MANCHESTER 1824
The University of Manchester

Inverse Semigroups and the Isomorphism Problem for Context-Free Trees

Jan Philipp **Wächter**

Department of Mathematics
University of Manchester

This research was supported by EPSRC

16 September 2026

Preliminaries

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ:

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and

Preliminaries

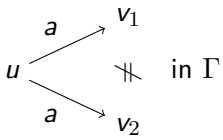
- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:

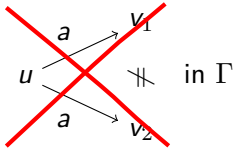
Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:



Preliminaries

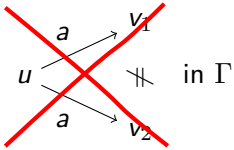
- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:



in Γ

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:

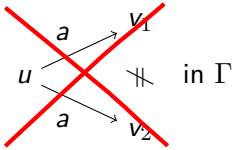


in Γ

- co-deterministic A-graph Γ:

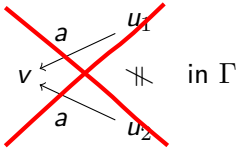
Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1} \colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:



in Γ

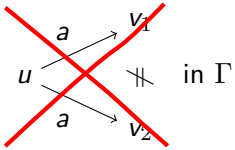
- co-deterministic A-graph Γ:



in Γ

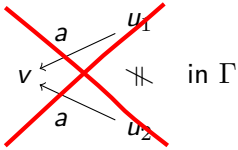
Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1}\colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $\quad A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:



in Γ

- co-deterministic A-graph Γ:

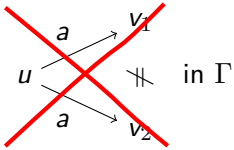


in Γ

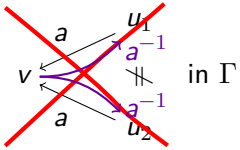
- inverse A-graph: involutive + strongly connected + deterministic + co−deterministic

Preliminaries

- involutive alphabet: alphabet A with involution $\cdot^{-1}\colon A \to A$ (i.e. $(a^{-1})^{-1} = a$ for all $a \in A$)
  $A^{-1} = \{a^{-1} \mid a \in A\}$: disjoint copy $A^{\pm 1} = A \uplus A^{-1}$: involutive alphabet
- A-graph: $\Gamma = (V, E)$ with $E \subseteq V \times A \times V$ (i.e. directed, A-edge-labeled)
  Today: usually with a root
- involutive A-graph Γ: A involutive alphabet and $u \xrightarrow{a} v$ in $\Gamma \iff u \xleftarrow{a^{-1}} v$ in Γ
- deterministic A-graph Γ:



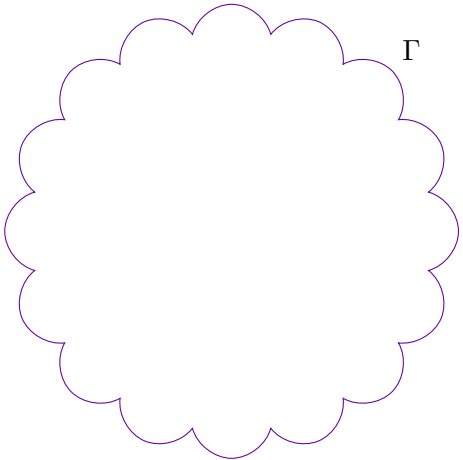
- co-deterministic A-graph Γ:



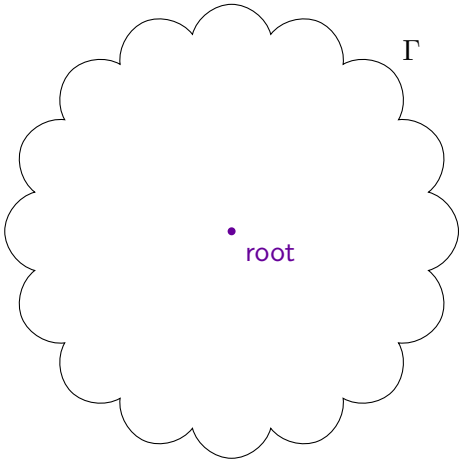
- inverse A-graph: involutive + strongly connected + deterministic + co-deterministic

End-Cones and End-Isomorphisms

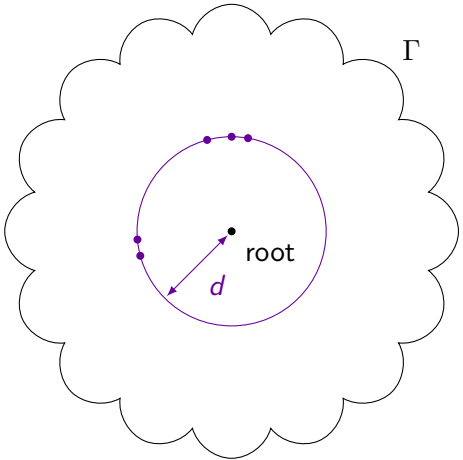
End-Cones and End-Isomorphisms



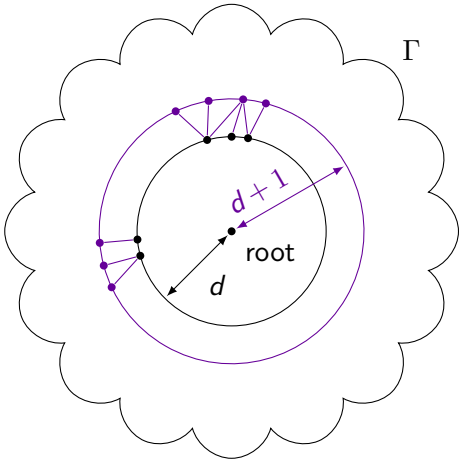
End-Cones and End-Isomorphisms



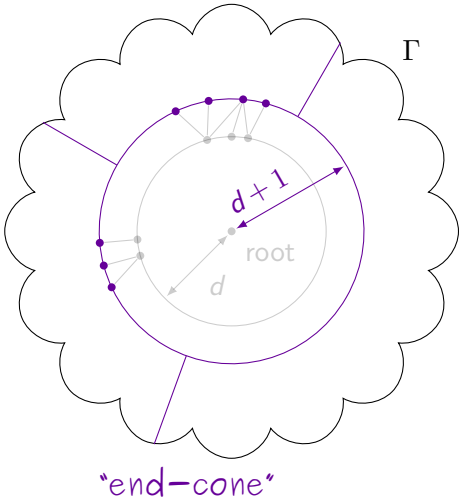
End-Cones and End-Isomorphisms



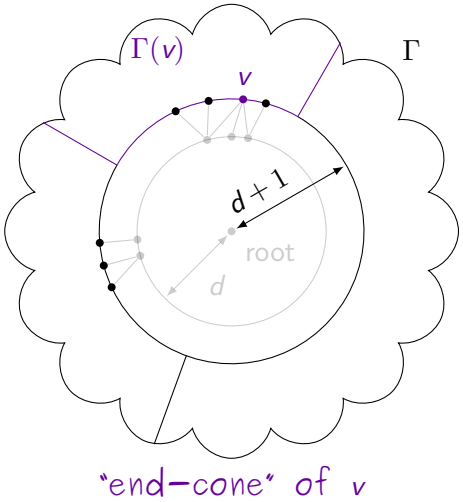
End-Cones and End-Isomorphisms



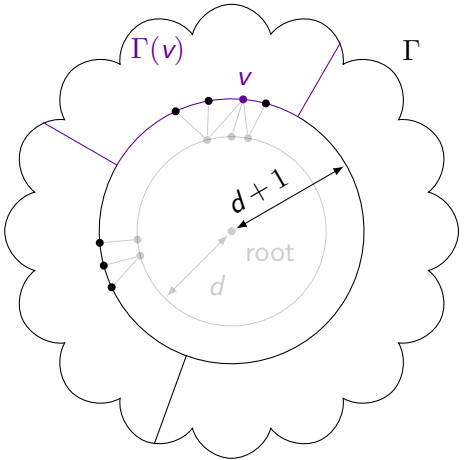
End-Cones and End-Isomorphisms

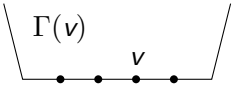


End-Cones and End-Isomorphisms

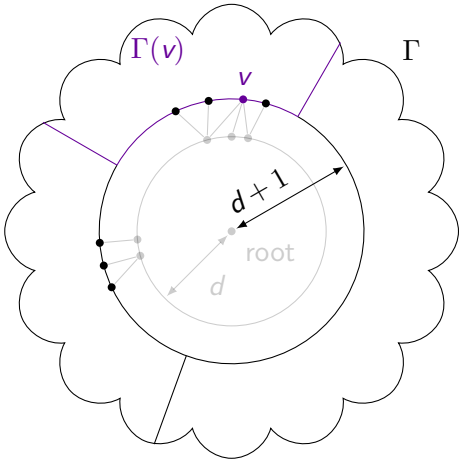


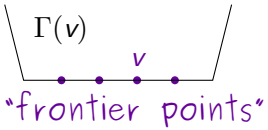
End-Cones and End-Isomorphisms



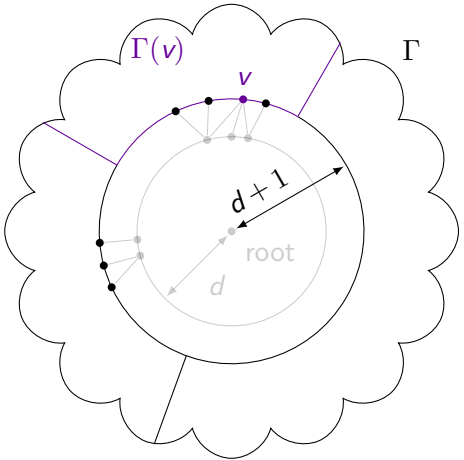


End-Cones and End-Isomorphisms





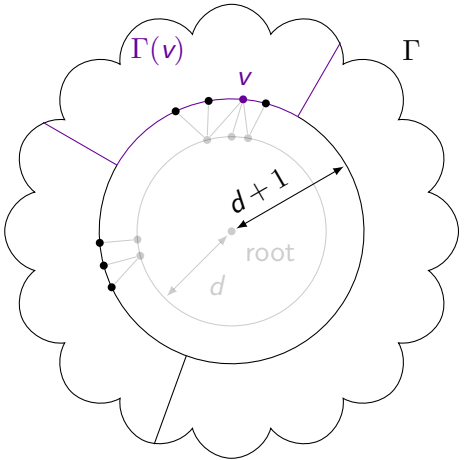
End-Cones and End-Isomorphisms



"end-cone" of v

"frontier points"

End-Cones and End-Isomorphisms



"end-cone" of v

"frontier points"

"end isomorphism":
maps frontier points to frontier points

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for
- finite alphabet A with

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for
- finite alphabet A with
- uniformly bounded degree and

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for
- finite alphabet A with
- uniformly bounded degree and
- finitely many end-isomorphism classes.

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for
- finite alphabet A with
- uniformly bounded degree and
- finitely many end-isomorphism classes.

Corollary (Muller, Schupp; 1985)

Being context-free is independent of the choice of the root.

Context-Free Graphs

Definition (Muller, Schupp; 1985)

A context-free graph is a

- rooted,
- connected and
- involutive
- A-graph for
- finite alphabet A with
- uniformly bounded degree and
- finitely many end-isomorphism classes.

Corollary (Muller, Schupp; 1985)

Being context-free is independent of the choice of the root.

Theorem (Muller, Schupp; 1985)

Γ *is context-free*

$\iff \Gamma = \Delta^{\pm 1}$ *for a configuration graph of a PDA*

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ *be finitely presented*

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ be finitely presented and $s \in M$. Then:

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ be finitely presented and $s \in M$. Then:
$S\Gamma(s)$ is tree-like

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ *be finitely presented and* $s \in M$. *Then:*
$S\Gamma(s)$ *is tree-like*

quasi-isometric to a tree

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ *be finitely presented and* $s \in M$. *Then:*

$$S\Gamma(s) \text{ is tree-like} \implies S\Gamma(s) \text{ is context-free}$$

quasi-isometric to a tree

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ *be finitely presented and* $s \in M$. *Then:*

$$S\Gamma(s) \text{ is tree-like} \implies S\Gamma(s) \text{ is context-free}$$

quasi-isometric to a tree

In fact: This is true for certain tree-like subgraphs of Schützenberger graphs as well!

Context-Free Graphs in the Wild

Context-free graphs appear naturally as

- Cayley graphs of context-free groups and
- Schützenberger graphs of some finitely presented inverse monoids.

Recall: the Schützenberger graph $S\Gamma(s)$ of s is
the strongly connected component of s in the Cayley graph.

Theorem (Gray, Silva, Szakács, 2022)

Let $M = \mathrm{Inv}\langle A \mid \ell_1 = r_1, \ldots, \ell_n = r_n \rangle$ *be finitely presented and* $s \in M$. *Then:*

$$S\Gamma(s) \text{ is tree-like} \implies S\Gamma(s) \text{ is context-free}$$

quasi-isometric to a tree

In fact: This is true for certain tree-like subgraphs of Schützenberger graphs as well!

These are inverse graphs!

Inverse Context-Free Graphs

Γ: inverse $A^{\pm 1}$-graph

Then:

Inverse Context-Free Graphs

Γ: inverse $A^{\pm 1}$-graph Then:

Γ is context-free $\iff$ Γ has finitely many end-isomorphism classes (definition)

$\iff$ Γ is the configuration graph of some PDA (Muller, Schupp, 1990)

Inverse Context-Free Graphs

Γ: inverse $A^{\pm 1}$-graph with root u Then:

$$\Gamma \text{ is context-free} \iff \Gamma \text{ has finitely many end-isomorphism classes} \quad \text{(definition)}$$
$$\iff \Gamma \text{ is the configuration graph of some PDA} \quad \text{(Muller, Schupp, 1990)}$$
$$\iff \mathscr{L}(\Gamma; u, u) \text{ is deterministic context-free} \quad \text{(Rodaro, 2024)}$$

Inverse Context-Free Graphs

Γ: inverse $A^{\pm 1}$-graph with root u Then:

Γ is context-free $\iff$ Γ has finitely many end-isomorphism classes (definition)

$\iff$ Γ is the configuration graph of some PDA (Muller, Schupp, 1990)

$\iff$ $\mathscr{L}(\Gamma; u, u)$ is deterministic context-free (Rodaro, 2024)

labels of cycles at u

Inverse Context-Free Graphs

Some details on ε-transitions and $\perp$ omitted

Γ: inverse $A^{\pm 1}$-graph with root u Then:

Γ is context-free $\iff$ Γ has finitely many end-isomorphism classes (definition)

$\iff$ Γ is the configuration graph of some PDA (Muller, Schupp, 1990)

$\iff$ $\mathscr{L}(\Gamma; u, u)$ is deterministic context-free (Rodaro, 2024)

labels of cycles at u

Inverse Context-Free Graphs

Some details on ε-transitions and $\perp$ omitted

Γ: inverse $A^{\pm 1}$-graph with root u Then:

$$\Gamma \text{ is context-free} \iff \Gamma \text{ has finitely many end-isomorphism classes} \quad \text{(definition)}$$
$$\iff \Gamma \text{ is the configuration graph of some PDA} \quad \text{(Muller, Schupp, 1990)}$$
$$\iff \mathscr{L}(\Gamma; u, u) \text{ is deterministic context-free} \quad \text{(Rodaro, 2024)}$$
$$\implies \Gamma \text{ is tree-like}$$

$\mathscr{L}(\Gamma; u, u)$: labels of cycles at u

Inverse Context-Free Graphs

Some details on ε-transitions and $\bot$ omitted

Γ: inverse $A^{\pm 1}$-graph with root u Then:

Γ is context-free $\iff$ Γ has finitely many end-isomorphism classes (definition)

$\iff$ Γ is the configuration graph of some PDA (Muller, Schupp, 1990)

$\iff$ $\mathscr{L}(\Gamma; u, u)$ is deterministic context-free (Rodaro, 2024)

$\iff$ Γ is tree-like

labels of cycles at u

"with some regularity condition"

Tree-Like Graphs

Tree-Like Graphs

A directed graph is tree-like

Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree

Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

Tree-Like Graphs

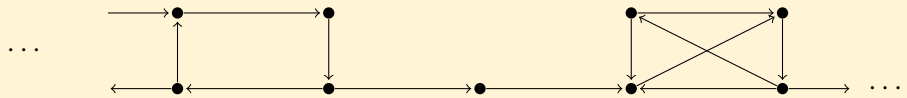
A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

Example

Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

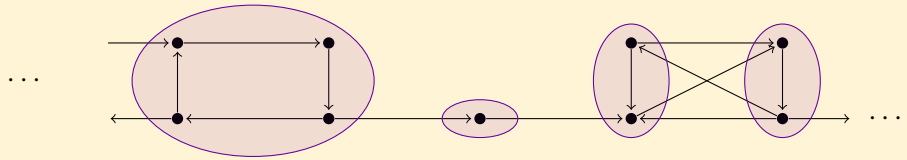
Example



Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

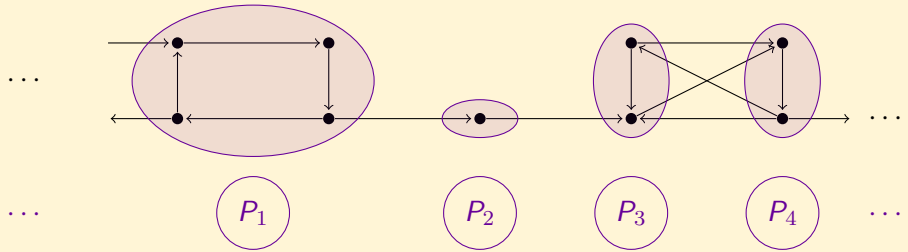
Example



Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

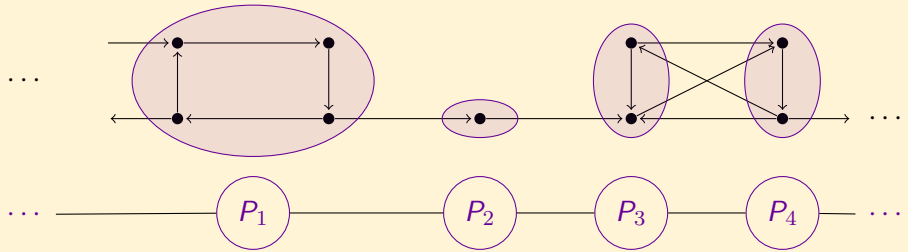
Example



Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

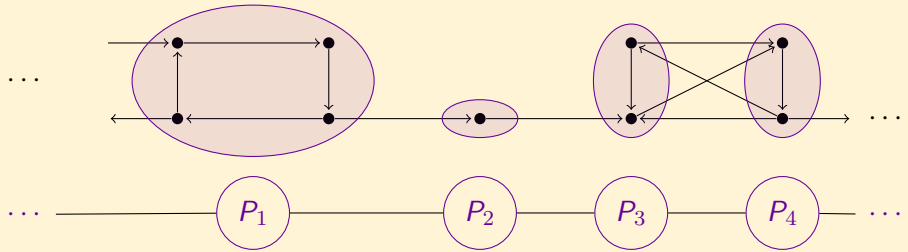
Example



Tree-Like Graphs

A directed graph is tree-like
$\iff$ it is quasi-isometric to a tree
$\iff$ it has a strong tree decomposition

Example



Idea: This tree needs to be "context-free" in some sense!

pDFAs Generating Trees

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

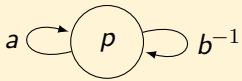
Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



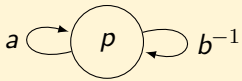
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

$p\ \varepsilon$ ← This is the root!



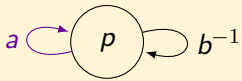
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

$p\ \varepsilon$ ← This is the root!

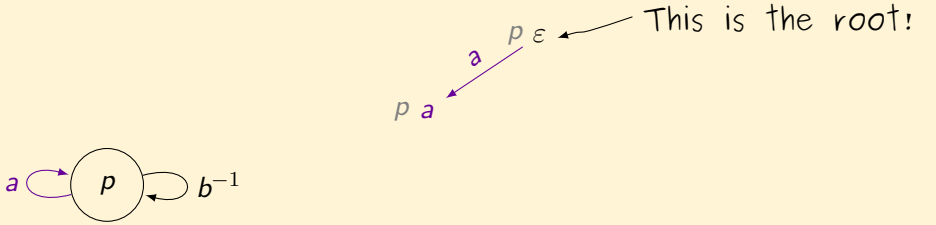


pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

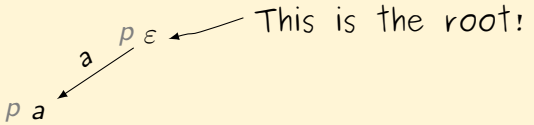


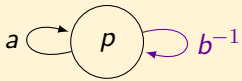
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



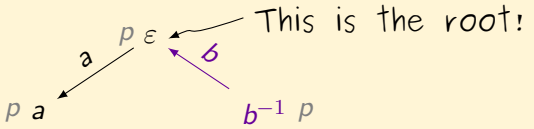


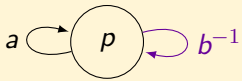
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



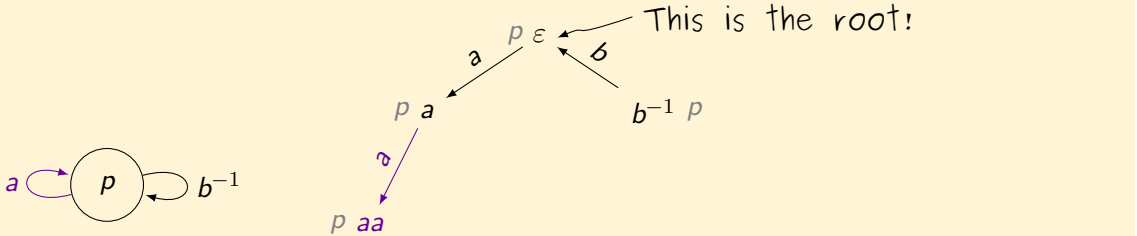


pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

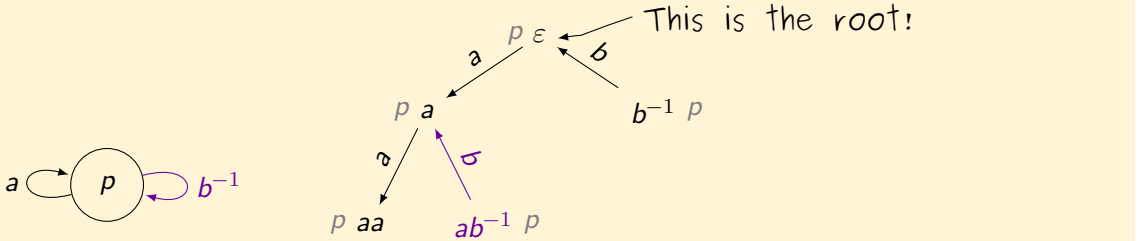


pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

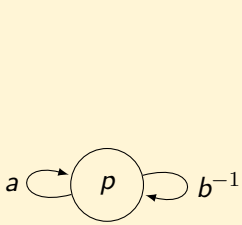


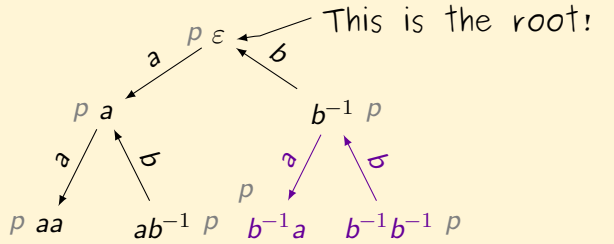
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



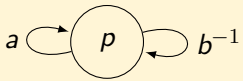


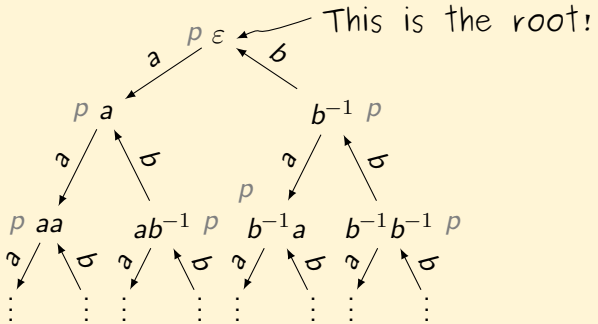
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



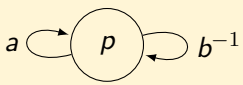


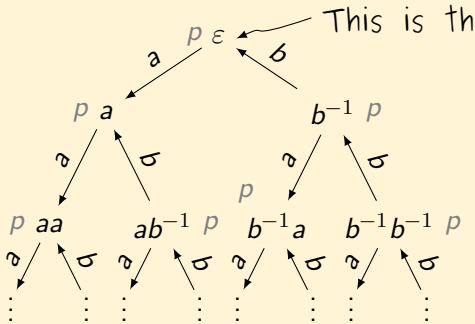
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





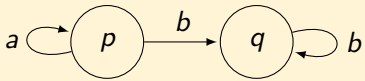
$\Gamma(p)$

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

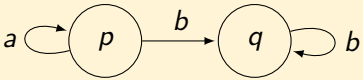


pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



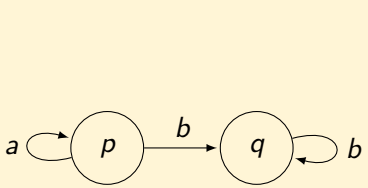
$\Gamma(p)$

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



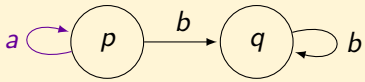
$\Gamma(p)$

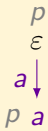
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





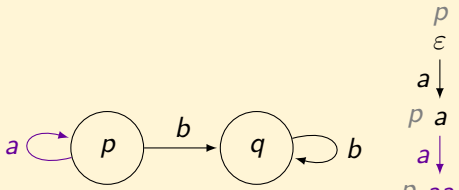
$\Gamma(p)$

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example



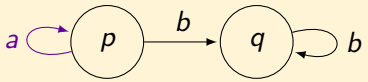
$\Gamma(p)$

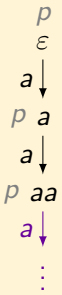
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





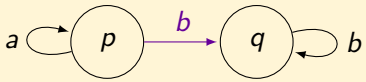
$\Gamma(p)$

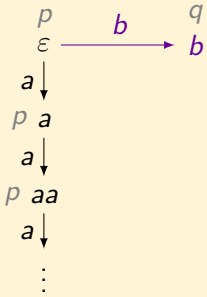
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





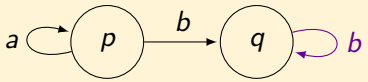
$\Gamma(p)$

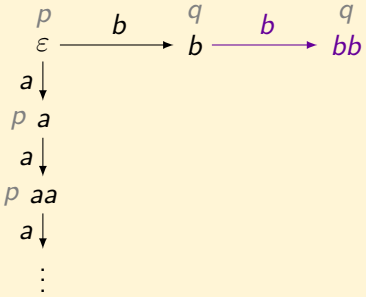
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





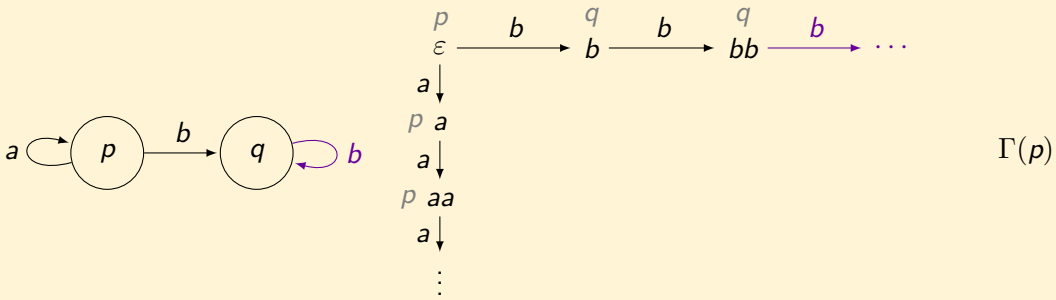
$\Gamma(p)$

pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example

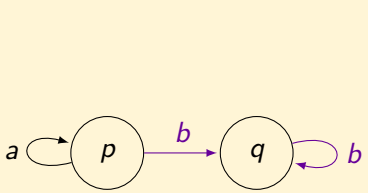


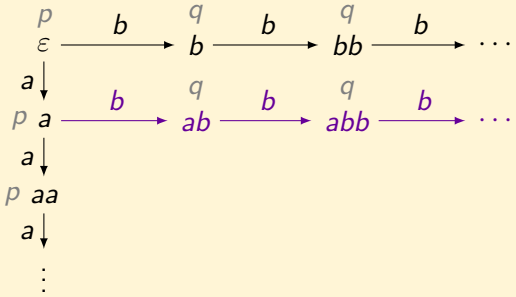
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





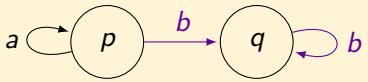
$\Gamma(p)$

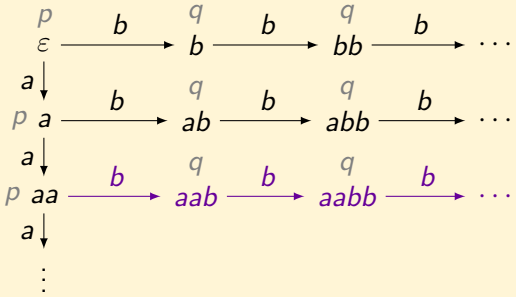
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





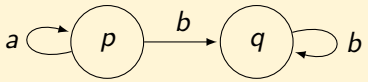
$\Gamma(p)$

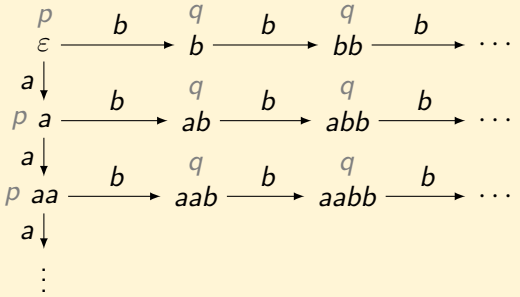
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





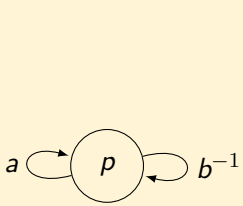
$\Gamma(p)$

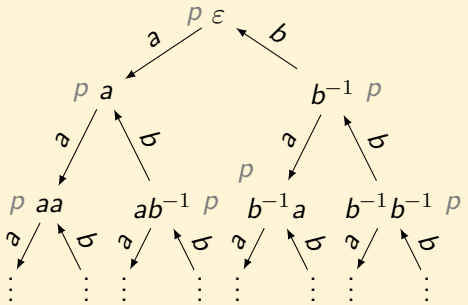
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





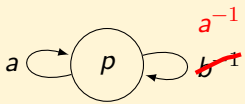
$\Gamma(p)$

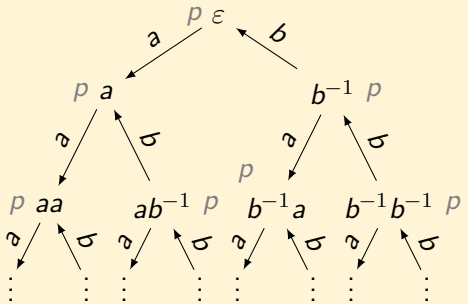
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





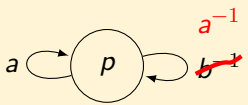
$\Gamma(p)$

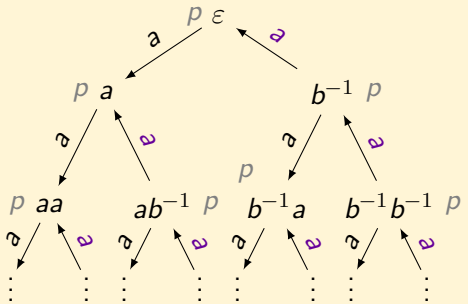
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





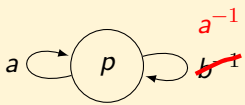
$\Gamma(p)$

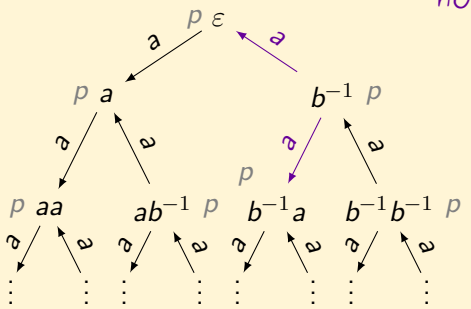
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





not deterministic!

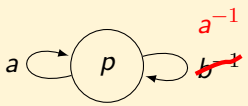
$\Gamma(p)$

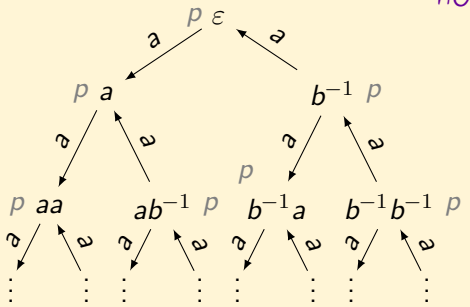
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





not deterministic!

$\Gamma(p)$

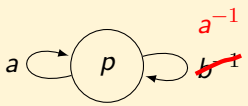
$\Gamma(p)$ is inverse and deterministic $\iff$ aa^{-1} cannot be read for any $a \in A^{\pm 1}$

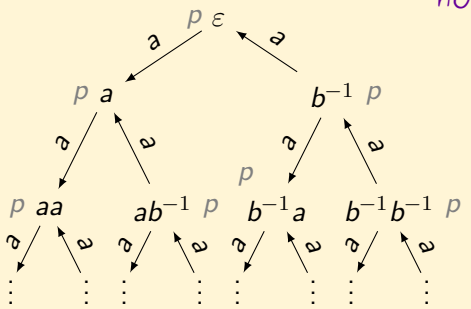
pDFAs Generating Trees

pDFA: partial deterministic finite automaton over $A^{\pm 1}$

Every state p of a pDFA generates an involutive tree $\Gamma(p)$ via its tree of runs/path space.

Example





not deterministic!

$\Gamma(p)$

$\Gamma(p)$ is inverse and deterministic $\iff$ aa^{-1} cannot be read for any $a \in A^{\pm 1}$

$\iff$: the pDFA is reduced

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free*

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

- free inverse monoids (Munn trees)

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

- free inverse monoids (Munn trees) and as Caley graphs of free groups.

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

- free inverse monoids (Munn trees) and as Caley graphs of free groups.
- inverse monoids with idempotent relators:

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

- free inverse monoids (Munn trees) and as Caley graphs of free groups.
- inverse monoids with idempotent relators:

Lemma (Margolis, Meakin, 1993)

$M = \mathrm{Inv}\langle A \mid \ell_i = r_i, i \in I \rangle$ *for* $\ell_i = 1 = r_i$ *in* $F(A)$

$\iff$ *all Schützenberger graphs of* M *are trees and, thus, context-free if* $|I| < \infty$

Context-Free Trees

Γ: inverse $A^{\pm 1}$-tree, i. e. a tree as an unlabeled, undirected graph

Theorem (W., arXiv 2026)

Γ *is context-free* $\iff$ Γ *arises as* $\Gamma(p)$ *for a state* p *of a reduced pDFA*

Context-free trees arise naturally as Schützenberger graphs of

- free inverse monoids (Munn trees) and as Caley graphs of free groups.
- inverse monoids with idempotent relators:

Lemma (Margolis, Meakin, 1993)

$M = \mathrm{Inv}\langle A \mid \ell_i = r_i, i \in I \rangle$ *for* $\ell_i = 1 = r_i$ *in* $F(A)$

$\iff$ *all Schützenberger graphs of* M *are trees and, thus, context-free if* $|I| < \infty$

What can we do with this?

Schützenberger Graphs and Green's Relations

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

Thus: We need to decide homomorphism/isomorphism problems!

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1} \iff ss^{-1} = tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t \iff s^{-1}s = t^{-1}t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

Thus: We need to decide homomorphism/isomorphism problems!

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1} \iff ss^{-1} = tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t \iff s^{-1}s = t^{-1}t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

We could use the word problem…

Thus: We need to decide homomorphism/isomorphism problems!

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1} \iff ss^{-1} = tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t \iff s^{-1}s = t^{-1}t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

We could use the word problem…

Thus: We need to decide homomorphism/isomorphism problems!

Theorem (Margolis, Meakin, 1993)

The word problem

Constant: $M = \mathrm{Inv}\langle A \mid \ell_i = r_i \rangle$ *for* $\ell_i = 1 = r_i$ *in* $F(A)$
Input: $u, v \in A^{\pm *}$
Question: *is* $u = v$ *in* M*?*

is decidable.

Schützenberger Graphs and Green's Relations

- $s \mathcal{R}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $ss^{-1} \mapsto tt^{-1} \iff ss^{-1} = tt^{-1}$
- $s \mathcal{L}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$ mapping $s \mapsto t \iff s^{-1}s = t^{-1}t$
- $s \mathcal{D}\ t \iff$ there is an isomorphism $S\Gamma(s) \to S\Gamma(t)$
- $s \geq_{\mathcal{J}} t \iff MsM \supseteq MtM$
  $\iff$ there is a homomorphism $S\Gamma(s) \to S\Gamma(t)$

We could use the word problem…

Thus: We need to decide homomorphism/isomorphism problems!

Theorem (Margolis, Meakin, 1993)

The word problem

Constant: $M = \mathrm{Inv}\langle A \mid \ell_i = r_i \rangle$ *for* $\ell_i = 1 = r_i$ *in* $F(A)$
Input: $u, v \in A^{\pm *}$
Question: *is* $u = v$ *in* M*?*

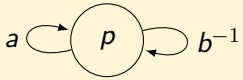
is decidable.

But: $\mathcal{D}$ and $\mathcal{J}$ remain open!

The Rooted Isomorphism Problem

The Rooted Isomorphism Problem

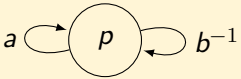
Example



The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

Example

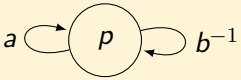


$\mathscr{L}(p) = \{a, b^{-1}\}^*$

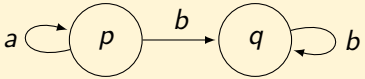
The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

Example



$\mathscr{L}(p) = \{a, b^{-1}\}^*$

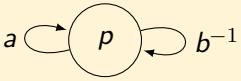


$\mathscr{L}(p) = a^* b^*$

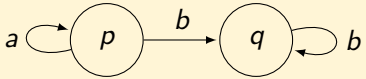
The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

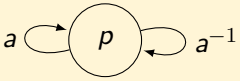
Example



$\mathscr{L}(p) = \{a, b^{-1}\}^*$



$\mathscr{L}(p) = a^* b^*$

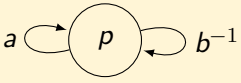


$\mathscr{L}(p) = \{a, a^{-1}\}^*$

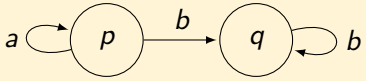
The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

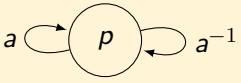
Example



$\mathscr{L}(p) = \{a, b^{-1}\}^*$



$\mathscr{L}(p) = a^*b^*$



$\mathscr{L}(p) = \{a, a^{-1}\}^*$

Fact

Let p and q be states of reduced pDFAs. Then:

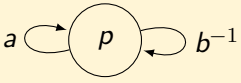
$$\Gamma(p) \doteq \Gamma(q) \iff \mathscr{L}(p) = \mathscr{L}(q)$$

isomorphic as **rooted** *graphs*

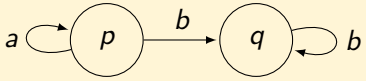
The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

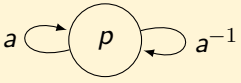
Example



$\mathscr{L}(p) = \{a, b^{-1}\}^*$



$\mathscr{L}(p) = a^*b^*$



$\mathscr{L}(p) = \{a, a^{-1}\}^*$

Fact

Let p and q be states of reduced pDFAs. Then:

$$\Gamma(p) \doteq \Gamma(q) \iff \mathscr{L}(p) = \mathscr{L}(q)$$

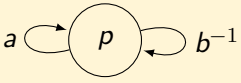
isomorphic as **rooted** *graphs*

It is well known that the latter is NL-complete.

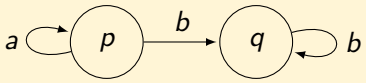
The Rooted Isomorphism Problem

$\mathscr{L}(p)$: words readable from state p in its pDFA

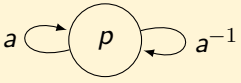
Example



$\mathscr{L}(p) = \{a, b^{-1}\}^*$



$\mathscr{L}(p) = a^*b^*$



$\mathscr{L}(p) = \{a, a^{-1}\}^*$

Fact

Let p and q be states of reduced pDFAs. Then:

$$\Gamma(p) \doteq \Gamma(q) \iff \mathscr{L}(p) = \mathscr{L}(q)$$

isomorphic as **rooted** *graphs*

It is well known that the latter is NL-complete.

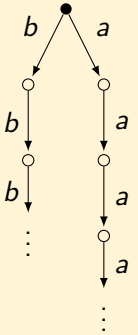
Adapting a construction by Cho & Huynh, we may assume every state to be accessible from p and q, respectively (for the lower bound).

The Non-Rooted Case: The Problem

The Non-Rooted Case: The Problem

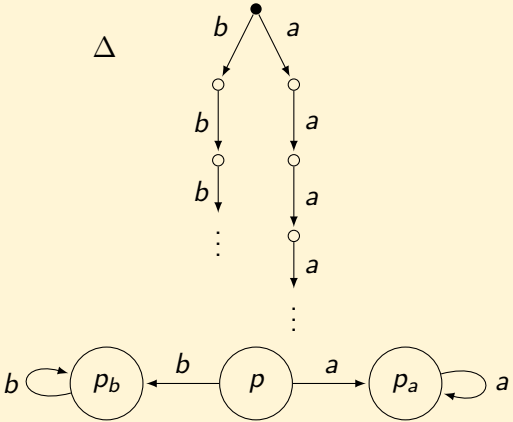
Example

Δ



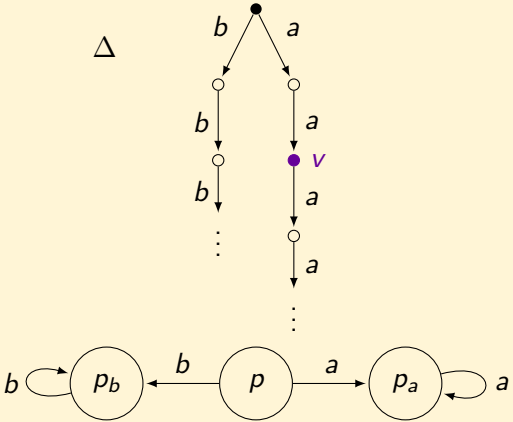
The Non-Rooted Case: The Problem

Example



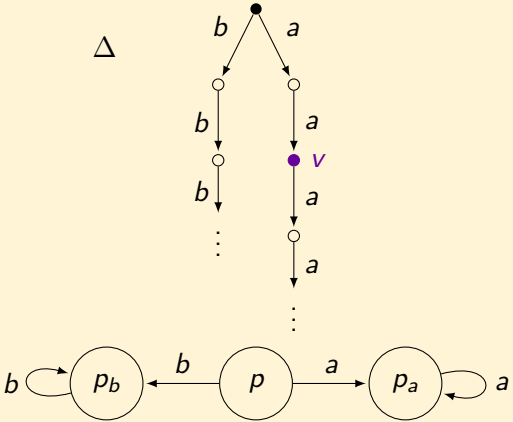
The Non-Rooted Case: The Problem

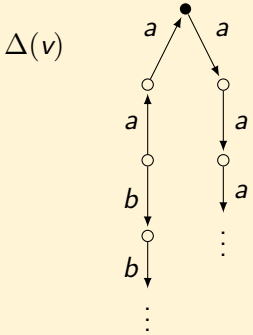
Example



The Non-Rooted Case: The Problem

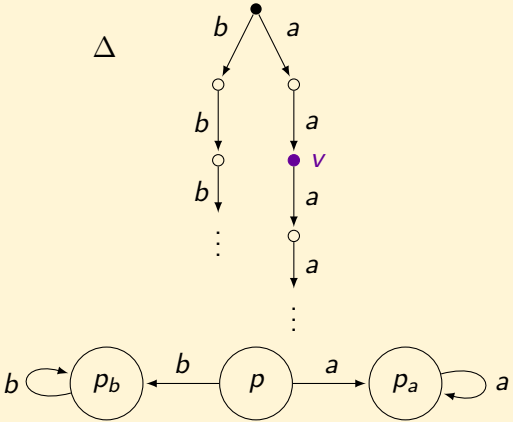
Example

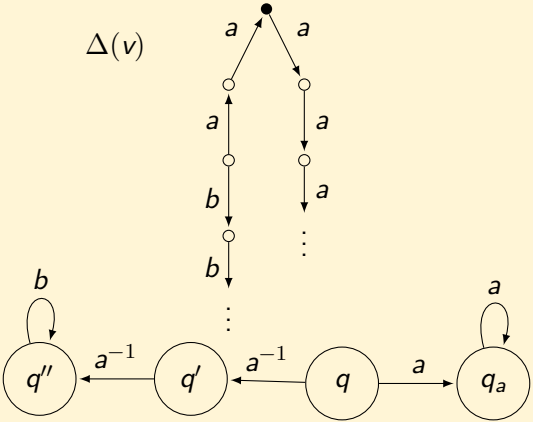




The Non-Rooted Case: The Problem

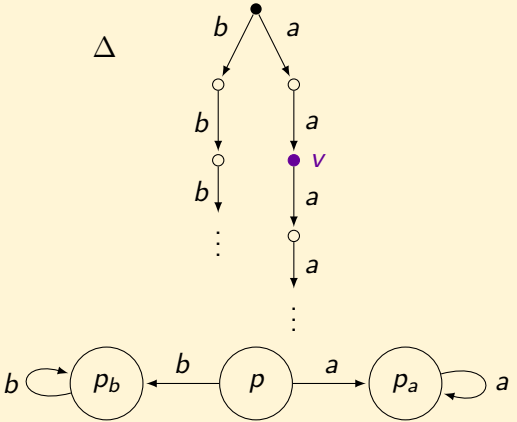
Example

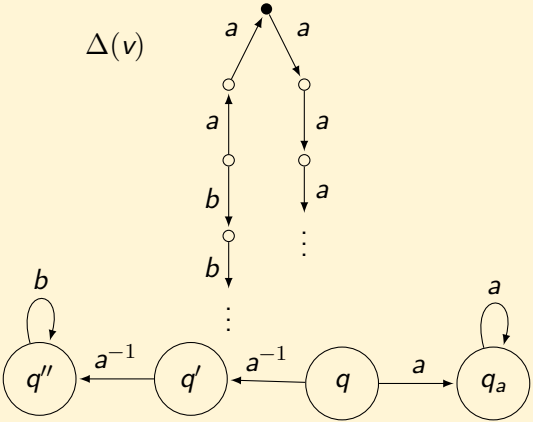




The Non-Rooted Case: The Problem

Example





The languages don't seem connected!

The Non-Rooted Isomorphism Problem

Theorem (W., arXiv 2026)

The non-rooted isomorphism problem for inverse context-free trees

Input: *states $\check{p}$ and $\check{q}$ of reduced pDFAs*
Question: *are $\Gamma(\check{p})$ and $\Gamma(\check{q})$ isomorphic as non-rooted graphs*

is NL*-complete*

The Non-Rooted Isomorphism Problem

Theorem (W., arXiv 2026)

The non-rooted isomorphism problem for inverse context-free trees

Input: *states $\check{p}$ and $\check{q}$ of reduced pDFAs*
Question: *are $\Gamma(\check{p})$ and $\Gamma(\check{q})$ isomorphic as non-rooted graphs*

is NL-complete and, thus, in polynomial time.

The Non-Rooted Isomorphism Problem

Theorem (W., arXiv 2026)

The non-rooted isomorphism problem for inverse context-free trees

Input: *states $\check{p}$ and $\check{q}$ of reduced pDFAs*
Question: *are $\Gamma(\check{p})$ and $\Gamma(\check{q})$ isomorphic as non-rooted graphs*

is NL-complete and, thus, in polynomial time.

The tricky part is "$\in \mathsf{NL} \subseteq \mathsf{P}$".

Proof Idea

Proof Idea

p: state of reduced pDFA $\mathcal{A}$,

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$,

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q)$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by q

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q}$,

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q},\ \Delta(v) = \Delta$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q},\ \Delta(v) = \Delta$, i.e. $U(p, \check{q})$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q}$, $\Delta(v) = \Delta$, i.e. $U(p, \check{q})$

$\Gamma(p) \doteq \Delta(v) = \Delta = \Gamma(\check{q})$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q}$, $\Delta(v) = \Delta$, i.e. $U(p, \check{q})$

$\Gamma(p) \doteq \Delta(v) = \Delta = \Gamma(\check{q}) \iff \mathscr{L}(p) = \mathscr{L}(\check{q})$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q}$, $\Delta(v) = \Delta$, i.e. $U(p, \check{q})$

$\Gamma(p) \doteq \Delta(v) = \Delta = \Gamma(\check{q}) \iff \mathscr{L}(p) = \mathscr{L}(\check{q})$

Thus: $U(p, q) \iff (q = \check{q}$ and $\mathscr{L}(p) = \mathscr{L}(\check{q}))$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is the root

$\implies q = \check{q}$, $\Delta(v) = \Delta$, i.e. $U(p, \check{q})$

$\Gamma(p) \doteq \Delta(v) = \Delta = \Gamma(\check{q}) \iff \mathscr{L}(p) = \mathscr{L}(\check{q})$

Thus: $U(p, q) \iff (q = \check{q}$ and $\mathscr{L}(p) = \mathscr{L}(\check{q}))$ or "v is not the root"

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

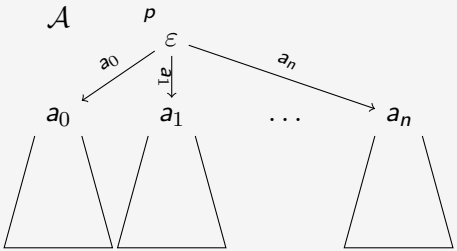
$\mathcal{A}$ p ε

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

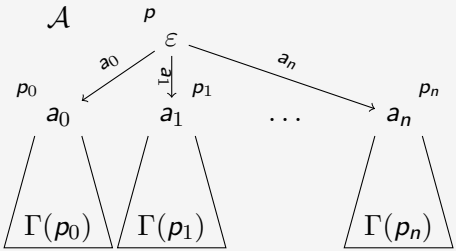


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

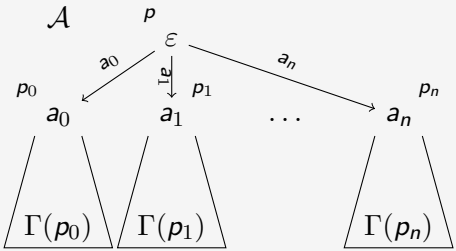


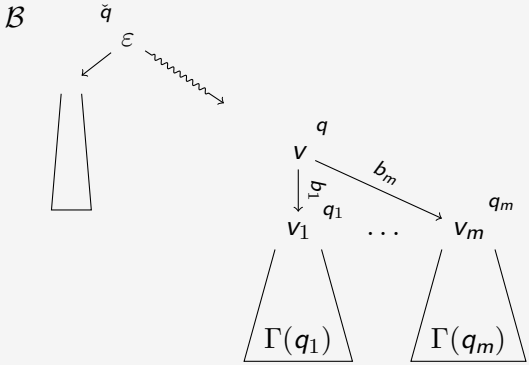
Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



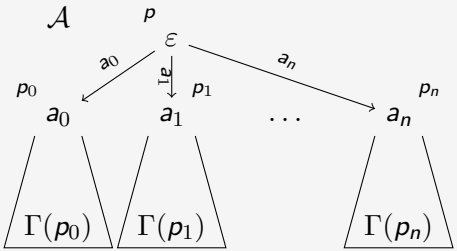


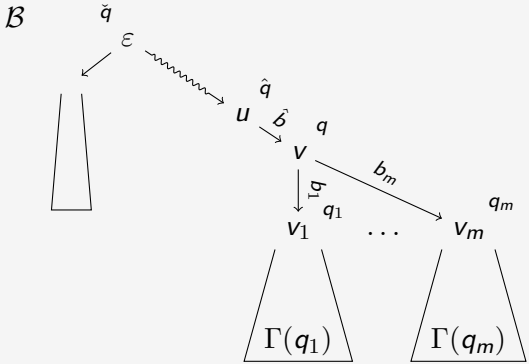
Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



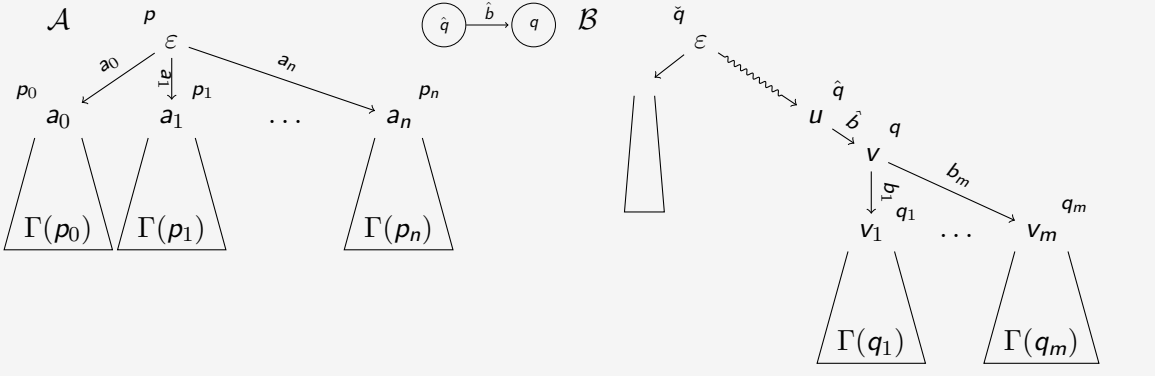


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

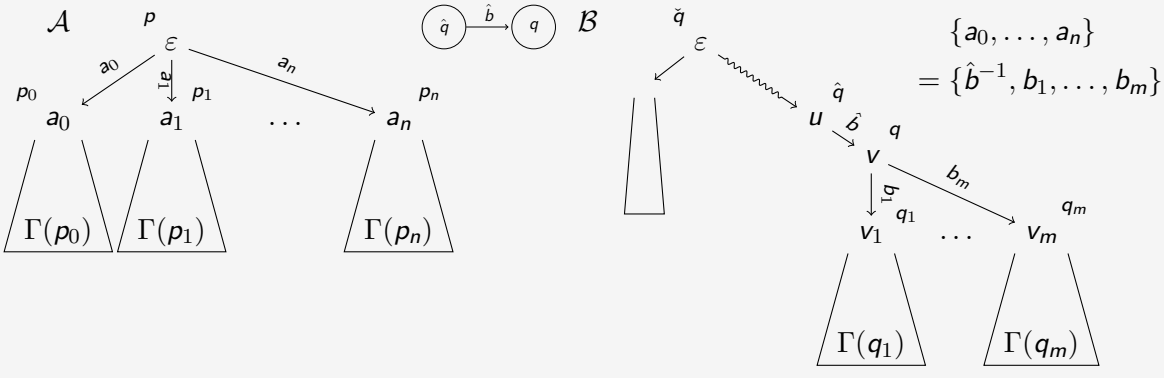


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

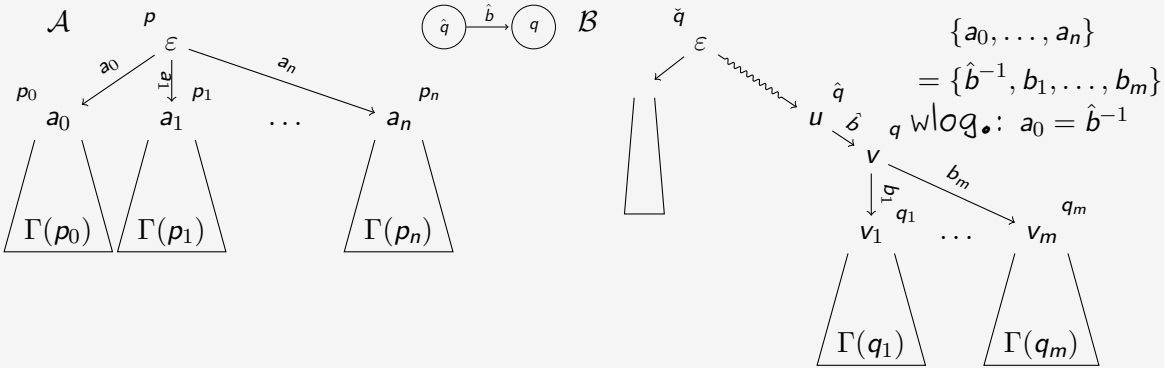


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

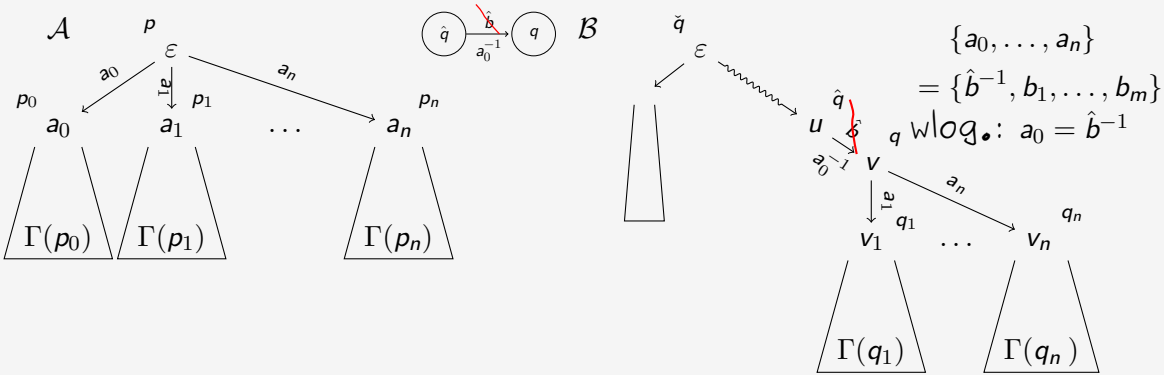


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

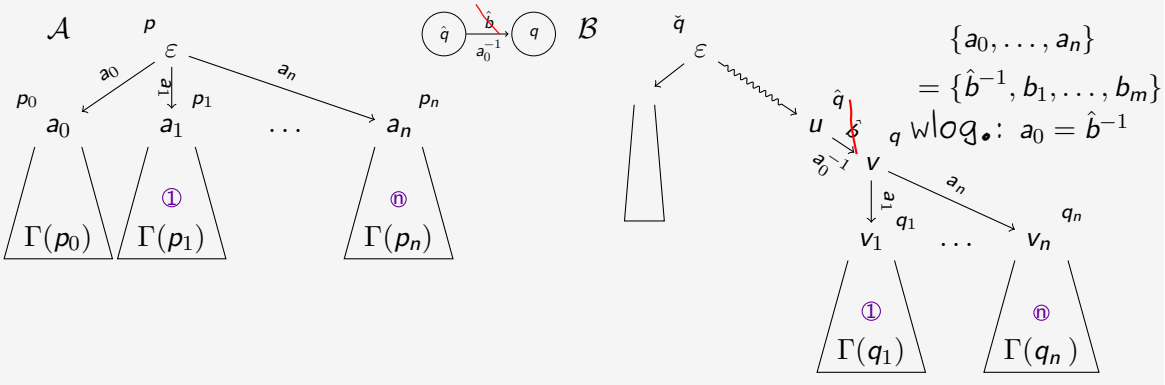


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root

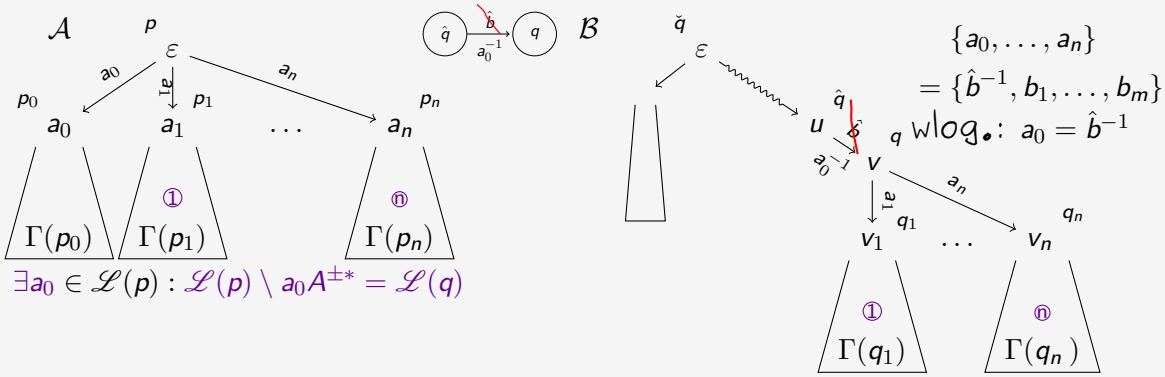


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



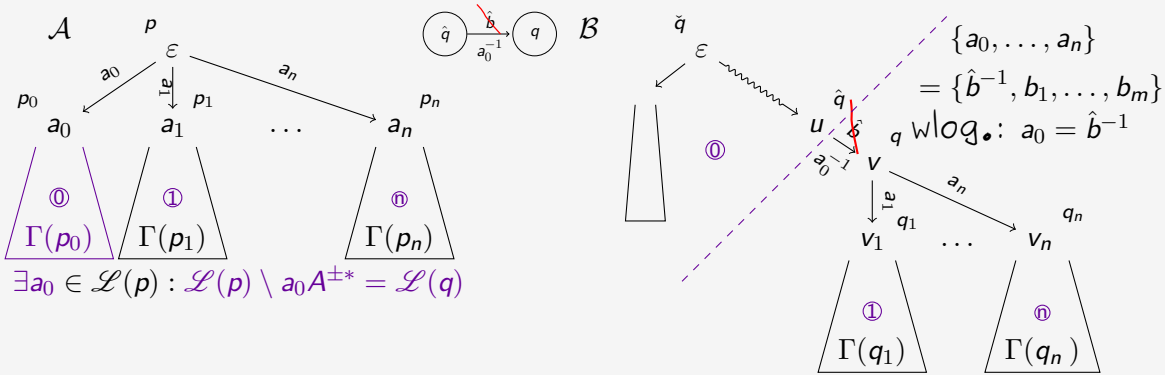
$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



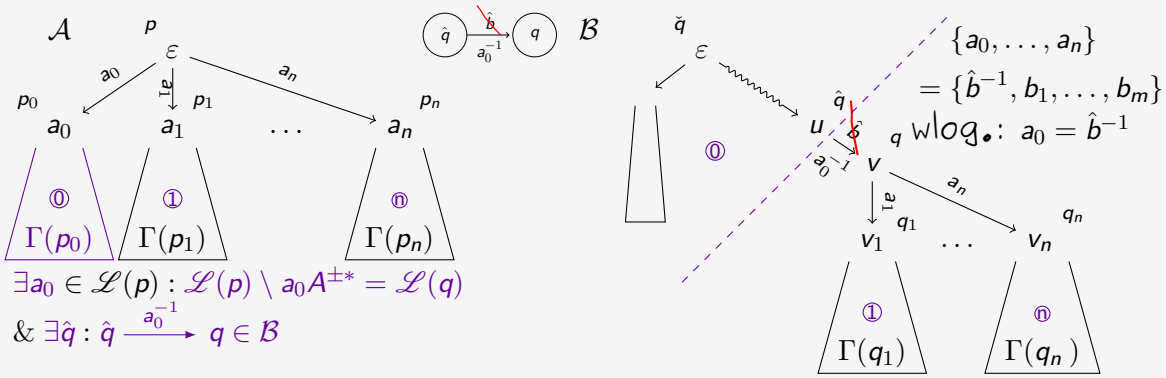
$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

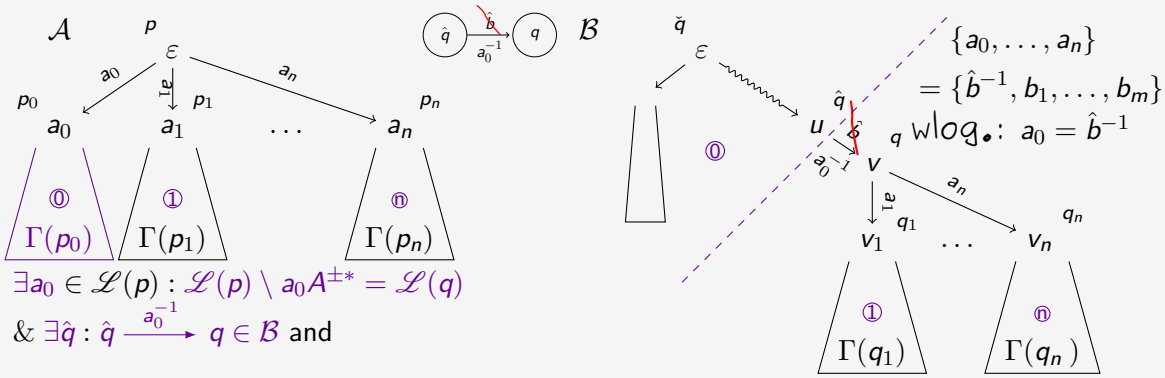
$\& \; \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B}$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v)$

Case: v is not the root



$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

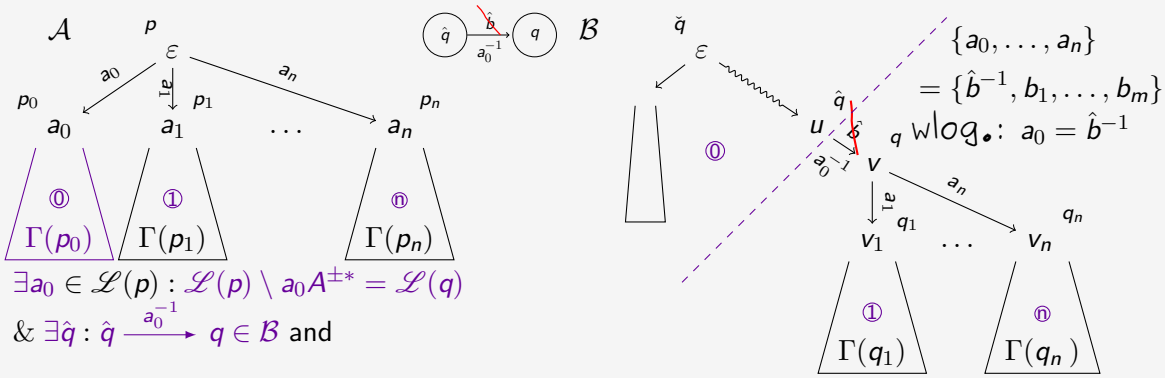
& $\exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B}$ and

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

Case: v is not the root



$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

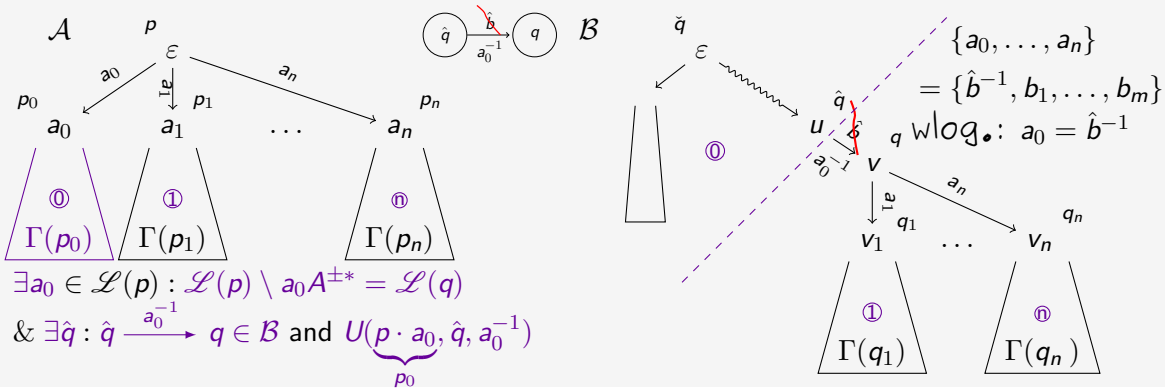
& $\exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B}$ and

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

Case: v is not the root



$\exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q)$

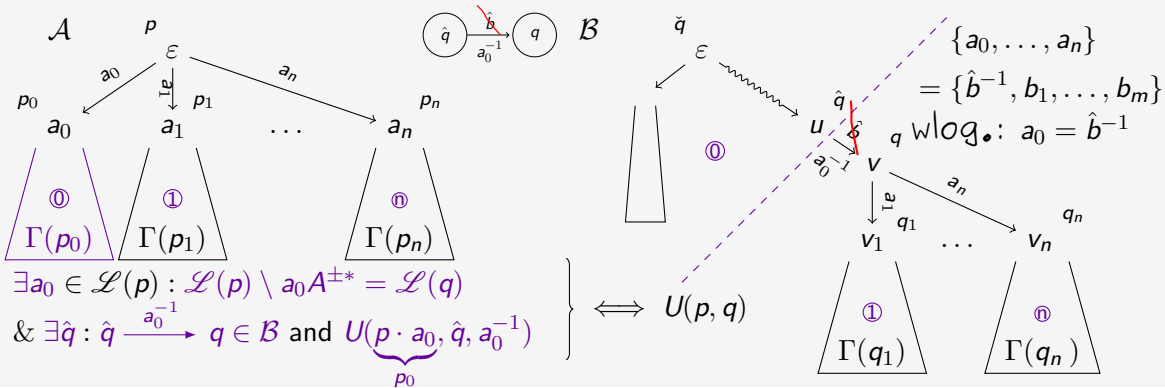
$\&\ \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B}$ and $U(\underbrace{p \cdot a_0}_{p_0}, \hat{q}, a_0^{-1})$

Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

Case: v is not the root

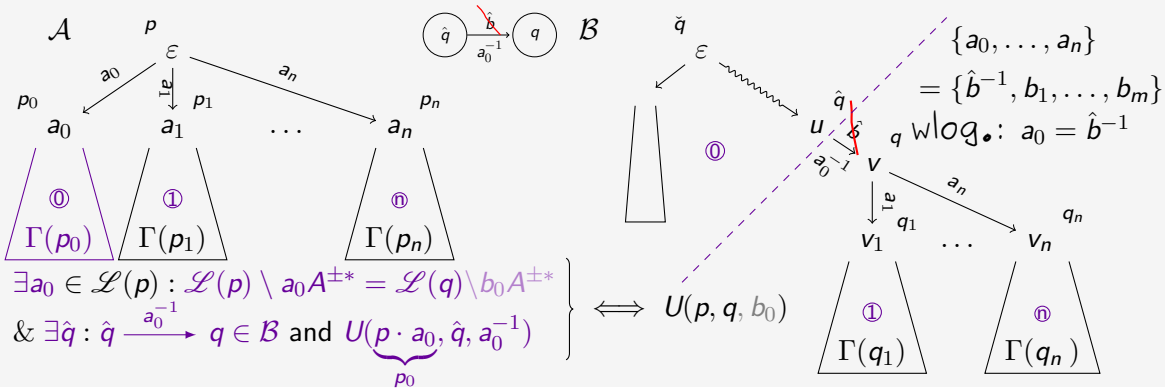


Proof Idea

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

Case: v is not the root



Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff$$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *}$$

or

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) :$$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \setminus b_0 A^{\pm *}$$

and

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \setminus b_0 A^{\pm *}$$
$$\text{and } \exists \hat{q} :$$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \backslash v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \backslash b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \backslash b_0 A^{\pm *}$$
$$\text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \;\&$$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \backslash v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \backslash b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \backslash b_0 A^{\pm *}$$
$$\text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \;\&\; U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *}$$
$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \setminus b_0 A^{\pm *}$$
$$\text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \;\&\; U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Algorithm:

- Construct a graph with nodes $U(p, q, b_0)$

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \backslash v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff \quad q = \check{q} \;\&\; \mathscr{L}(p) = \mathscr{L}(\check{q}) \backslash b_0 A^{\pm *} \quad \text{(c1)}$$

$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \backslash b_0 A^{\pm *}$$

$$\text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \;\&\; U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Algorithm:

- Construct a graph with nodes $U(p, q, b_0)$
- Mark $U(p, q, b_0)$ if (c1) holds.

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \backslash v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \,\&\, \mathscr{L}(p) = \mathscr{L}(\check{q}) \backslash b_0 A^{\pm *} \quad \text{(c1)}$$

$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \backslash b_0 A^{\pm *}$$

$$\text{(c2)} \quad \text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \,\&\, U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Algorithm:

- Construct a graph with nodes $U(p, q, b_0)$
- Mark $U(p, q, b_0)$ if (c1) holds.
- Add edge $U(p \cdot a_0, \hat{q}, a_0^{-1}) \longrightarrow U(p, q, b_0)$ if (c2) holds.

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \,\&\, \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *} \quad \text{(c1)}$$

$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \setminus b_0 A^{\pm *}$$

$$\text{(c2)} \quad \text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \,\&\, U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Algorithm:

- Construct a graph with nodes $U(p, q, b_0)$
- Mark $U(p, q, b_0)$ if (c1) holds.
- Add edge $U(p \cdot a_0, \hat{q}, a_0^{-1}) \longrightarrow U(p, q, b_0)$ if (c2) holds.

Summary

p: state of reduced pDFA $\mathcal{A}$, $\check{q}$: state of a reduced pDFA $\mathcal{B}$, $\Delta = \Gamma(\check{q})$

Predicate: $U(p, q, b_0) \iff \exists$ node $v \in \Delta$ labeled by $q : \Gamma(p) \doteq \Delta(v) \setminus v\, b_0 A^{\pm *}$

$$U(p, q, b_0) \iff q = \check{q} \,\&\, \mathscr{L}(p) = \mathscr{L}(\check{q}) \setminus b_0 A^{\pm *} \quad \text{(c1)}$$

$$\text{or} \quad \exists a_0 \in \mathscr{L}(p) : \mathscr{L}(p) \setminus a_0 A^{\pm *} = \mathscr{L}(q) \setminus b_0 A^{\pm *}$$

$$\text{(c2)} \quad \text{and } \exists \hat{q} : \hat{q} \xrightarrow{a_0^{-1}} q \in \mathcal{B} \,\&\, U(p \cdot a_0, \hat{q}, a_0^{-1})$$

Algorithm:

- Construct a graph with nodes $U(p, q, b_0)$
- Mark $U(p, q, b_0)$ if (c1) holds.
- Add edge $U(p \cdot a_0, \hat{q}, a_0^{-1}) \longrightarrow U(p, q, b_0)$ if (c2) holds.
- Check whether some $U(\check{p}, q)$ can be reached from a marked node.

Thank you!