Termination of Injective DPO Graph Rewriting Systems using Subgraph Counting

Qi QIU
Supervisor: Xavier URBAIN
Universite Claude Bernard Lyon 1, CNRS, INSA Lyon,
LIRIS, UMR5205, 69622 Villeurbanne, France
Univ. Grenoble Alpes, CNRS, Grenoble INP,
LIG, 38000 Grenoble, France

October 23, 2025

Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

SAT solver: miniSAT
Parallelization
Single Program Multi Data (SPMD)
Message Passing Interface (MPI) / C++
A processor / a miniSAT instance / a strategy
Impact of Clause Sharing Strategies

- limit the size of the clauses to be shared
- share with different number of neighbours
- ...

Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

Automated Theorem Proving by Saturation

Proving $\vdash F$ by proving $\neg F \vdash \bot$

Example:

- for proving $\vdash p \vee \neg p$
- calculate $\neg(p \vee \neg p) = \neg p \wedge p$
- we have $\{\neg p, p\} \vdash \bot$
- thus $\vdash p \vee \neg p$

Given Clause Procedure

An automated theorem prover by saturation needs to

- maintain a set of clauses S
- deduce new clauses from S
- delete redundant clauses from S

Given Clause Procedure (GC) is a set of rules for deciding

- which clauses to keep in S
- which clauses to use for deducing new clauses

Framework for Saturation Theorem Proving

Given Clause Procedure (GC)

- Process:
  - $\mathcal{N} \cup \mathcal{M} \implies_{\mathsf{GC}} \mathcal{N} \cup \mathcal{M}'$ where ...
- INFER:
  - $\mathcal{N} \cup \{(C, l)\} \implies_{\mathsf{GC}} \mathcal{N} \cup \{(C, \mathsf{active})\} \cup \mathcal{M}$ where ...

A Comprehensive Framework for Saturation Theorem Proving, Waldmann U., Tourret S., Robillard S., Blanchette J. (IJCAR 2020)

- Isabelle/HOL framework
- formalization of GC
- formalization of LGC

4 Variants employed by different Theorem Provers

1. Otter-loop(OL) : a refinement of GC
2. iProver-loop(IL) : an extension of OL
3. Discount-loop(DL) : a refinement of LGC
4. Zipperposition-loop(ZL) : an extension of DL

Correctness: Can any formula provable by a variant also be proved by GC?

Contribution

Formalization of the variants in Isabelle/HOL
Proving their correctness in Isabelle/HOL
International Conference on Automated Deduction 2023 (CADE-29)

Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

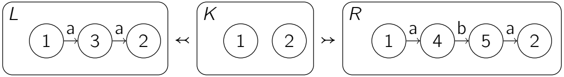
- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

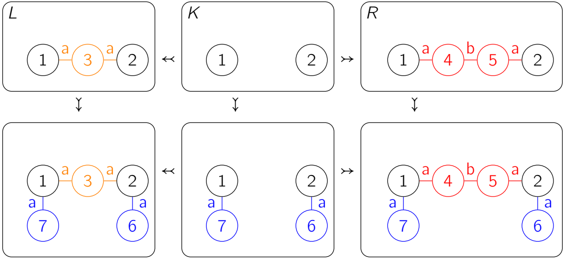
DPO Graph Rewriting and Termination

Edge-labeled directed graphs

DPO graph rewriting rule

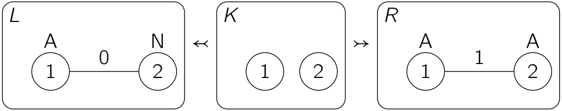


Graph transformation using DPO rewriting rule

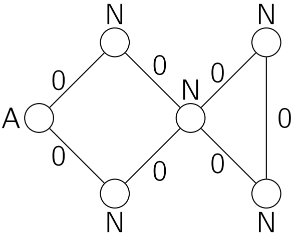


Termination of DPO graph rewriting rule

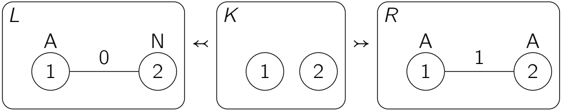
DPO Graph Rewriting System: Example



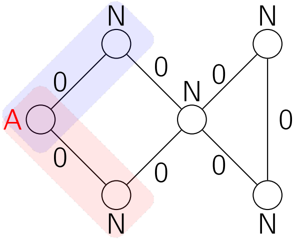
Example: the construction of a spanning tree



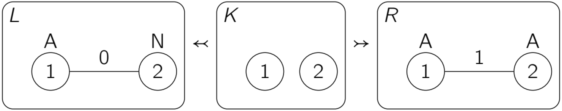
DPO Graph Rewriting System: Example



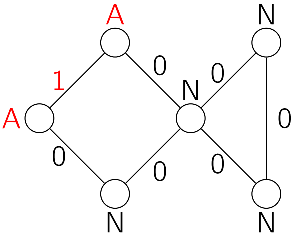
Example: the construction of a spanning tree



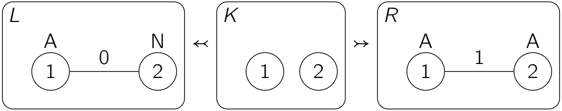
DPO Graph Rewriting System: Example



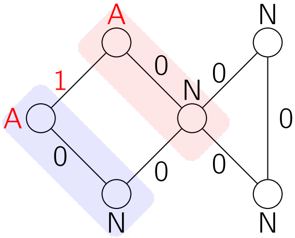
Example: the construction of a spanning tree



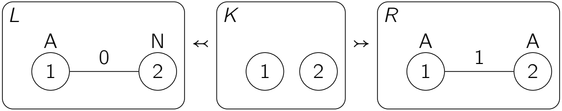
DPO Graph Rewriting System: Example



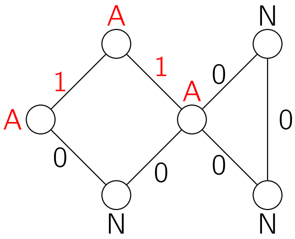
Example: the construction of a spanning tree



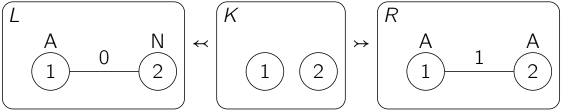
DPO Graph Rewriting System: Example



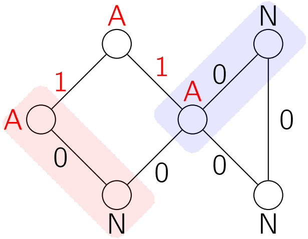
Example: the construction of a spanning tree



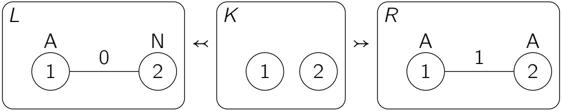
DPO Graph Rewriting System: Example



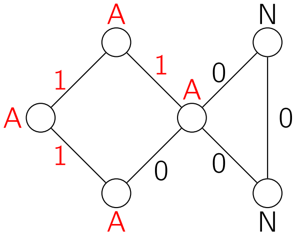
Example: the construction of a spanning tree



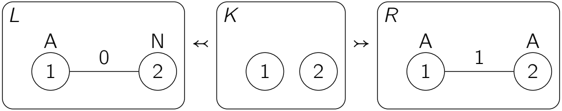
DPO Graph Rewriting System: Example



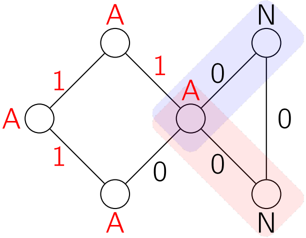
Example: the construction of a spanning tree



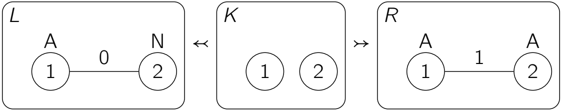
DPO Graph Rewriting System: Example



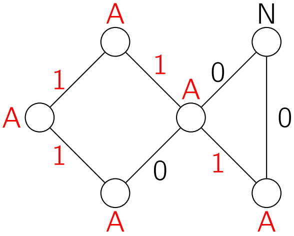
Example: the construction of a spanning tree



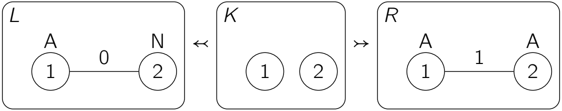
DPO Graph Rewriting System: Example



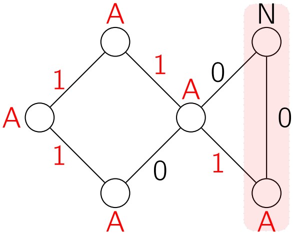
Example: the construction of a spanning tree



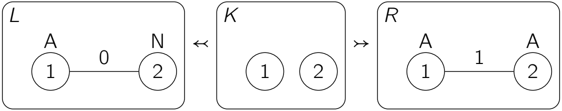
DPO Graph Rewriting System: Example



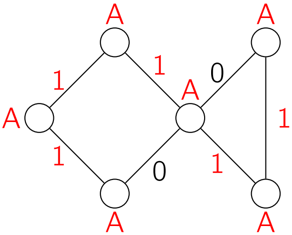
Example: the construction of a spanning tree



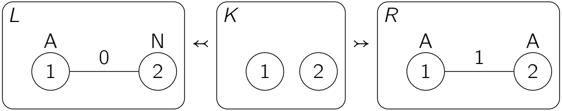
DPO Graph Rewriting System: Example



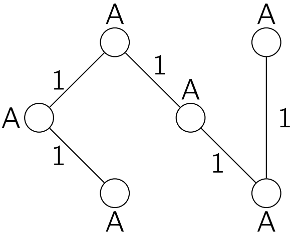
Example: the construction of a spanning tree



DPO Graph Rewriting System: Example



Example: the construction of a spanning tree



Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

Term Rewriting Systems (TRS)

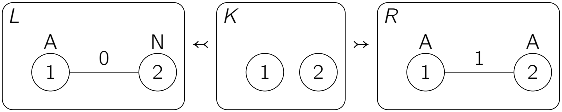
- Rule based term transformation
- Example of a TRS:

$$r_1 : f(x) \longrightarrow g(x)$$
$$r_2 : a \longrightarrow b$$

- Example of an execution:
  $f(a) \xrightarrow{r_1} g(a) \xrightarrow{r_2} g(b)$
- Can we make use existing advanced termination techniques for TRS?

Translation into TRS with AC Symbols

Example:



Translation :

$$\lambda(1, A) * \lambda(2, N) * \lambda(\{1, 2\}, 0)$$
$$\longrightarrow$$
$$\lambda(1, A) * \lambda(2, A) * \lambda(\{1, 2\}, 1)$$

Preservation of the termination property:

$$\begin{tikzcd}
\text{A terminates} & \text{GRS terminates} \arrow[l] & \text{TRS terminates} \arrow[l] \\
\text{A} \arrow[r] \arrow[u, dashed] & \text{GRS} \arrow[r] \arrow[u, dashed] & \text{TRS} \arrow[u]
\end{tikzcd}$$

Other Translations

Translation into

- Normalized Term Rewriting Systems (NTRS)
- Hierarchical Term Rewriting Systems (HTRS) with innermost strategy
- two others translations

Limitations

- Termination techniques for Term Rewriting Systems rely heavily on the tree structure of terms
- Graph has no tree structure
- New difficulties introduced by translation

Termination techniques for DPO Graph Rewriting Systems

Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

Well-founded semirings

A mathematical structure $(S, \oplus, \otimes, 0, 1, \prec, \preceq)$

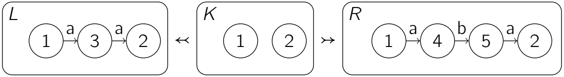
- $\otimes$, $\oplus$: binary operations
- 0, 1 : neutral elements for $\oplus$, $\otimes$
- $\prec, \preceq$: orders
- $\prec$ / $\preceq$: well-founded
- satisfies some conditions

Examples:

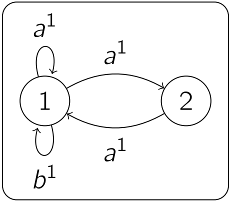
- The natural arithmetic semiring $(\mathbb{N}, +, *, 0, 1, <, \leq)$
- The natural tropical semiring: $(\mathbb{N} \cup \{+\infty\}, \mathbf{min}, +, +\infty, 0, <, \leq)$
- The natural arctic semiring: $(\mathbb{N} \cup \{-\infty\}, \mathbf{max}, +, -\infty, 0, <, \leq)$

Termination using Weighted Type Graphs

Termination of



by the weighted type graphs over the natural arithmetic semiring.



Existence of suitable weighted type graphs:

- undecidable in general

Practical Solution, its Limitation, and our Solution

Σ : finite set of labels

Searching a weighted type graph with

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

1. Decide if $s \xrightarrow{l} t$ exists for every pair of nodes s, t and label l
2. Assign a weight to every existing edge
3. Check if the weighted type graph satisfy requirements

Complexity $O(2^{2n})$ or undecidable

- $n = k^2 \cdot |\Sigma|$

Solution: weighted type graph over real numbers

Result: $O(2^{2n}) \Rightarrow O(2^n)$ or undecidable $\Rightarrow$ decidable

Intuition

Rewriting system $\mathcal{R}$ defines binary rewriting relation $\Rightarrow_{\mathcal{R}}$

A weighted type graph over natural numbers defines

- a homomorphism $h : (\mathbf{Graph}, \Rightarrow_{\mathcal{R}}) \to (\mathbb{N}, <)$

Codomain of h can be $(\mathbb{R}, <)$ if

- there is $\delta > 0$
- for all rewriting steps $G \Rightarrow H$:
  - $h(G) \geq 0$
  - $h(G) - h(H) \geq \delta$

Non-well-founded Semirings

Non-well-founded semiring $(S, \oplus, \otimes, 0, 1, \prec, \preceq, \mu)$

- $\otimes$, $\oplus$: binary operations
- 0, 1 : neutral elements for $\oplus$, $\otimes$
- $\prec, \preceq$: orders
- homomorphism $\mu : (S, \prec) \to (\overline{\mathbb{R}}, <)$
- $\prec$, $\otimes$, $\oplus$ satisfy some conditions

Well-founded semiring $(S, \oplus, \otimes, 0, 1, \prec, \preceq)$

- $\otimes$, $\oplus$: binary operations
- 0, 1 : neutral elements for $\oplus$, $\otimes$
- $\prec, \preceq$: orders
- $\prec / \preceq$: well-founded
- $\prec, \preceq$, $\otimes$, $\oplus$ satisfy some conditions

The tropical, arctic, and arithmetic semirings are instances of non-well-founded semirings.

- The natural tropical semiring:
  $\mathfrak{T} = (\mathbb{N} \cup \{+\infty\}, \mathsf{min}, +, +\infty, 0, <, \mathsf{id}_{\mathbb{N}\cup\{+\infty\}})$
- The natural arctic semiring:
  $\mathfrak{A} = (\mathbb{N} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0, <, \mathsf{id}_{\mathbb{N}\cup\{-\infty\}})$
- The natural arithmetic semiring $\mathfrak{N} = (\mathbb{N}, +, *, 0, 1, <, \mathsf{id}_{\mathbb{N}})$

Termination result

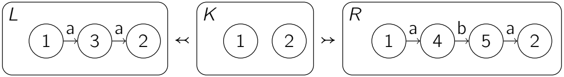
Theorem (Termination of DPO rewriting system)

Let φ be a DPO rewriting rule, T a finite weighted type graph over one of our non-well-founded semirings over real numbers such that

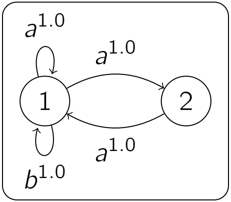
1. *edge weights are positive real numbers*
2. *the rule satisfies some conditions.*
3. *there is $\delta > 0$ such that: for all rewriting step $G \Rightarrow H$, we have $w(G) - w(H) \geq \delta$*

then φ terminates.

Example with Non-well-founded Semirings

Termination of 

by the weighted type graphs over the real arithmetic semiring.



For every rewriting step $G \Rightarrow H$, we have $w(G) - w(H) \geq 1.0$.

Contribution

Extension of the existing approach to real numbers

Automated Termination Prover: LyonParallel

- our approach
- previous approach
- parallel execution
- cooperation
- more user-friendly for non-experts than existing tools
- OCaml

Accepted for publication

- International Workshop on Graph Computation Models 2025

Master's Thesis Projects

- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

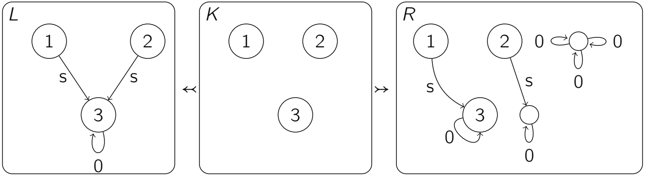
Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

Limitation of Existing Techniques

Injective DPO graph rewriting rule:

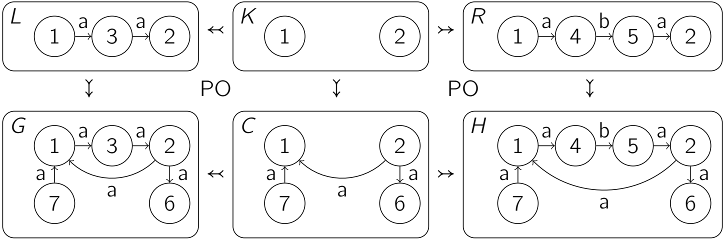


Terminating because of the strict decrease of the number of occurrences $\bullet \xrightarrow{s} \bullet \xleftarrow{s} \bullet$

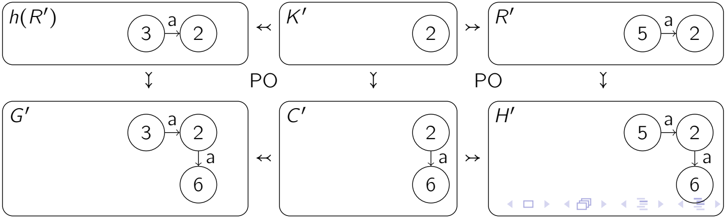
Termination cannot be proved by existing techniques

Implicit Occurrences: Occurrences depending on the context of the rewriting step

Injectively DPO graph rewriting rule:



An implicit occurrence of chain $\bullet \xrightarrow[a]{} \bullet \xrightarrow[a]{} \bullet$



X-non-increasing rule

$\varphi = L \overset{l}{\leftarrowtail} K \overset{r}{\rightarrowtail} R$: a rule
$X \subseteq R$: a graph
$D(R, X)$: subgraphs of R which can form an implicit X-occurrence in some rewriting step
φ is X-non-increasing rule if

1. For every $R_i \in D(R, X)$, there is a morphism $h_i : R_i \rightarrowtail L$ which preserves the interface elements,
2. three more conditions on h_i

Condition 1: for every implicit X-occurrence in H, there is a corresponding implicit X-occurrence in G with the same interface elements,

Other conditions: different implicit X-occurrences in H are mapped to different implicit X-occurrences in G,

for every rewriting step $G \Rightarrow H$.

Main results

$\varphi = L \twoheadleftarrow K \rightarrowtail R$: a rule
$X \subseteq R$: a graph

Lemma

For every rewriting step $G \Rightarrow H$ using φ, there are more implicit X-occurrences in G than in H if

- *φ is X-non-increasing*

Theorem (Sufficient termination condition)

φ terminates if

- *φ is X-non-increasing*
- *There are strictly more X-occurrences in L than in R*

Contribution

Machine-checkable sufficient termination condition

Termination of new classes of graph rewriting systems

Implementation in **LyonParallel**

International Conference on Graph Transformation (ICGT 2025)

Master's Thesis Projects

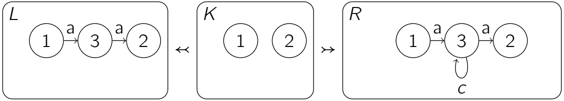
- Impact of Clause Sharing Strategies in Parallel SAT Solvers
- Formal Verification of Theorem Proving Methods

Doctoral Research: Termination of Graph Relabeling Systems

- DPO Graph Rewriting and Termination
- Termination via Translation to Term Rewriting Systems
- Termination using Weighted Type Graphs
- Termination using Subgraph Counting
- Termination using Subgraph Counting with antipatterns

Future Work

Limitation of Subgraph Counting and Solution



Solution : Counting occurrences of L not including in an occurrence of R.

Contribution

Counting subgraphs with antipatterns

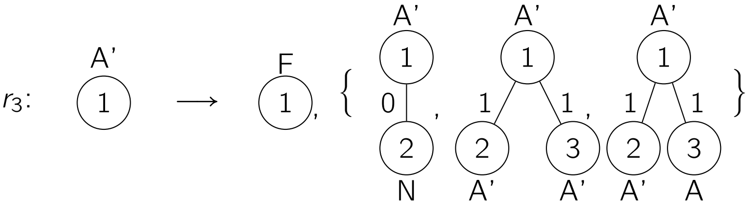
Termination of new classes of injective DPO graph rewriting systems

Tool available : **LyonParallel**

Under revision for resubmission

Future Work

Graph transformation rule with negative application conditions:



Extending the technique to DPO graph rewriting systems with negative application conditions