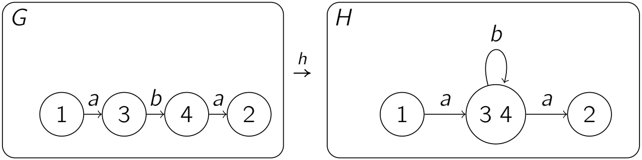
Termination of Graph Rewriting using Weighted Type Graphs over Non-well-founded Semirings

Qi QIU

Supervisor: Xavier URBAIN
Universite Claude Bernard Lyon 1, France
Univ. Grenoble Alpes, France

Graph rewriting

- Finite directed multigraphs with edge labels from a finite set Σ.
- Visualization of graph morphisms:



- Double pushout (DPO) graph rewriting
- Rules $\varphi = (L \xleftarrow{l} K \xrightarrow{r} R)$ consist of two graph morphisms $l : K \to L$ and $r : K \to R$.
- Rewriting steps $G \Rightarrow_\varphi H$ are double-pushouts:

$$\begin{tikzcd}
L \arrow[d] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d] \arrow[dl, phantom, "\mathrm{PO}"] \arrow[dr, phantom, "\mathrm{PO}"] & R \arrow[d] \\
G & C \arrow[l] \arrow[r] & H
\end{tikzcd}$$

Termination

- $\mathcal{R}$: set of DPO graph rewriting rules
- impossibility of transforming any graph G_0 indefinitely

  $$G_0 \Rightarrow_{\mathcal{R}} G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \cdots$$

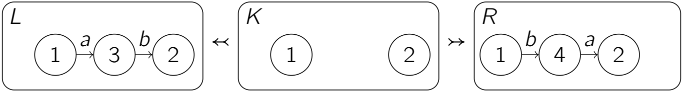
  with the non-deterministic strategy

  "apply rules as long as possible"

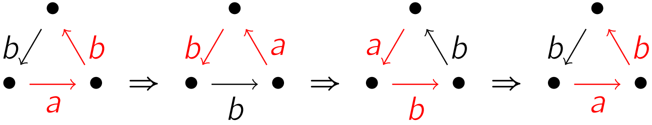
- Corresponds to program termination on all inputs in conventional programming languages
- Undecidable in general [**plump1998terminationundecidable**]

One-rule examples

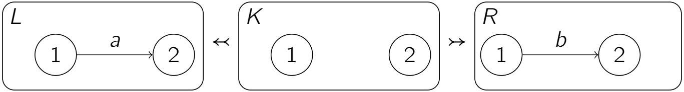
Rule a :



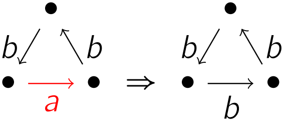
Looping:



Rule b:



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



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

- Parameter : a finite graph T with weighted edges
- Assigns weights to graphs according to T
- Weights are well-founded
- A rewriting system terminates if every rewriting step strictly decreases the weight.
- The existence of suitable weighted type graphs is undecidable in general

Type Graph Search in Practice

Previous work [**zantema2014termination**; **bruggink2014termination**; **bruggink2015proving**] search for a weighted type graph T with

- weights in $\mathbb{N}$
- $k \in \mathbb{N}$ nodes
- no parallel edges of the same label

Problem reduced to solving an integer program

- $k^2|\Sigma|$ binary variables
- $k^2|\Sigma|$ integer variables
- with or without multiplication of integer variables

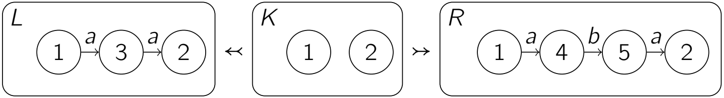
Complexity

- $O(2^{2k^2|\Sigma|})$ if without multiplication
- undecidable otherwise

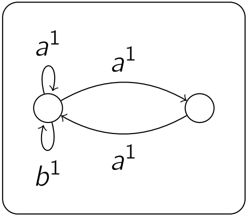
Can we reduce the complexity?

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

- Rule α:



- Weighted type graph T over natural numbers:

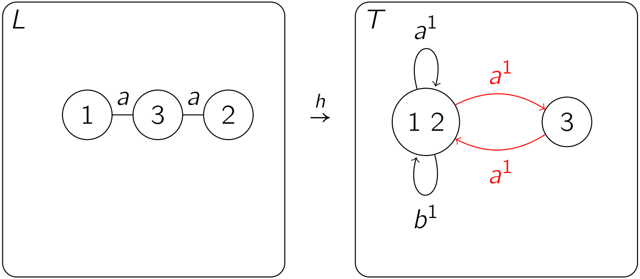


Weight $w_T(h)$ of Morphism $h : L \to T$

Weight function $w : \mathsf{Edges}(T) \to \mathbb{N}$
The weight $w_T(h)$ is defined as:

$$w_T(h) \stackrel{def}{=} \prod_{e \in \mathsf{Edges}(L)} w(h(e)),$$

the product of the weights of the images of the edges in L.
Example:

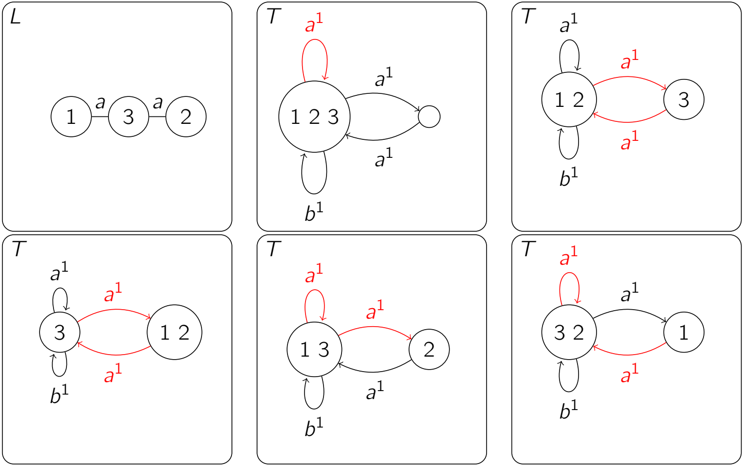


$w_T(h) = 1 * 1 = 1$

Weight w_T of Graph L

$w_T(L)$: the sum of the weights $w_T(h)$ of all morphisms $h : L \to T$

Example :



$w_T(L) = 1 + 1 + 1 + 1 + 1 = 5$

Not every weighted graph can be a weighted type graph

For our running example α,
there must be a morphism $c : L \to T$ such that
for all $G \Rightarrow_\alpha H$ defined by:

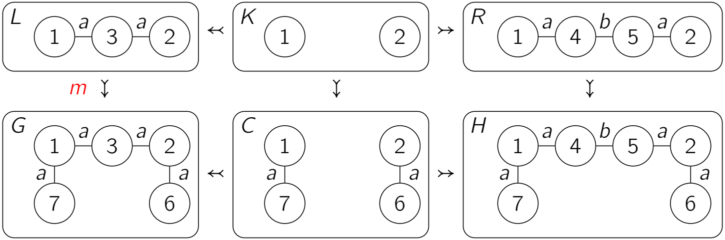
$$\begin{tikzcd} L \arrow[d, "m"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d] & R \arrow[d] \\ G & C \arrow[l] \arrow[r] & H \end{tikzcd}$$

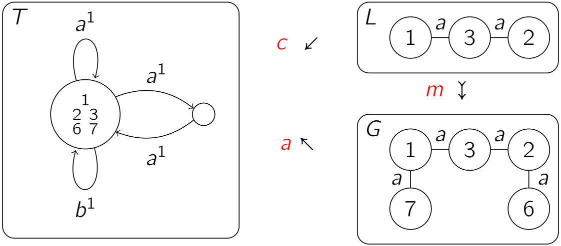
there is a morphism $a : G \to T$ such that $c = a \circ m$:

$$\begin{tikzcd} & L \arrow[dl, "c"'] \arrow[dd, "m"] \\ T & \\ & G \arrow[ul, "a"] \end{tikzcd}$$

It guarantees: for all $G \Rightarrow_\alpha H$, the weight of G is defined and $w_T(G) \in \mathbb{N}$

Example





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.

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 every $t_K : K \to T$,

$$S(t_K, L) \geq S(t_K, R)$$

.

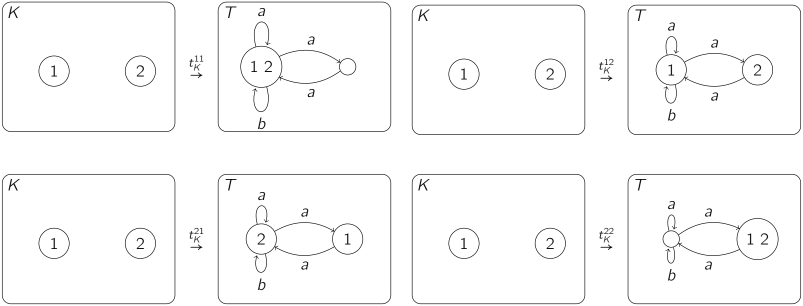
- for some $t_K : K \to T$,

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

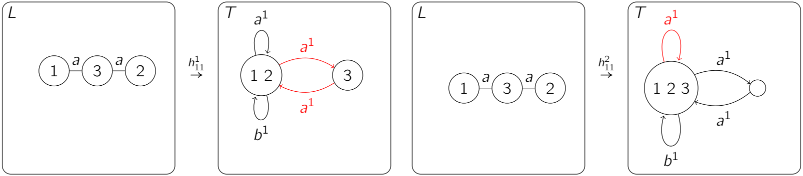
.

Example

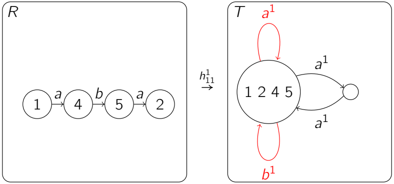
There are $t_K^{11}, t_K^{12}, t_K^{21}, t_K^{22} : K \to T$ as depicted below:



Detail for t_K^{11}



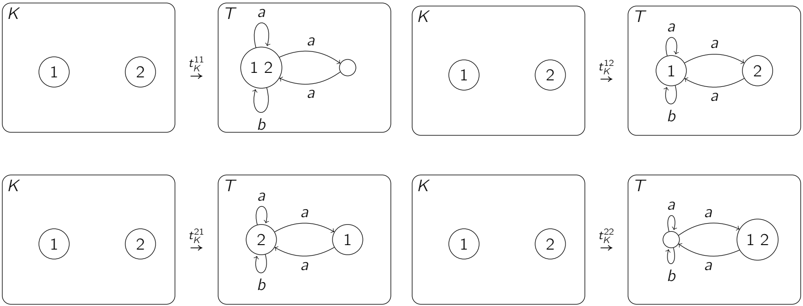
We have $S(t_k^{11}, L) = w_{\mathcal{T}}(h_{11}^1) + w_{\mathcal{T}}(h_{11}^2) = (1 * 1) + (1 * 1) = 2$



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.

Idea

Since $w_T(G) \geq 0$ for all $G \Rightarrow_\alpha H$ by definition of the type graph method.
We can replace the condition

"weights are well founded"

by

"each rewriting step decreases the weight by a constant $\delta > 0$"

Proposition

A rule α terminates, if

1. *there exists $\delta \in \mathbb{R}$, $\delta > 0$*
2. *$w_T(G) - w_T(H) \geq \delta$ for every rewriting step $G \Rightarrow_\alpha H$*

Condition for Termination

Every rewriting step strictly decreases the weight if

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

$$S(t_K, L) \geq S(t_K, R)$$

.

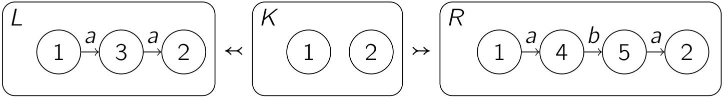
- for some $t_K : K \to T$,

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

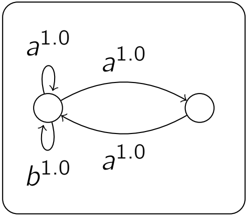
.

Termination of the running example with a weighted type graph over real numbers

- Rule φ



- Termination proved by the weighted type graph over real numbers:



Searching Weighted Type Graphs over the natural vs. real numbers

- k nodes
- no parallel edges of the same label from Σ

Weighted type graph searching problem:

1. Decide whether, for each pair of nodes s, t and label l, the edge $s \xrightarrow{l} t$ exists
2. Assign a ~~natural~~ real number weight to each existing edge
3. Check if the sufficient condition is satisfied

Mixed Integer Program:

- $k^2|\Sigma|$ binary variables
- $k^2|\Sigma|$ ~~integer~~ real variables
- with or without multiplication of real variables

Complexity:

- $\cancel{O(2^{2k^2|\Sigma|})}$ $O(2^{k^2|\Sigma|})$ if no multiplication, or
- ~~undecidable~~ decidable (but double exponential) otherwise

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:

- Acceleration when without multiplication for most cases
- Many timeouts (200 seconds) with multiplication because of
  - double exponential complexity
  - impossibility of further reducing the search space by restricting the weights

Conclusion

- Type Graph Method
  - termination technique
  - for DPO rewriting systems
  - using weighted type graphs
- Weights can be real numbers
- In theory: complexity reduced
- In practice: acceleration when without multiplication
- Tool: LyonParallel [1]

[1] `github.com/Qi-tchi/LyonParallel/tree/icgt2025`, MIT License

Acknowledgements

- Thanks to the anonymous reviewers for their valuable comments and suggestions.
- Graph visualization from [**overbeek2023apbpotutorial**].
- All concepts are from or adapted from [**endrullis2024generalized˙arxiv˙v2**].
- Termination section inspired by the slides of a presentation by Plump for [**plump2018modular**].

References I