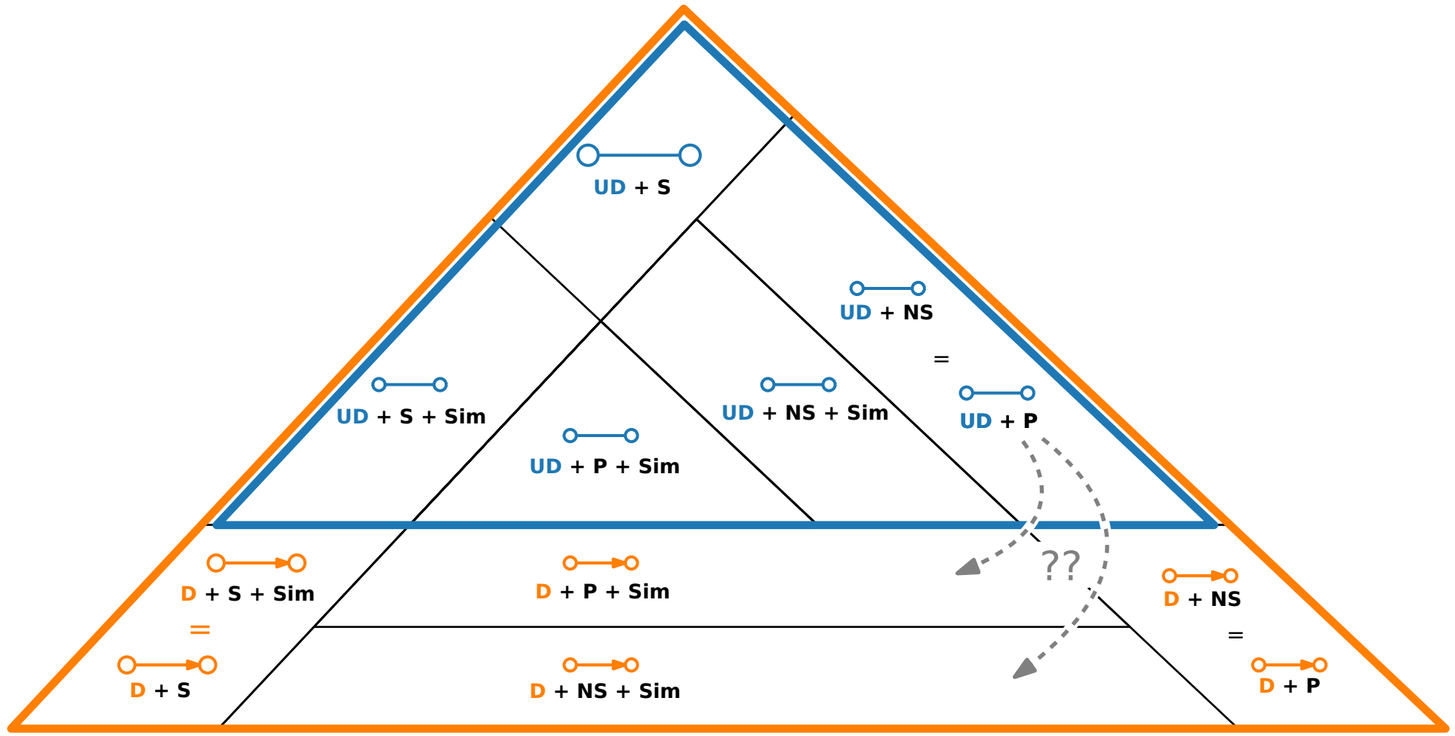



# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

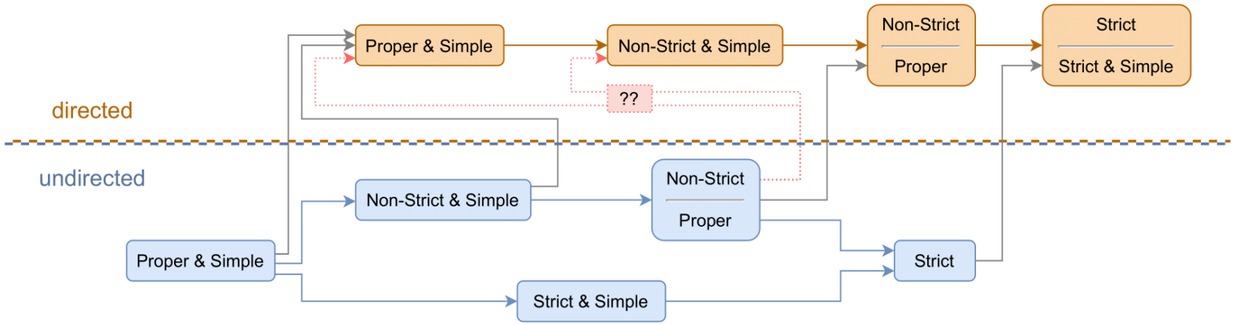



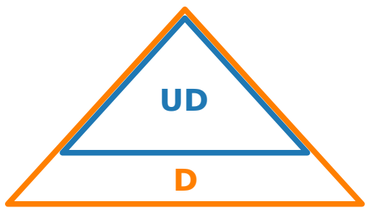



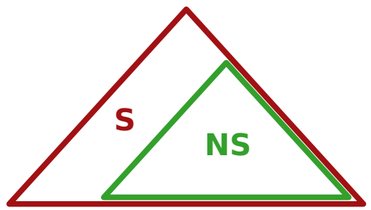



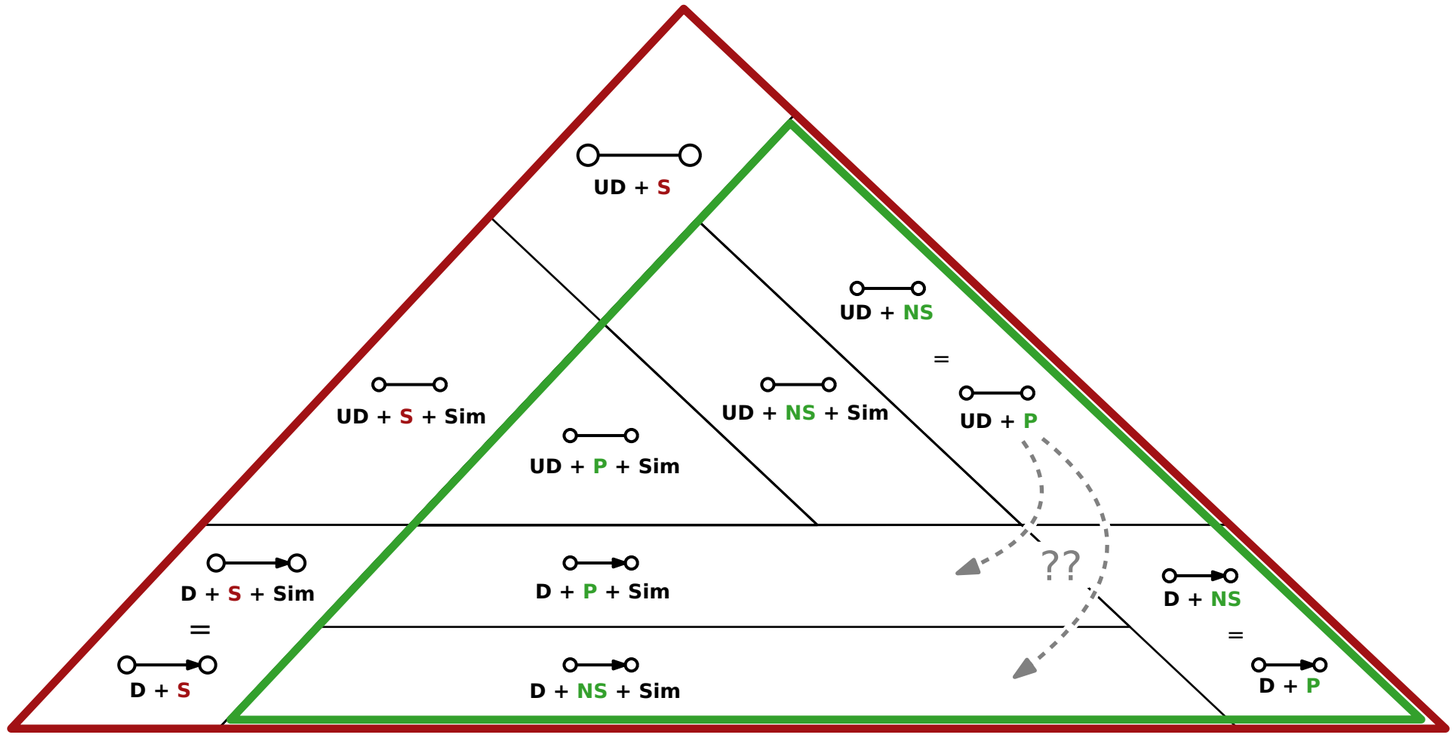
# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

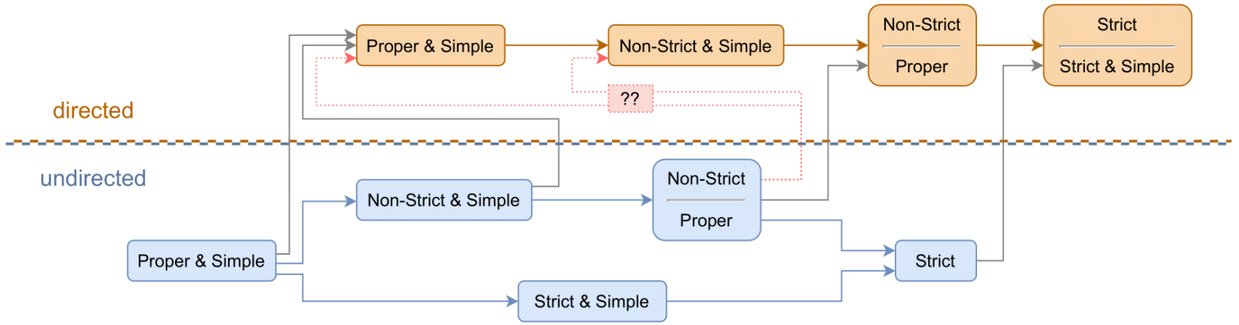



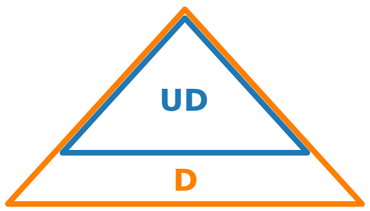



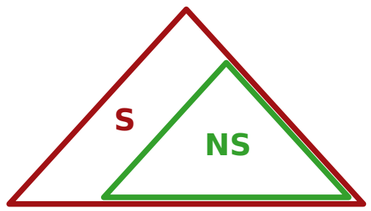



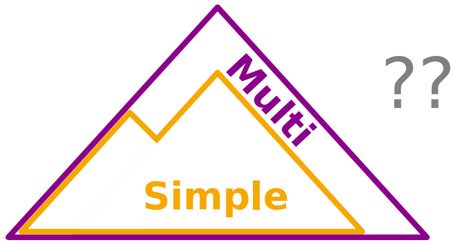
# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

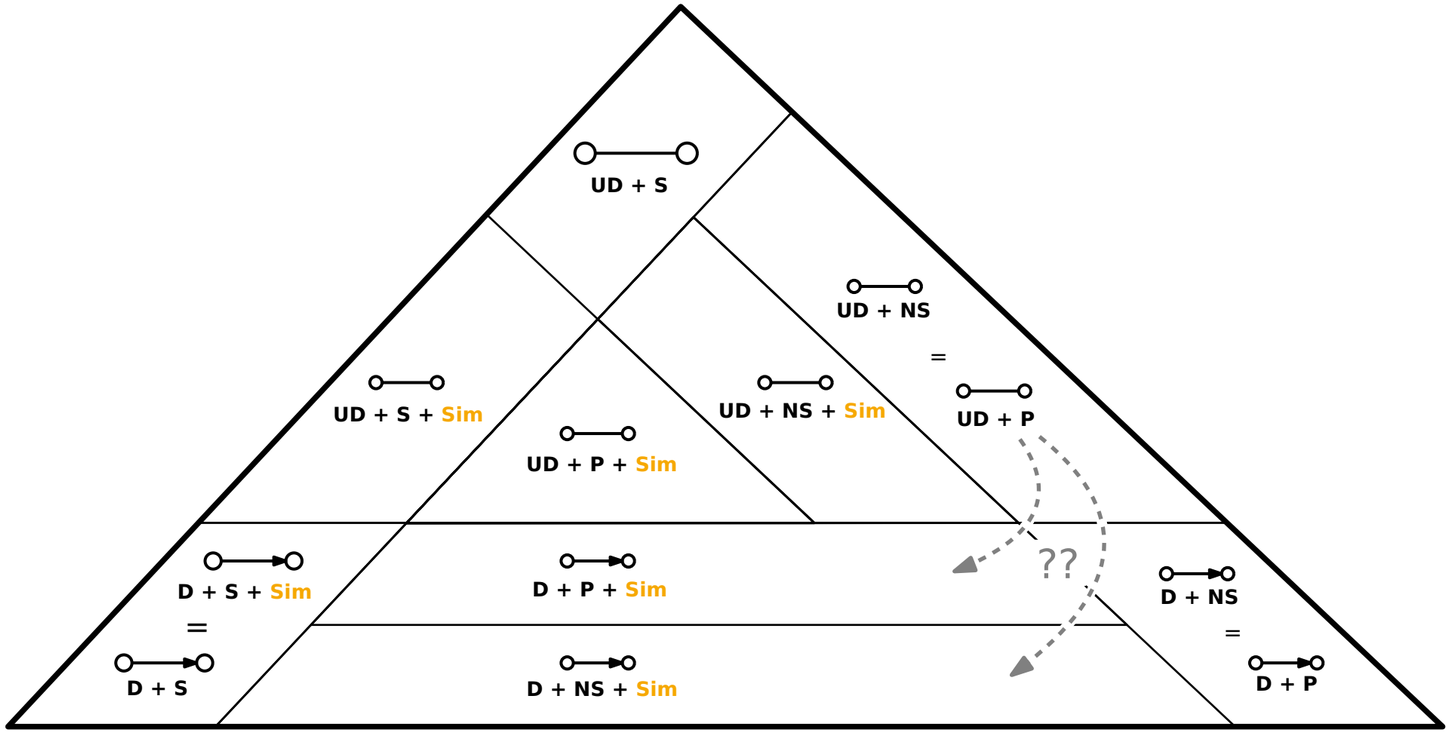



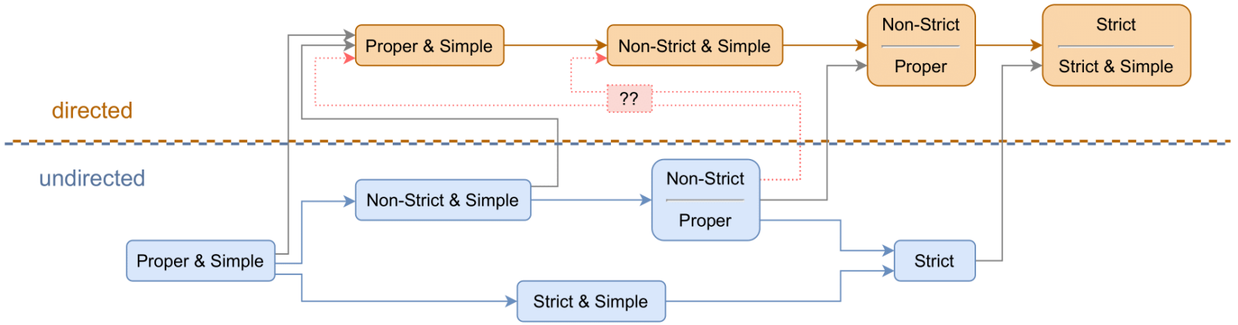



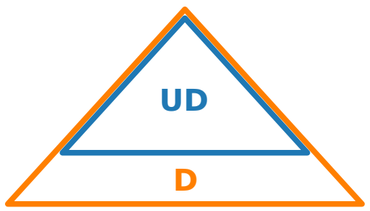

??

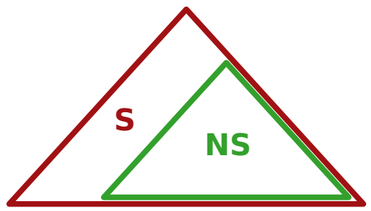


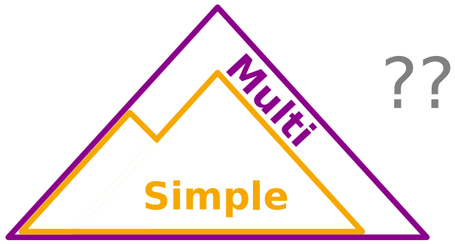
# Summary, but Make it Triangular

Two temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *reachability equivalent* if their reachability graphs are isomorphic (the same).

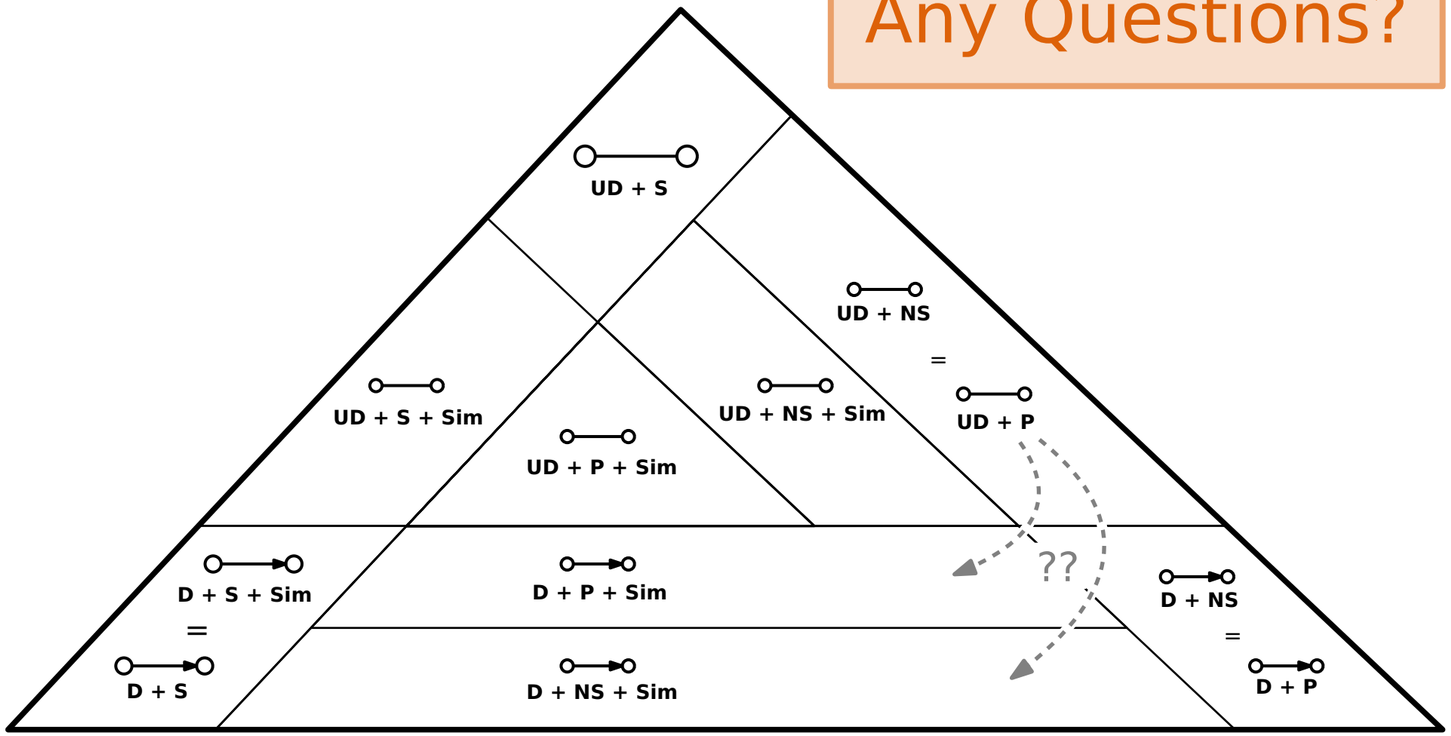




??

Any Questions?



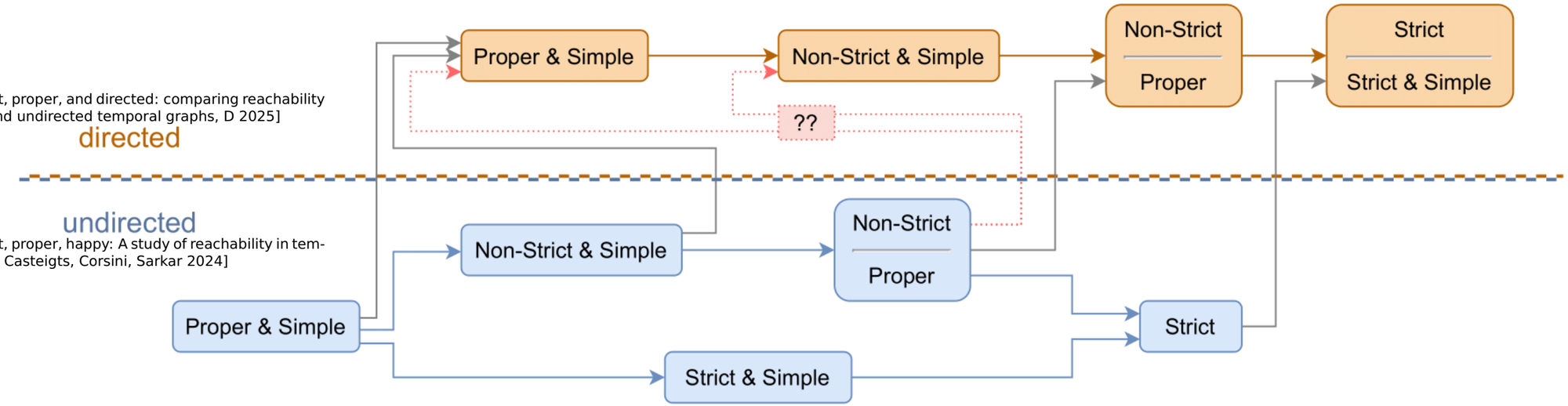
# Directed and Undirected Reachability Hierarchies

[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

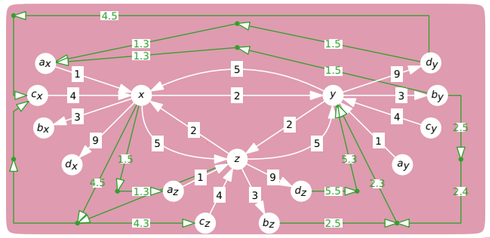
directed

undirected

[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]



# Directed and Undirected Reachability Hierarchies

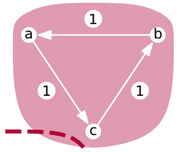


[Simple, strict, proper, and directed: comparing reachability in directed and undirected temporal graphs, D 2025]

directed

undirected

[Simple, strict, proper, happy: A study of reachability in temporal graphs, Casteigts, Corsini, Sarkar 2024]



Proper & Simple

Non-Strict & Simple

Non-Strict

Proper

Strict

Strict & Simple

??

$(\mathcal{R}(\mathcal{G}), \lambda : e \mapsto 1)$

Proper & Simple

Non-Strict & Simple

Non-Strict

Proper

Strict

Strict & Simple