Simple, strict, proper, happy: A study of reachability in temporal graphs

Timothée Corsini
joint work with A. Casteigts and W. Sarkar

LaBRI - University of Bordeaux

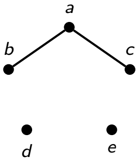
ICALP Workshop 2023

July 10th 2023

Motivation : temporal graphs

Temporal graph

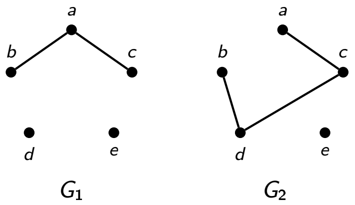
A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \to 2^{\tau}$. n vertices, m edges, τ the lifetime.



G_1

Temporal graph

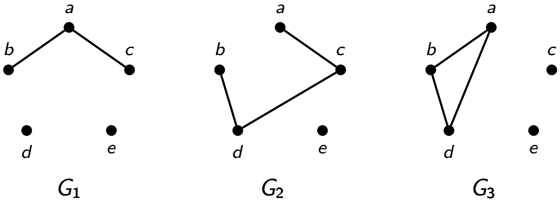
A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \to 2^{\tau}$. n vertices, m edges, τ the lifetime.



Motivation : temporal graphs

Temporal graph

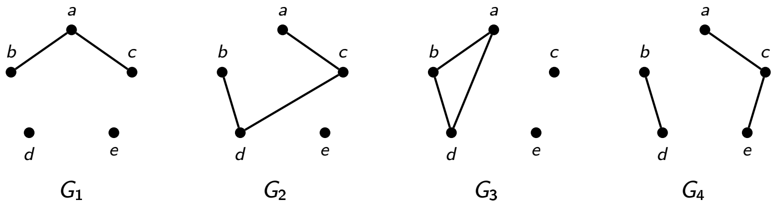
A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \to 2^{\tau}$. n vertices, m edges, τ the lifetime.



Motivation : temporal graphs

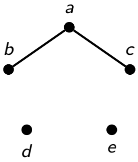
Temporal graph

A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \to 2^{\tau}$. n vertices, m edges, τ the lifetime.

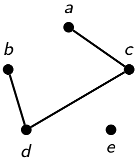


Temporal graph

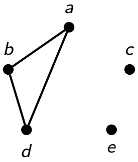
A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \rightarrow 2^{\tau}$. n vertices, m edges, τ the lifetime.



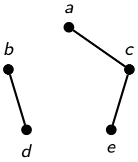
G_1



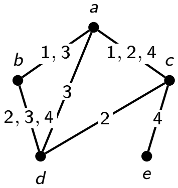
G_2



G_3



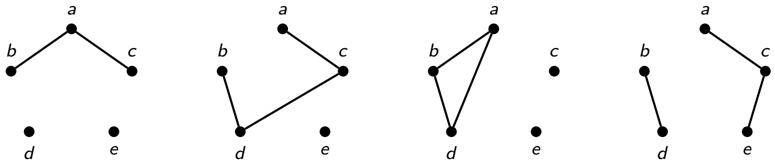
G_4



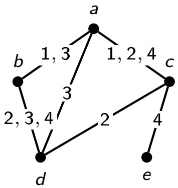
$\mathcal{G}$

Temporal graph

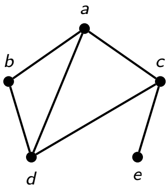
A temporal graph is a graph that changes with time.
$\mathcal{G} = (V, E, \lambda)$ with $\lambda : E \rightarrow 2^{\tau}$. n vertices, m edges, τ the lifetime.



G_1 G_2 G_3 G_4



$\mathcal{G}$



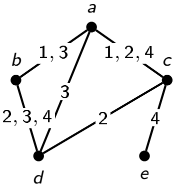
The footprint G

Journey

A journey (or temporal path) is a path that traverses the edges chronologically.

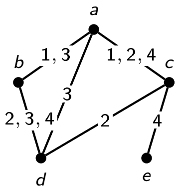
Journey

A journey (or temporal path) is a path that traverses the edges chronologically.



Journey

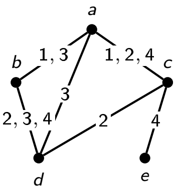
A journey (or temporal path) is a path that traverses the edges chronologically.



- journey from b to e but not from e to b

Journey

A journey (or temporal path) is a path that traverses the edges chronologically.



- journey from b to e but not from e to b
- $\mathcal{G}$ is not temporally connected

Temporal Reachability

u can reach v if there is a journey from u to v.

Temporal connectivity

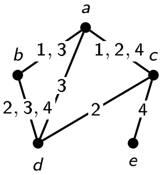
Every vertex can reach the others.

Important subtleties

Important subtleties

Strictness

Journeys use at most one edge at a timestamp.

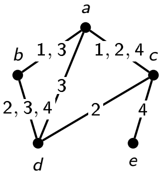


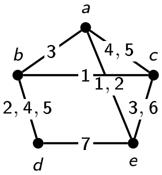
Strictness

Journeys use at most one edge at a timestamp.

Properness

No adjacent edges at the same time.





Strictness

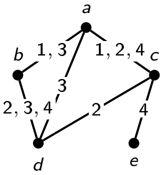
Journeys use at most one edge at a timestamp.

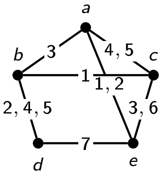
Properness

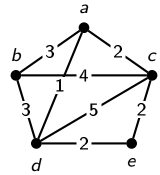
No adjacent edges at the same time.

Simpleness

Each edge has exactly one label.







Important subtleties

Strictness

Journeys use at most one edge at a timestamp.

Properness

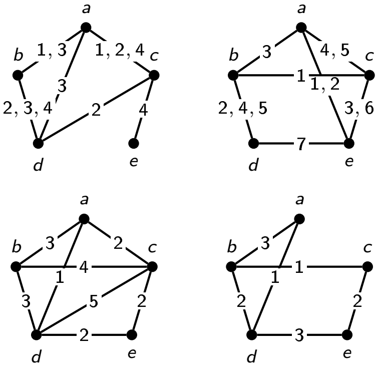
No adjacent edges at the same time.

Simpleness

Each edge has exactly one label.

"Happiness"

Both simple and proper.



Strictness

Journeys use at most one edge at a timestamp.

Properness

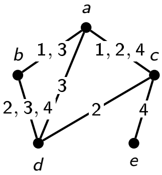
No adjacent edges at the same time.

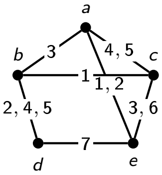
Simpleness

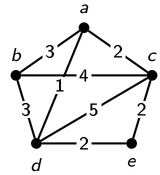
Each edge has exactly one label.

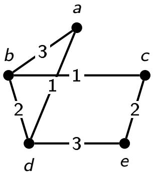
"Happiness"

Both simple and proper.









More Inportant than it seems !

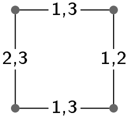
Illustration : The spanner problem

Temporal spanner

Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner

Temporal spanner

Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner



$\mathcal{G}_1$

Illustration : The spanner problem

Temporal spanner

Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner

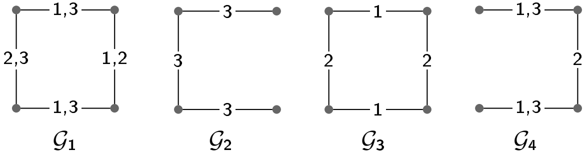
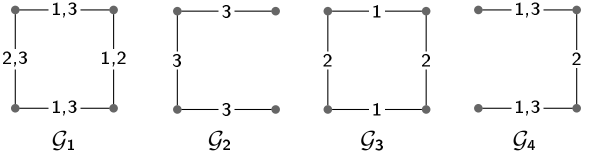


Illustration : The spanner problem

Temporal spanner

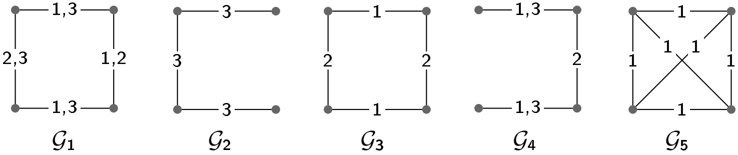
Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner



- $\mathcal{G}_2$ is a non-strict min-labelled spanner
- $\mathcal{G}_3$ is a strict min-labelled spanner
- $\mathcal{G}_4$ is a strict min-edge spanner

Temporal spanner

Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner

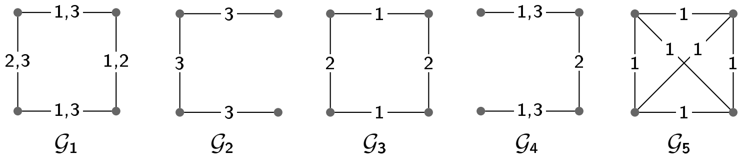


- $\mathcal{G}_2$ is a non-strict min-labelled spanner
- $\mathcal{G}_3$ is a strict min-labelled spanner
- $\mathcal{G}_4$ is a strict min-edge spanner

Illustration : The spanner problem

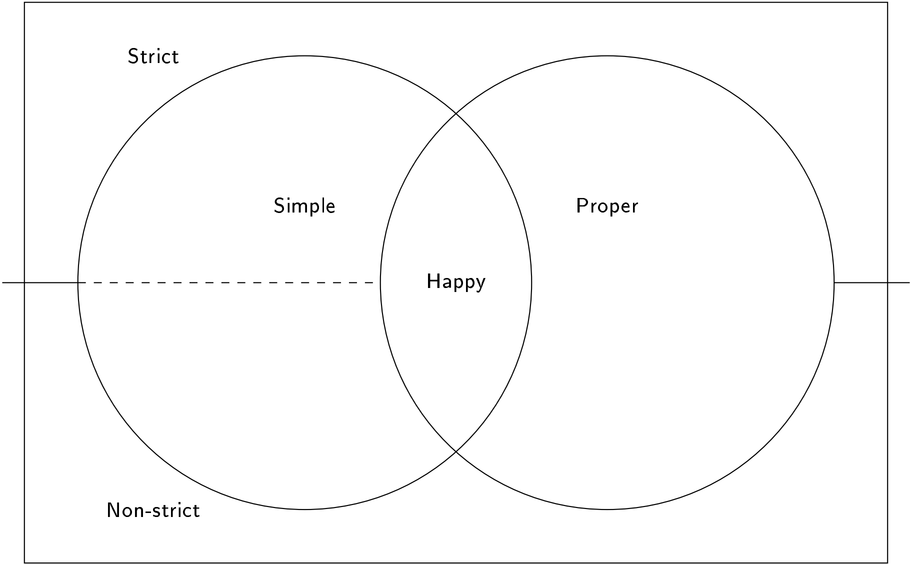
Temporal spanner

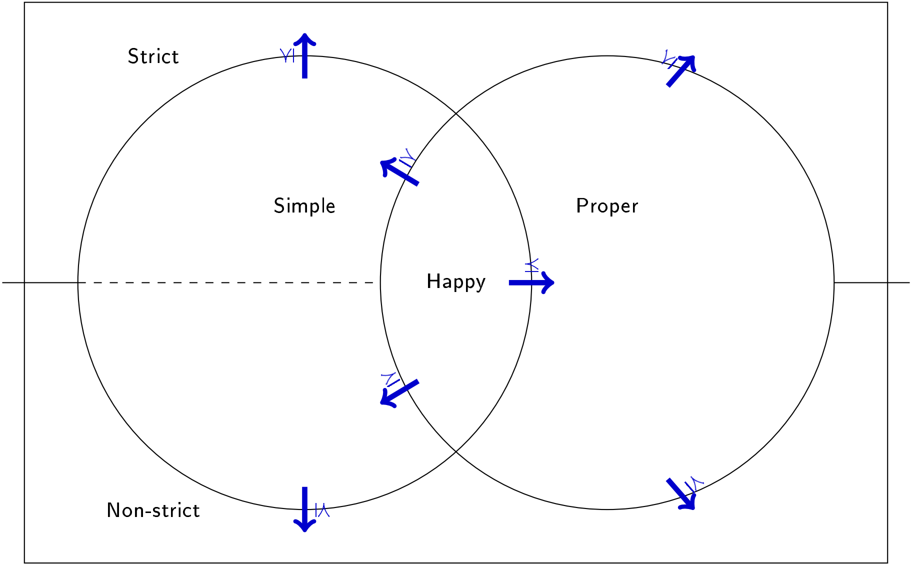
Input : a graph $\mathcal{G}$ that is temporally connected
Output : a graph $\mathcal{G}' \subseteq \mathcal{G}$ that preserves temporal connectivity
Cost measure : size of the spanner



- $\mathcal{G}_2$ is a non-strict min-labelled spanner
- $\mathcal{G}_3$ is a strict min-labelled spanner
- $\mathcal{G}_4$ is a strict min-edge spanner
- $\mathcal{G}_5$ is a strict min spanner of itself

Outline




Strict
Simple
Proper
Happy
Non-strict

How to compare expressitivy ?

What concept captures expressitivy ?

How to compare expressitivy ?

What concept captures expressitivy ?

Reachability graph

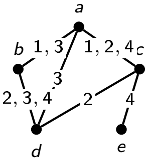
A directed graph $H = (V, E')$ s.t. $(u, v) \in E'$ iff u can reach v in the original graph.

How to compare expressitivy ?

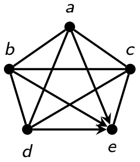
What concept captures expressitivy ?

Reachability graph

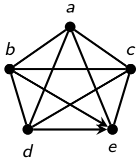
A directed graph $H = (V, E')$ s.t. $(u, v) \in E'$ iff u can reach v in the original graph.



$\mathcal{G}$



strict reachability



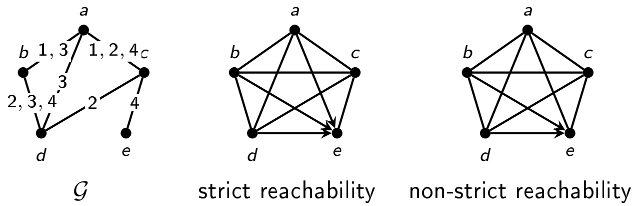
non-strict reachability

How to compare expressitivy ?

What concept captures expressitivy ?

Reachability graph

A directed graph $H = (V, E')$ s.t. $(u, v) \in E'$ iff u can reach v in the original graph.

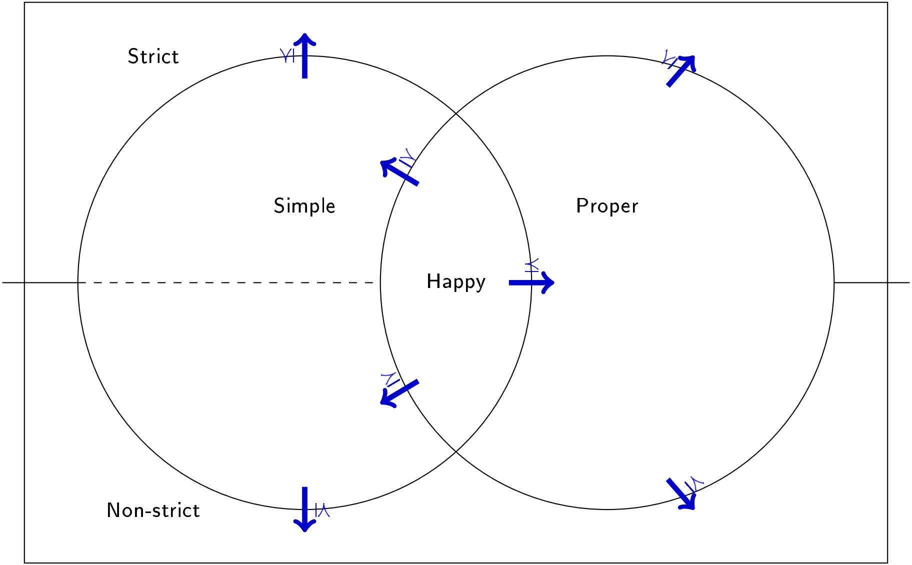


Reachability graph expressitivy

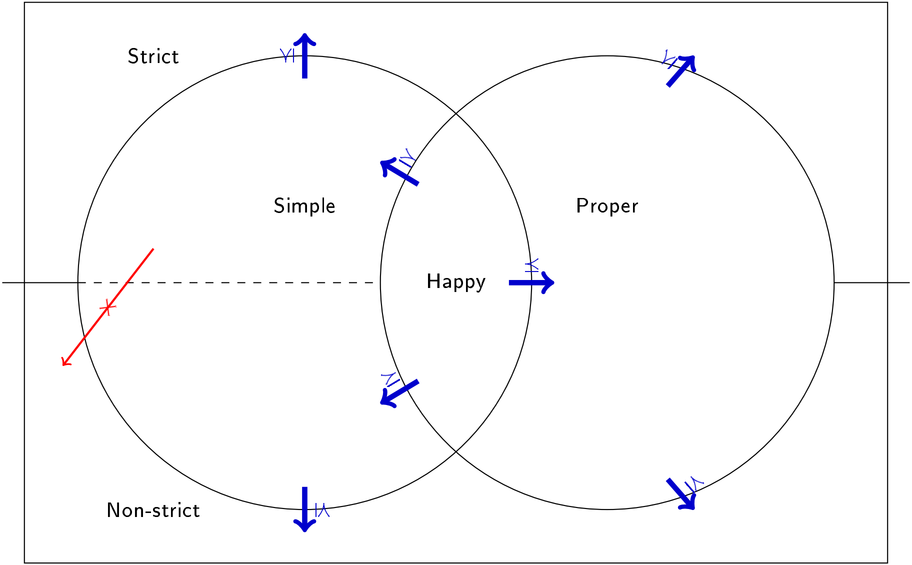
Can a given reachability graph be obtained from a given setting ?

Separating the settings

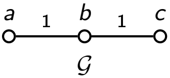
Outline

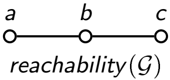


Outline

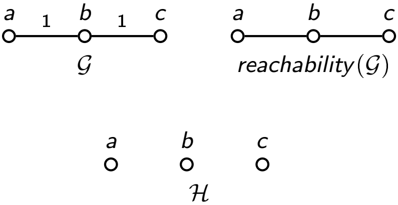


From "strict and simple" to "non-strict"

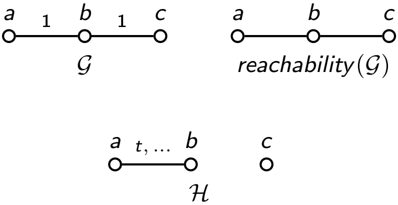




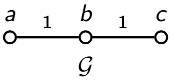
From "strict and simple" to "non-strict"

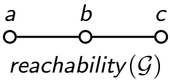


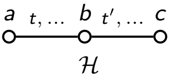
From "strict and simple" to "non-strict"



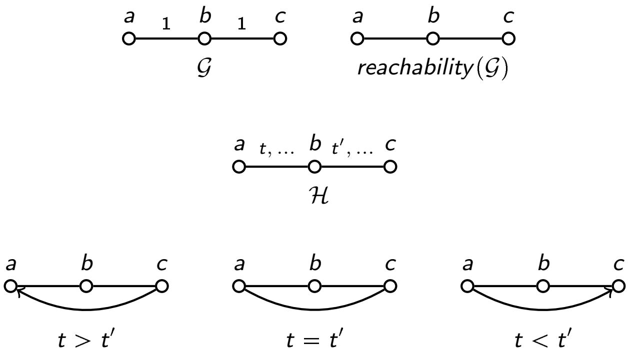
From "strict and simple" to "non-strict"



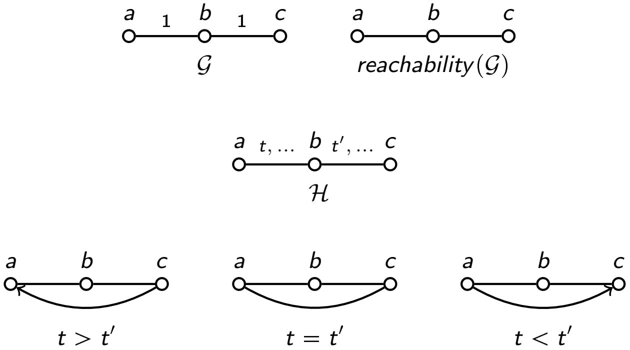




From "strict and simple" to "non-strict"

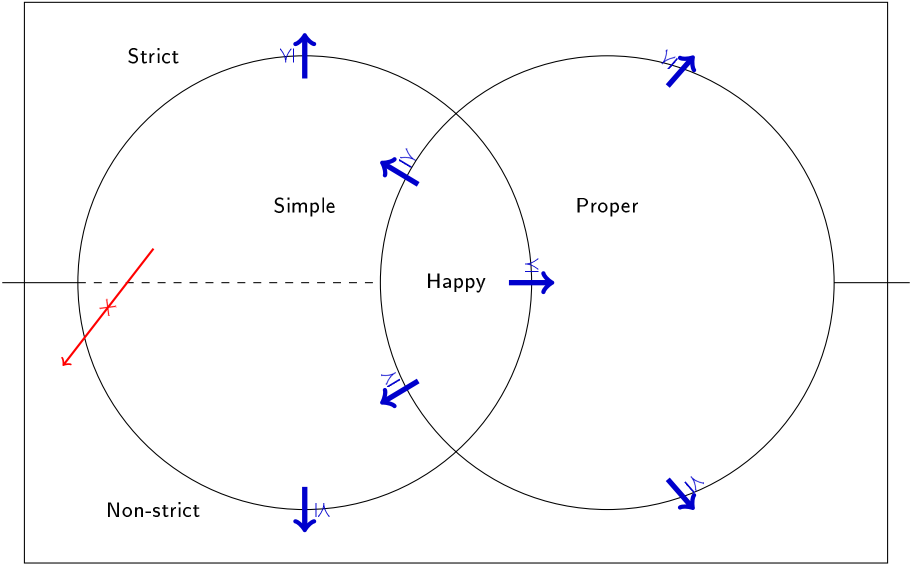


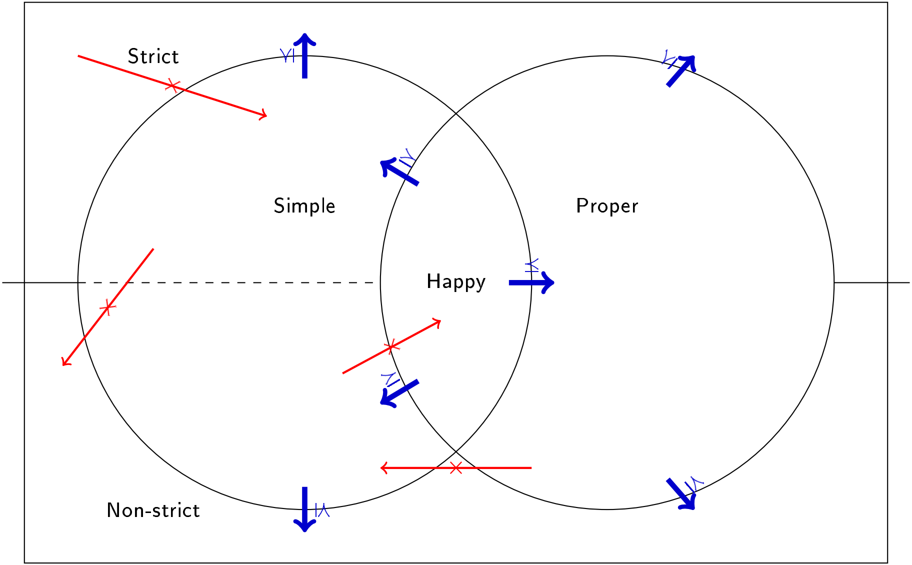
From "strict and simple" to "non-strict"



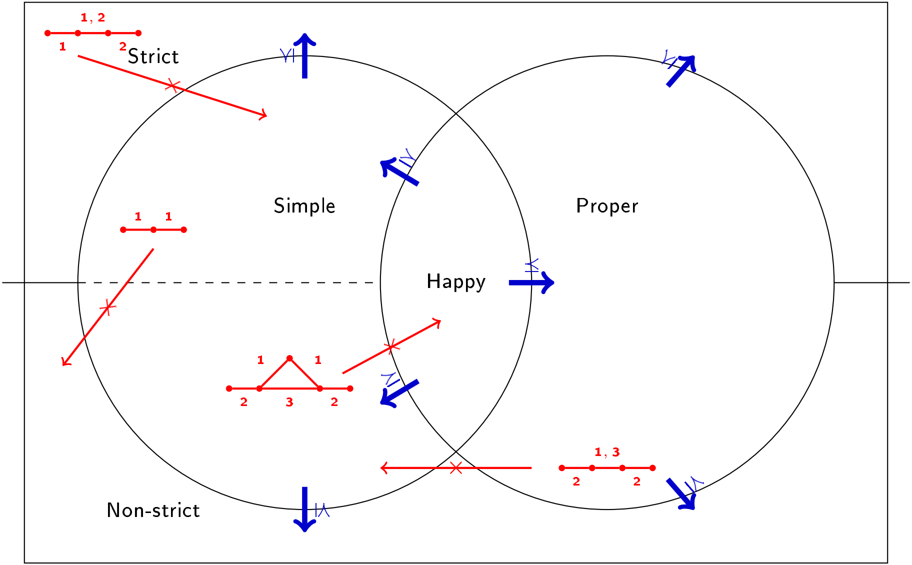
$reachability(\mathcal{H}) \neq reachability(\mathcal{G})$

The "simple and strict" setting cannot be realized in the "non-strict" setting.


Strict
Simple
Proper
Happy
Non-strict

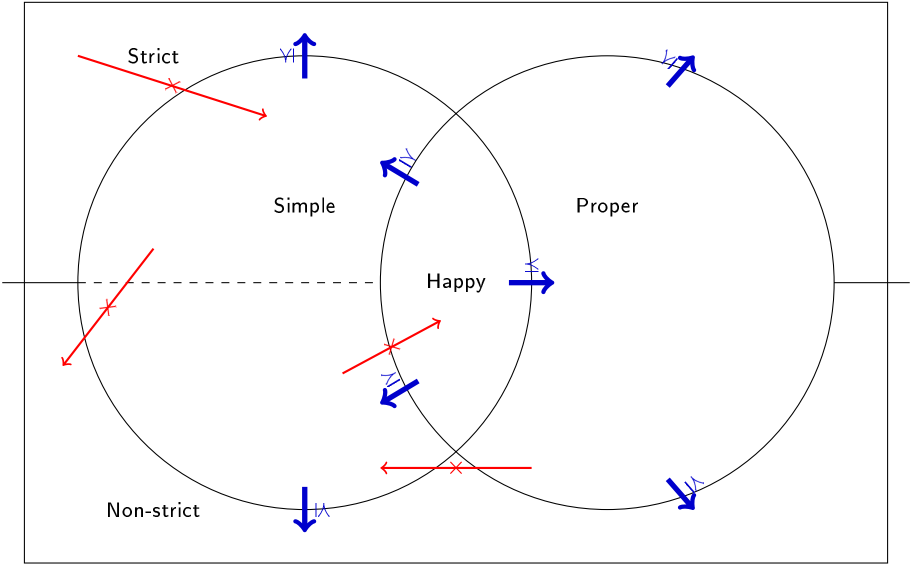

Strict
Simple
Proper
Happy
Non-strict

Outline

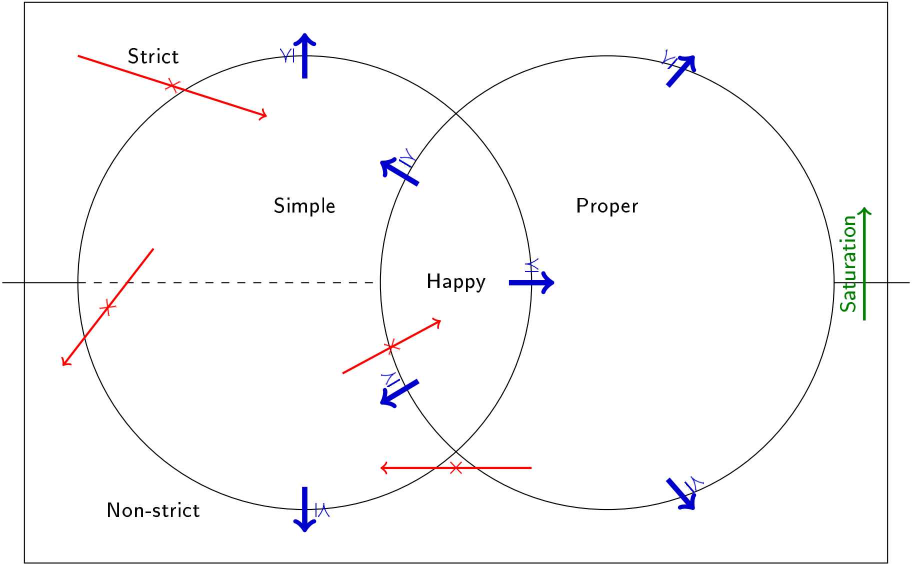


Transformations between settings

Outline



Outline



Saturation method : from "non-strict" to "strict"

Goal : Turn a graph from "non-strict" to "strict" with the same reachability.

Saturation method

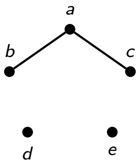
$\mathcal{G} \to \mathcal{H}$ such that there is a contact $(\{u, v\}, t)$ in $\mathcal{H}$ if and only if $\{u, v\}$ are connected at time t in $\mathcal{G}$.

Saturation method : from "non-strict" to "strict"

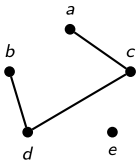
Goal : Turn a graph from "non-strict" to "strict" with the same reachability.

Saturation method

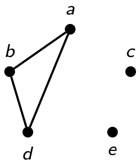
$\mathcal{G} \rightarrow \mathcal{H}$ such that there is a contact $(\{u, v\}, t)$ in $\mathcal{H}$ if and only if $\{u, v\}$ are connected at time t in $\mathcal{G}$.



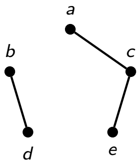
G_1



G_2



G_3



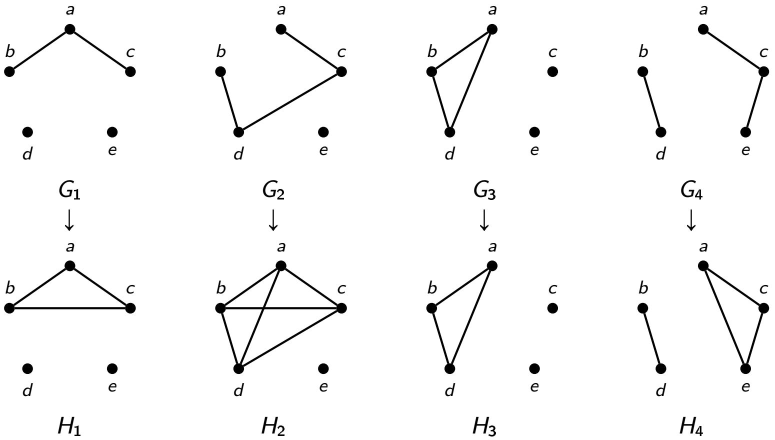
G_4

Saturation method : from "non-strict" to "strict"

Goal : Turn a graph from "non-strict" to "strict" with the same reachability.

Saturation method

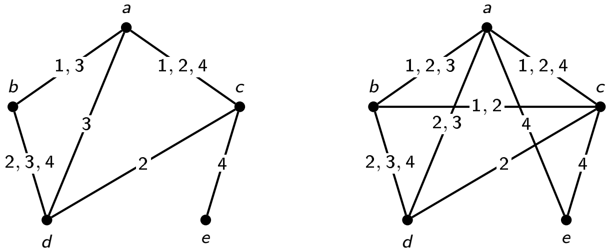
$\mathcal{G} \rightarrow \mathcal{H}$ such that there is a contact $(\{u, v\}, t)$ in $\mathcal{H}$ if and only if $\{u, v\}$ are connected at time t in $\mathcal{G}$.



Goal : Turn a graph from "non-strict" to "strict" with the same reachability.

Saturation method

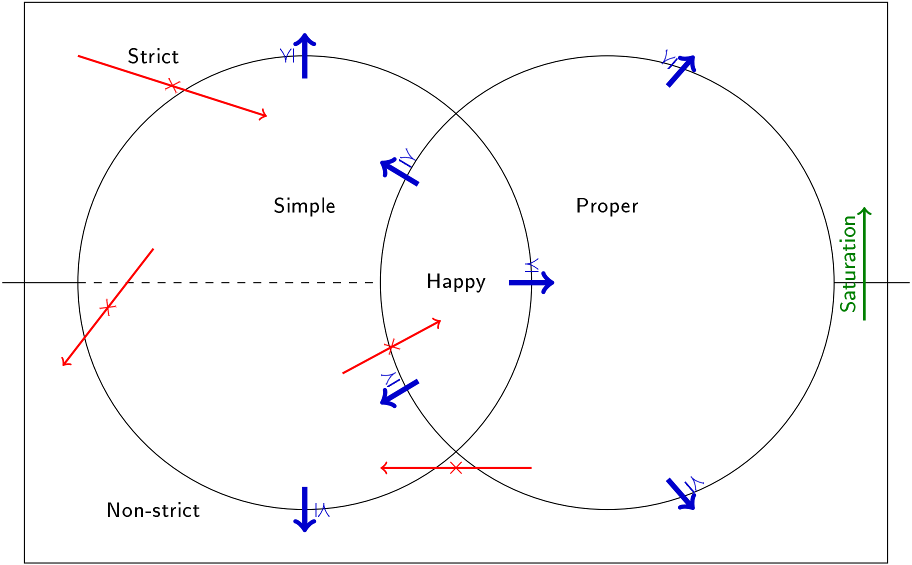
$\mathcal{G} \to \mathcal{H}$ such that there is a contact $(\{u, v\}, t)$ in $\mathcal{H}$ if and only if $\{u, v\}$ are connected at time t in $\mathcal{G}$.



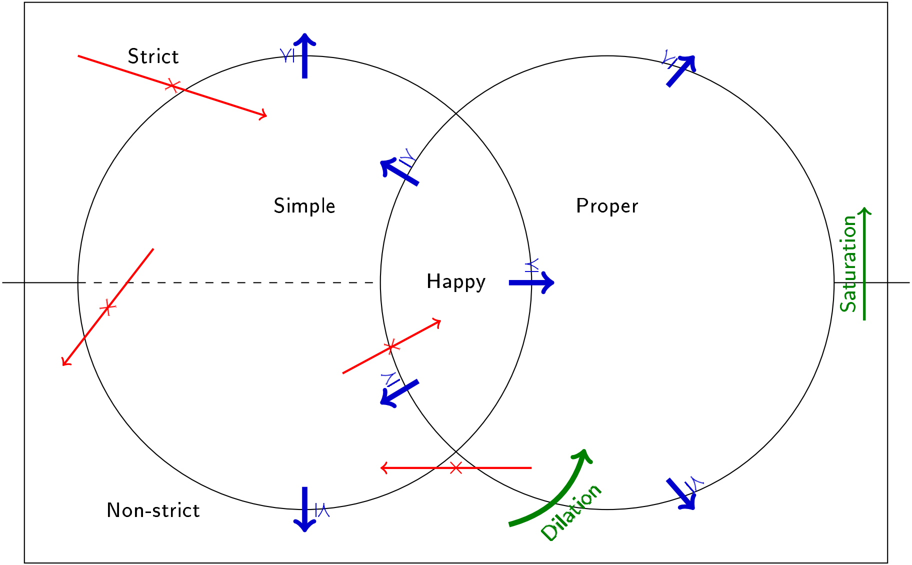
$\mathcal{G}$ $\mathcal{H}$

$\exists$ Strict (u, v)-journey in $\mathcal{H}$ if and only if $\exists$ non-strict (u, v)-journey in $\mathcal{G}$

Outline



Outline



Strict
Simple
Proper
Happy
Saturation
Non-strict
Dilation

Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

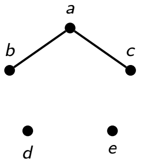
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

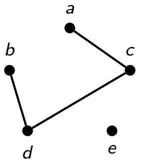
Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 1 : Duplicate the snapshots



G_1



G_2

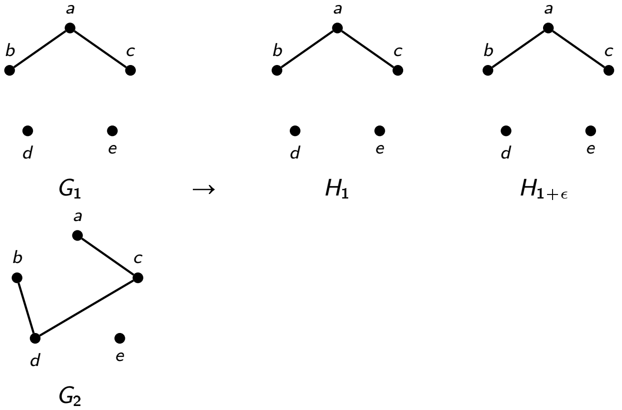
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 1 : Duplicate the snapshots



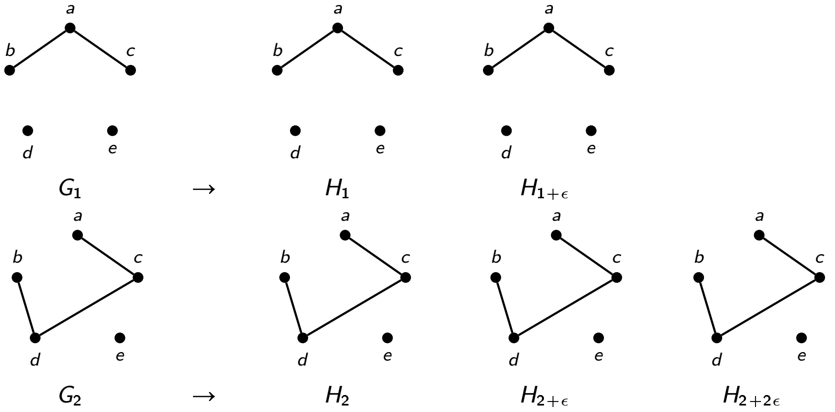
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 1 : Duplicate the snapshots



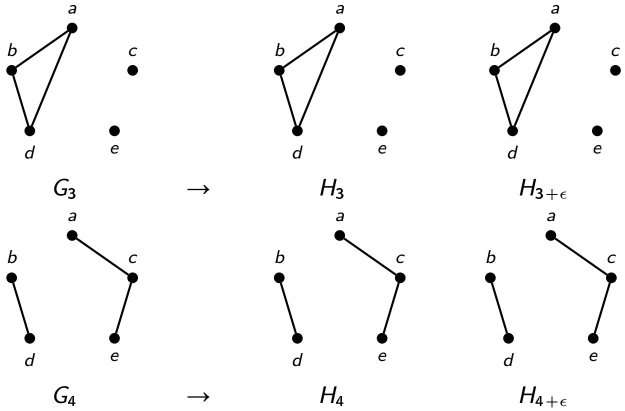
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 1 : Duplicate the snapshots



Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 2 : Tilt the time labels to make it proper

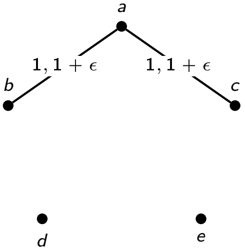
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 2 : Tilt the time labels to make it proper



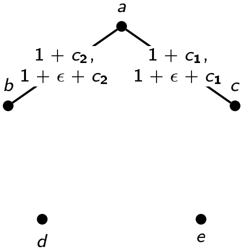
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 2 : Tilt the time labels to make it proper



Add a constant value c_i to each label

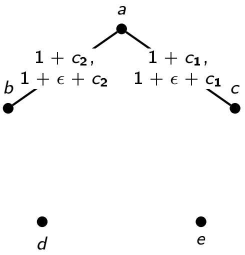
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 2 : Tilt the time labels to make it proper



Add a constant value c_i to each label

c_i is the edge-color

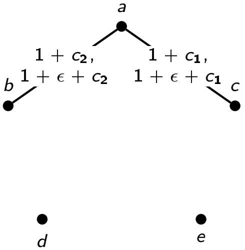
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 2 : Tilt the time labels to make it proper



Add a constant value c_i to each label

c_i is the edge-color

at most $\chi' \leqslant \Delta + 1$ colors

Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 3 : Normalize the time labels

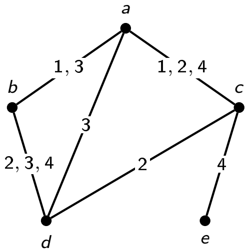
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

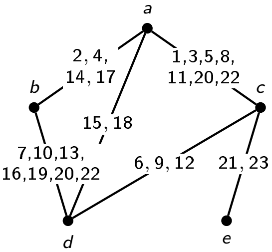
Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 3 : Normalize the time labels



$\mathcal{G}$



$\mathcal{H}$

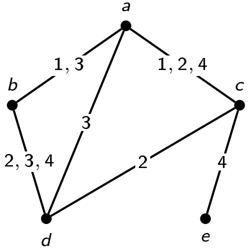
Time dilation method : from "non-strict" to "proper"

Goal : Turn a graph from "non-strict" to "proper" with the same reachability.

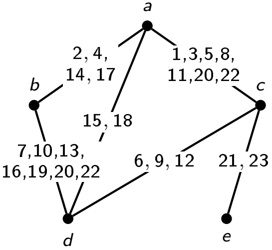
Time dilation method

Turn $\mathcal{G}$ into $\mathcal{H}$ such that the journeys use the same support (same sequence of edges).

Step 3 : Normalize the time labels



$\mathcal{G}$



$\mathcal{H}$

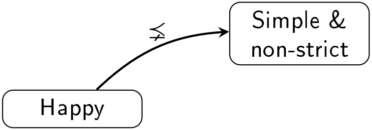
Lemma

$\mathcal{G}$ and $\mathcal{H}$ have the same reachability.

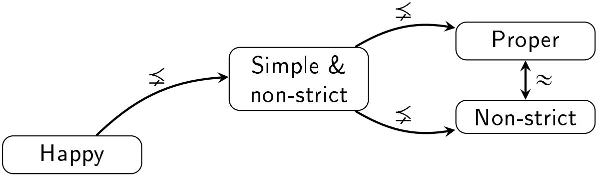
Summary : Ordering of the settings

Happy

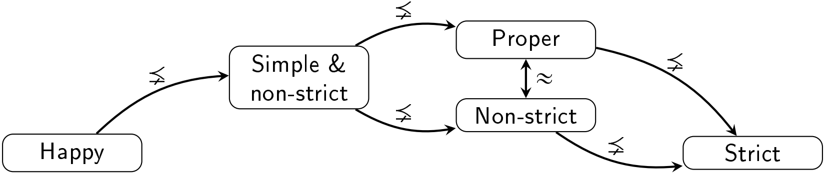
Summary : Ordering of the settings



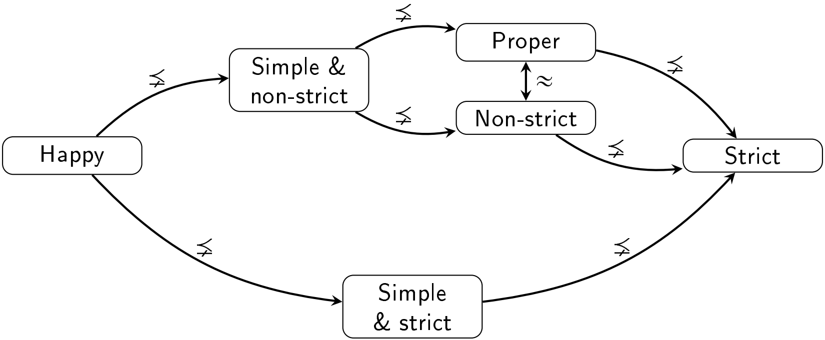
Summary : Ordering of the settings



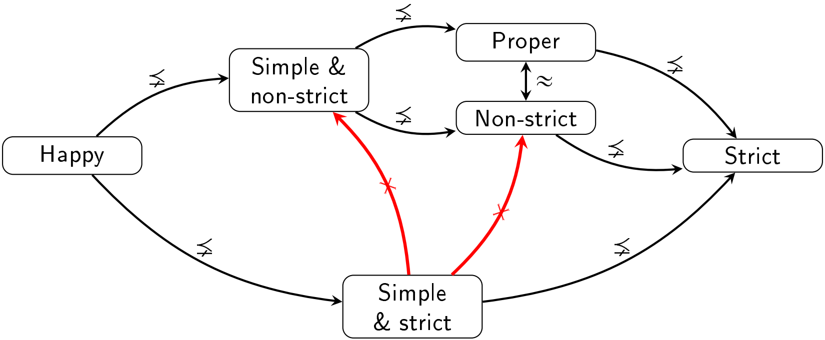
Summary : Ordering of the settings



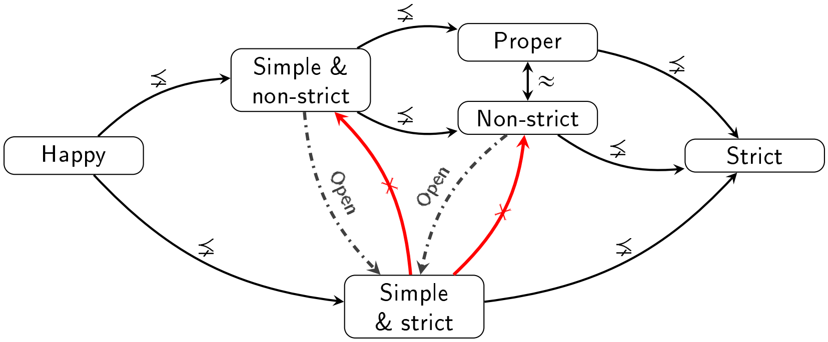
Summary : Ordering of the settings

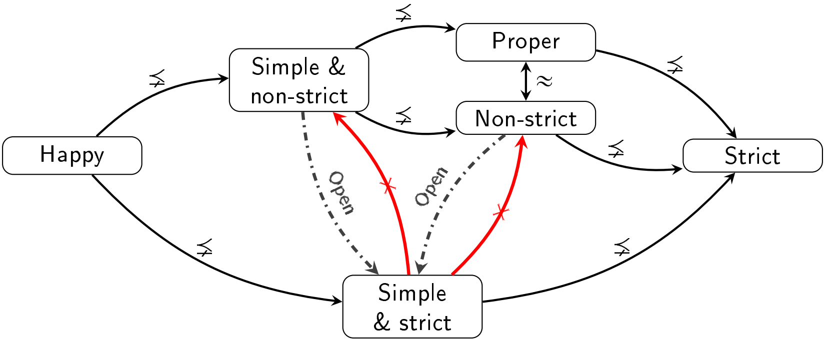


Summary : Ordering of the settings



Summary : Ordering of the settings

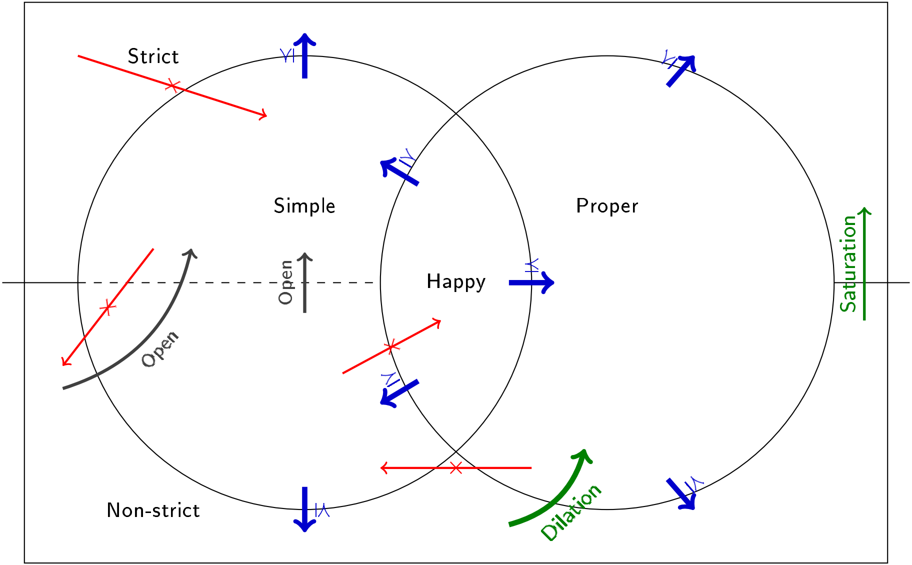




Observations

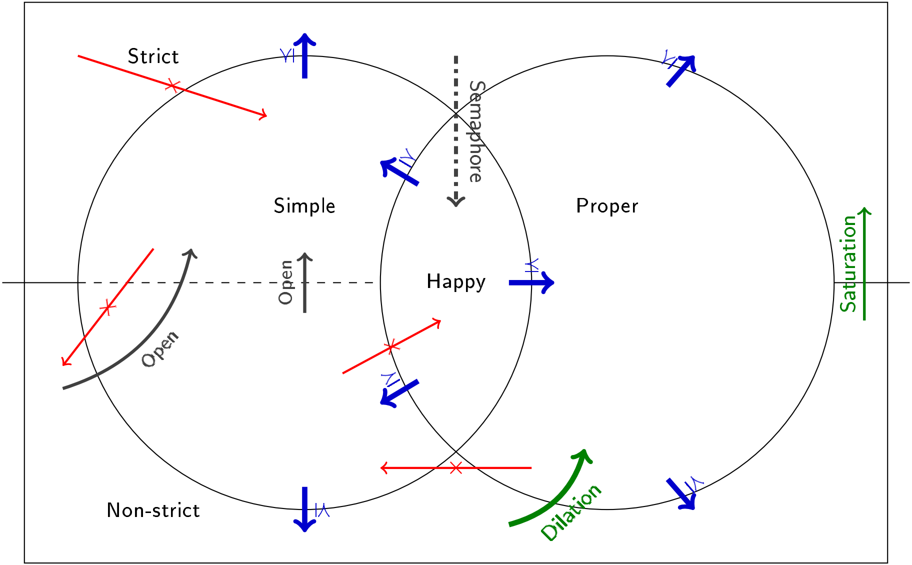
- Happy is the least expressive
- Strict is the most expressive

Conclusion



Strict
Simple
Proper
Open
Happy
Saturation
Open
Non-strict
Dilation

Conclusion



Strict
Semaphore
Simple
Proper
Open
Happy
Saturation
Open
Dilation
Non-strict

Thanks for your attention !