# Termination of Graph Rewriting using Weighted Type Graphs over Non-well-founded Semirings

Qi QIU

Supervisor: Xavier URBAIN
Université Claude Bernard Lyon 1, France
Université de Lorraine, France

# Plan (for myself)

- Introduction
  - Graphs and Graph Morphisms
  - Double Pushout Diagram (DPO)
  - Graph rewriting using DPO
    - Example of DPO graph rewriting : grsaa
  - Termination of DPO graph rewriting systems
    - Example: looping and terminating examples
- Type graph method in theory
- Type graph method in practice and the introduction of the main problem
- Morphism weight (by a example)
- Graph weight (by a example)
- A condition that weighted type graphs must satisfy
- Rule $\varphi$
- Example with rule $\varphi$
- Sufficient condition for termination using weighted type graph
- Sufficient condition simplified
- Example

# Outline

Introduction

Termination of Injective DPO Graph Rewriting using Morphism Counting

Etending the Type Graph Method to Non-well-founded Semirings

Termination of Injective DPO Graph Rewriting using Morphism Counting with antipatterns

Introduction
Graphs and Graph Morphisms
Termination of DPO Graph Rewriting Systems

Termination of Injective DPO Graph Rewriting using Morphism Counting

Etending the Type Graph Method to Non-well-founded Semirings

Termination of Injective DPO Graph Rewriting using Morphism Counting with antipatterns

# Graphs

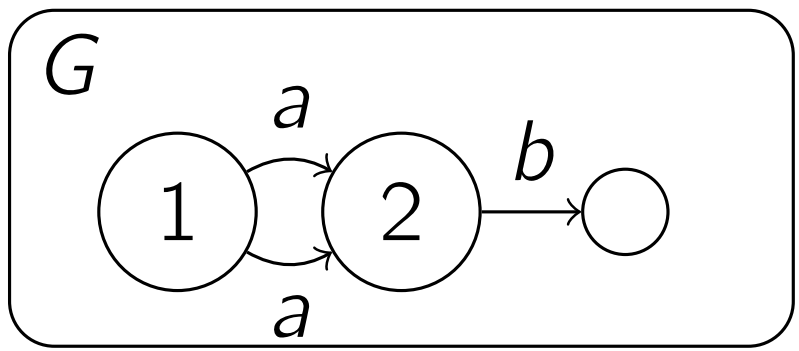
Graphs are finite and directed with:

- Parallel edges,
- Labeled edges,
- Finite labels.

Example:

- Graph $G$ with 3 nodes.
- 2 edges from node 1 to node 2 labeled by $a$.
- 1 edge from node 2 to node 3 labeled by $b$.

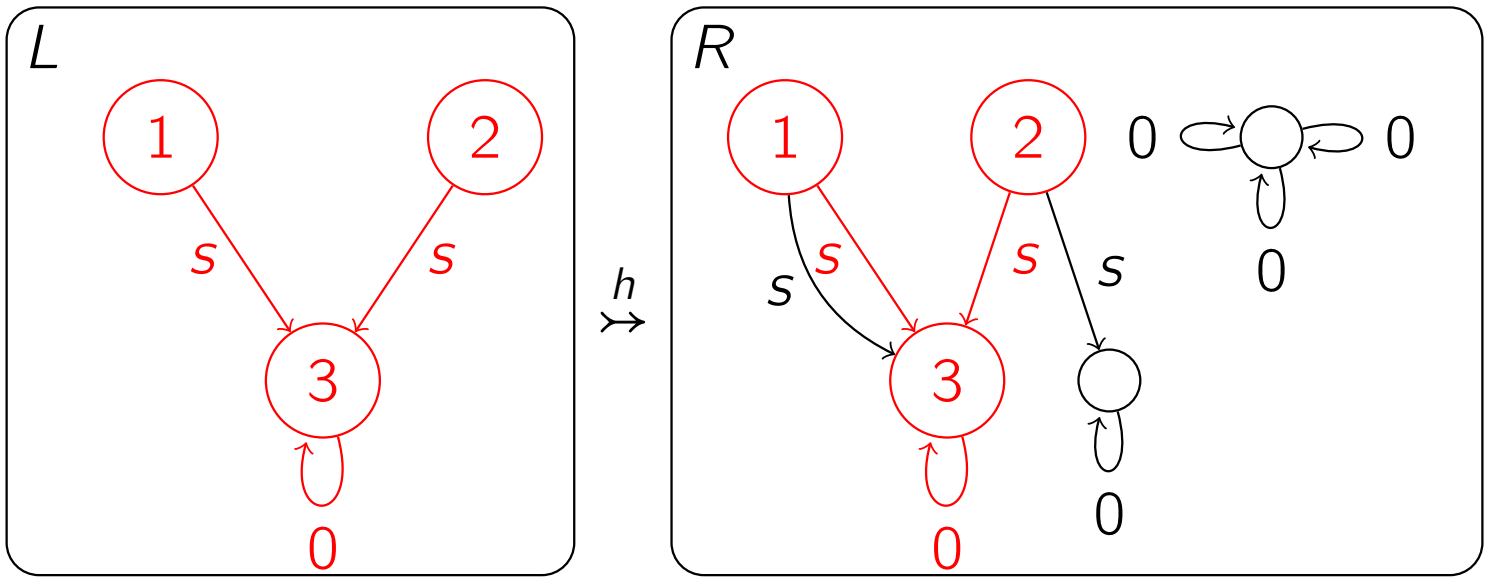
Notation:




- Graphs are contained in boxes
- Graph name $G$ is placed in the top left corner
- Numbers in nodes are identifiers, ommitted when not relevant

# Inclusion (morphism)

Morphisms: structure-preserving functions between graphs
Isomorphisms: bijective morphisms
**Inclusions**: morphisms that are indentity functions
Example in a viasual notion:

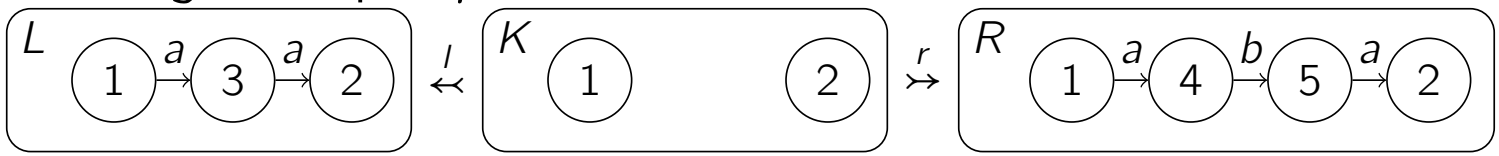



- $\overset{h}{\rightarrowtail}$ between the boxes marques an inclusion named $h$.

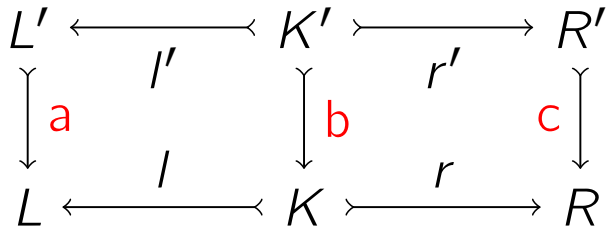
# Graph rewriting rule

**Rules** $\varphi = (L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R)$ consist of inclusions $l$ and $r$.

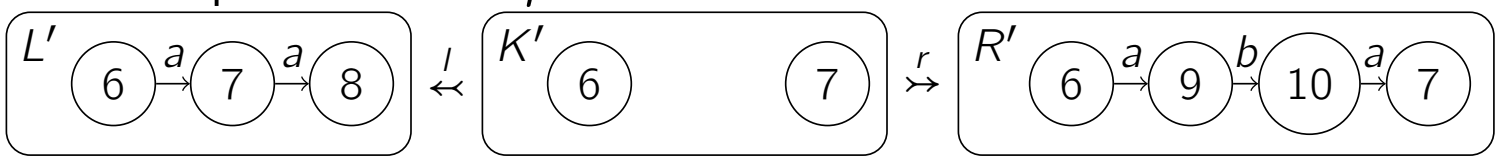
- Running example $\varphi$:




Rule $\varphi' = (L' \overset{l'}{\hookleftarrow} K' \overset{r'}{\hookrightarrow} R')$ and $\varphi$ are equivalent if there are isomorphisms $a, b, c$ such that $a \circ l' = l \circ b$ and $c \circ r' = r \circ b$:

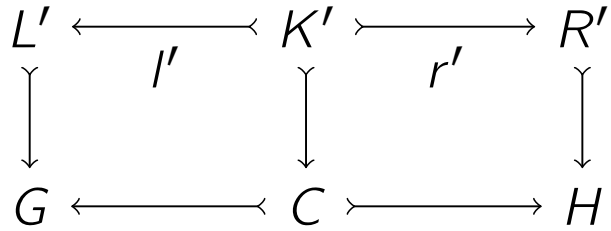



A rule equivalent to $\varphi$:

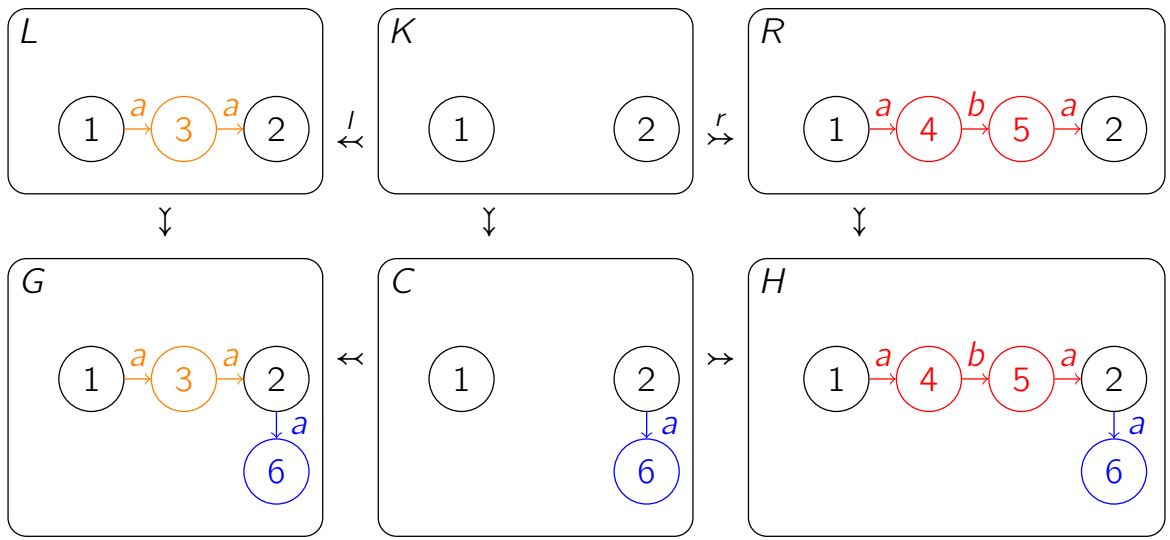


# Graph Rewriting

Rewriting steps $G \Rightarrow_\varphi H$ using rule $\varphi$ are commutative diagrams with an equivalent rule $L' \overset{l'}{\hookleftarrow} K' \overset{r'}{\hookrightarrow} R'$ where all morphisms are inclusions:

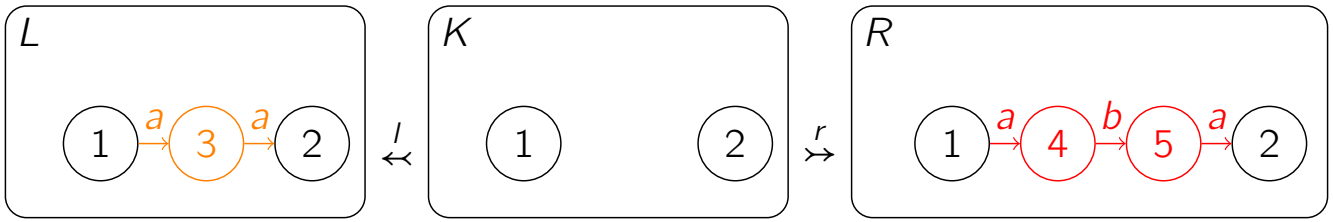



A rewriting step using $\varphi$:

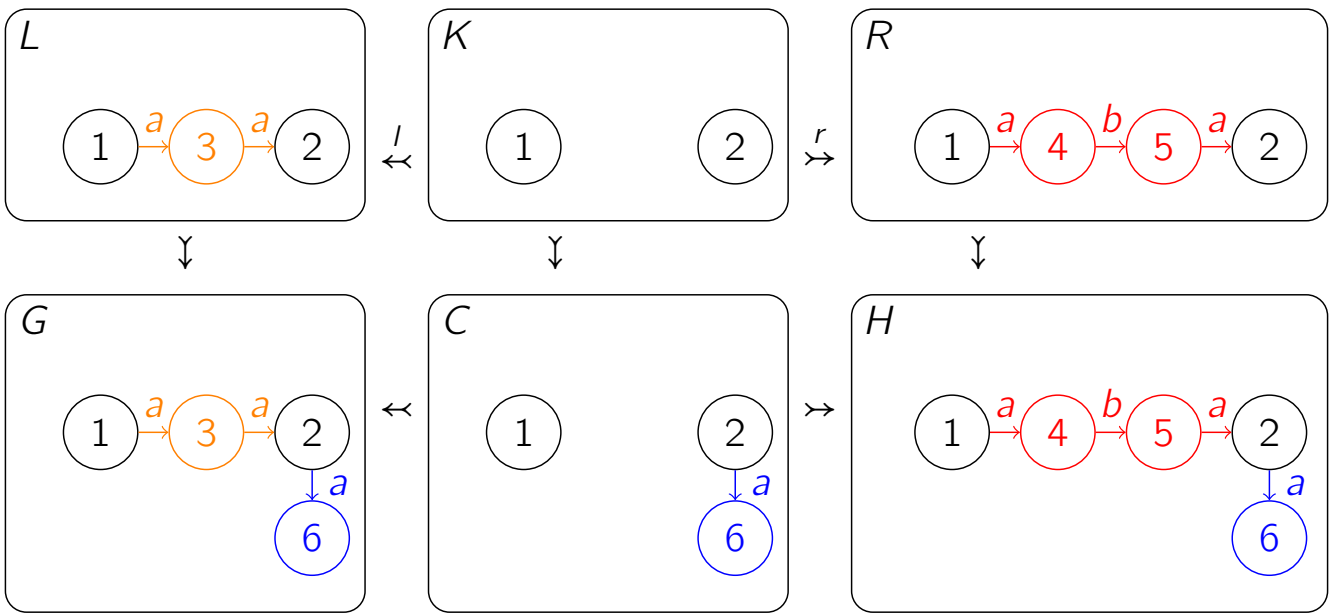


# A rewriting step of the running rule (to do : delete)

Rewriting rule $\varphi$:




A rewriting step $G \Rightarrow_\varphi H$:



# Termination of DPO Graph Rewriting Systems

- $\mathcal{R}$ : a set of rules
- No graph $G_0$ can be rewritten forever:

$$G_0 \underset{\mathcal{R}}{\Rightarrow} G_1 \underset{\mathcal{R}}{\Rightarrow} G_2 \underset{\mathcal{R}}{\Rightarrow} \cdots$$

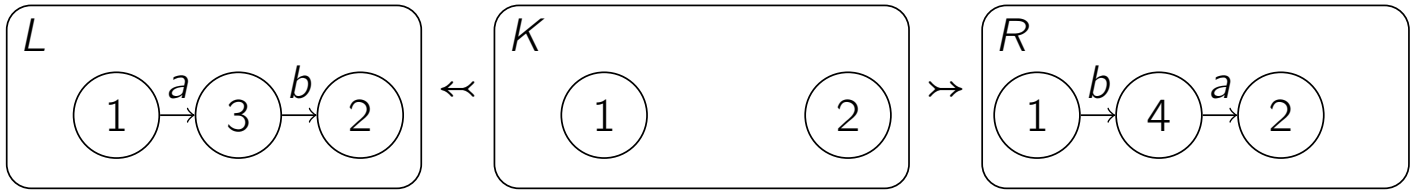
  when using the non-deterministic strategy

  "apply rules as long as possible"
- Aligns with the standard notion of program termination:

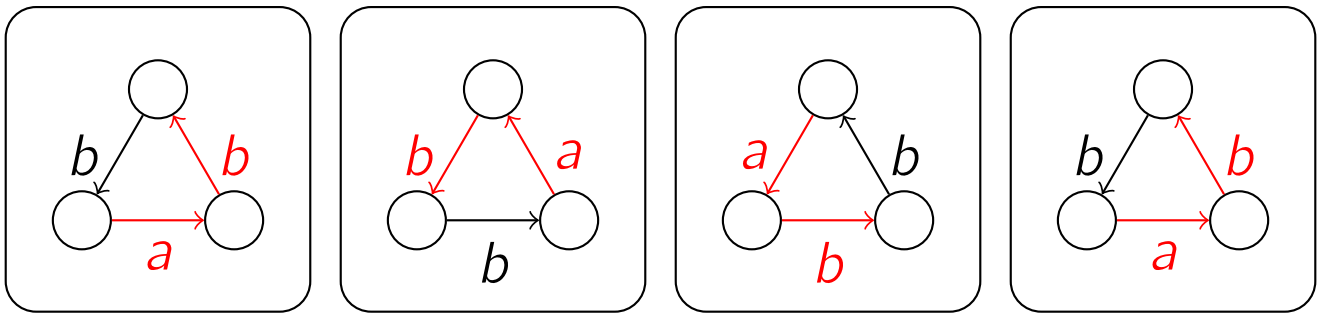
  "every execution (on any input) eventually halts."
- Undecidable in general

# One-rule examples

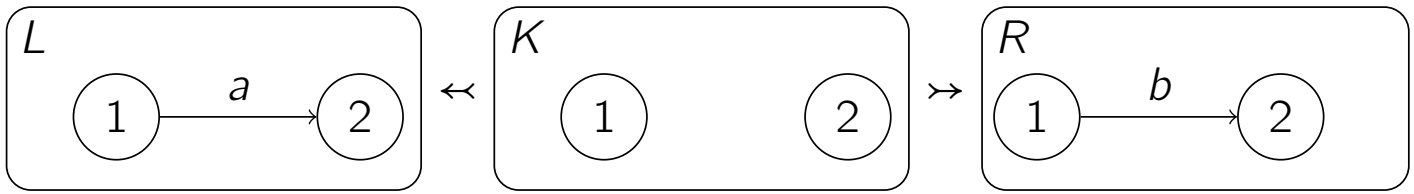
Rule $a$ :



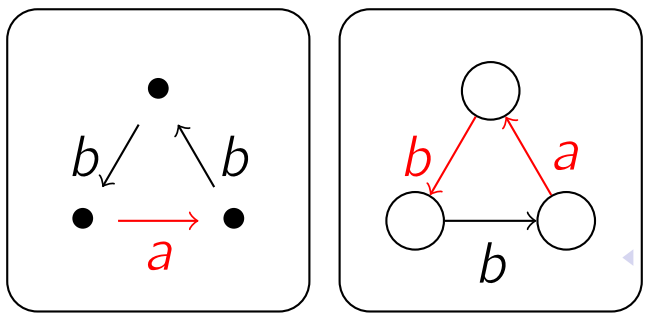

Looping:




Rule $b$:




Terminating: the number of edges labeled by "a" strictly decreases.



Introduction

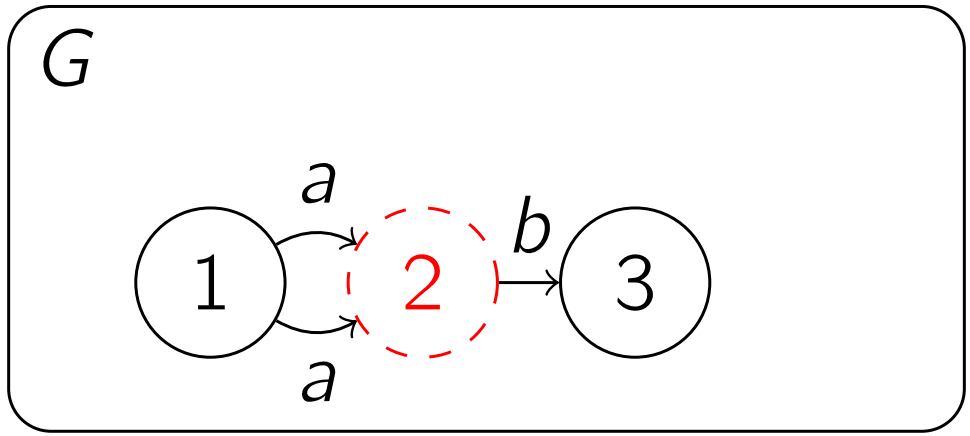
Termination of Injective DPO Graph Rewriting using Morphism Counting

Etending the Type Graph Method to Non-well-founded Semirings

Termination of Injective DPO Graph Rewriting using Morphism Counting with antipatterns

# Pre-graphs

Pre-graphs are graphs with missing nodes and dangling edges.
Example:

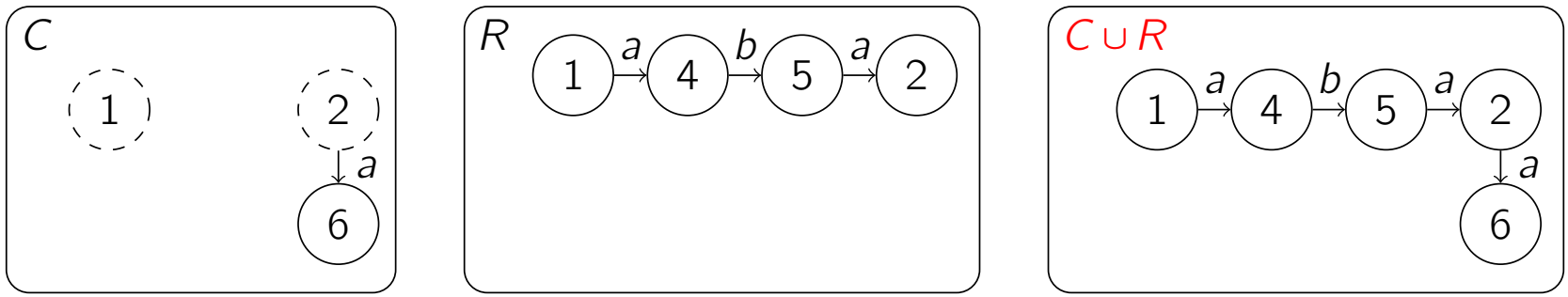



The pregraph $G$ has

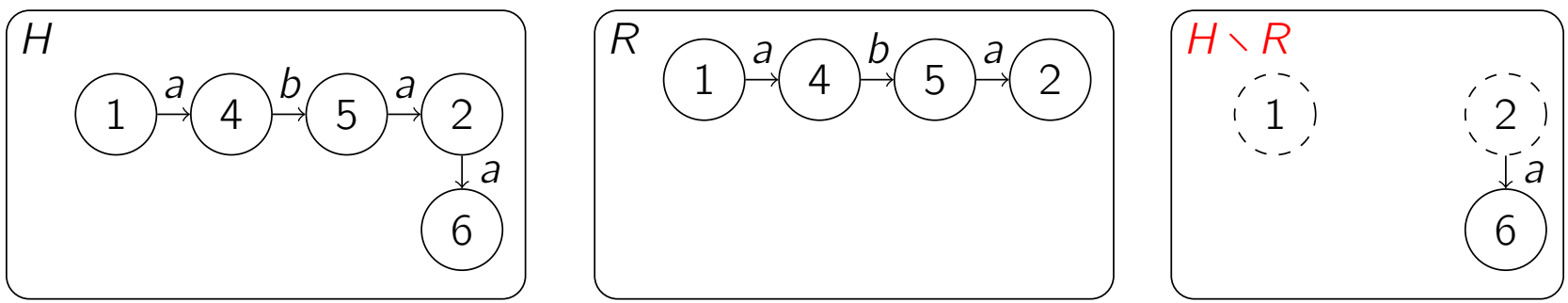
- 2 existing nodes,
- 1 missing node,
- 3 dangling edges.

# Pre-graph operations

Union of two pre-graphs $C \subseteq G$ and $R \subseteq G$, denoted $C \cup R$:




Relative complement of $R$ in $H$ where $R \subseteq H$, denoted $H \setminus R$:



Introduction

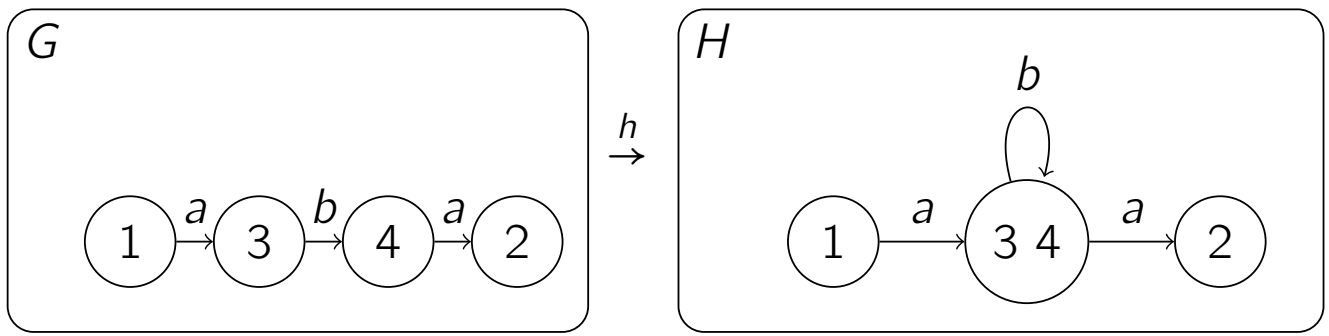
Termination of Injective DPO Graph Rewriting using Morphism Counting

Etending the Type Graph Method to Non-well-founded Semirings

Termination of Injective DPO Graph Rewriting using Morphism Counting with antipatterns

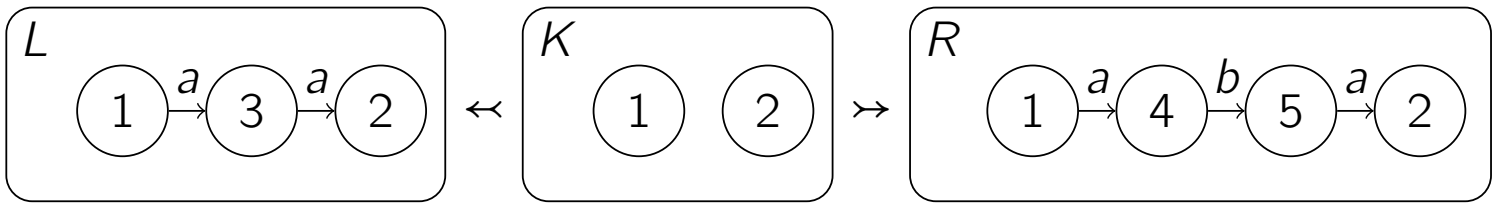
# Graphs and Graph Morphisms

- **Graph morphisms**: structure-preserving mappings.
  Ex:

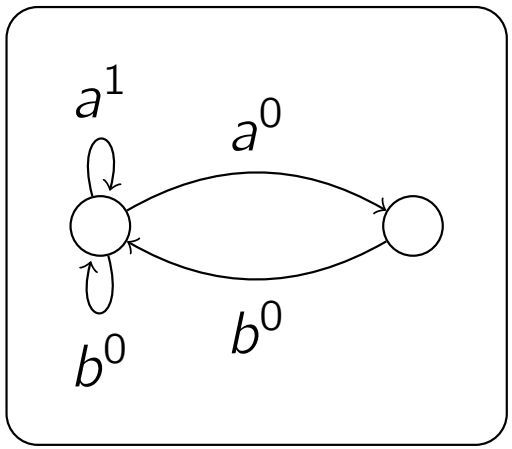


# Type graph method with weighted type graphs over natural numbers with an example

- Rule $\alpha$:

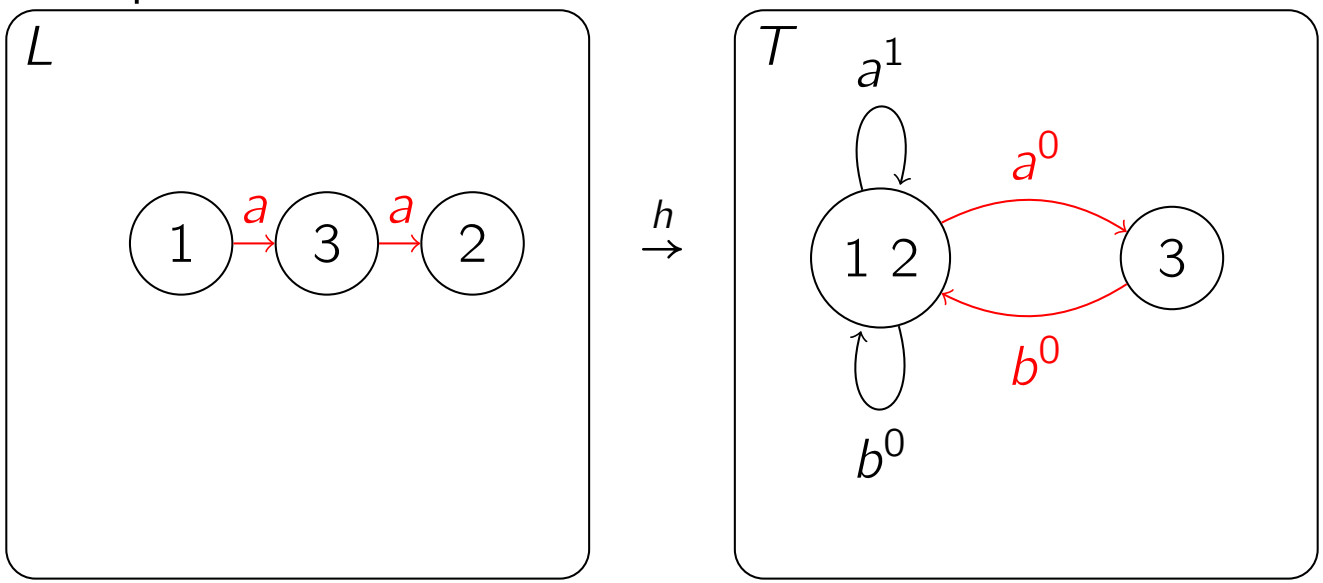



- Weighted type graph $T$ over natural numbers:



# Morphism Weight

- The weight of $h : L \to T$ is the sum of weights of all edges in $\mathsf{Im}(h)$.
- Example:

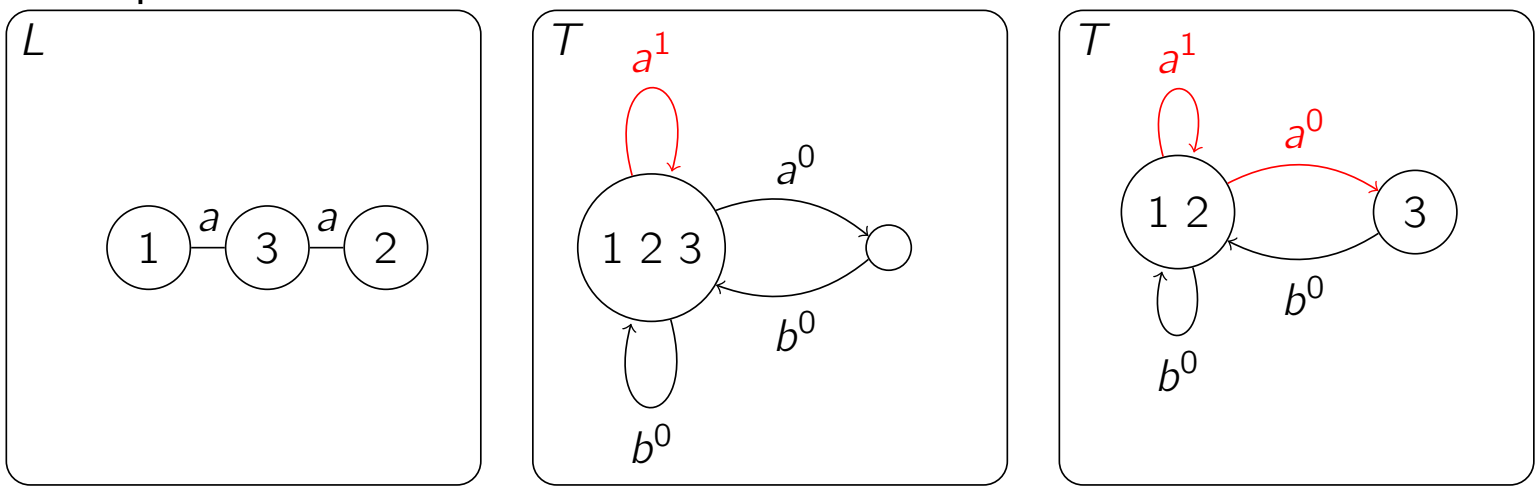



$w_T(h) = 0 + 0 = 0$

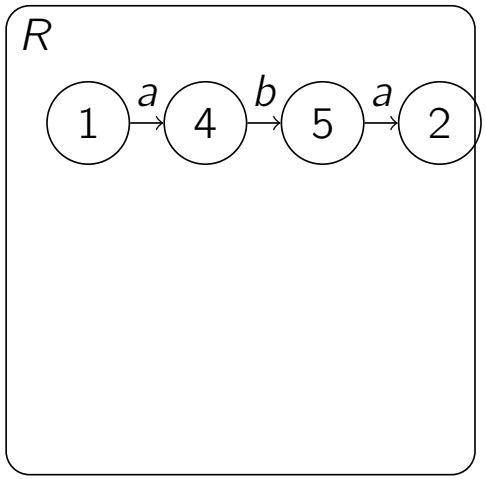
# Graph Weight

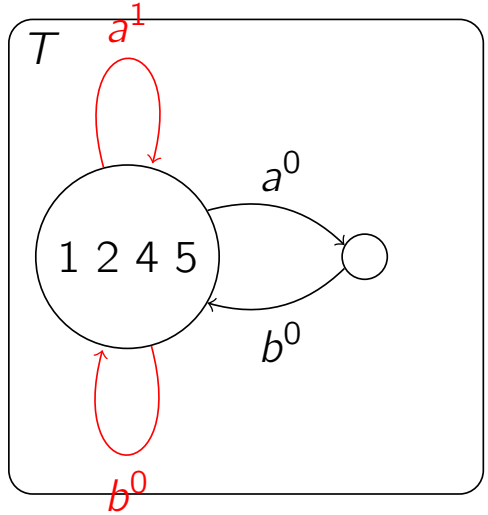
$w_T(L)$: the minimum weight $w_T(h)$ of all morphisms $h : L \to T$

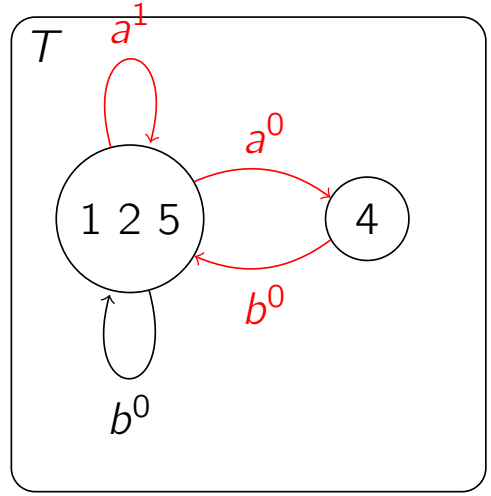
Example :

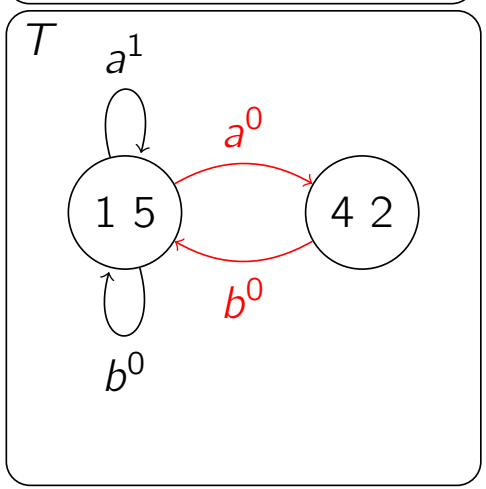



$w_T(L) = \min\{1 + 1, 1 + 0\} = 1$



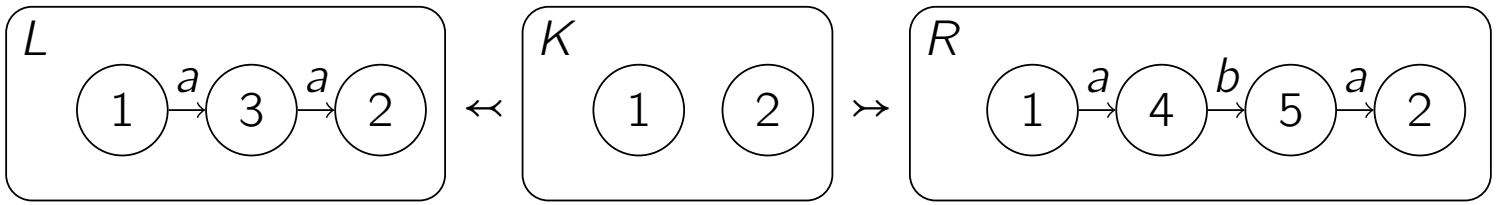




The weight of $R$ is

$$\min\{1+0+1, 0+0+1, 0+0+0\} = 0$$

Since for all $G \Rightarrow_\alpha H$, we have $w_T(G) \in \mathbb{N}$, it remains to show that every rewriting step strictly decreases the weight.
Not every weighted graph can be a weighted type graph It guarantees: for all $G \Rightarrow_\alpha H$, the weight of $G$ is defined and $w_T(G) \in \mathbb{N}$

# Termination Condition




For every morphism $t_K : K \to T$, we define

- $S(t_k, L)$ : the sum of the weights of the morphisms $t_L$ that extend $t_K$
- $S(t_k, R)$ : the sum of the weights of the morphisms $t_R$ that extend $t_K$

By [**endrullis2024generalized˙arxiv˙v2**], every rewriting step strictly decreases the weight if

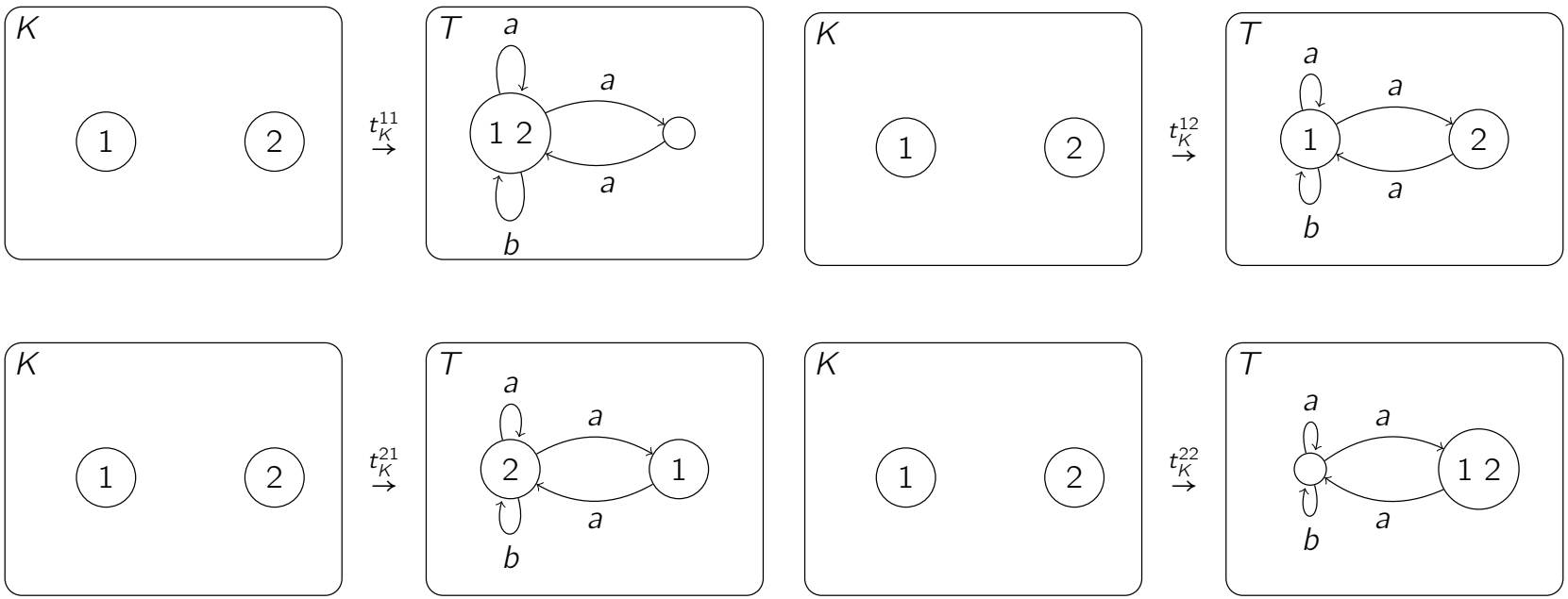
- for all $t_K : K \to T$,

$$S(t_K, L) > S(t_K, R)$$

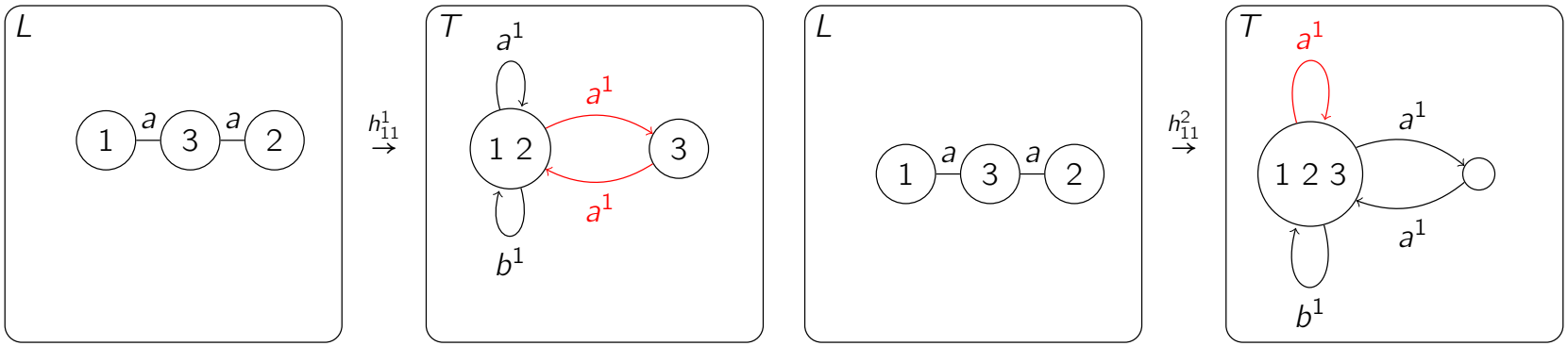
.

# Example

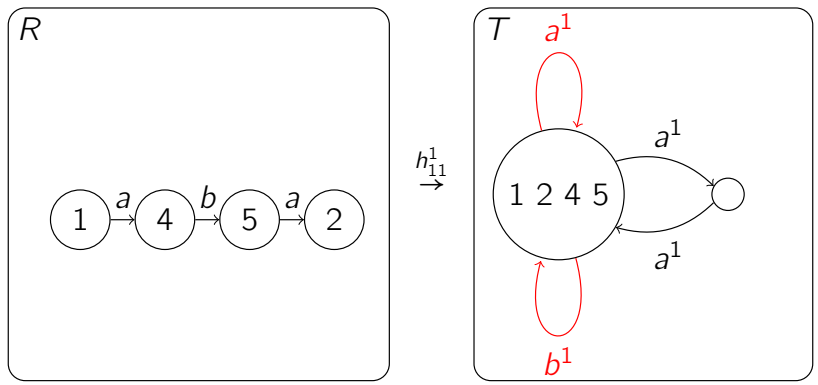
There are $t_K^{11}, t_K^{12}, t_K^{21}, t_K^{22} : K \to T$ as depicted below:



# Detail for $t_K^{11}$



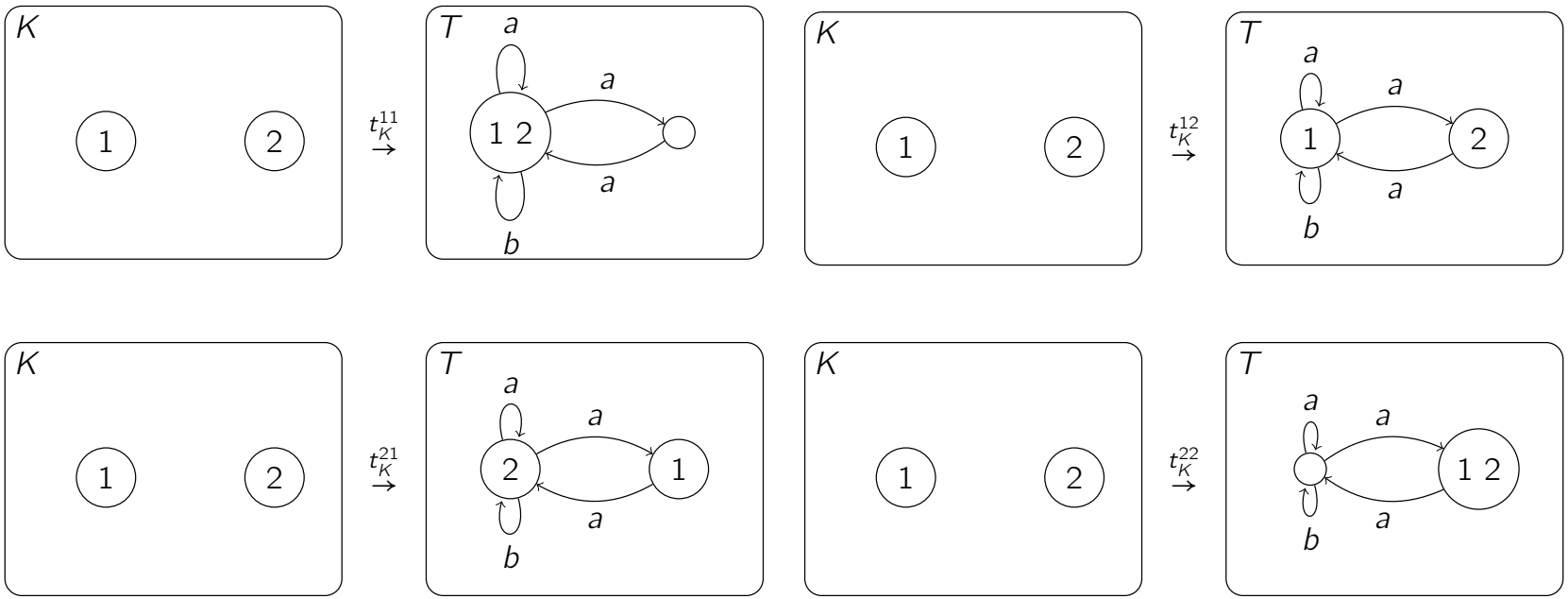

We have $S(t_k^{11}, L) = w_{\mathcal{T}}(h_{11}^1) + w_{\mathcal{T}}(h_{11}^2) = (1 * 1) + (1 * 1) = 2$




we have $S(t_k^{11}, R) = w_{\mathcal{T}}(h_{11}^3) = 1 * 1 * 1 = 1$.
Therefore, for $t_K^{11}$, we have: $S(t_k^{11}, L) = 2 > 1 = S(t_k^{11}, R)$.

# Example

There are $t_K^{11}, t_K^{12}, t_K^{21}, t_K^{22} : K \to T$ as depicted below:




- $t_K^{11} : 2 > 1$
- $t_K^{12} : 1 \geq 1$
- $t_K^{21} : 1 \geq 1$
- $t_K^{22} : 1 \geq 1$

Therefore, our Running Example is terminating by the weighted type graph $T$ over natural numbers.

# Type Graph Method with Weighted Type Graphs over Well-founded Semirings

The existence of suitable weighted type graphs is undecidable in general.

# Searching for Weighted Type Graphs over Natural Numbers

User-specified parameters:

- $k$ nodes
- edge weights in $\{0, 1, \ldots, n\}$

Assumption:

- no parallel edges of the same label

The problem amounts to checking the satisfiability of an existential Presburger arithmetic formula:

- $k^2|\Sigma|$ binary variables
- $k^2|\Sigma|$ integer variables

Challenge:

- there are $2^{k^2|\Sigma|} \cdot n^{k^2|\Sigma|}$ possible assignments of weights
- $k$ and $n$ unknown à priori

# Solution: using real numbers instead of natural numbers.

Every rewriting step strictly decreases the weight if

- for every $t_K : K \to T$,

$$S(t_K, L) > S(t_K, R)$$

.

- there is $\delta > 0$ such that for some $t_K : K \to T$,

$$S(t_K, L) > S(t_K, R) + \delta$$

.

# Searching for Weighted Type Graphs over Real Numbers

User-specified parameters:

- $k$ nodes
- ~~edge weights in $\{0, 1, \ldots, n\}$~~

Assumption:

- no parallel edges of the same label

The problem amounts to checking the satisfiability of an existential Presburger arithmetic formula:

- $k^2|\Sigma|$ binary variables
- $k^2|\Sigma|$ real variables

Challenge:

- ~~there are $2^{k^2|\Sigma|} \cdot n^{k^2|\Sigma|}$ possible assignments of weights~~
- there are $2^{k^2|\Sigma|}$ linear programs which have polynomial-time average-case complexity

# Implementation and Experimental Results

Implemented in the tool LyonParallel

- supports natural and real numbers
- searches with natural and real numbers in parallel and cooperates between them

Tested on examples from previous work:

- no need of user-specified upper bound on weights
- Acceleration

Introduction

Termination of Injective DPO Graph Rewriting using Morphism Counting

Etending the Type Graph Method to Non-well-founded Semirings

Termination of Injective DPO Graph Rewriting using Morphism Counting with antipatterns