FM Update

On the Satisfiability of Context-free String Constraints with Subword-Ordering

C. Aiswarya

CMI

Prakash Saivasan

IMSc

Soumodev Mal

CMI

accepted at LICS 2022

What is a String Constraint?

What is a String Constraint?

Set of variables V

What is a String Constraint?

Set of variables V

Alphabet A

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$\boxed{x \in L}$$

What is a String Constraint?

Set of variables V

Alphabet A

REGULAR

- **Membership Constraints**

$$x \in L$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

REGULAR

CONTEXT-FREE

⋮

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$\boxed{x \in L}$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$\boxed{x \in L}$$

$$L = a^n b^n$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$\boxed{x \in L}$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$\boxed{x \in L}$$

- **Relational Constraints**

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y$$

$$x \preceq yy$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y$$

$$x \preceq yy$$

...

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y$$

$$x \preceq yy$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz \qquad \texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y \qquad x \preceq yy$$

What is a String Constraint?

Set of variables V

Alphabet A

- **Membership Constraints** $x \in L$
- **Relational Constraints** $x = yz$, $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$, $T(x) = y$, $x \preceq yy$

$\bigwedge$

What is a String Constraint?

Set of variables V

Alphabet A

"Satisfiability"

- **Membership Constraints** $x \in L$

- **Relational Constraints** $x = yz$, $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$, $T(x) = y$, $x \preceq yy$

$$\bigwedge$$

Some known results

- **Membership Constraints** $x \in L$

- **Relational Constraints** $x = yz$ $\quad \texttt{Len}(x) = 3 \times \texttt{Len}(y)$

  $T(x) = y$ $\quad x \preceq yy$ ...

Some known results

- Membership Constraints

$x \in L$

- Relational Constraints

$x = yz$

$\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$

$x \preceq yy$

...

Some known results

- Membership Constraints

$x \in L$

- Relational Constraints

$x = yz$

$\mathrm{Len}(x) = 3 \times \mathrm{Len}(y)$

$T(x) = y$

$x \preceq yy$

...

Decidable! [Makanin 1977]

G S Makanin. The problem of solvability of equations in a free semigroup. Mathematics of the USSR-Sbornik, 32(2), 1977. doi: 10.1070/SM1977v032n02ABEH002376.

Some known results

- Membership Constraints

$x \in L$ ← **Regular**

- Relational Constraints

$x = yz$ $\quad$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $\quad$ $x \preceq yy$ $\quad$ ...

Some known results

- Membership Constraints

$x \in L$ ← **Regular**

- Relational Constraints

$x = yz$

$\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$

$x \preceq yy$

...

Decidable [Schulz 1992]

Klaus U. Schulz. Makanin's algorithm for word equations-two improvements and a gener- alization. In K. U. Schulz, editor, Word Equations and Related Topics, pages 85–150, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. ISBN 978-3-540-46737-3.

Some known results

- Membership Constraints

$x \in L$

- Relational Constraints

$x = yz$

$\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$

$x \preceq yy$

...

Some known results

- Membership Constraints

$x \in L$

- Relational Constraints

$x = yz$

$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$

$x \preceq yy$

...

Open!

Some known results

- Membership Constraints $x \in L$
- Relational Constraints $x = yz$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $x \preceq yy$ ...

Some known results

- Membership Constraints $x \in L$
- Relational Constraints $x = yz$ $\mathrm{Len}(x) = 3 \times \mathrm{Len}(y)$

$T(x) = y$ $x \preceq yy$ ...

Undecidable [Bjorner et. al. 2009]

Bjørner, N., Tillmann, N., Voronkov, A. (2009). Path Feasibility Analysis for String-Manipulating Programs. In: Kowalewski, S., Philippou, A. (eds) Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2009. Lecture Notes in Computer Science, vol 5505. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-00768-2_27

Some known results

- Membership Constraints $x \in L$
- Relational Constraints $x = yz$ $\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$ $x \preceq yy$ ...

Our Setting

- **Membership Constraints**

$$x \in L$$

- **Relational Constraints**

$$x = yz$$

$$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$$

$$T(x) = y$$

$$x \preceq yy$$

...

Our Setting

- **Membership Constraints** $x \in L$
- **Relational Constraints** $x = yz$ $\text{Len}(x) = 3 \times \text{Len}(y)$ $T(x) = y$ $x \preceq yy$ ...

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints**

$x = yz$ $\mathrm{Len}(x) = 3 \times \mathrm{Len}(y)$

$T(x) = y$ $x \preceq yy$ ...

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints**

$x = yz$

$\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$

$x \preceq \mathtt{Shuffle}(yy)$

...

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $x \preceq \texttt{Shuffle}(yy)$ **...**

`Shuffle`

Our Setting

- **Membership Constraints** $x \in L$ ⟵ **Context-free**

- **Relational Constraints** $x = yz$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

  $T(x) = y$ $x \preceq \texttt{Shuffle}(yy)$ **...**

`Shuffle`

u = abab

v = baabaa

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints**

$x = yz$ | $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ | $x \preceq \texttt{Shuffle}(yy)$ ...

`Shuffle`

u = abab

{ }

v = baabaa

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $x \preceq \texttt{Shuffle}(yy)$ ...

Shuffle

u = abab

{ abbaababaa ,

}

v = baabaa

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\quad$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $\quad$ $x \preceq \texttt{Shuffle}(yy)$ ...

Shuffle

u = abab

{ abbaababaa ,

baaabbabaa ,

}

v = baabaa

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\quad$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

  $T(x) = y$ $\quad$ $x \preceq \texttt{Shuffle}(yy)$ ...

`Shuffle`

u = abab

{ abbaababaa ,

baaabbabaa ,

... }

v = baabaa

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints**

$x = yz$

$\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$

$x \preceq \texttt{Shuffle}(yy)$

...

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**
- **Relational Constraints** $x = yz$ $\quad \texttt{Len}(x) = 3 \times \texttt{Len}(y)$ $\quad T(x) = y$ $\quad x \preceq \texttt{Shuffle}(yy)$ ...

Subword

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

  $T(x) = y$ $x \preceq \texttt{Shuffle}(yy)$ ...

Subword $\preceq$

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints** $x = yz$ $\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$ $x \preceq \mathtt{Shuffle}(yy)$ **...**

abbaaababab

Subword $\preceq$

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**
- **Relational Constraints**

$x = yz$ $\quad$ $\mathtt{Len}(x) = 3 \times \mathtt{Len}(y)$

$T(x) = y$ $\quad$ $x \preceq \mathtt{Shuffle}(yy)$ ...

abbaaababab

Subword $\preceq$

bbabb

Our Setting

- **Membership Constraints** $x \in L$ ← **Context-free**

- **Relational Constraints**

$x = yz$ $\quad$ $\texttt{Len}(x) = 3 \times \texttt{Len}(y)$

$T(x) = y$ $\quad$ $x \preceq \texttt{Shuffle}(yy)$ $\quad$ **...**

Subword $\preceq$

abbaaababab

bbabb

Satisfiability is Undecidable!

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

$V = \{x, y\}$ Alphabet A

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

$V = \{x, y\}$ Alphabet A

$x \in L_1$ $y \in L_2$

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

$V = \{x, y\}$	Alphabet A
$x \in L_1$	$y \in L_2$
$x \preceq y$	$y \preceq x$

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

$V = \{x, y\}$ Alphabet A

$x \in L_1$ $y \in L_2$

$x \preceq y$ $y \preceq x$ $\implies x = y$

Satisfiability is Undecidable!

Intersection non-emptiness of CFL

$V = \{x, y\}$ Alphabet A

$x \in L_1$ $y \in L_2$

$x \preceq y$ $y \preceq x$ $\implies x = y$

Undecidable!

Satisfiability is Undecidable!

Even for regular membership

Satisfiability is Undecidable!

Even for regular membership

Reduction from PCP

Satisfiability is Undecidable!

Even for regular membership

Reduction from PCP

Satisfiability is Undecidable!

Even for regular membership

Reduction from PCP

PCP instance over alphabet Σ:

Satisfiability is Undecidable!

PCP instance over alphabet Σ:

Satisfiability is Undecidable!

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

Satisfiability is Undecidable!

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

Satisfiability is Undecidable!

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \; i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \quad i_1, i_2, \ldots, i_k$

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$$

$$=$$

$$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

sequence of indices

?

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$$

=

$$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

?

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \; i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	3	4
U	baab	ab	bab	ba
V	b	aa	babb	aba

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	3	4
U	baab	ab	bab	ba
V	b	aa	babb	aba

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i \in (1 + 2 + 3 + 4)^*$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	3	4
U	baab	ab	bab	ba
V	b	aa	babb	aba

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$x_i \in (1 + 2 + 3 + 4)^*$

$x_s \in \Sigma^*$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	3	4
U	baab	ab	bab	ba
V	b	aa	babb	aba

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i \in (1 + 2 + 3 + 4)^*$$
$$x_s \in \Sigma^*$$
$$x_u \in (1.baab + 2.ab + 3.bab + 4.ba)^*$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	3	4
U	baab	ab	bab	ba
V	b	aa	babb	aba

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i \in (1 + 2 + 3 + 4)^*$$
$$x_s \in \Sigma^*$$
$$x_u \in (1.baab + 2.ab + 3.bab + 4.ba)^*$$
$$x_v \in (1.b + 2.aa + 3.babb + 4.aba)^*$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \; i_1, i_2, \ldots, i_k$

sequence of indices

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k} \; ?$$

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j \right)^* \qquad x_s = \Sigma^*$$

$$x_u = \left(\bigcup_{j \in [n]} j.u_j \right)^* \qquad x_v = \left(\bigcup_{j \in [n]} j.v_j \right)^*$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \quad i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j\right)^*$$

$$x_s = \Sigma^*$$

$$x_u = \left(\bigcup_{j \in [n]} j.u_j\right)^*$$

$$x_v = \left(\bigcup_{j \in [n]} j.v_j\right)^*$$

$$x_i \preceq x_u$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists \quad i_1, i_2, \ldots, i_k$

sequence of indices

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j \right)^* \qquad x_s = \Sigma^*$$

$$x_u = \left(\bigcup_{j \in [n]} j.u_j \right)^* \qquad x_v = \left(\bigcup_{j \in [n]} j.v_j \right)^*$$

$$x_i \preceq x_u \qquad x_s \preceq x_u$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j\right)^* \qquad x_s = \Sigma^*$$

$$x_u = \left(\bigcup_{j \in [n]} j.u_j\right)^* \qquad x_v = \left(\bigcup_{j \in [n]} j.v_j\right)^*$$

$$x_i \preceq x_u \qquad x_s \preceq x_u \qquad x_u \preceq \texttt{Shuffle}(x_i, x_s)$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$x_i = \left(\bigcup_{j \in [n]} j \right)^*$ $\qquad$ $x_s = \Sigma^*$

$x_u = \left(\bigcup_{j \in [n]} j.u_j \right)^*$ $\qquad$ $x_v = \left(\bigcup_{j \in [n]} j.v_j \right)^*$

$x_i \preceq x_u$ $\qquad$ $x_s \preceq x_u$ $\qquad$ $x_u \preceq \texttt{Shuffle}(x_i, x_s)$

$x_i \preceq x_v$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$ = $v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j\right)^* \qquad x_s = \Sigma^*$$

$$x_u = \left(\bigcup_{j \in [n]} j.u_j\right)^* \qquad x_v = \left(\bigcup_{j \in [n]} j.v_j\right)^*$$

$$x_i \preceq x_u \qquad x_s \preceq x_u \qquad x_u \preceq \texttt{Shuffle}(x_i, x_s)$$

$$x_i \preceq x_v \qquad x_s \preceq x_v$$

Satisfiability is Undecidable!

Reduction from PCP

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

$\exists$ $i_1, i_2, \ldots, i_k$

sequence of indices

$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$

=

$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

?

Regular String Constraint:

Variable set: $\{x_i, x_s, x_u, x_v\}$

$$x_i = \left(\bigcup_{j \in [n]} j \right)^*$$

$$x_s = \Sigma^*$$

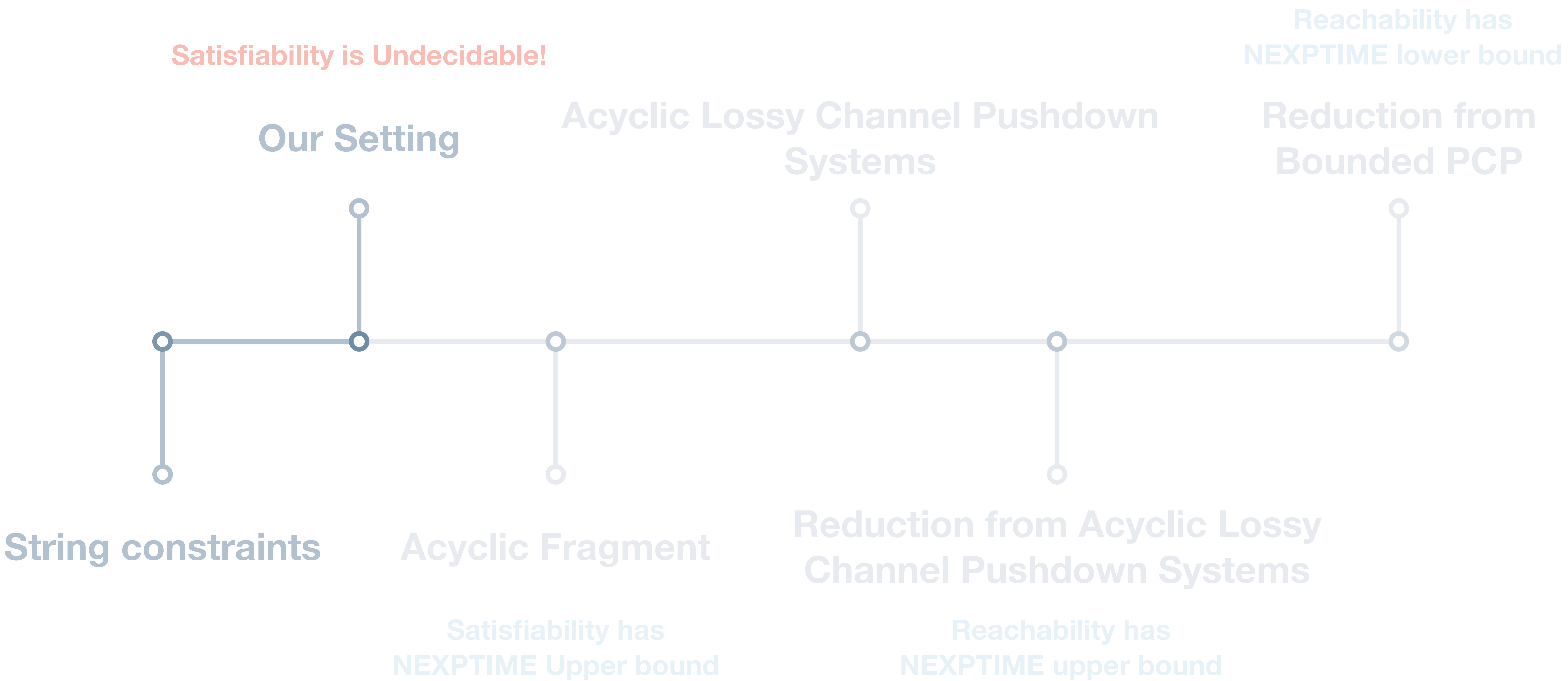
$$x_u = \left(\bigcup_{j \in [n]} j.u_j \right)^*$$

$$x_v = \left(\bigcup_{j \in [n]} j.v_j \right)^*$$

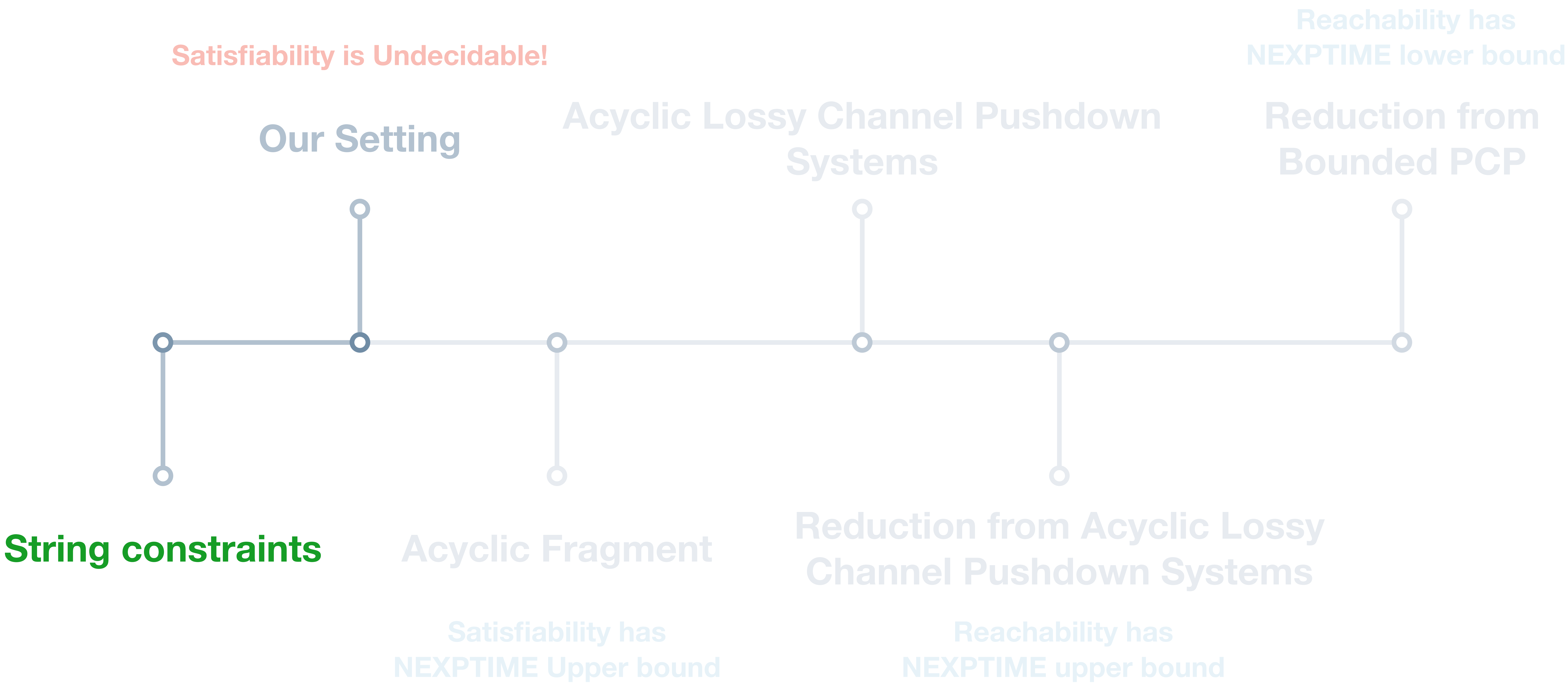
$x_i \preceq x_u$ $\quad$ $x_s \preceq x_u$ $\quad$ $x_u \preceq \texttt{Shuffle}(x_i, x_s)$

$x_i \preceq x_v$ $\quad$ $x_s \preceq x_v$ $\quad$ $x_v \preceq \texttt{Shuffle}(x_i, x_s)$

Outline

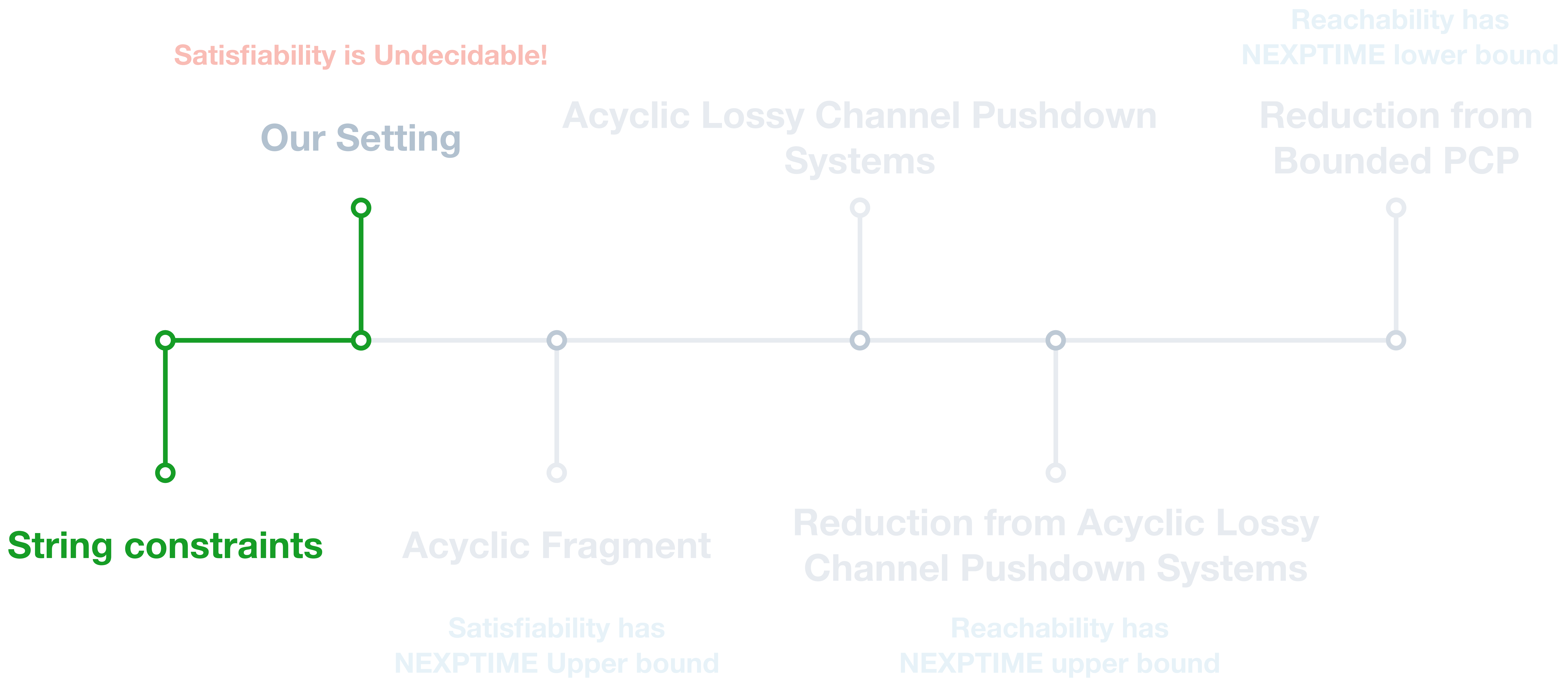

Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME upper bound

Outline



Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME upper bound

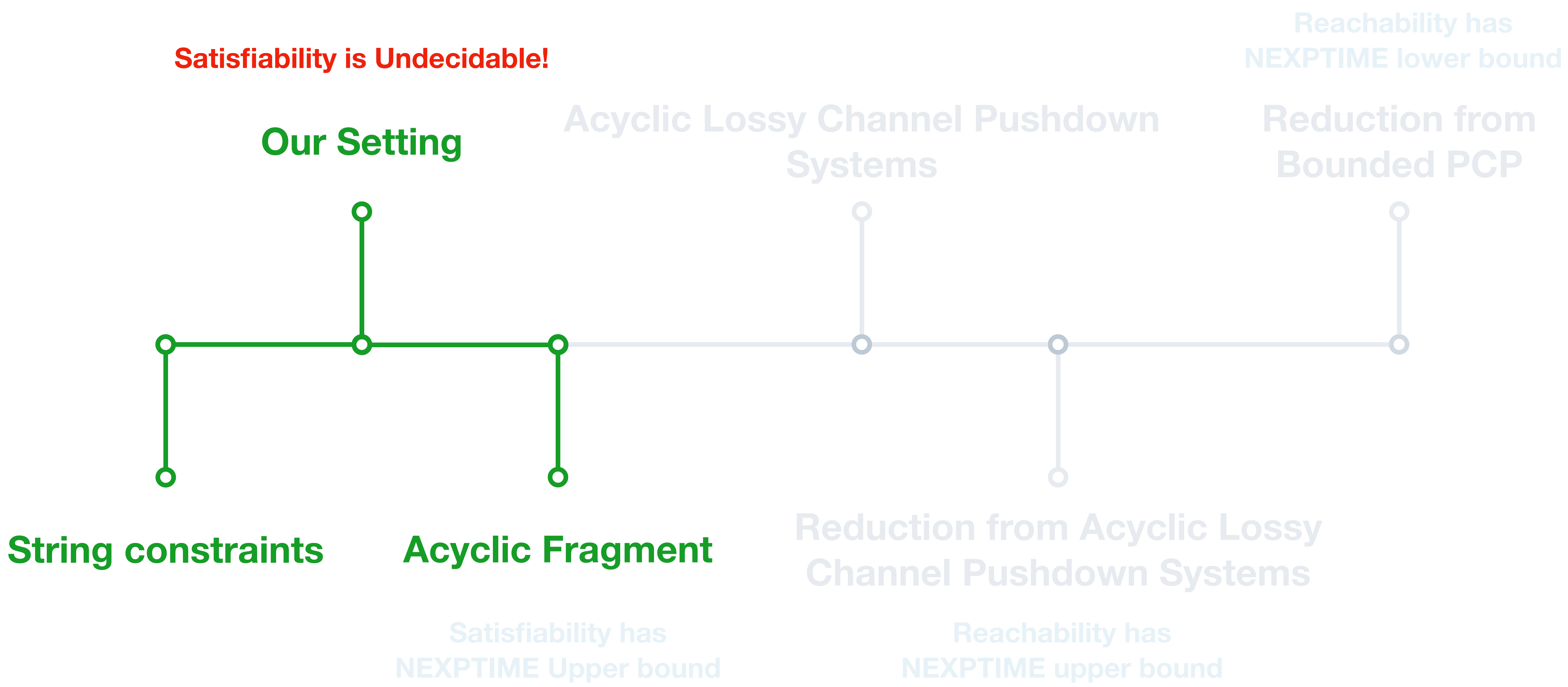
Outline


Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME upper bound

Outline

Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME upper bound

Outline


Reachability has
NEXPTIME lower bound
Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown
Systems
Reduction from
Bounded PCP
String constraints
Acyclic Fragment
Reduction from Acyclic Lossy
Channel Pushdown Systems
Satisfiability has
NEXPTIME Upper bound
Reachability has
NEXPTIME upper bound

Acyclic Fragment

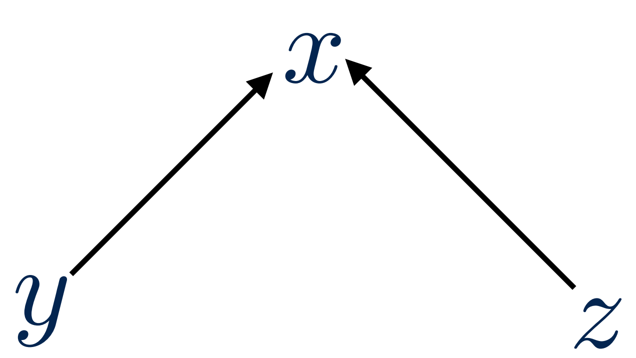
Acyclic Fragment

Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

Acyclic Fragment

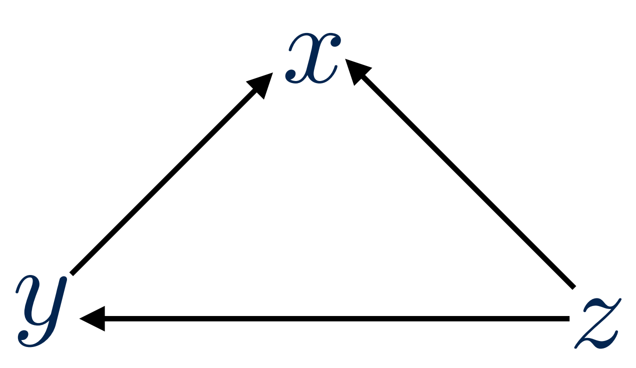
$x \preceq \texttt{Shuffle}(yz)$



Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

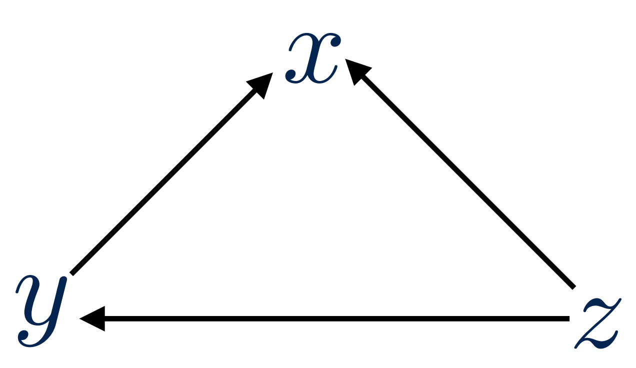
$y \preceq z$



Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$

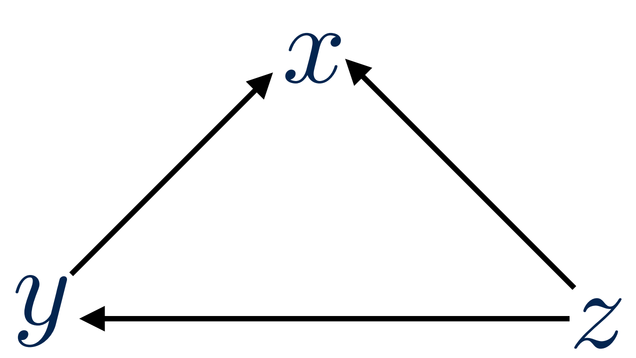


Acyclic!

Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



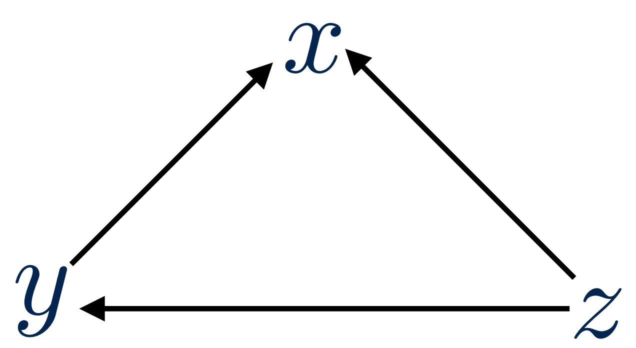
✔

Acyclic!

Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$

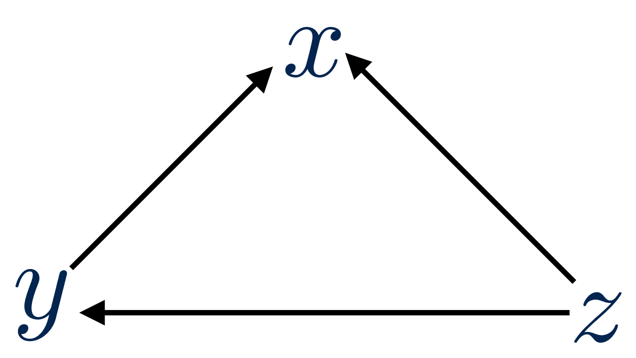


Acyclic!

Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



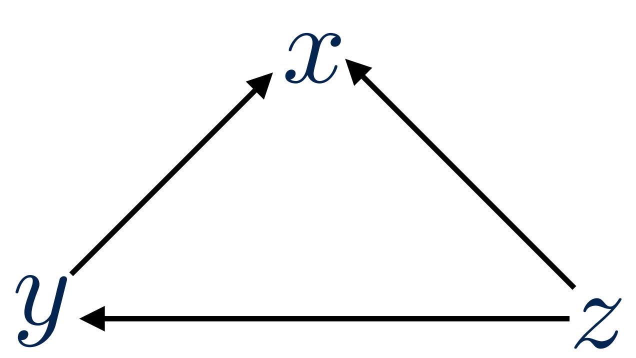
Acyclic!

$x \preceq \texttt{Shuffle}(yz)$

Acyclic Fragment

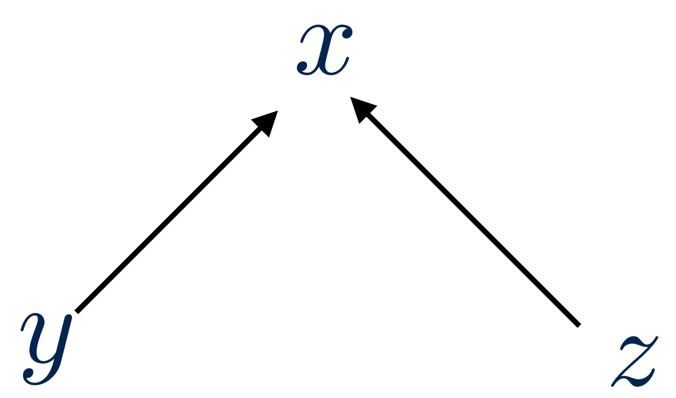
$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

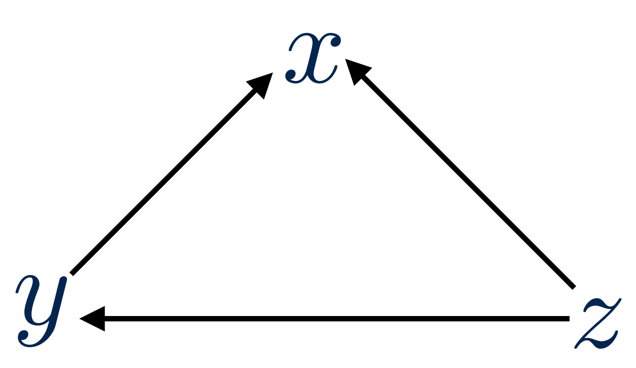
$x \preceq \texttt{Shuffle}(yz)$



Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

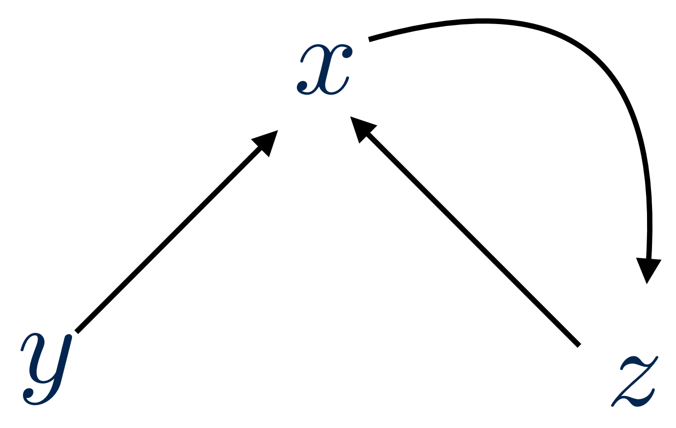
$y \preceq z$



Acyclic!

$x \preceq \texttt{Shuffle}(yz)$

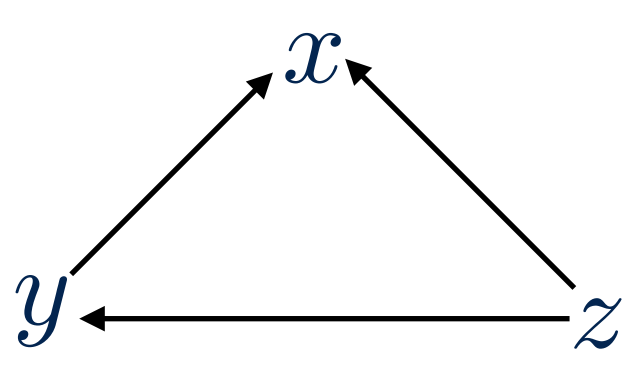
$z \preceq x$



Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

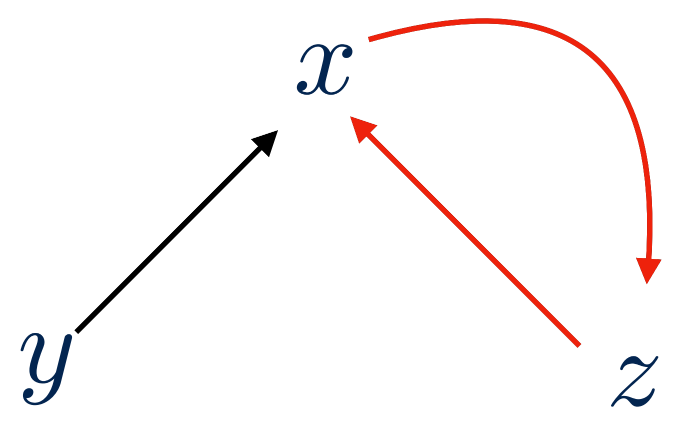
$y \preceq z$



Acyclic!

$x \preceq \texttt{Shuffle}(yz)$

$z \preceq x$

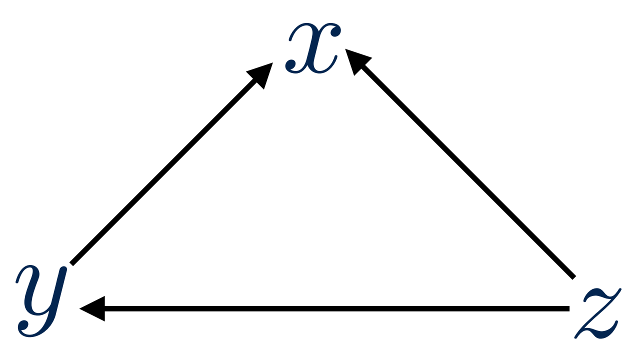


Cycle!

Acyclic Fragment

$x \preceq \texttt{Shuffle}(yz)$

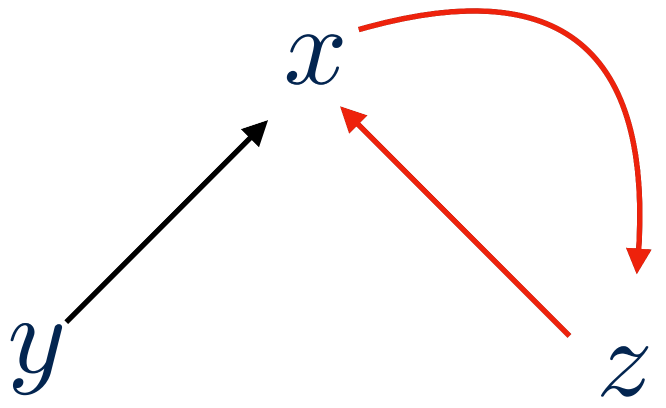
$y \preceq z$



Acyclic!

$x \preceq \texttt{Shuffle}(yz)$

$z \preceq x$



Cycle!

Main Result

Main Result

Satisfiability of Acyclic Fragment is NEXPTIME Complete!

NEXPTIME Upper bound

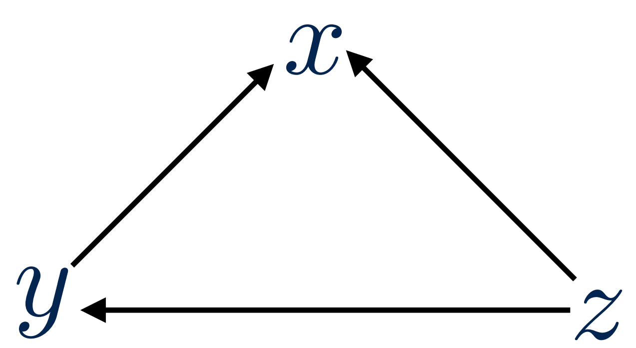
NEXPTIME Upper bound

small model property.

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



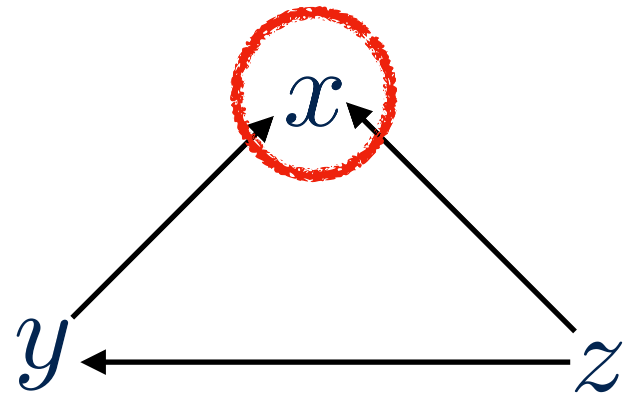
Acyclic!

small model property.

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



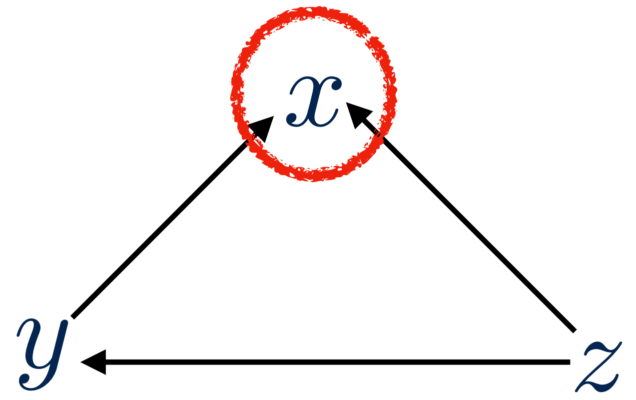
Acyclic!

small model property.

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

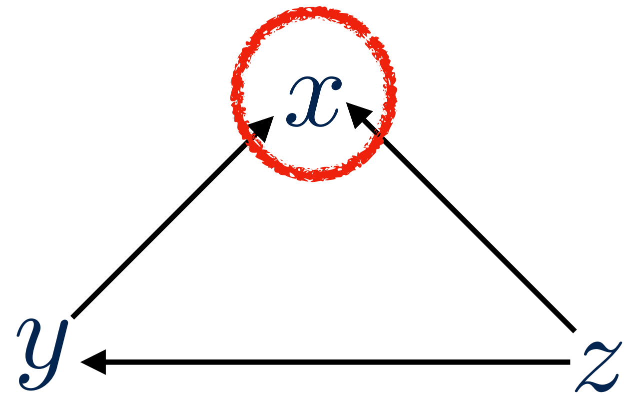
small model property.

Grammar in CNF form

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

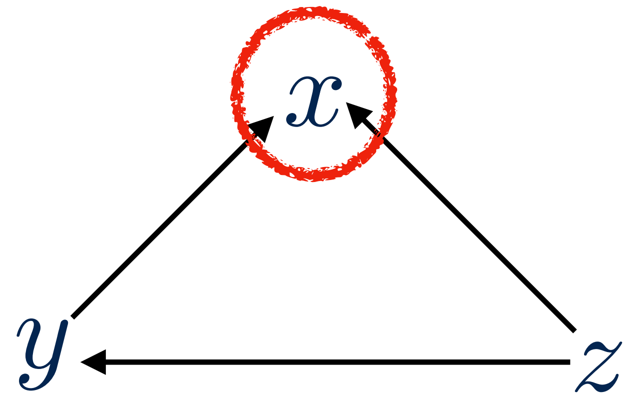
Grammar in CNF form

Parse tree

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

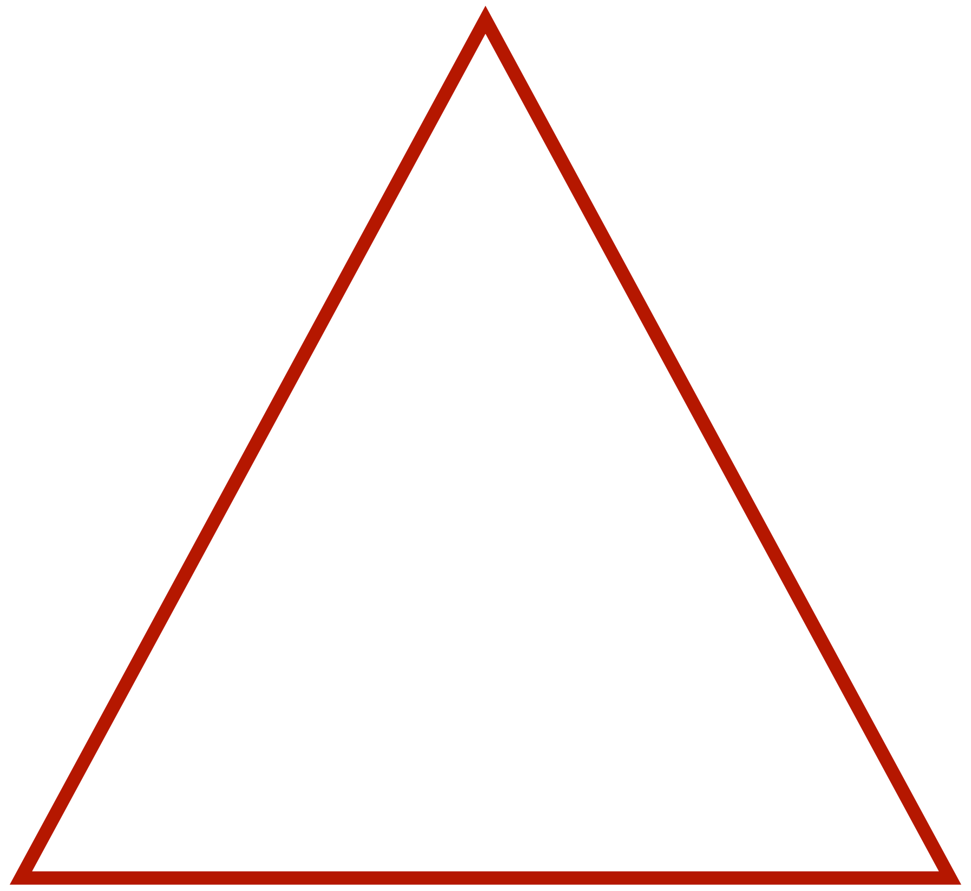
$y \preceq z$



Acyclic!

small model property.

Grammar in CNF form

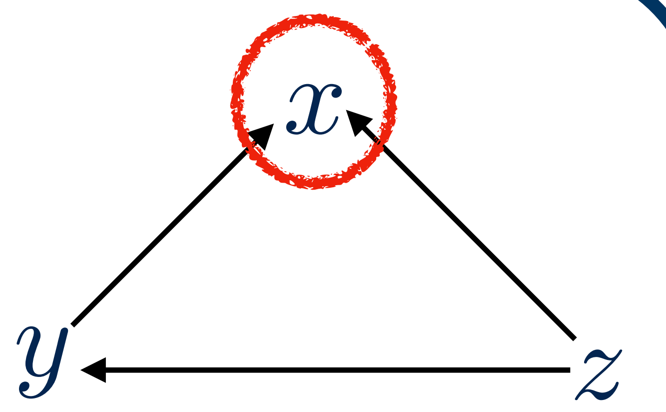


Parse tree

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

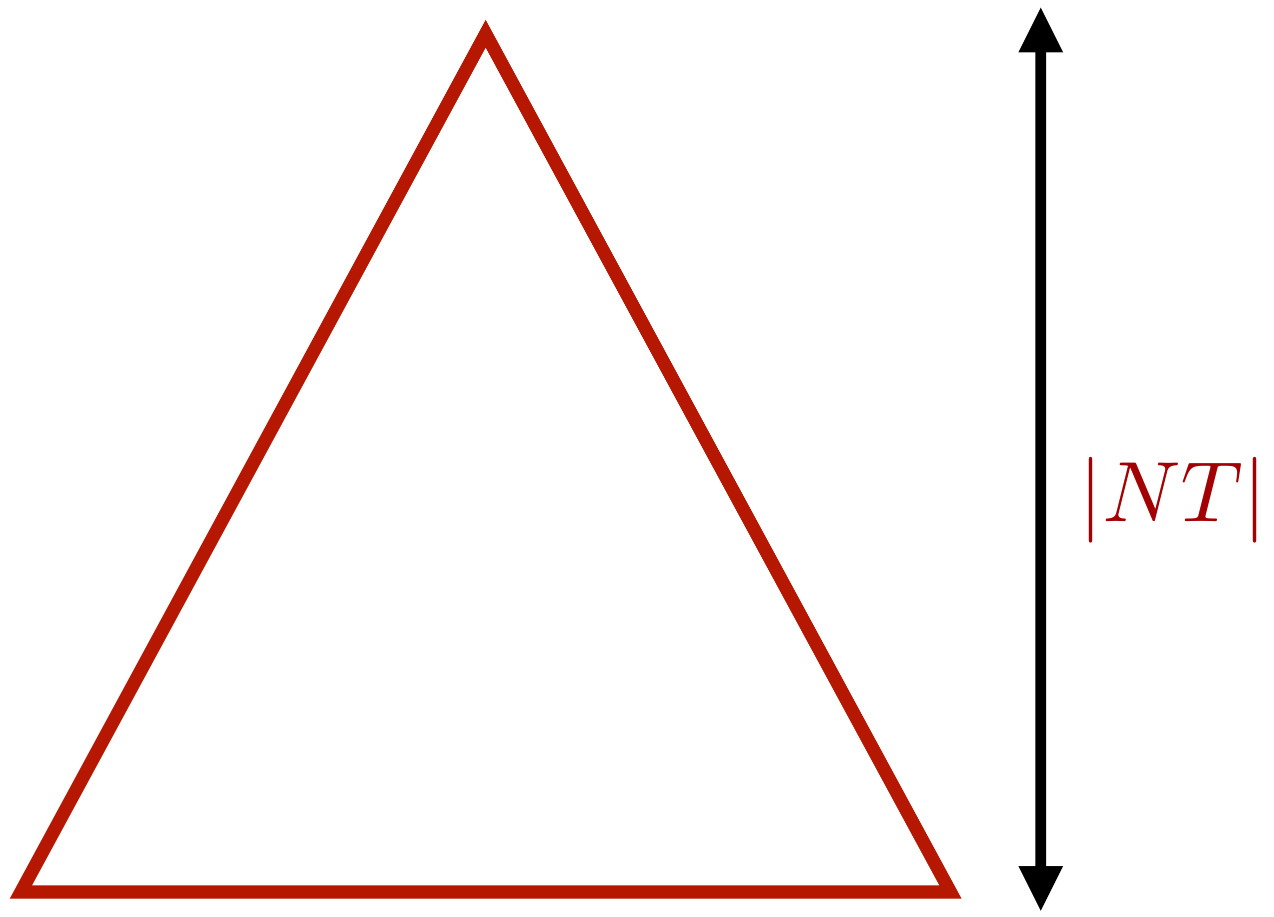
$y \preceq z$



Acyclic!

small model property.

Grammar in CNF form

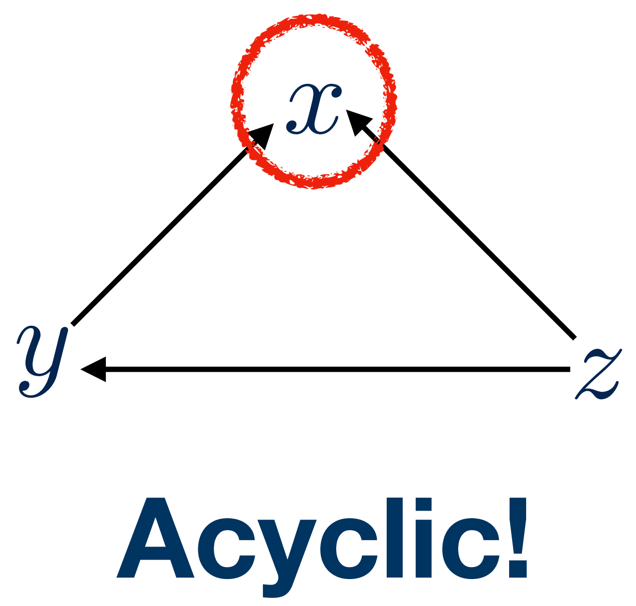


Parse tree

NEXPTIME Upper bound

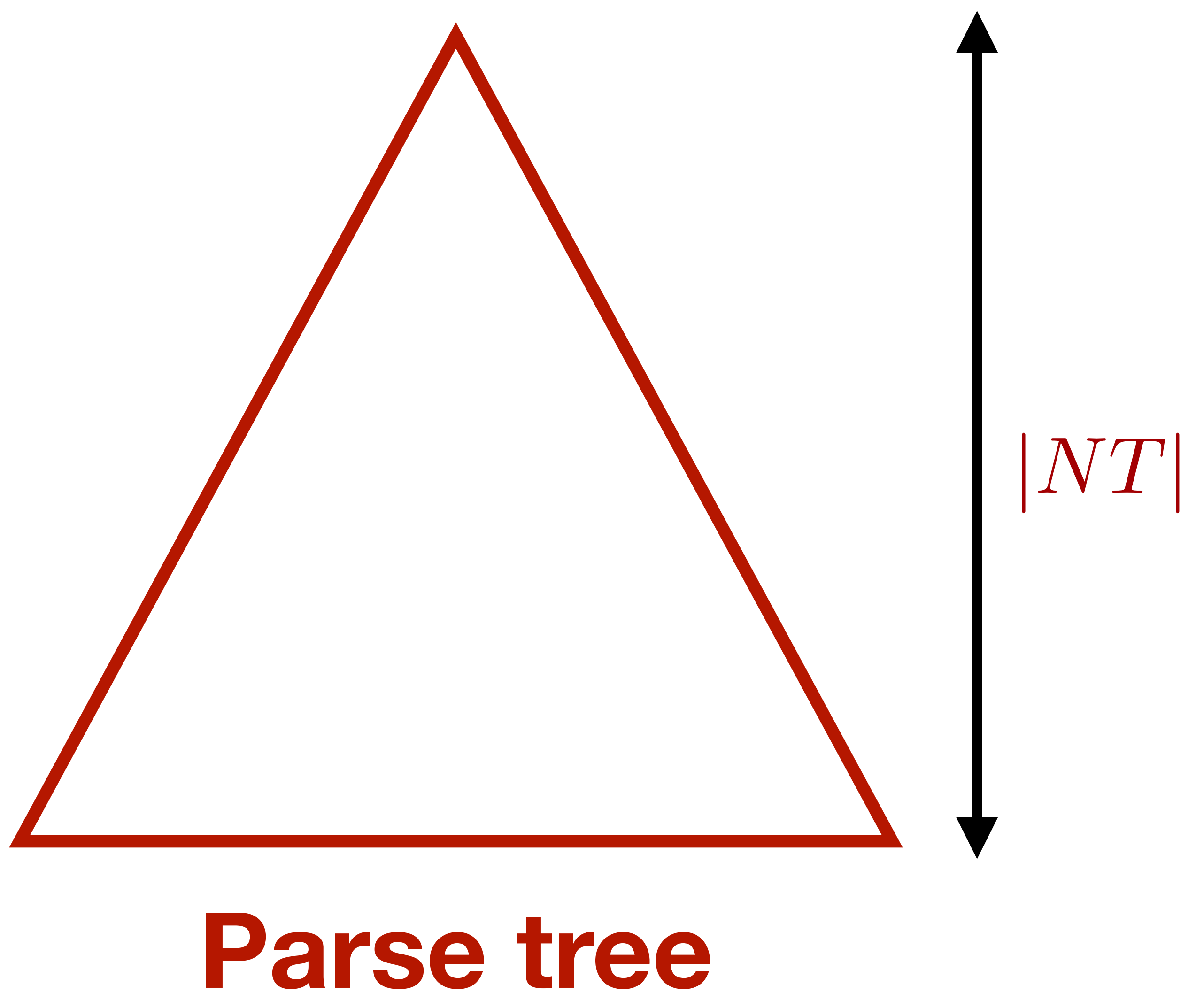
$x \preceq \mathtt{Shuffle}(yz)$

$y \preceq z$



small model property.

Grammar in CNF form

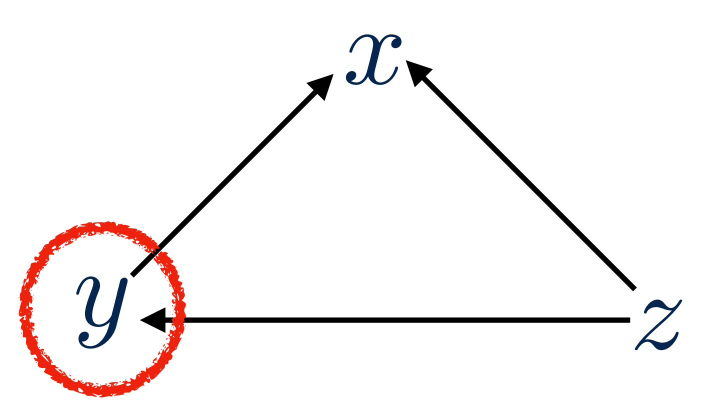


size of x $= \mathcal{O}(2^{|NT|})$

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



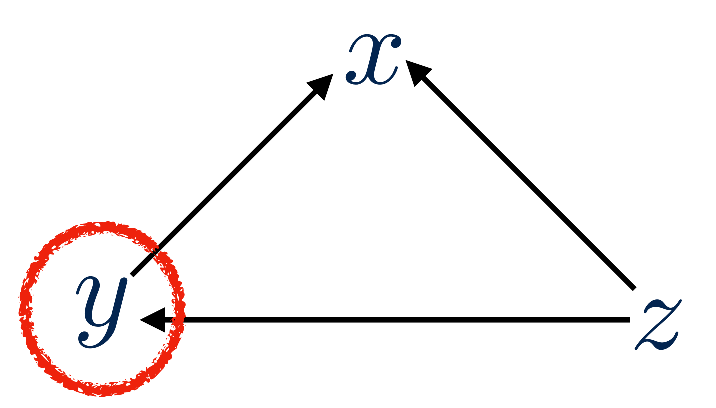
Acyclic!

small model property.

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

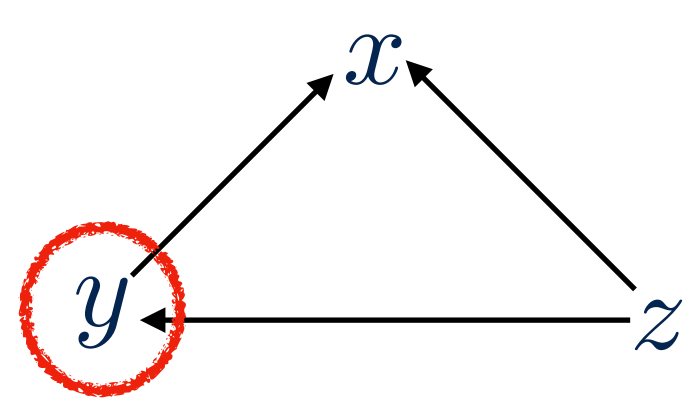
small model property.

Will x still be embedded in y?

NEXTPTIME Upper bound

$x \preceq \mathtt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

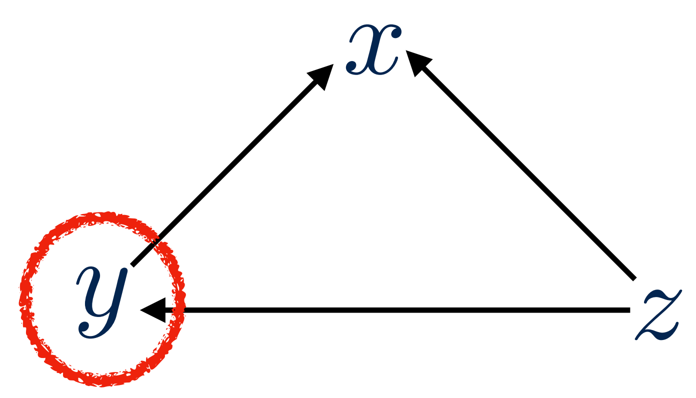
Will x still be embedded in y?

Lemma:

NEXPTIME Upper bound

$x \preceq \mathtt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

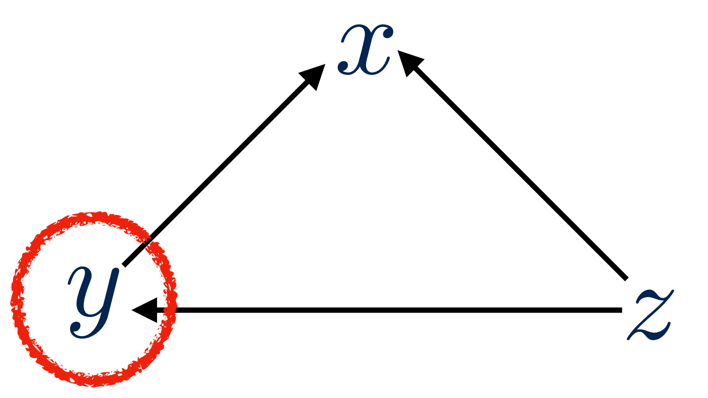
Will x still be embedded in y?

Lemma: if $w_1 \preceq w_2$ and $w_2 \in L$

NEXPTIME Upper bound

$x \preceq \mathtt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

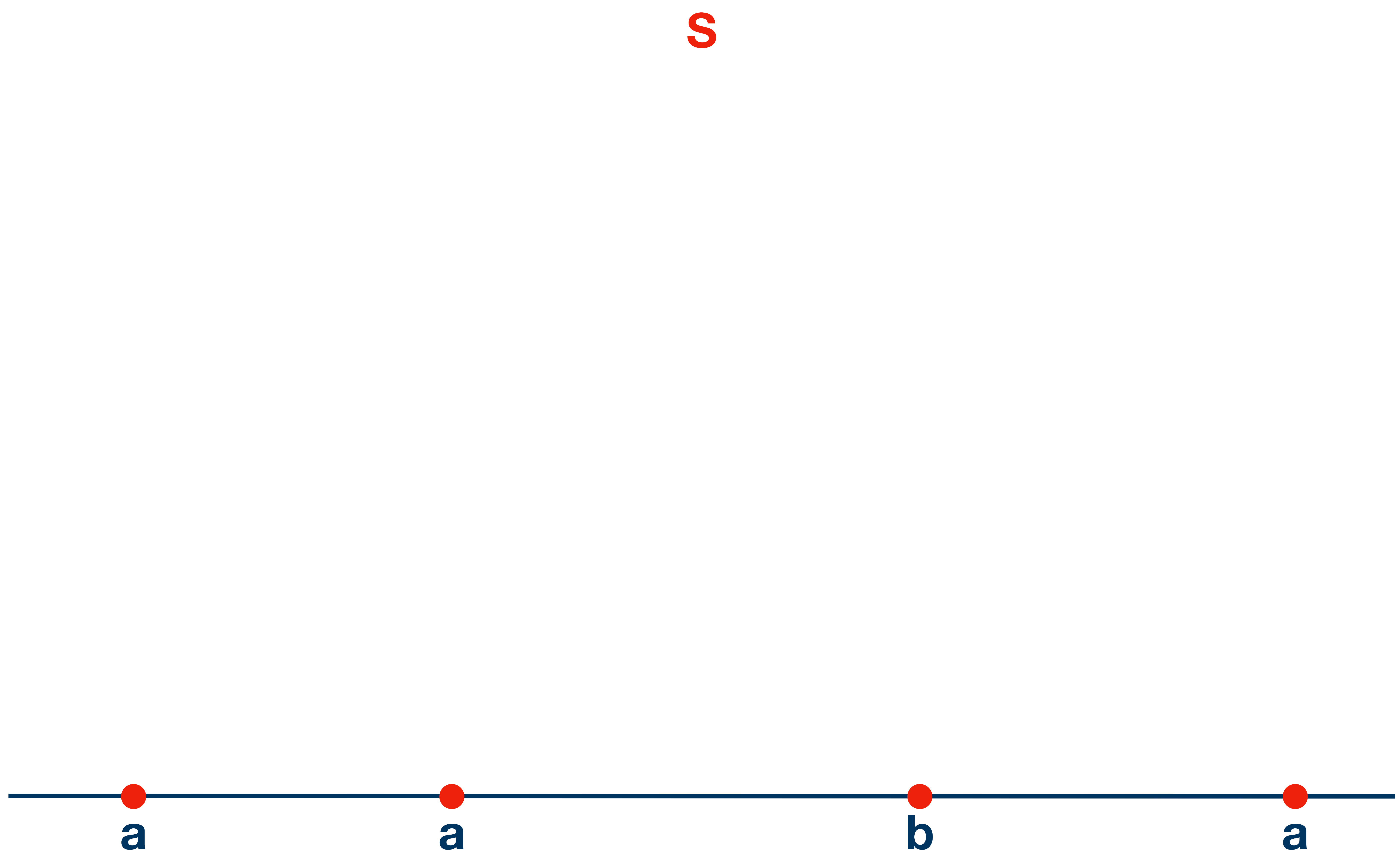
small model property.

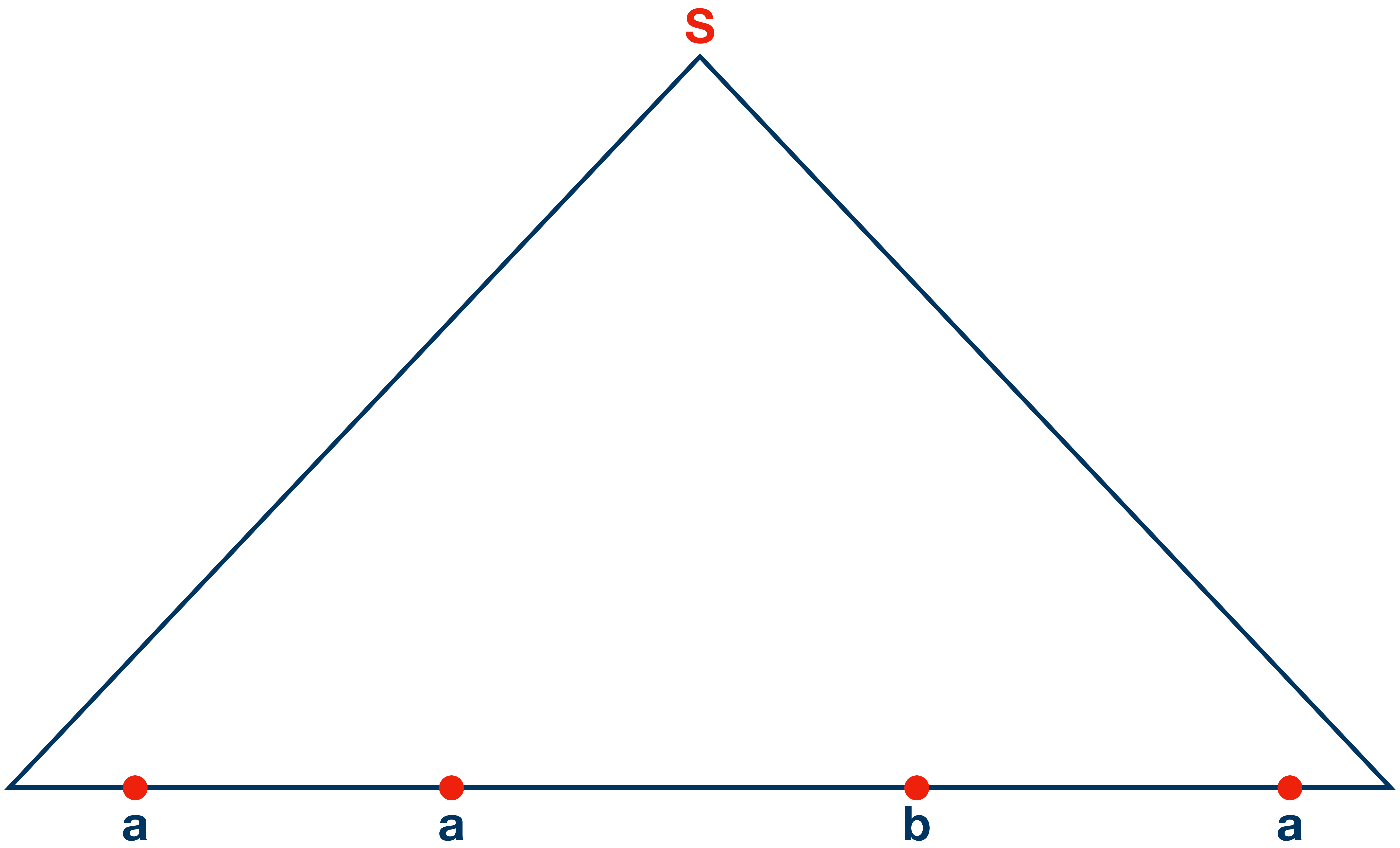
Will x still be embedded in y?

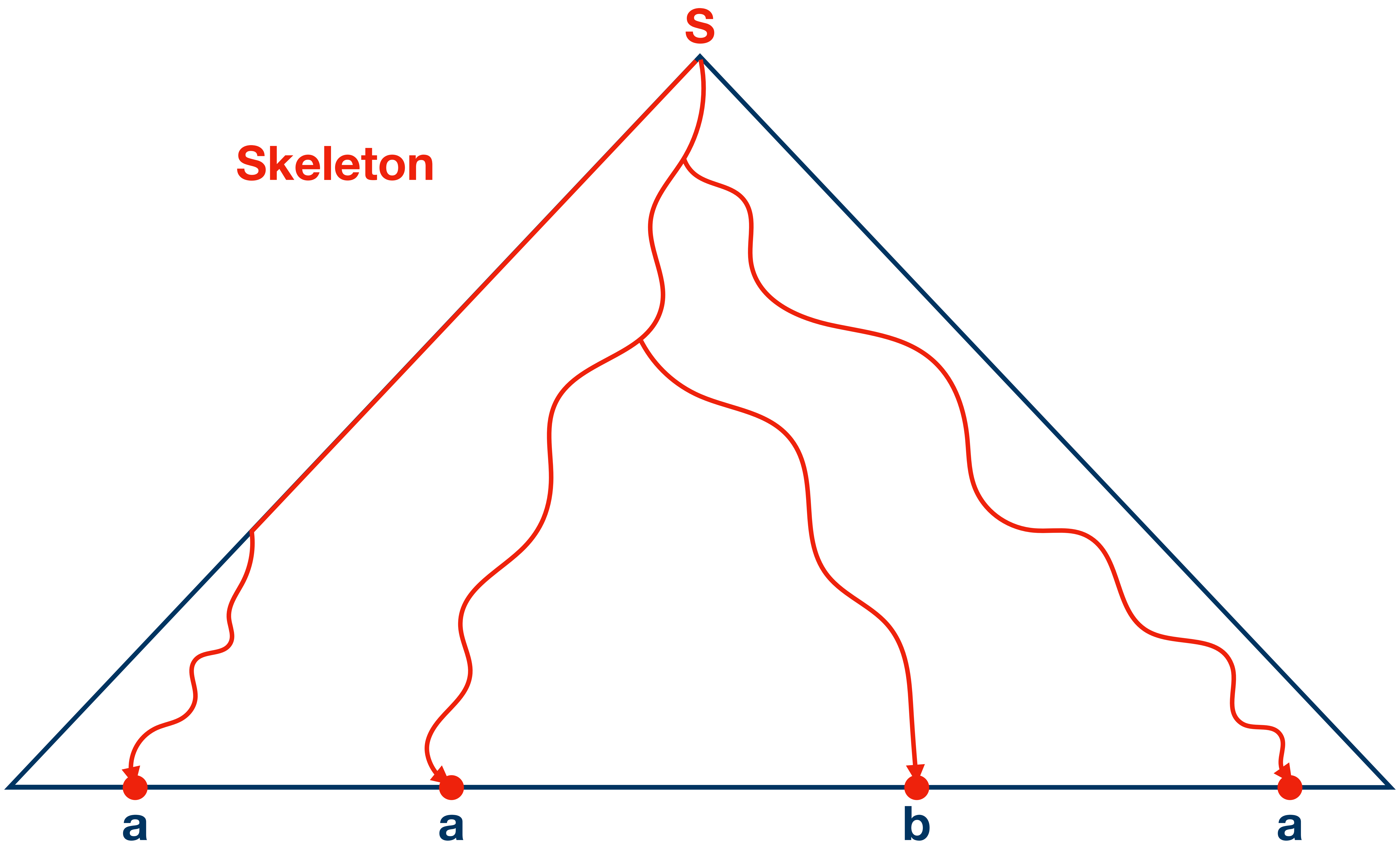
Lemma: if $w_1 \preceq w_2$ and $w_2 \in L$

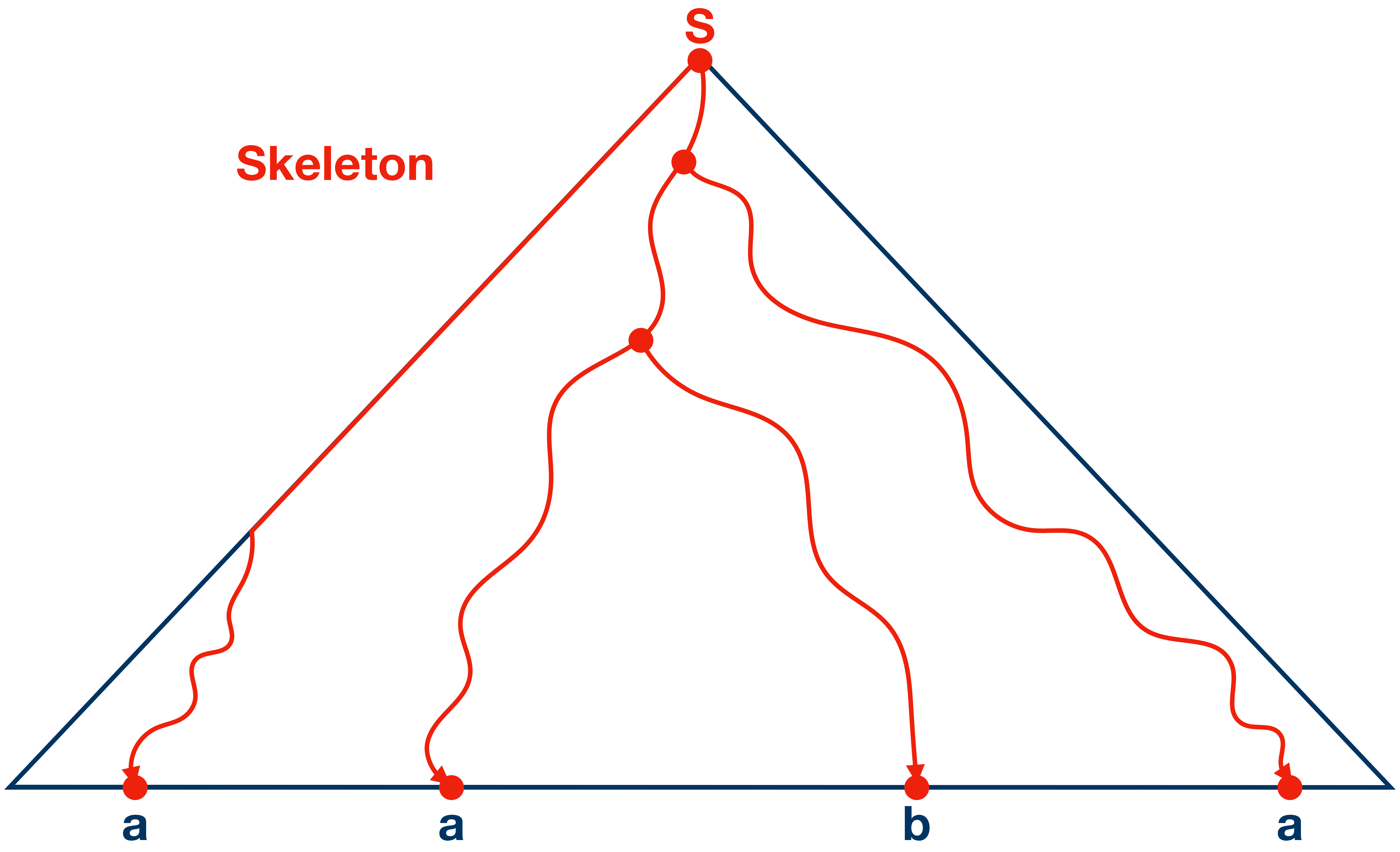
then $\exists\, w_3$: $w_1 \preceq w_3$, $w_3 \in L$ and w_3 has bounded size.

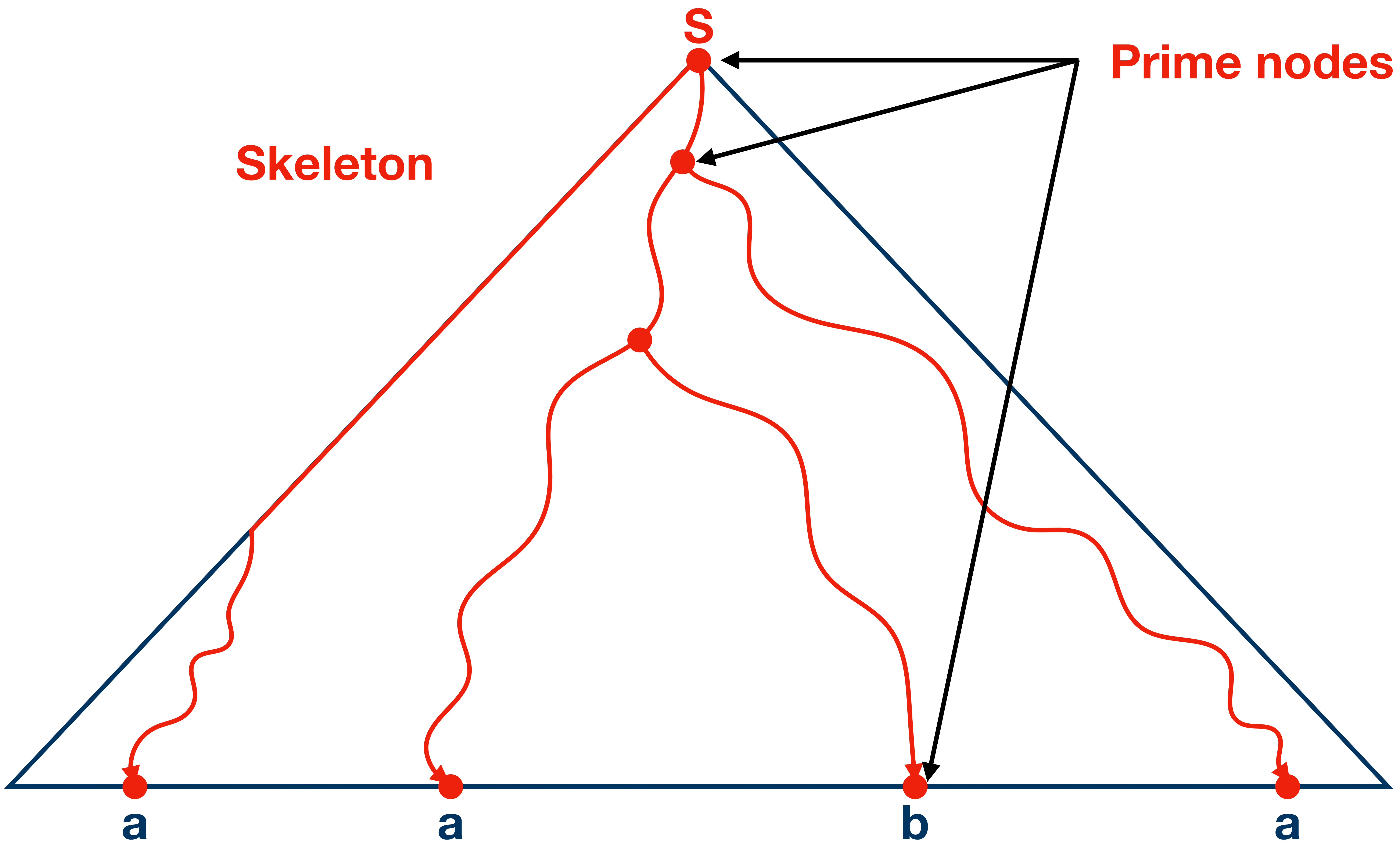

a
a
b
a

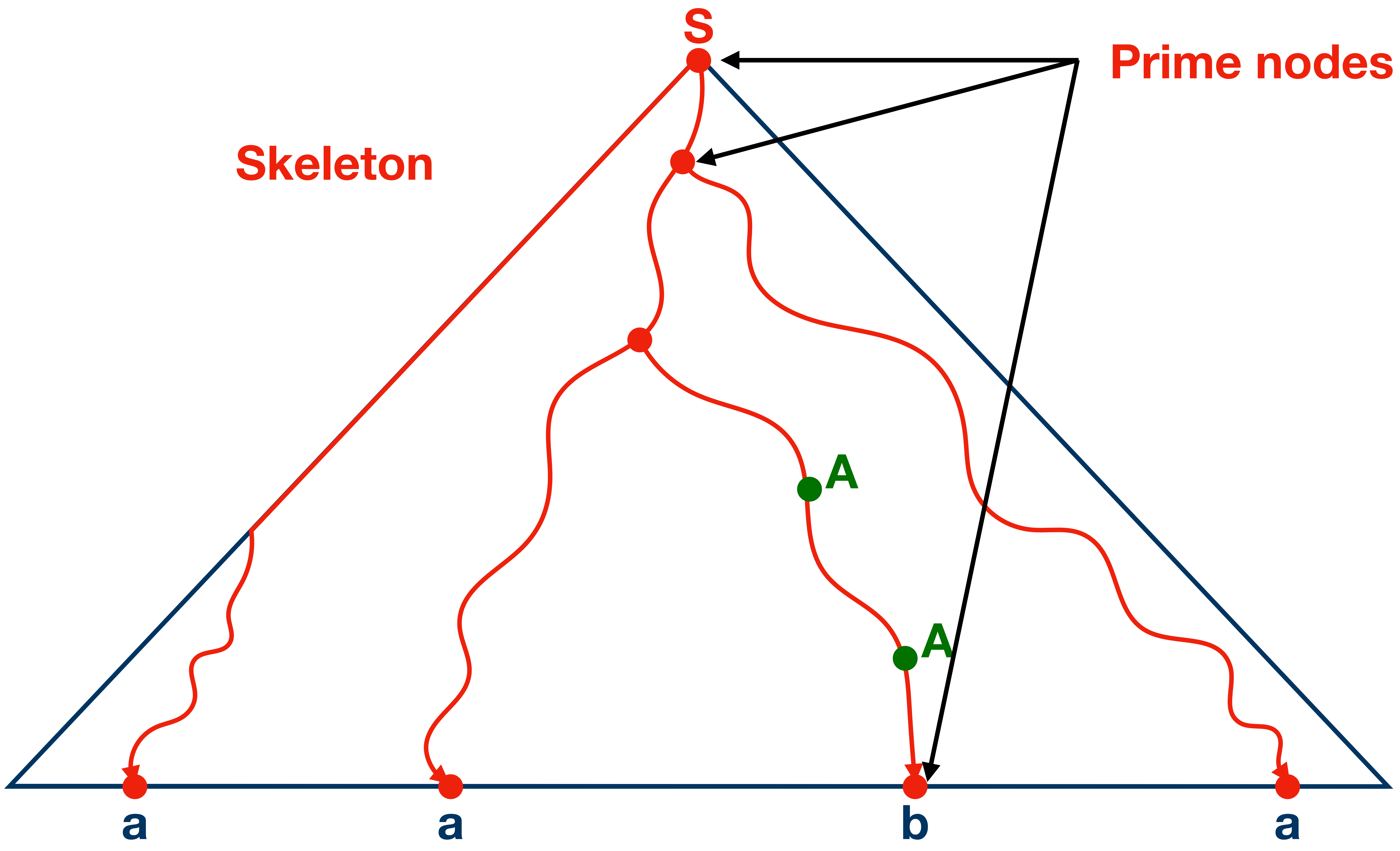

s
a
a
b
a


S
a
a
b
a

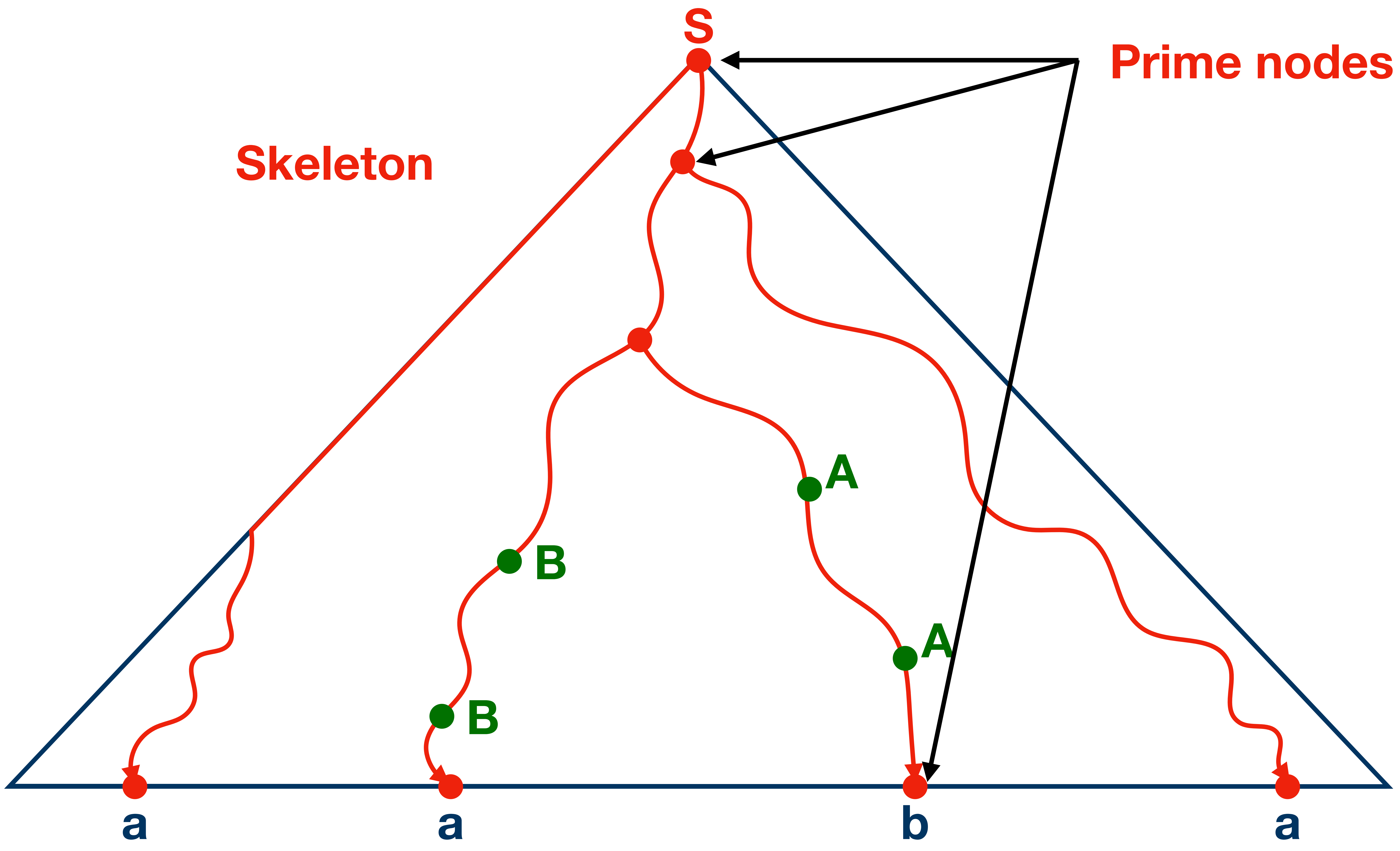

S
Skeleton
a
a
b
a

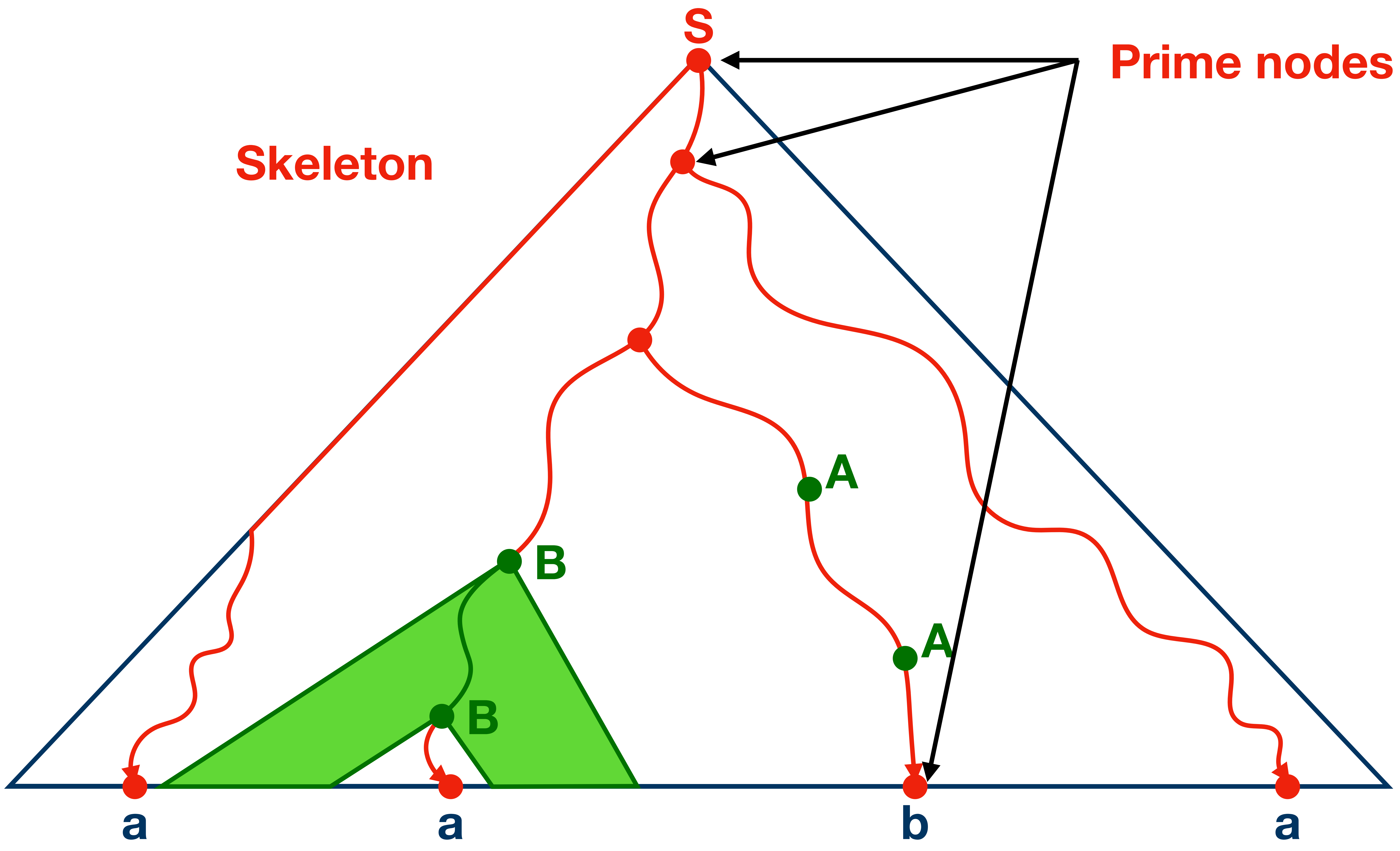

S
Skeleton
a
a
b
a

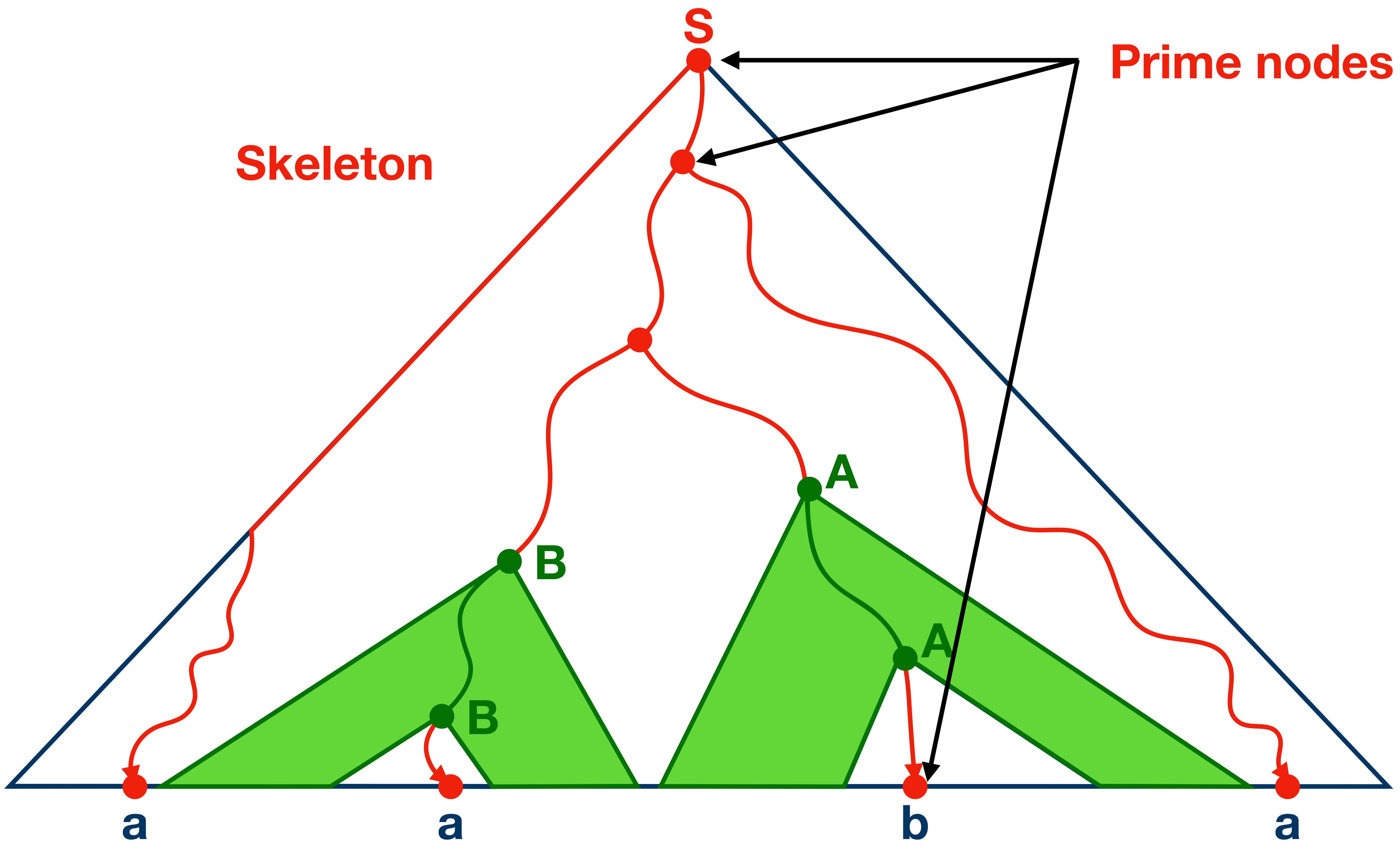

S
Prime nodes
Skeleton
a
a
b
a



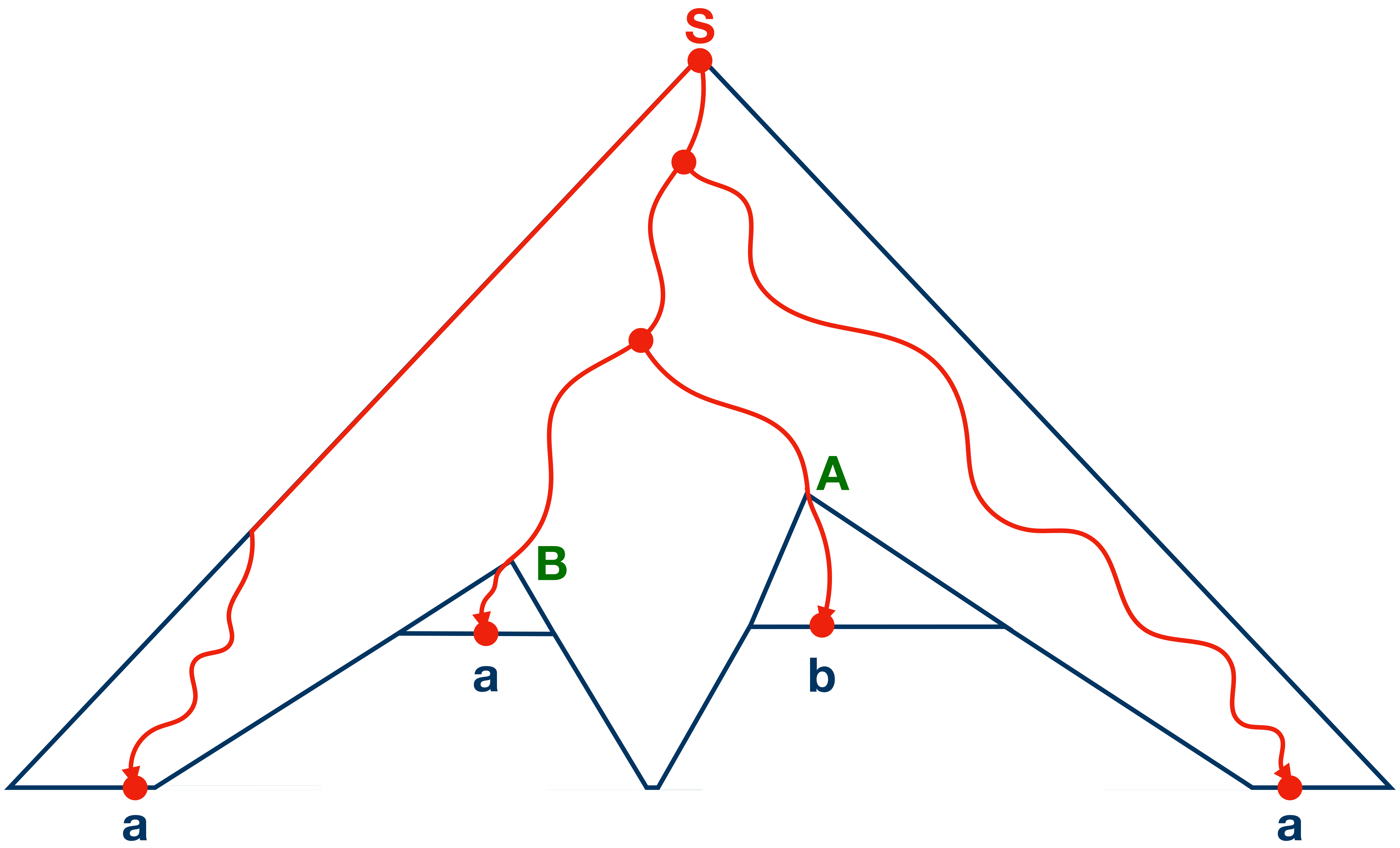
S
Prime nodes
Skeleton
A
A
a
a
b
a

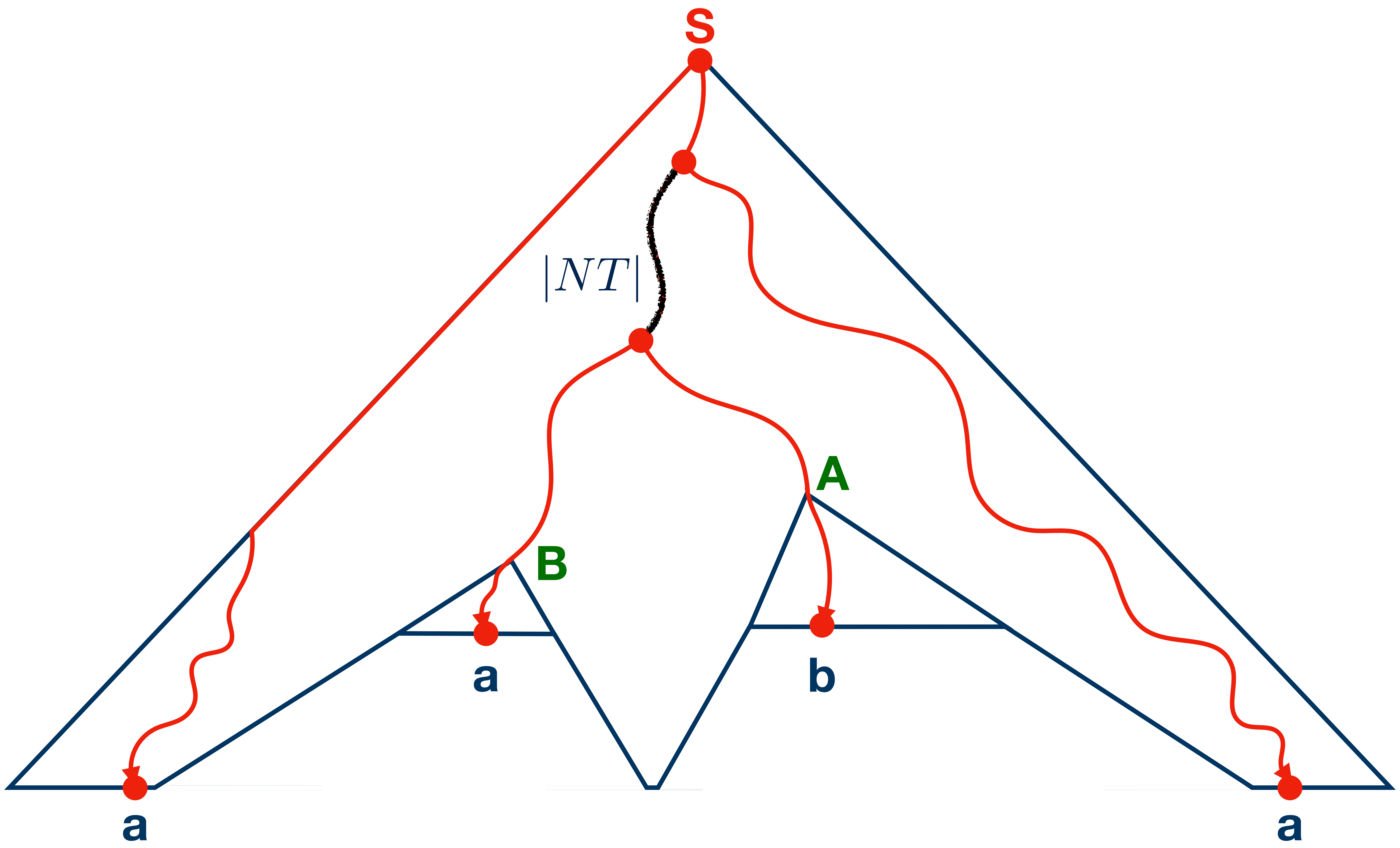

S
Prime nodes
Skeleton
A
B
A
B
a
a
b
a

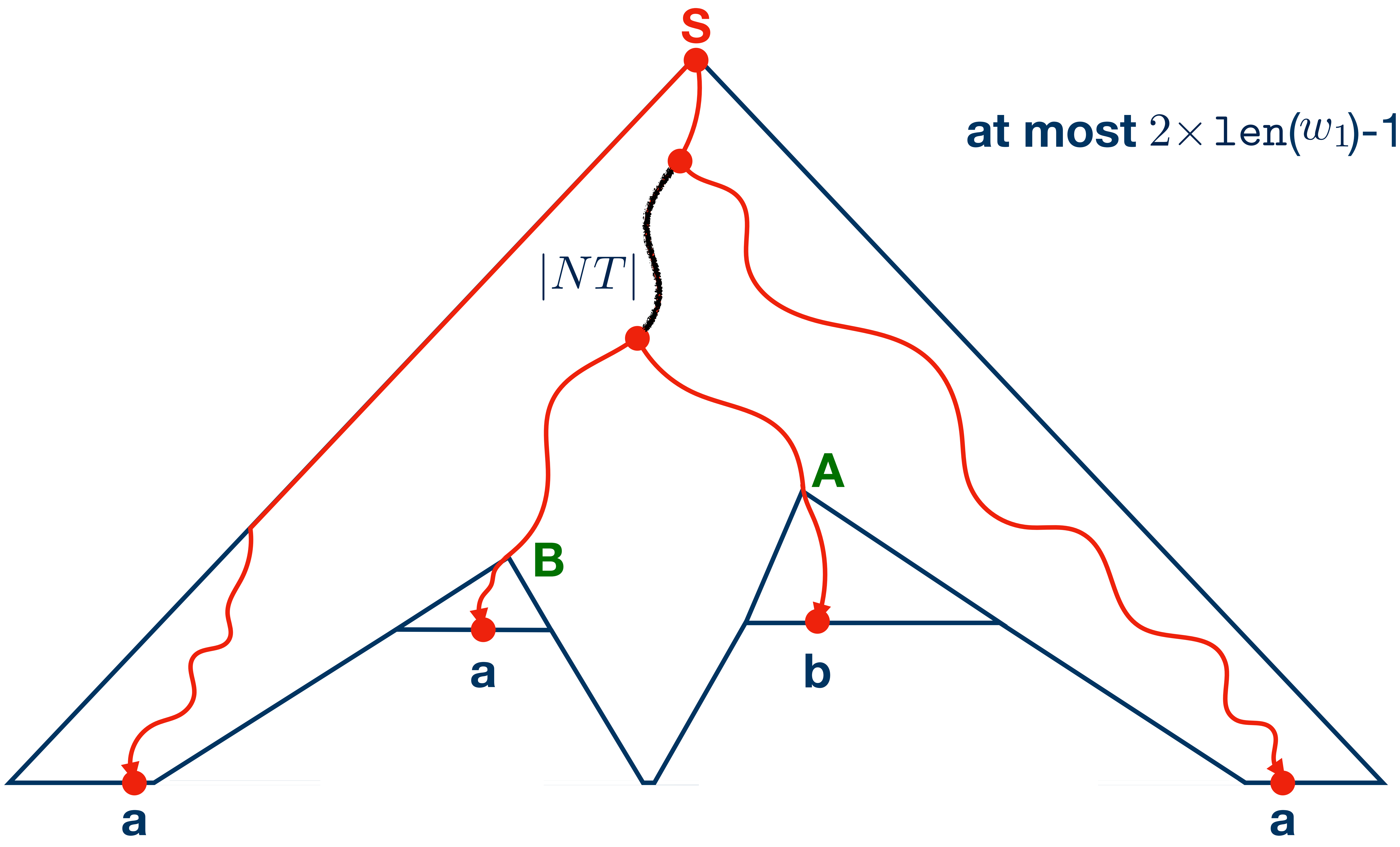

S
Prime nodes
Skeleton
A
A
B
B
a
a
b
a



S
Prime nodes
Skeleton
A
B
A
B
a
a
b
a


S
A
B
a
b
a
a


S
$|NT|$
A
B
a
b
a
a

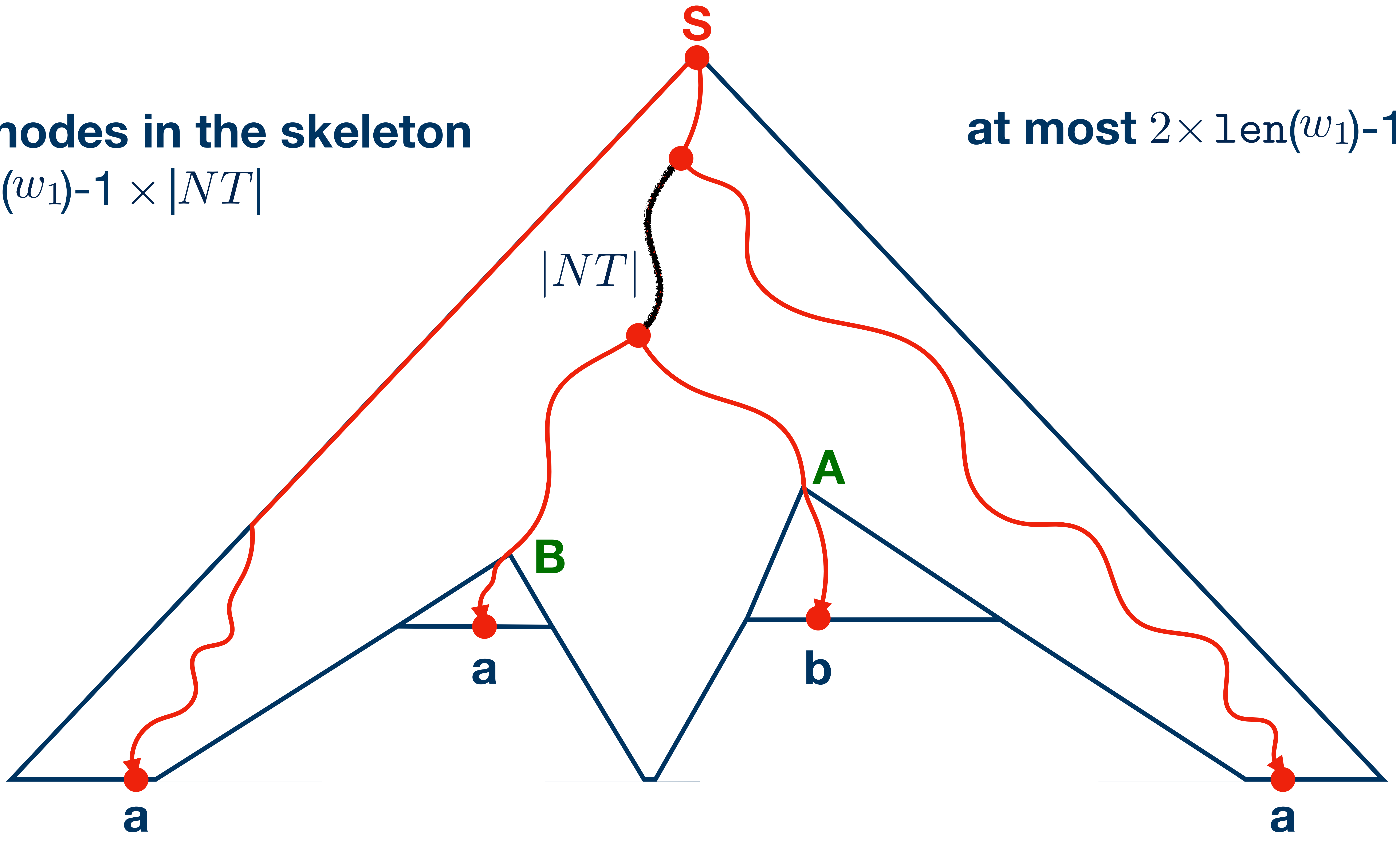


at most $2\times \texttt{len}(w_1)$-1 prime nodes

Total number of nodes in the skeleton

$= 2 \times \texttt{len}(w_1) - 1 \times |NT|$

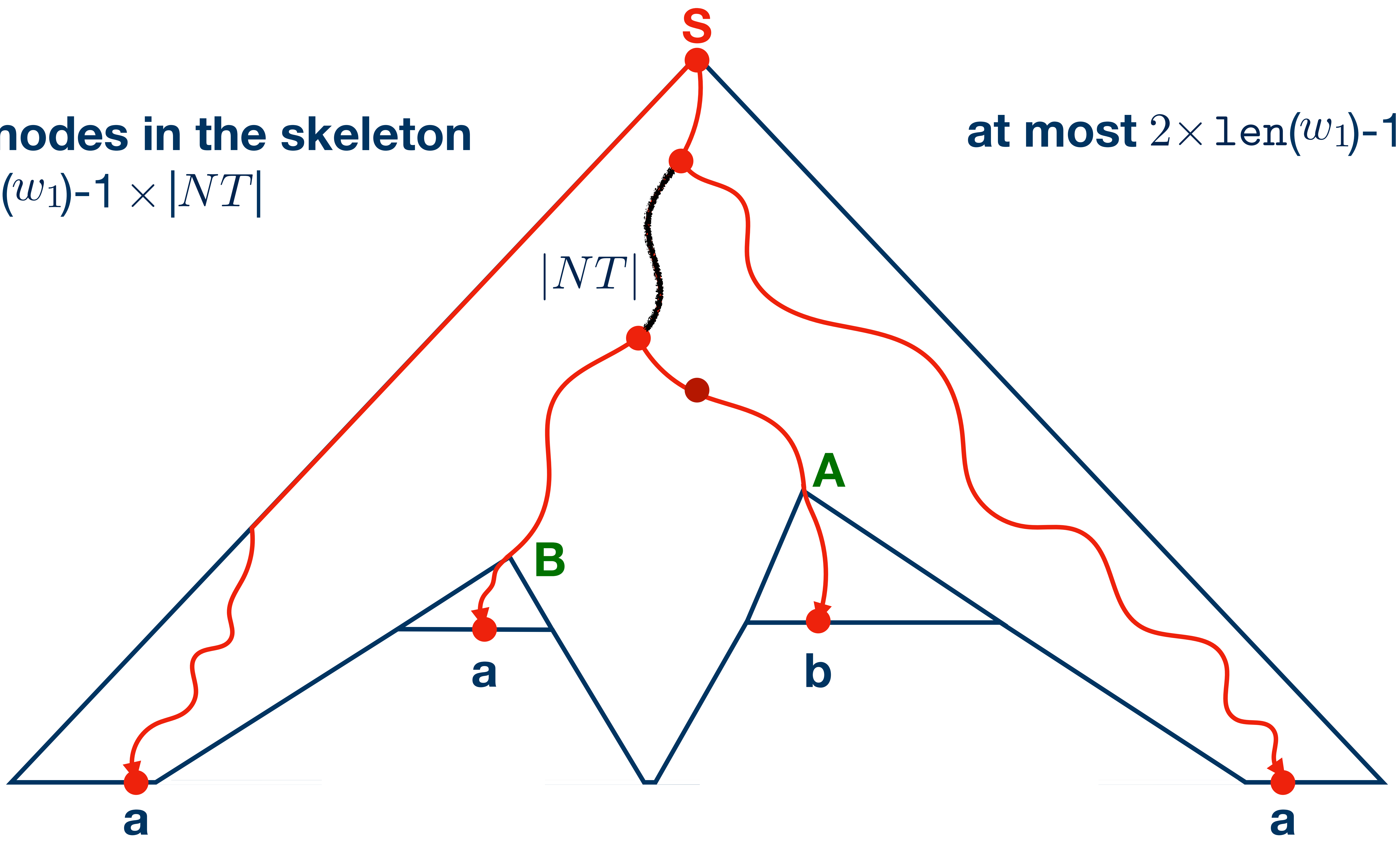
at most $2 \times \texttt{len}(w_1) - 1$ prime nodes



Total number of nodes in the skeleton

$= 2 \times \texttt{len}(w_1) - 1 \times |NT|$

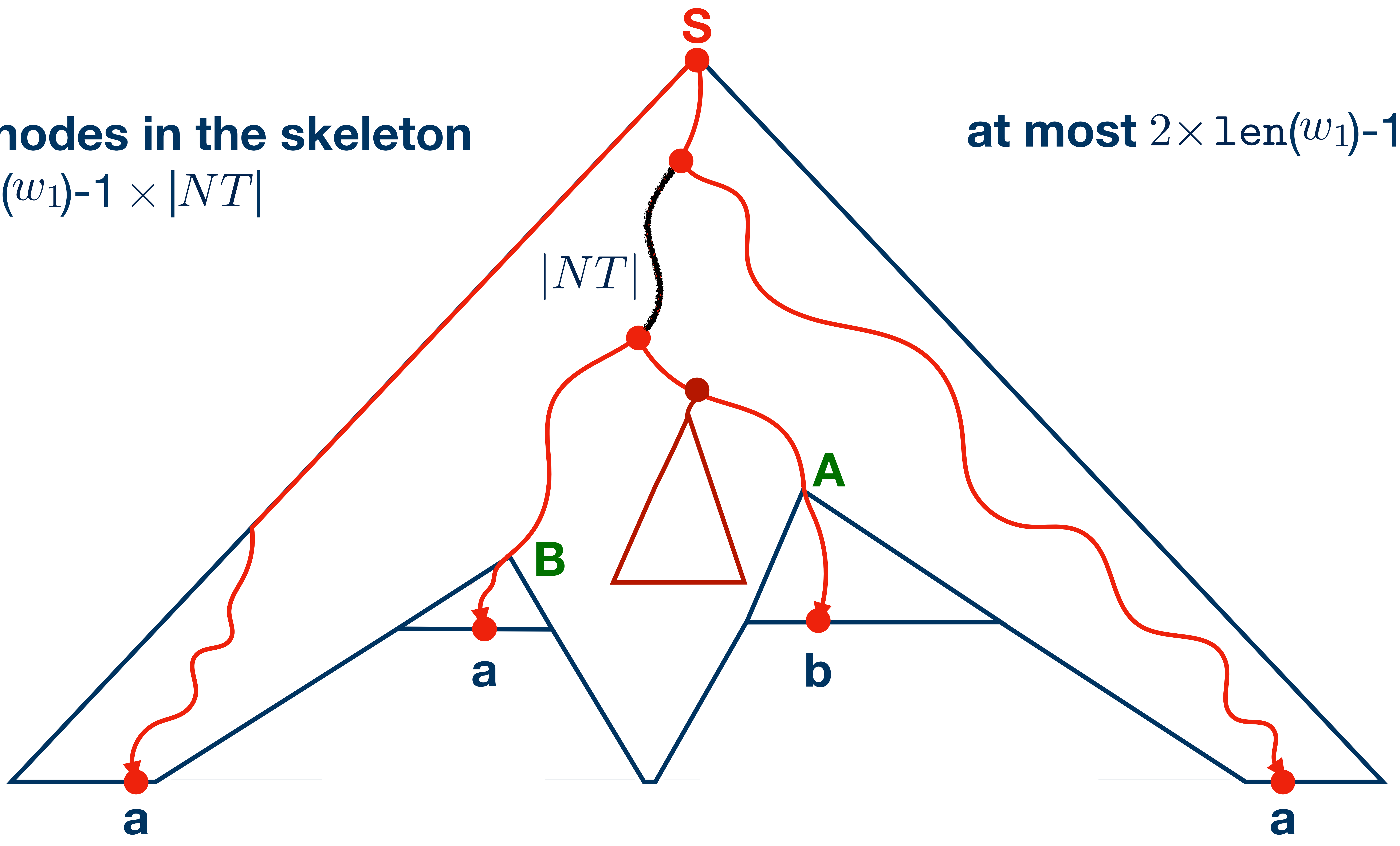
at most $2 \times \texttt{len}(w_1)$-1 prime nodes

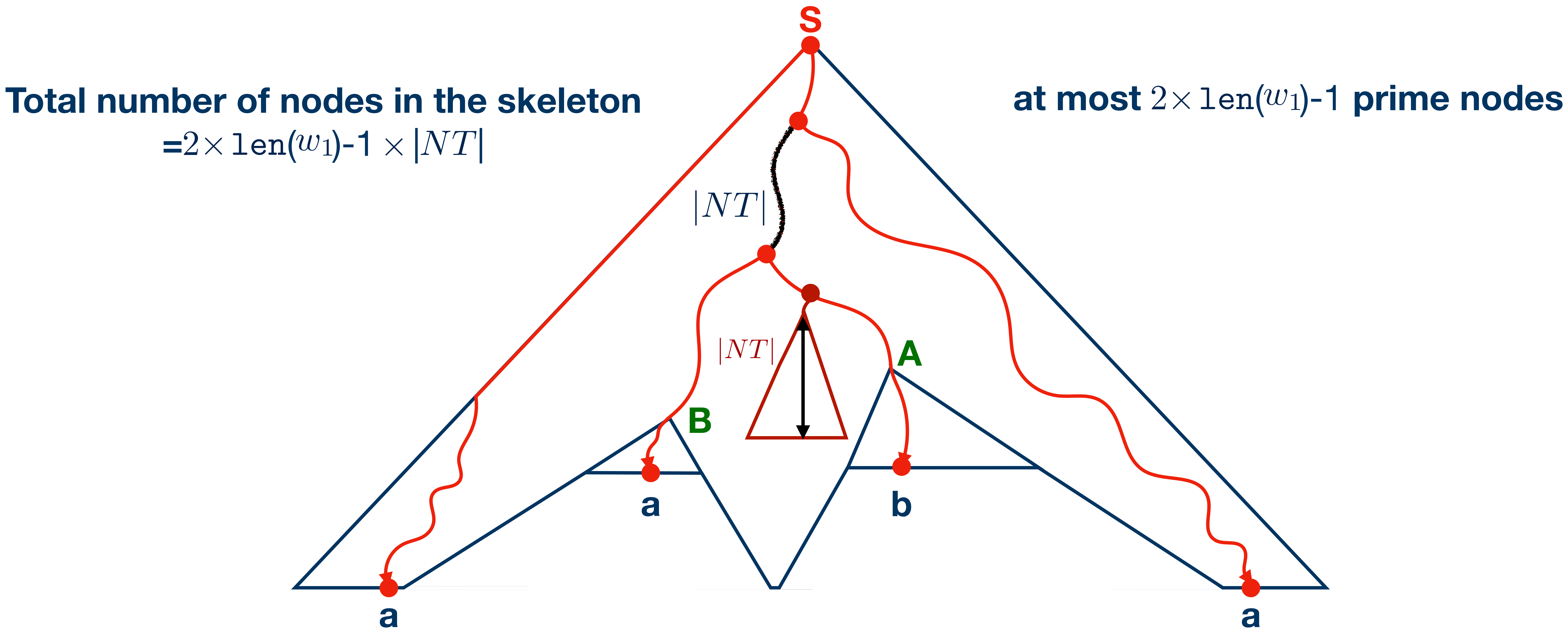


Total number of nodes in the skeleton

=$2\times \texttt{len}(w_1)$-1 $\times |NT|$

at most $2\times \texttt{len}(w_1)$-1 prime nodes




Total number of nodes in the skeleton
$=2\times \texttt{len}(w_1)-1 \times |NT|$
at most $2\times \texttt{len}(w_1)-1$ prime nodes
S
$|NT|$
$|NT|$
A
B
a
b
a
a

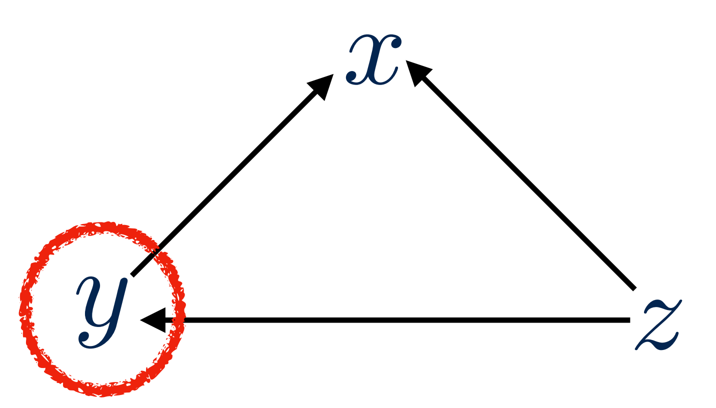
Total number of nodes in the skeleton
=$2\times$ len(w_1)-1 $\times |NT|$
at most $2\times$ len(w_1)-1 prime nodes
S
$|NT|$
subtree of size atmost $2^{|NT|}$
$|NT|$
A
B
a
b
a
a

S
Total number of nodes in the skeleton
$=2\times$ len(w_1)-1 $\times |NT|$
at most $2\times$ len(w_1)-1 prime nodes
$|NT|$
Size of the parse tree
$\mathcal{O}(Len(w_1) \times |NT| \times 2^{|NT|})$
subtree of size atmost $2^{|NT|}$
$|NT|$
A
B
a
b
a
a

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

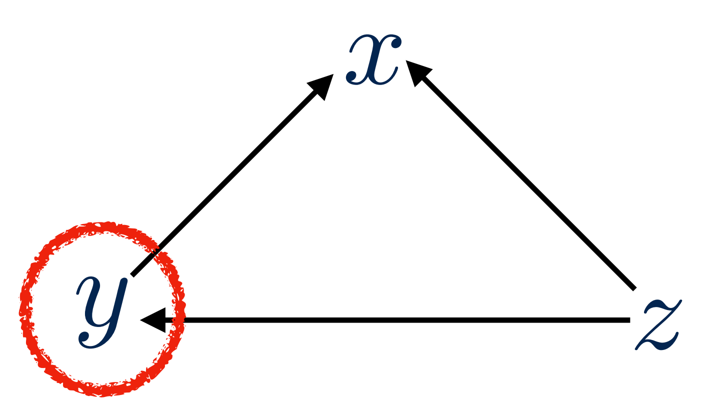
Size of the parse tree

$\mathcal{O}(Len(x) \times |NT| \times 2^{|NT|})$

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

Size of the parse tree

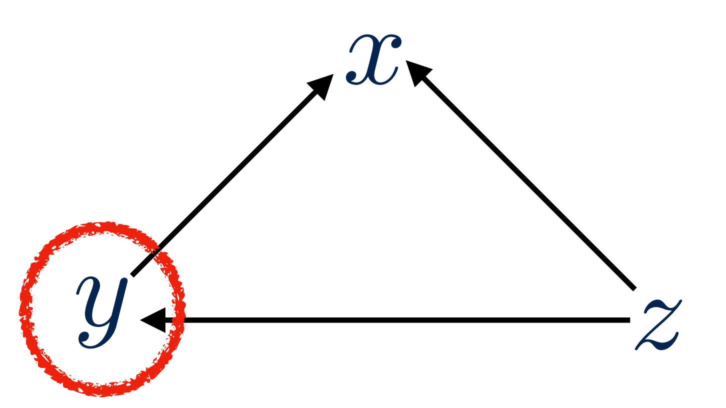
$\mathcal{O}(Len(x) \times |NT| \times 2^{|NT|})$

$2^{|NT|}$

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

Size of the parse tree

$\mathcal{O}(Len(x) \times |NT| \times 2^{|NT|})$

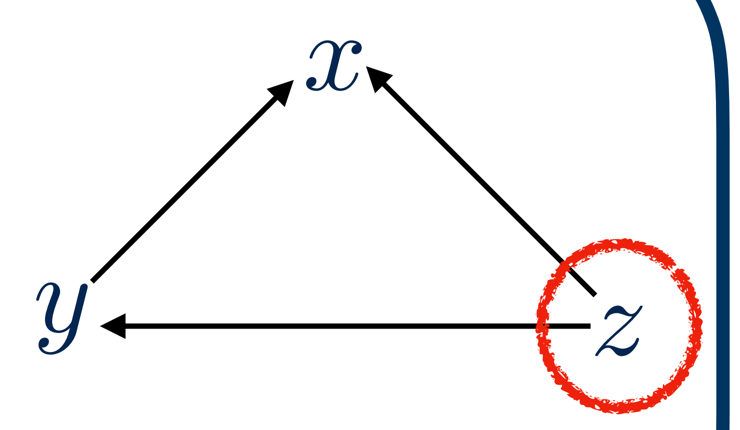
$2^{|NT|}$

y is bounded by $\mathcal{O}(2^{2^{2 \times |NT|}} \times |NT|)$

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

Size of the parse tree

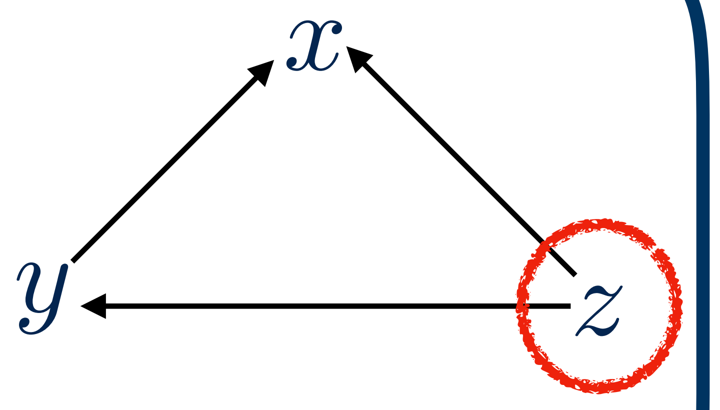
$\mathcal{O}(\texttt{Len}(y) \times |NT| \times 2^{|NT|})$

y is bounded by $\mathcal{O}(2^{2\times|NT|} \times |NT|)$

NEXPTIME Upper bound

$x \preceq \texttt{Shuffle}(yz)$

$y \preceq z$



Acyclic!

small model property.

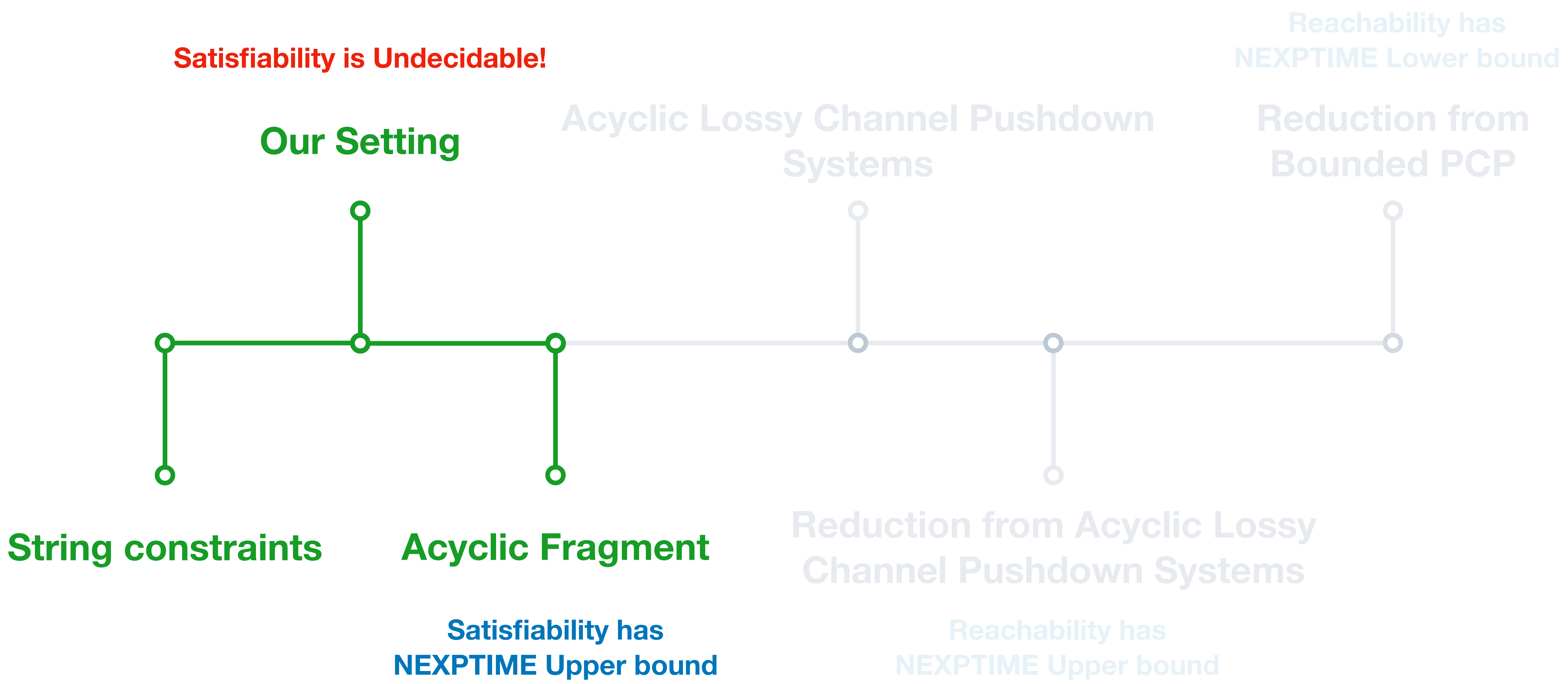
Size of the parse tree

$\mathcal{O}(\texttt{Len}(y) \times |NT| \times 2^{|NT|})$

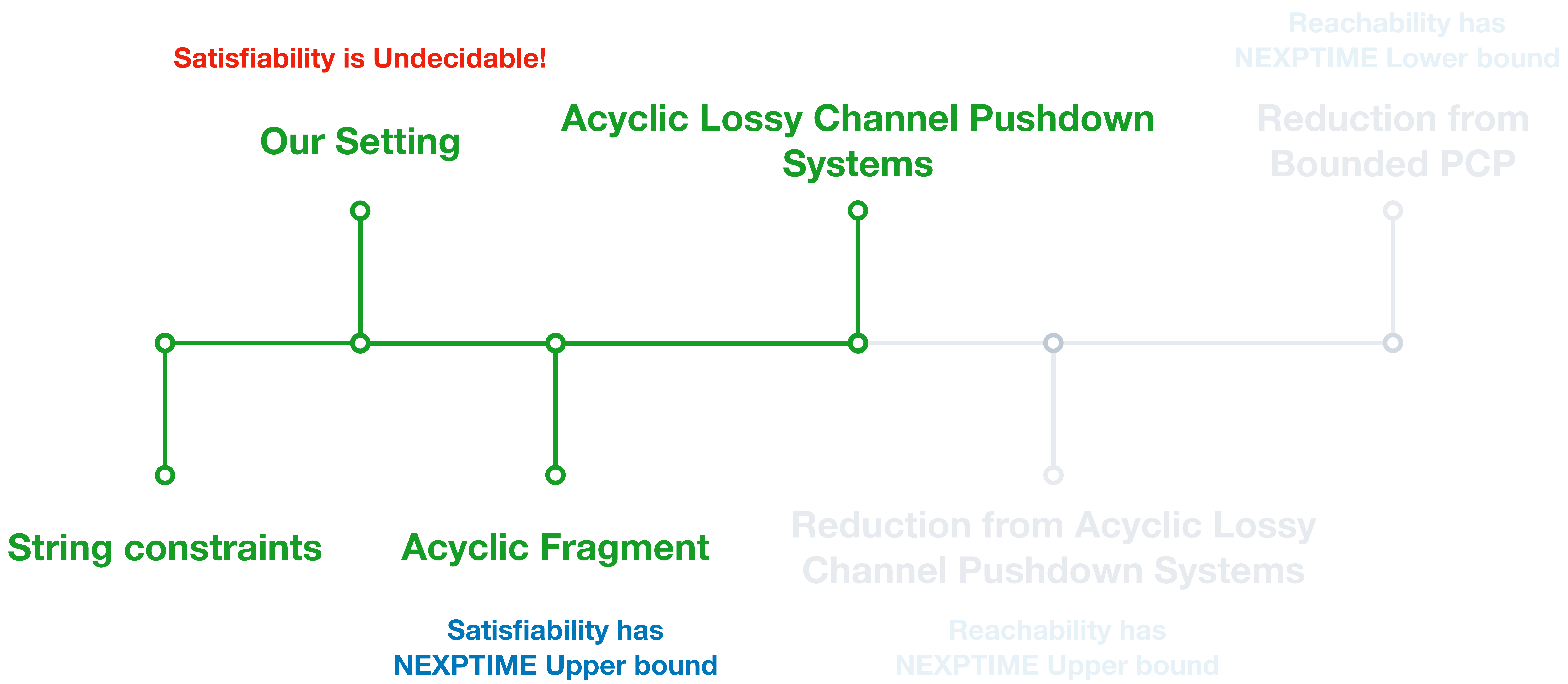
y is bounded by $\mathcal{O}(2^{2\times|NT|} \times |NT|)$

z is bounded by $\mathcal{O}(2 \times 2^{3\times|NT|} \times |NT|^2)$

Outline

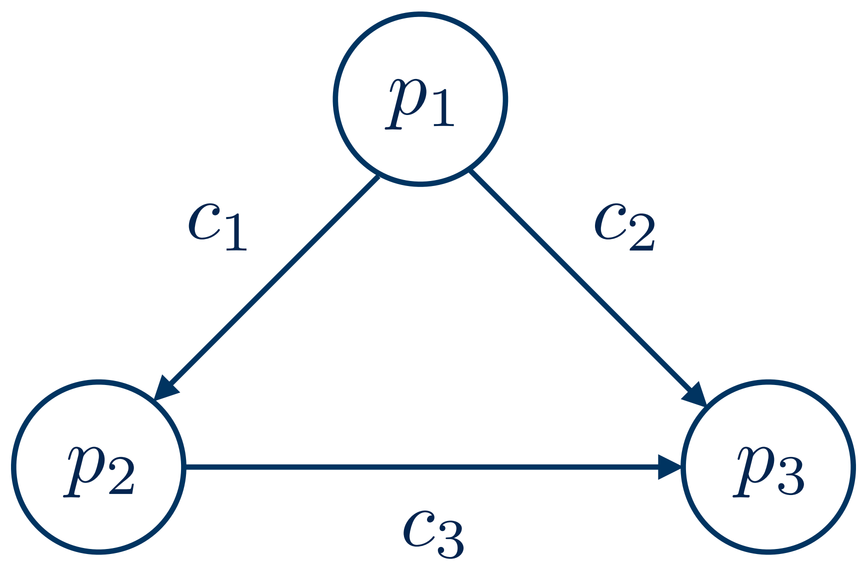

Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound

Outline

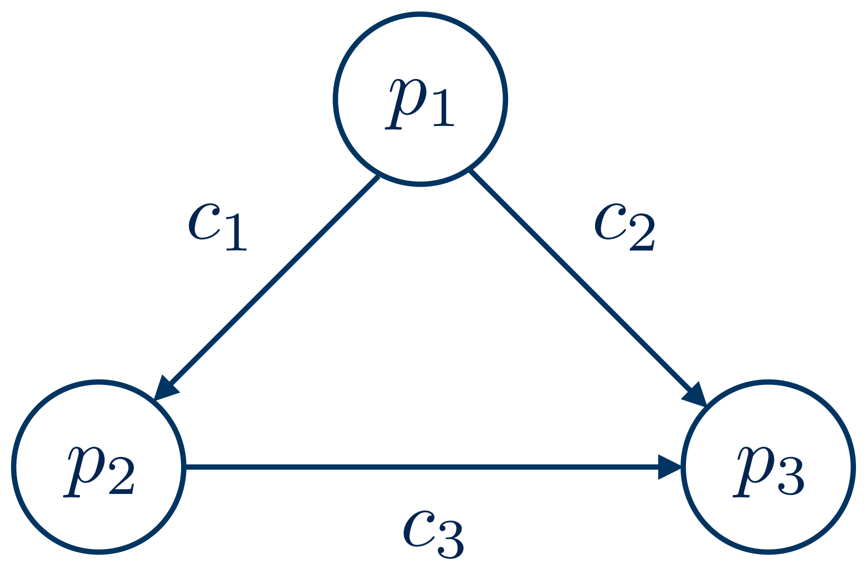


Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound

Acyclic Lossy Channel Pushdown Systems



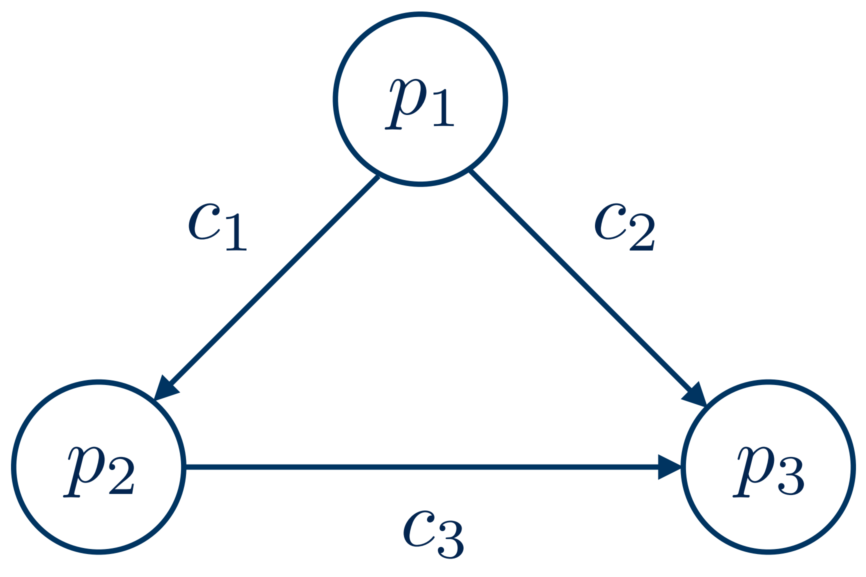
Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]



Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

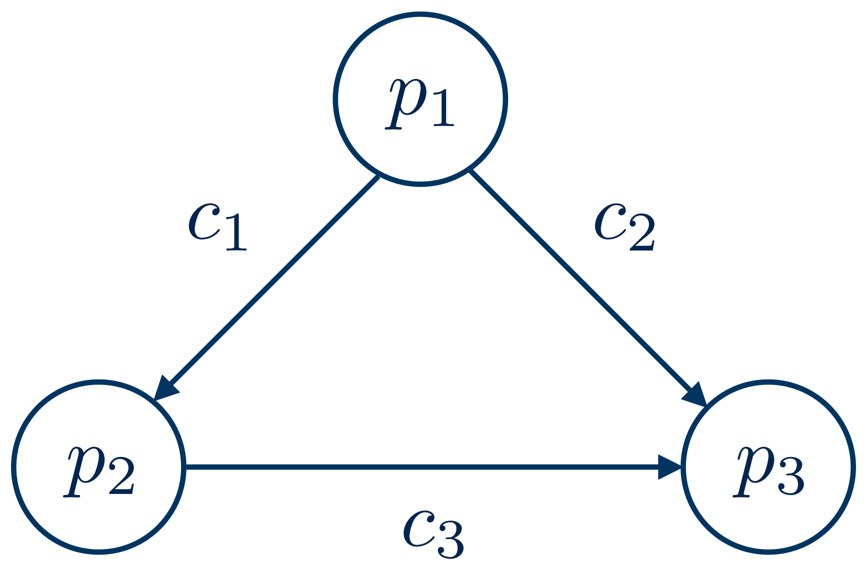


Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

Lossy FIFO channels



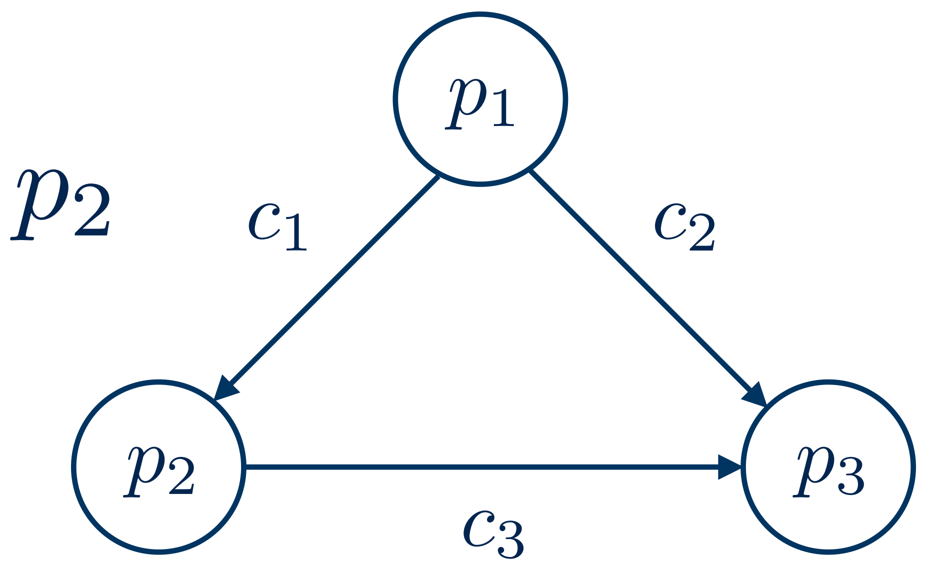
Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

Lossy FIFO channels

$$\texttt{Reader}(c_1) = p_2$$

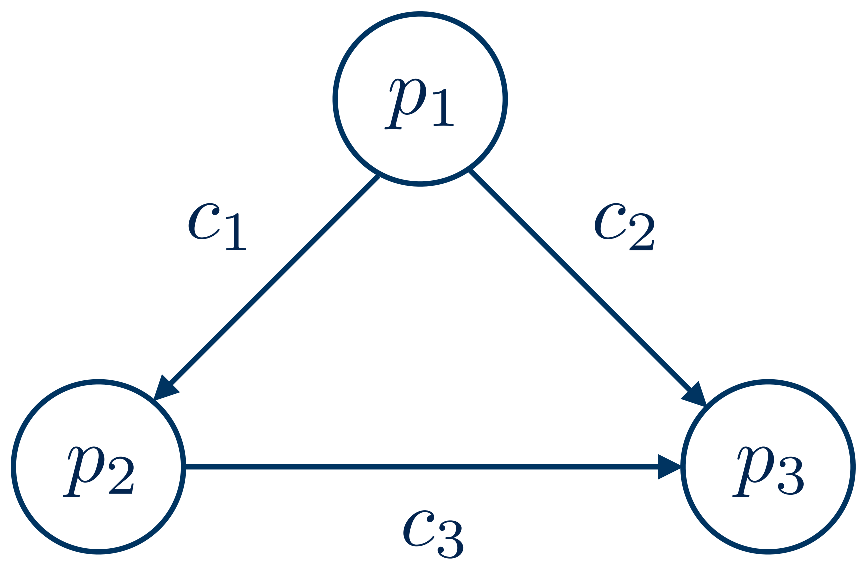


Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

Lossy FIFO channels



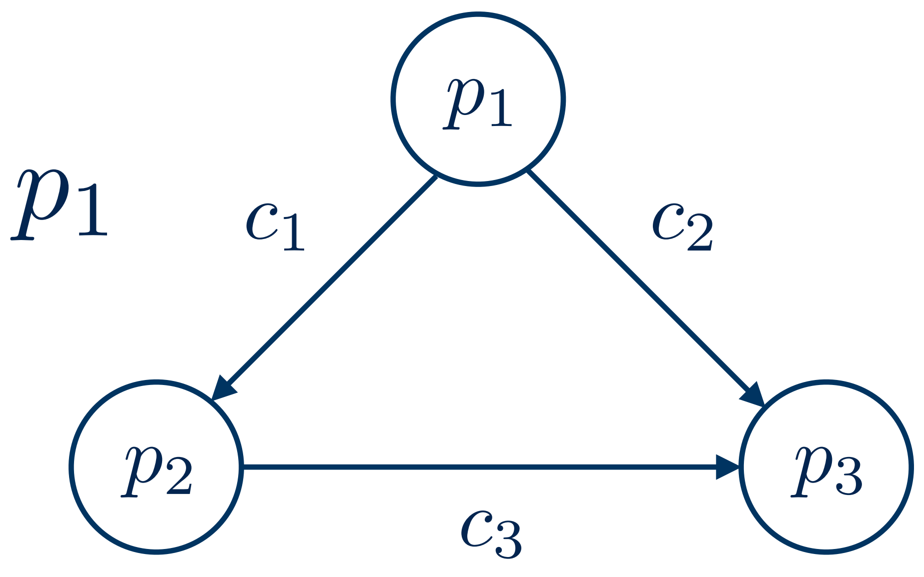
Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

Lossy FIFO channels

$$\texttt{Writer}(c_1) = p_1$$

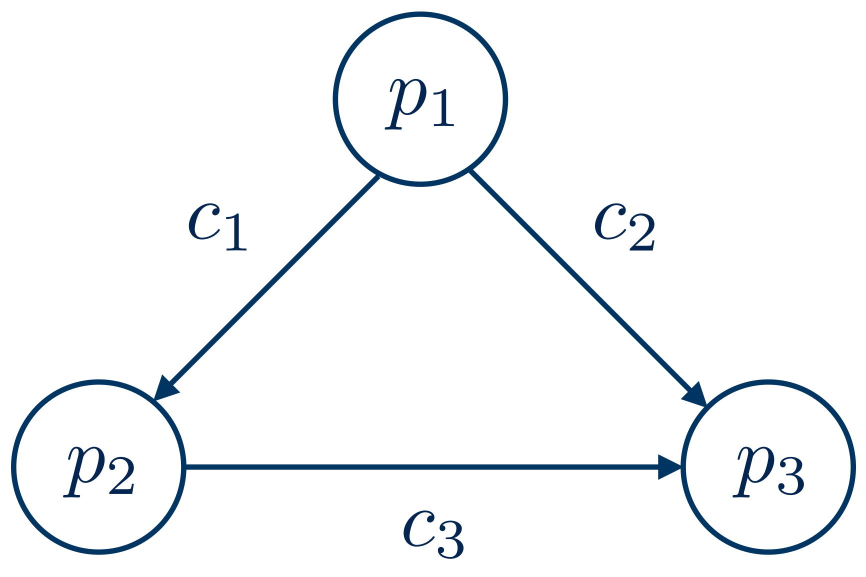


Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems [Atig et. al. 2008]

Processes - Pushdown Systems

Lossy FIFO channels



Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems

Acyclic Lossy Channel Pushdown Systems

Control State Reachability Problem

Acyclic Lossy Channel Pushdown Systems

[Atig et. al. 2008]

Control State Reachability Problem

Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Acyclic Lossy Channel Pushdown Systems

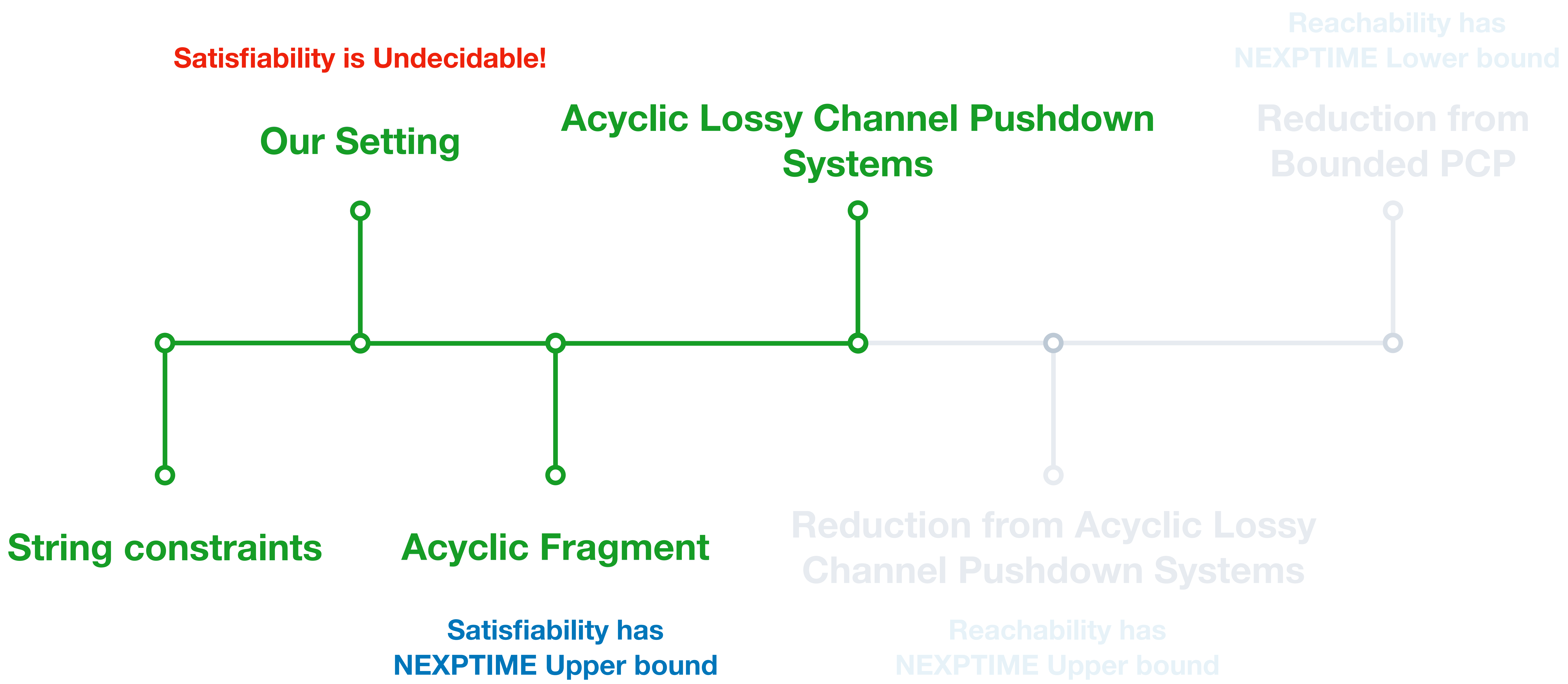
[Atig et. al. 2008]

Control State Reachability Problem

NEXPTIME Upper bound

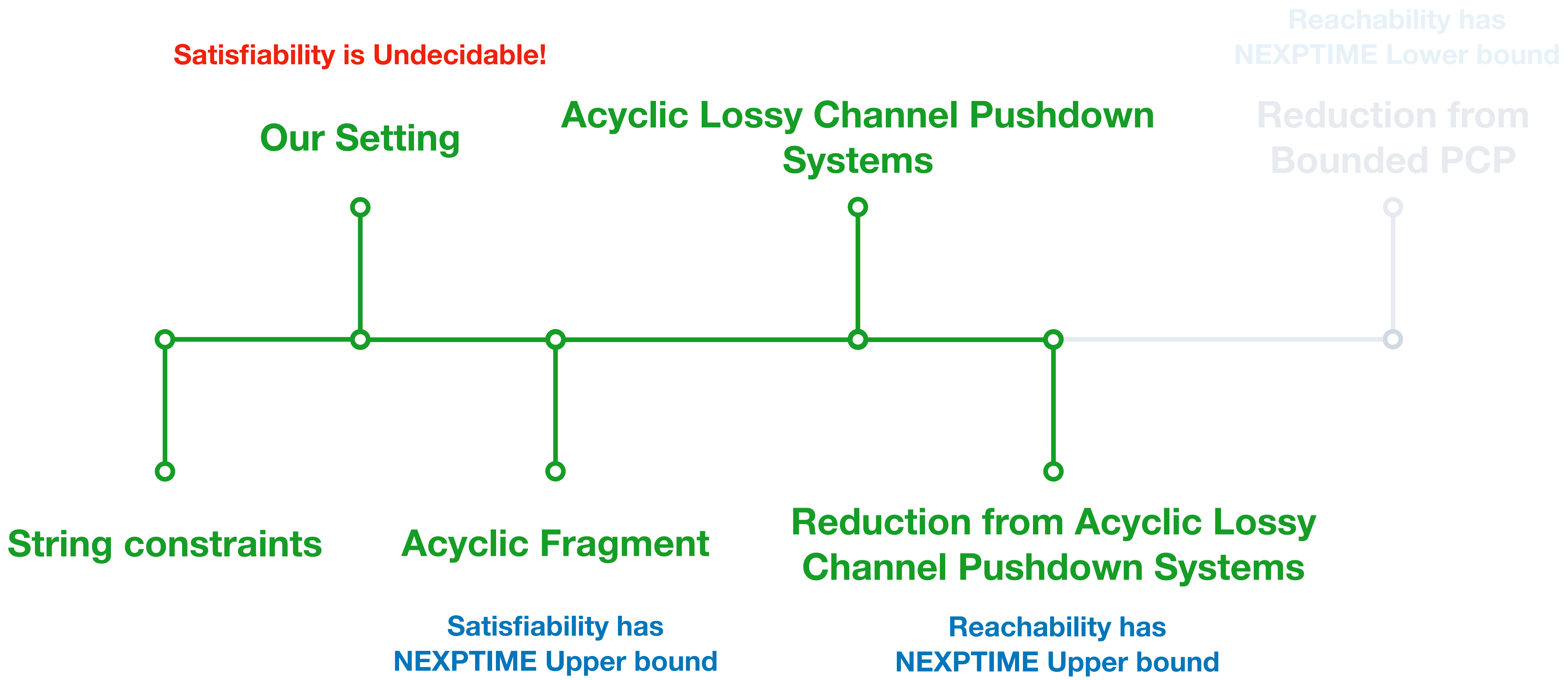
Mohamed Faouzi Atig, Ahmed Bouajjani, and Tayssir Touili. 2008. On the Reachability Analysis of Acyclic Networks of Pushdown Systems. In Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings (Lecture Notes in Computer Science, Vol. 5201), Franck van Breugel and Marsha Chechik (Eds.). Springer. https://doi.org/10.1007/978-3-540- 85361- 9_29

Outline



Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound

Outline


Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound

NEXPTIME Upper bound

NEXPTIME Upper bound

From

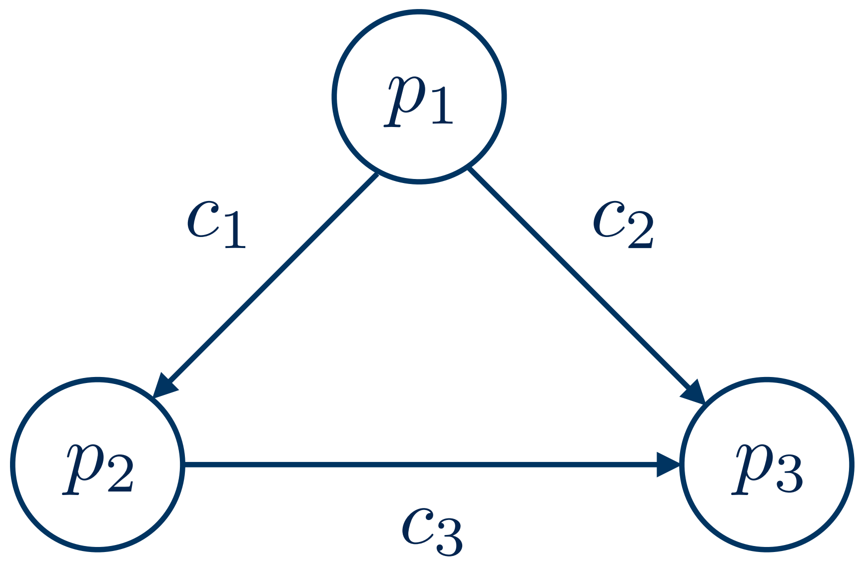
Reachability problem of acyclic LCS

to

Satisfiability problem of acyclic String Constraints

NEXPTIME Upper bound

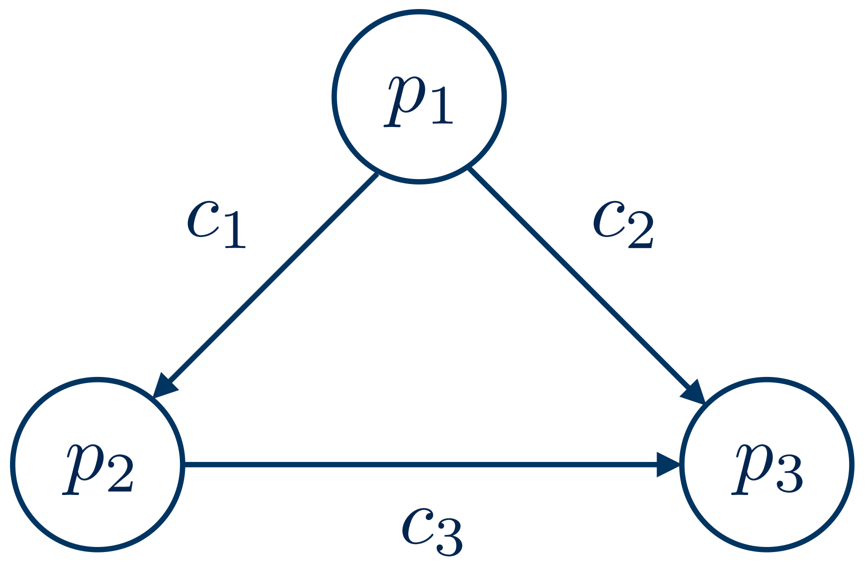
NEXPTIME Upper bound



An Acyclic LCS

NEXPTIME Upper bound

Equivalent Acyclic String constraints

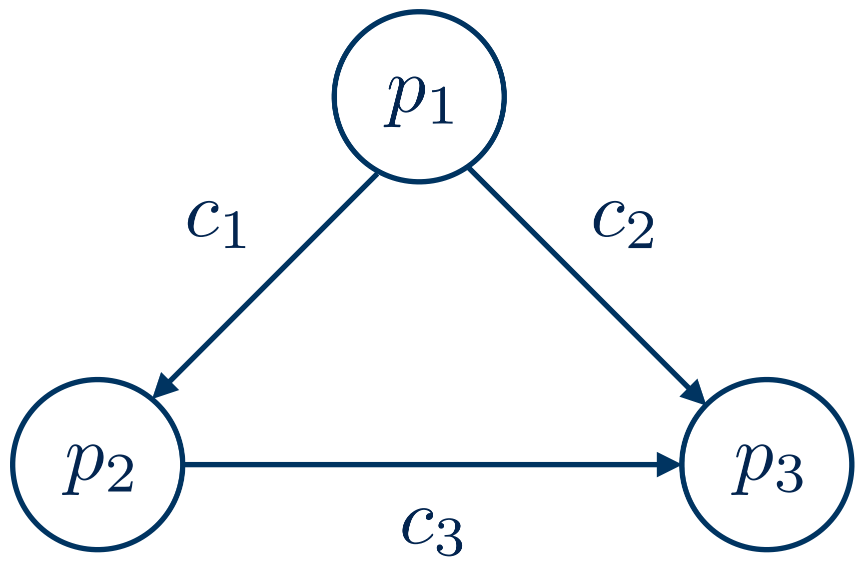


An Acyclic LCS

NEXTPTIME Upper bound

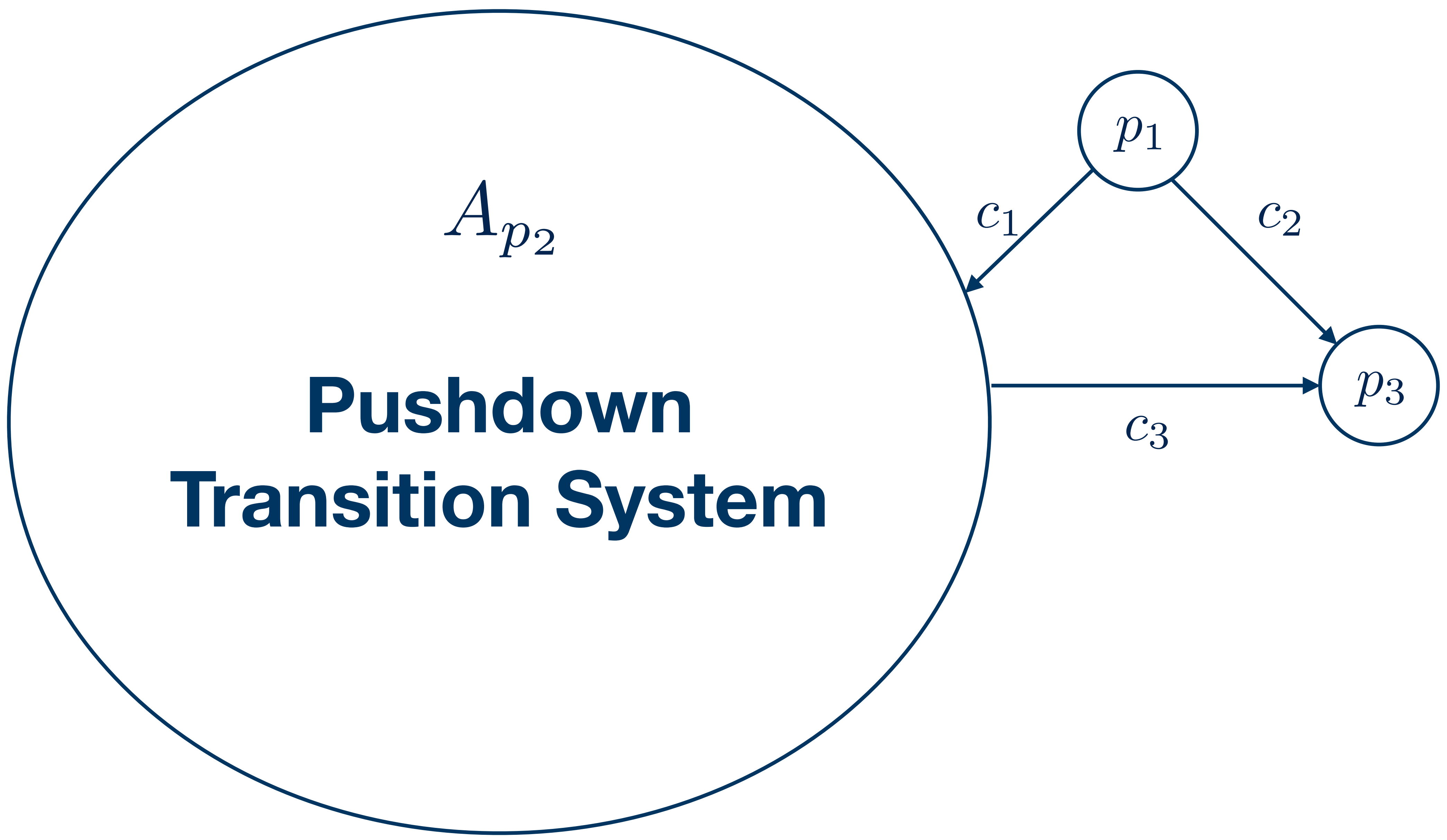
Equivalent Acyclic String constraints

Variable set V = $\{\text{one variable for each process}\}$

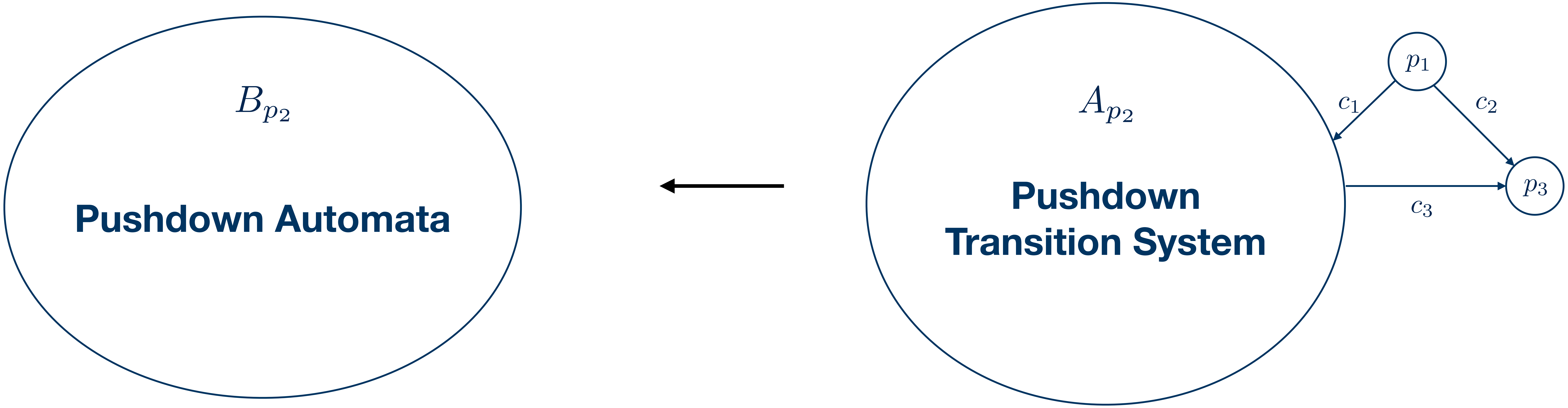


An Acyclic LCS

NEXPTIME Upper bound

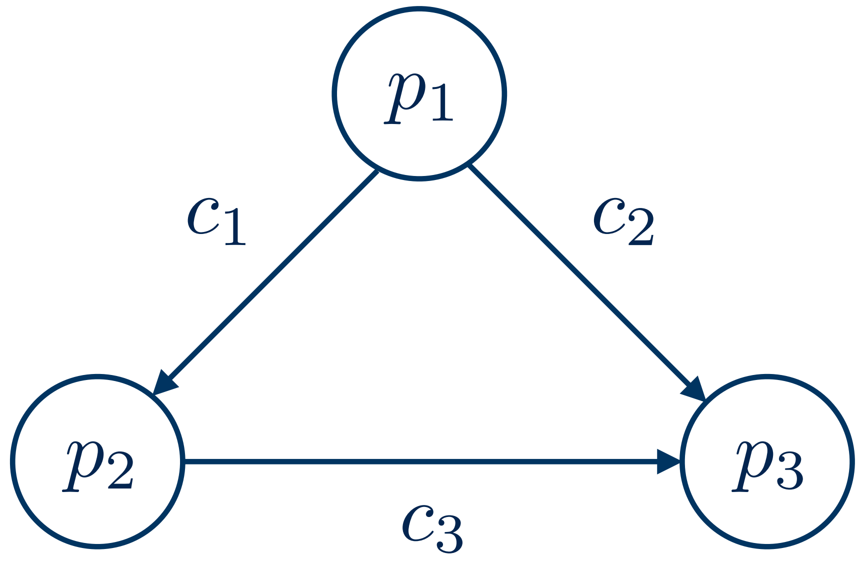


NEXPTIME Upper bound



NEXPTIME Upper bound

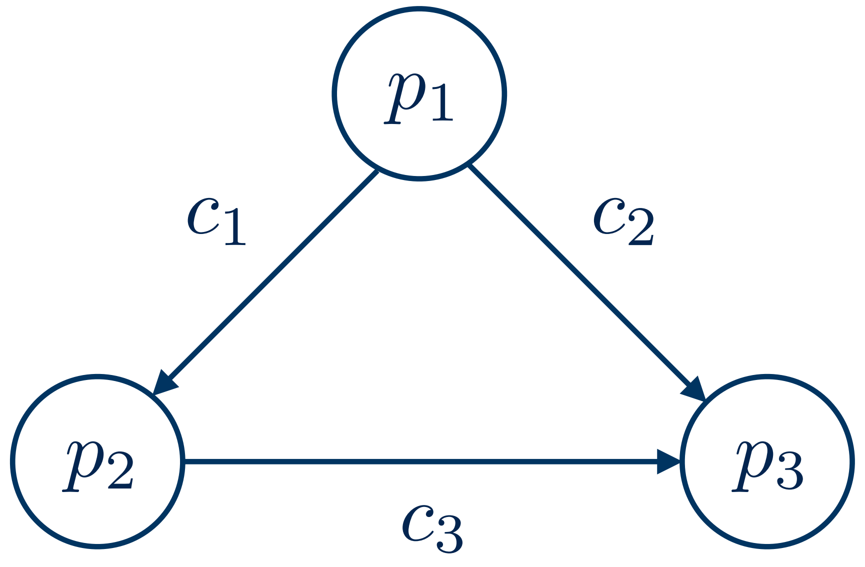
NEXPTIME Upper bound



An Acyclic LCS

NEXPTIME Upper bound

Subword ordered Constraints:

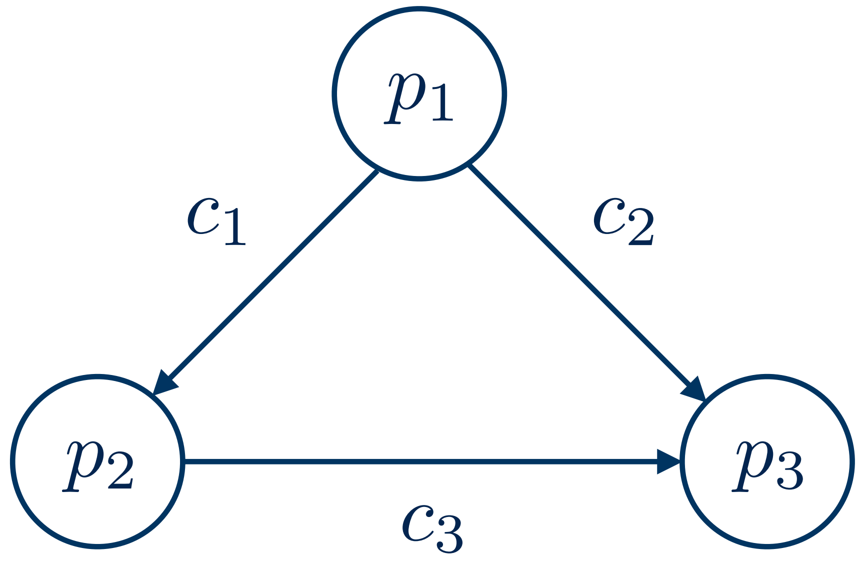


An Acyclic LCS

NEXPTIME Upper bound

Subword ordered Constraints:

For each channel c, $\texttt{Reader}(c) = p$ $\texttt{Writer}(c) = q$



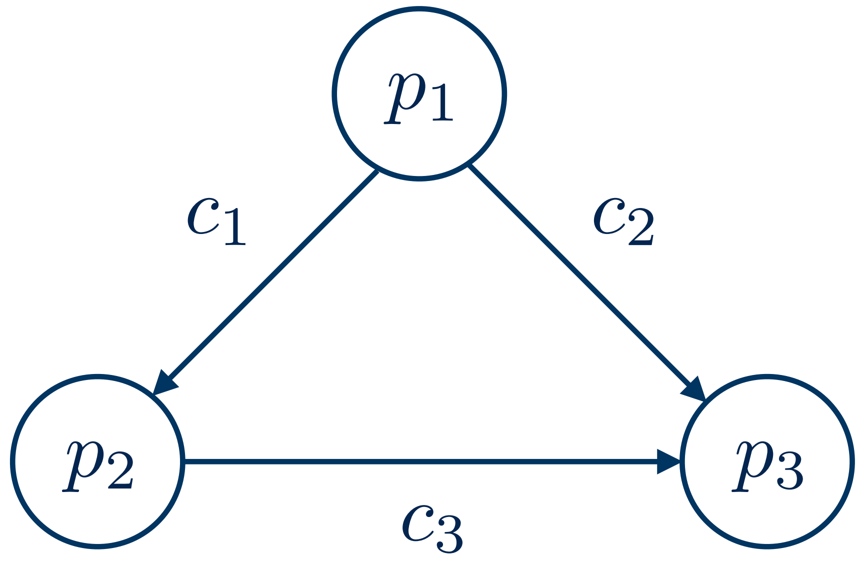
An Acyclic LCS

NEXPTIME Upper bound

Subword ordered Constraints:

For each channel c, $\texttt{Reader}(c) = p$ $\texttt{Writer}(c) = q$

$$x_p \preceq x_q$$



An Acyclic LCS

NEXPTIME Upper bound

Equivalent Acyclic String constraints

Variable set V = $\{\text{one variable for each process}\}$

Membership constraints :

$$x_p = \mathcal{L}(B_p)$$

Subword ordered Constraints:

For each channel c, $\texttt{Reader}(c) = p$ $\texttt{Writer}(c) = q$

$$x_p \preceq x_q$$

NEXPTIME Upper bound

Equivalent Acyclic String constraints

Variable set V = $\{\text{one variable for each process}\}$

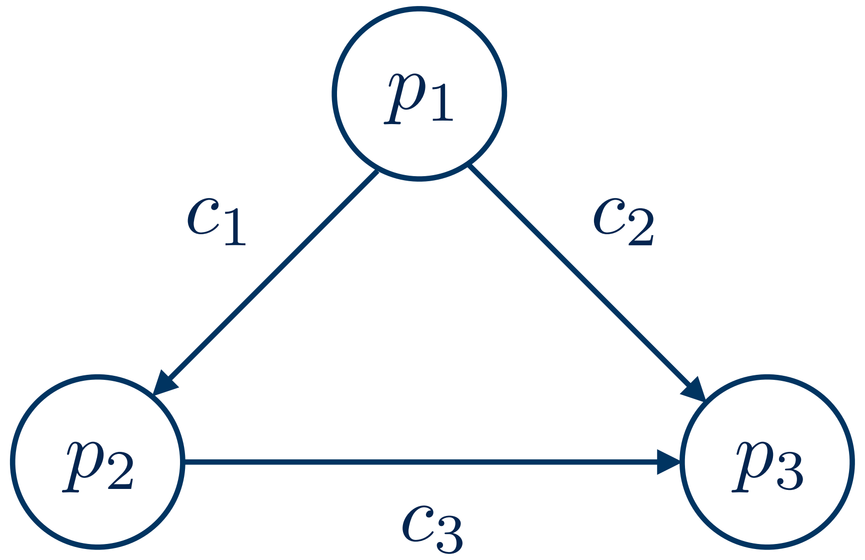
Membership constraints :

$$x_p = \mathcal{L}(B_p)$$

Subword ordered Constraints:

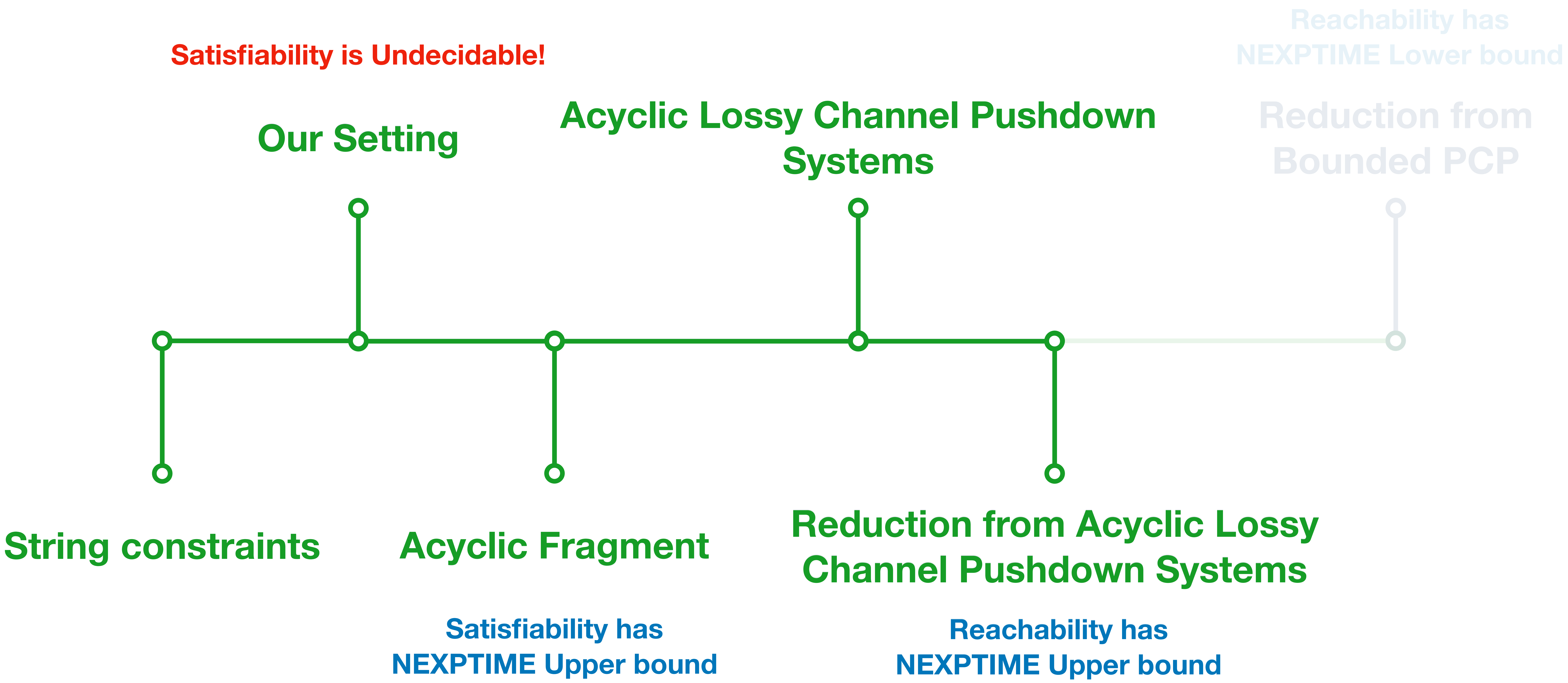
For each channel c, $\texttt{Reader}(c) = p$ $\texttt{Writer}(c) = q$

$$x_p \preceq x_q$$

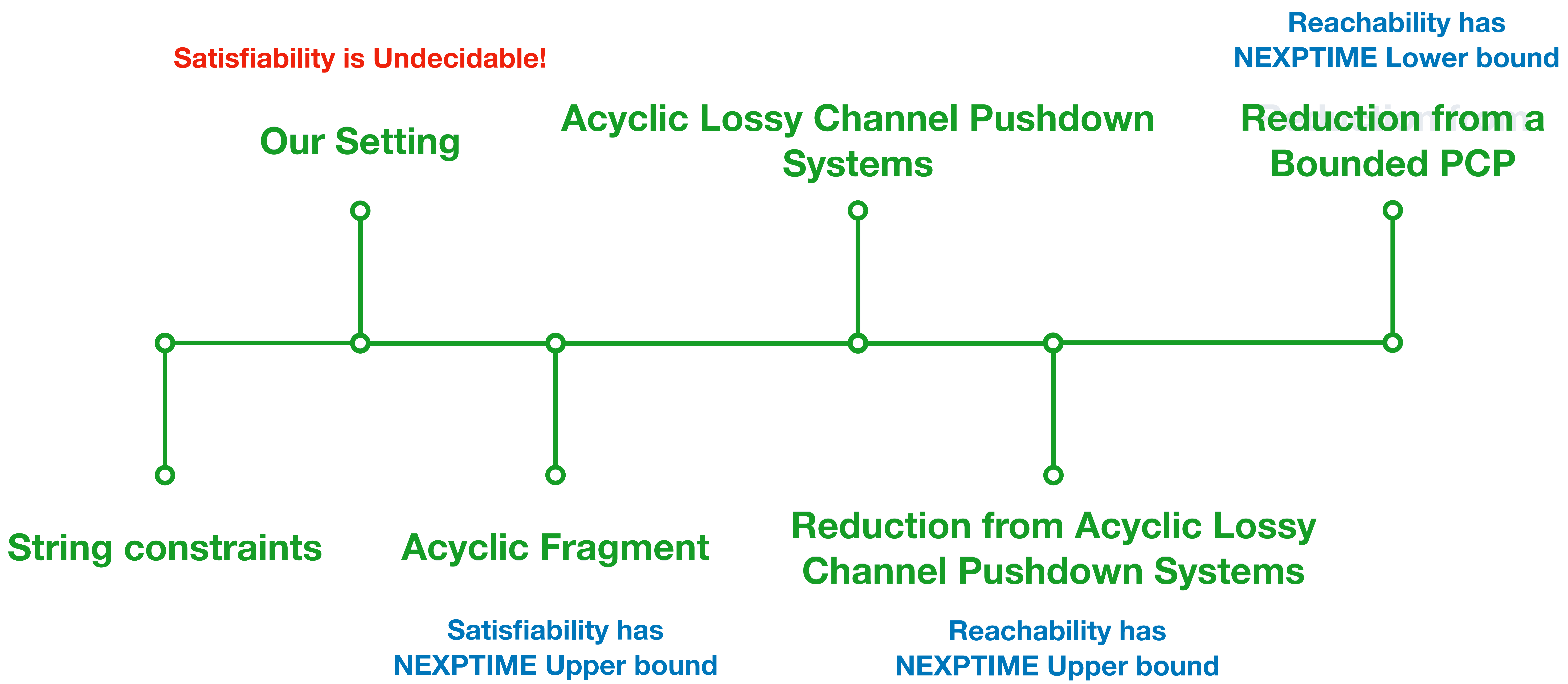


An Acyclic LCS

Outline


Satisfiability is Undecidable!
Our Setting
Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Lower bound
Reduction from Bounded PCP
String constraints
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound

Outline


String constraints
Satisfiability is Undecidable!
Our Setting
Acyclic Fragment
Satisfiability has NEXPTIME Upper bound
Acyclic Lossy Channel Pushdown Systems
Reduction from Acyclic Lossy Channel Pushdown Systems
Reachability has NEXPTIME Upper bound
Reachability has NEXPTIME Lower bound
Reduction from a Bounded PCP

NEXPTIME Lower bound

NEXPTIME Lower bound

Bounded variant of PCP problem - NEXPTIME complete.

NEXPTIME Lower bound

Bounded variant of PCP problem - NEXPTIME complete.

Reduction from it to the Reachability problem for acyclic LCS

NEXPTIME Lower bound

NEXPTIME Lower bound

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

Given two equi-dimensional vector of strings U and V:

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

Given two equi-dimensional vector of strings U and V:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

Given two equi-dimensional vector of strings U and V:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

and an integer ℓ given in binary

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

Given two equi-dimensional vector of strings U and V:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

and an integer ℓ given in binary

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k$

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$$
$$=$$
$$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

NEXPTIME Lower bound

Bounded PCP instance over alphabet Σ:

Given two equi-dimensional vector of strings U and V:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

and an integer ℓ given in binary

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k$

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$$

=

$$v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

AND

$$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$$

?

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at most 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

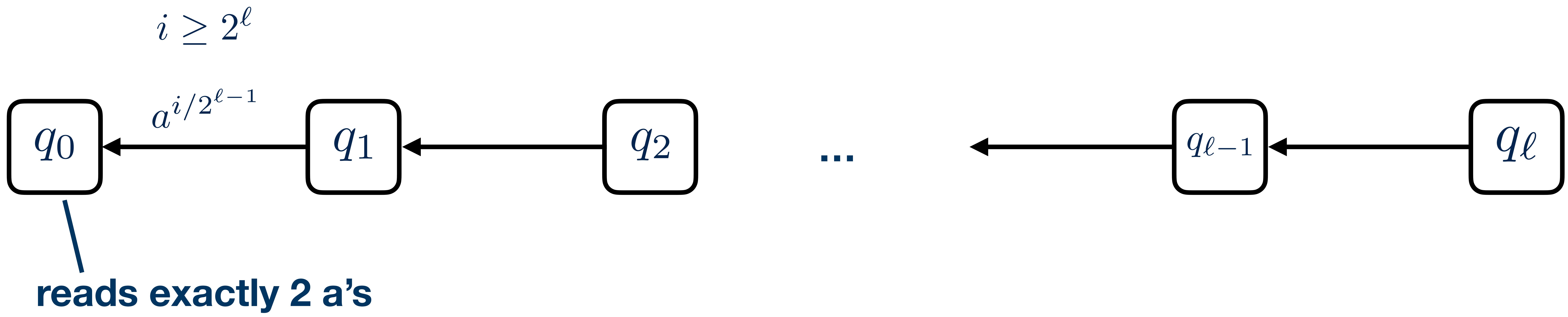
NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count at least 2^ℓ

NEXPTIME lower bound

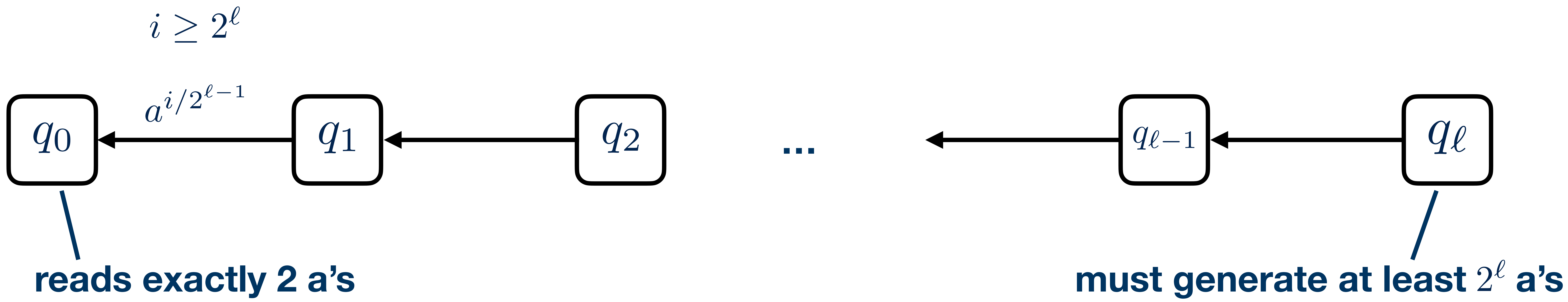
To give a reduction, we look at the following gadgets:



Gadget to count at least 2^ℓ

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:



Gadget to count at least 2^ℓ

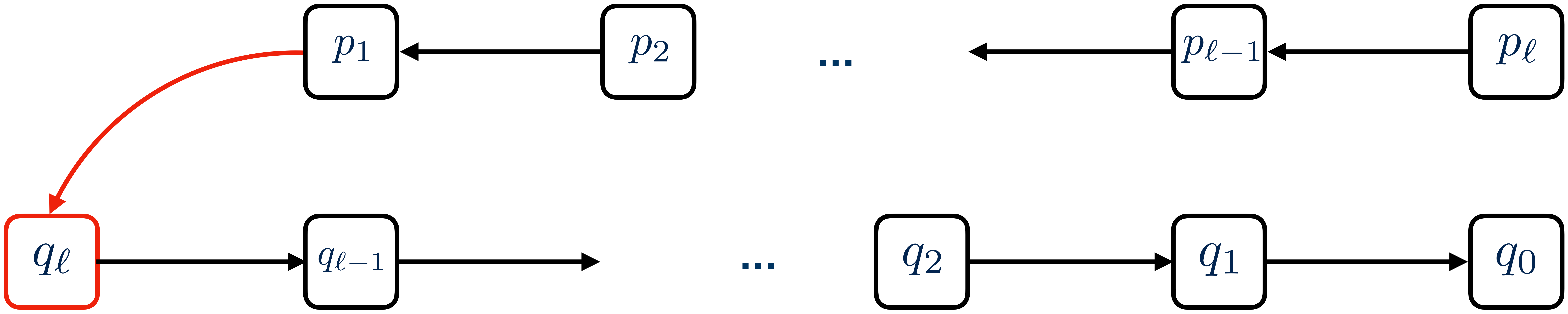
NEXPTIME lower bound

To give a reduction, we look at the following gadgets:

Gadget to count exactly 2^ℓ

NEXPTIME lower bound

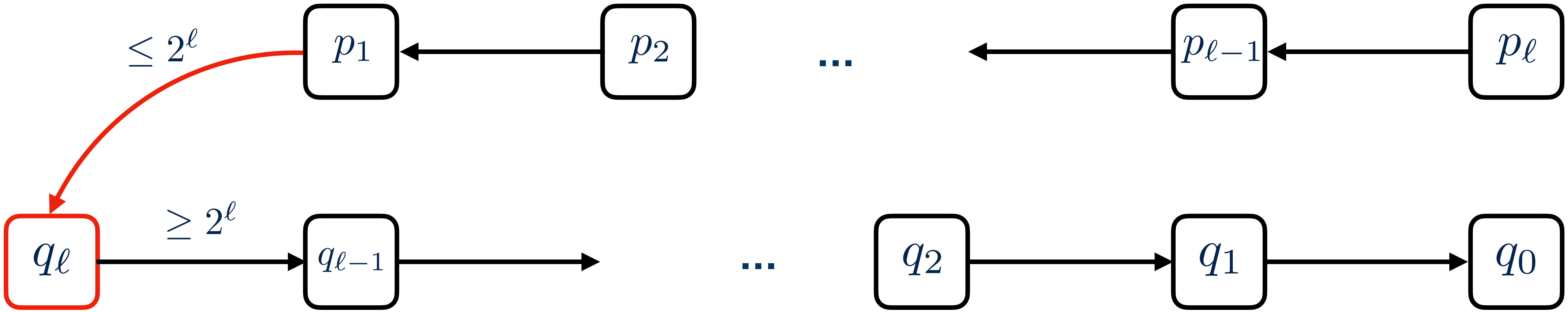
To give a reduction, we look at the following gadgets:



Gadget to count exactly 2^ℓ

NEXPTIME lower bound

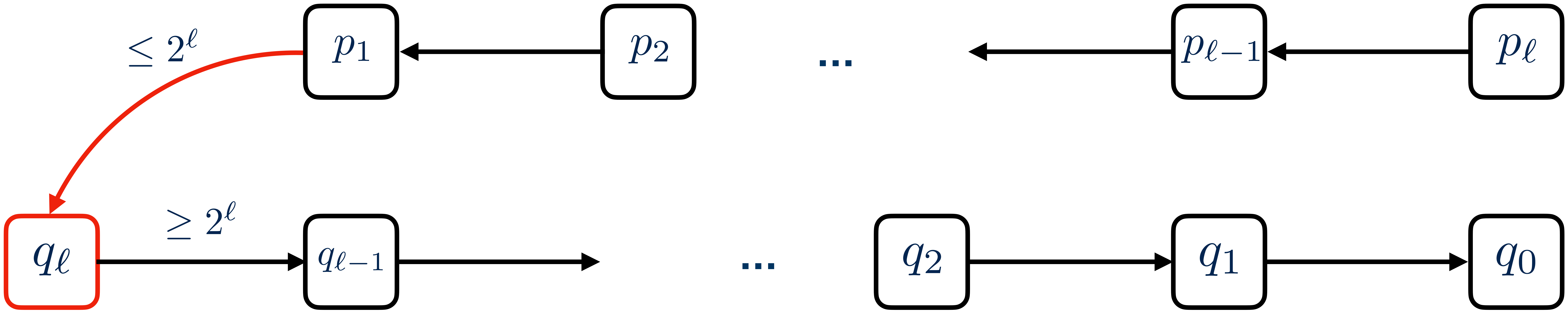
To give a reduction, we look at the following gadgets:



Gadget to count exactly 2^ℓ

NEXPTIME lower bound

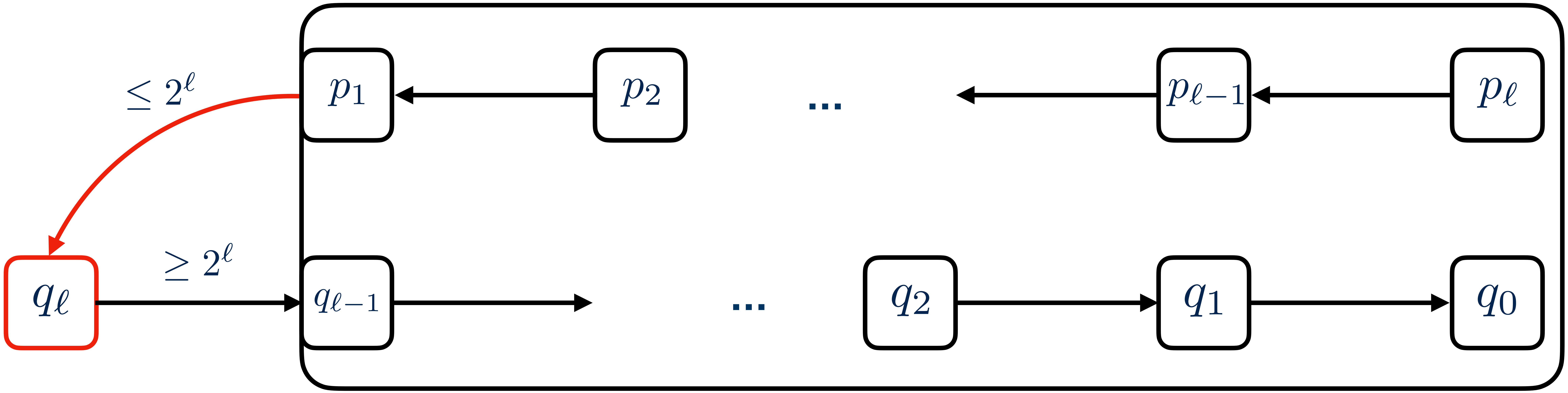
To give a reduction, we look at the following gadgets:



Topology for the PCP simulation

NEXPTIME lower bound

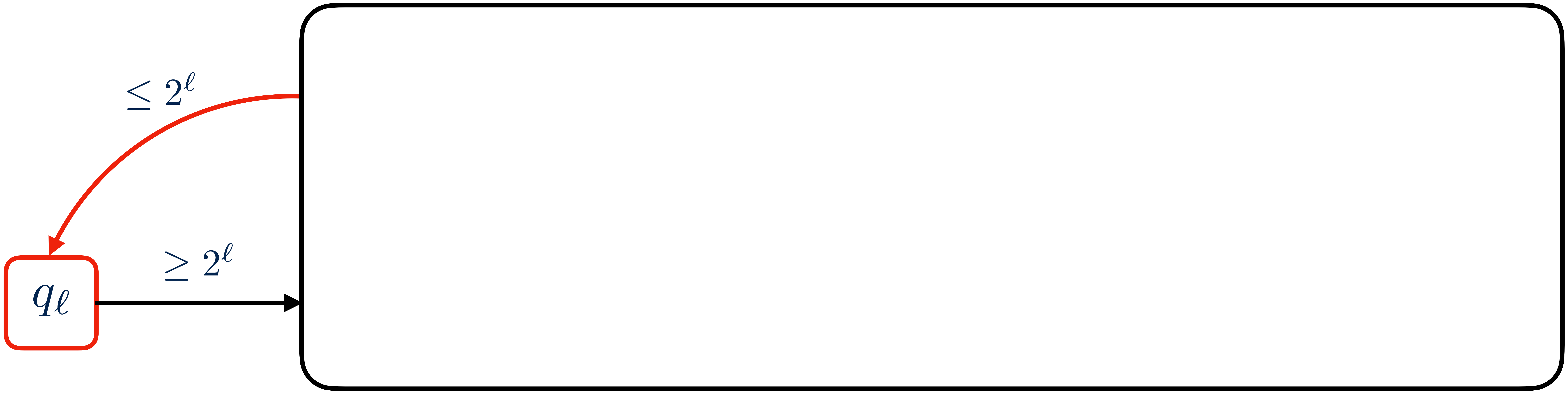
To give a reduction, we look at the following gadgets:



Topology for the PCP simulation

NEXPTIME lower bound

To give a reduction, we look at the following gadgets:



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

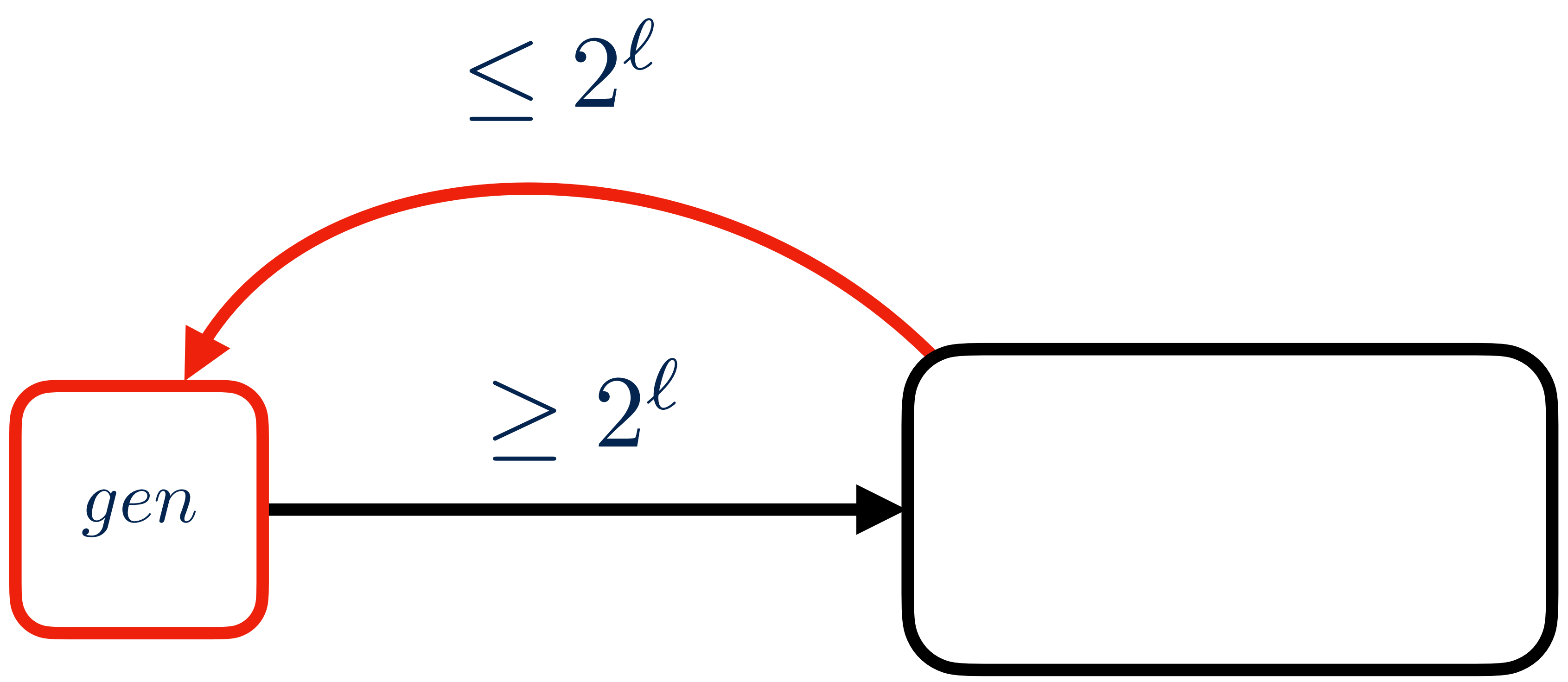
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

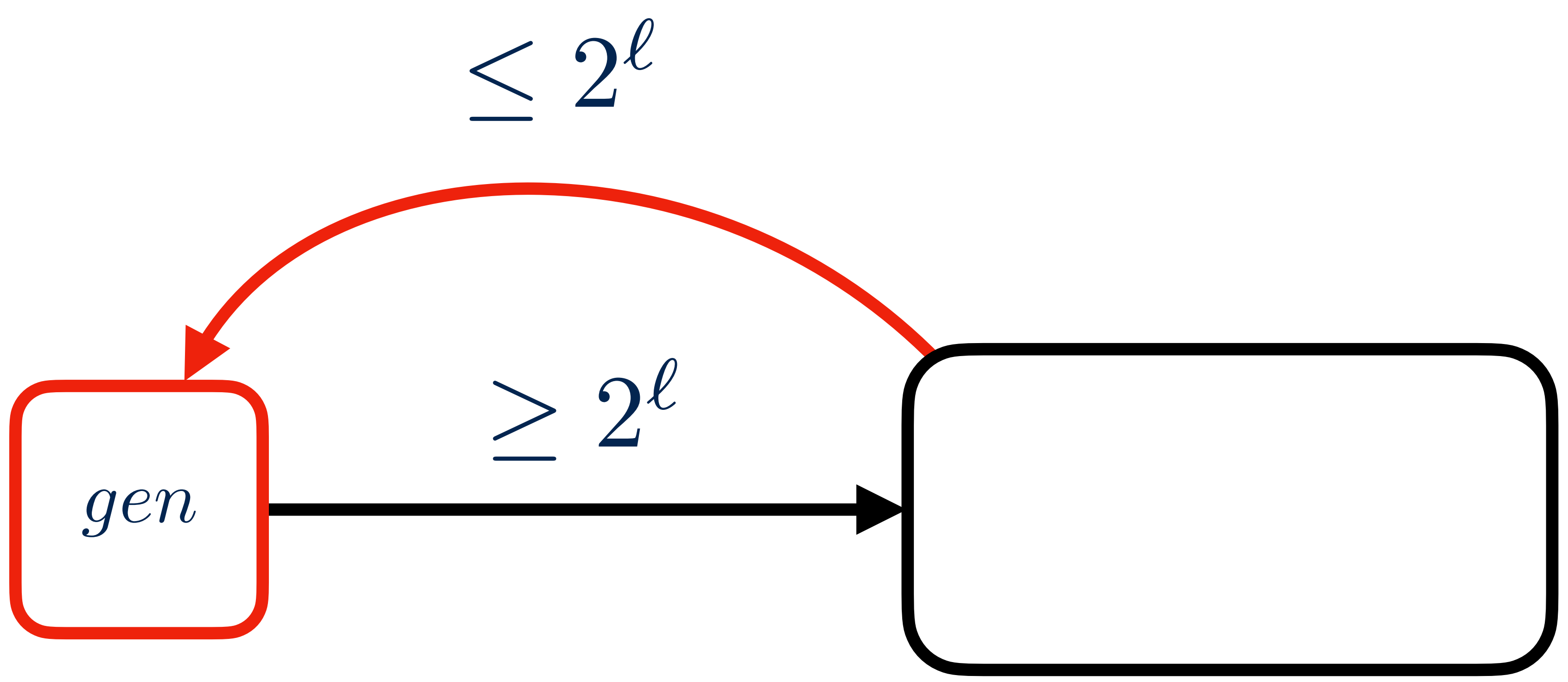
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$ **=** $v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

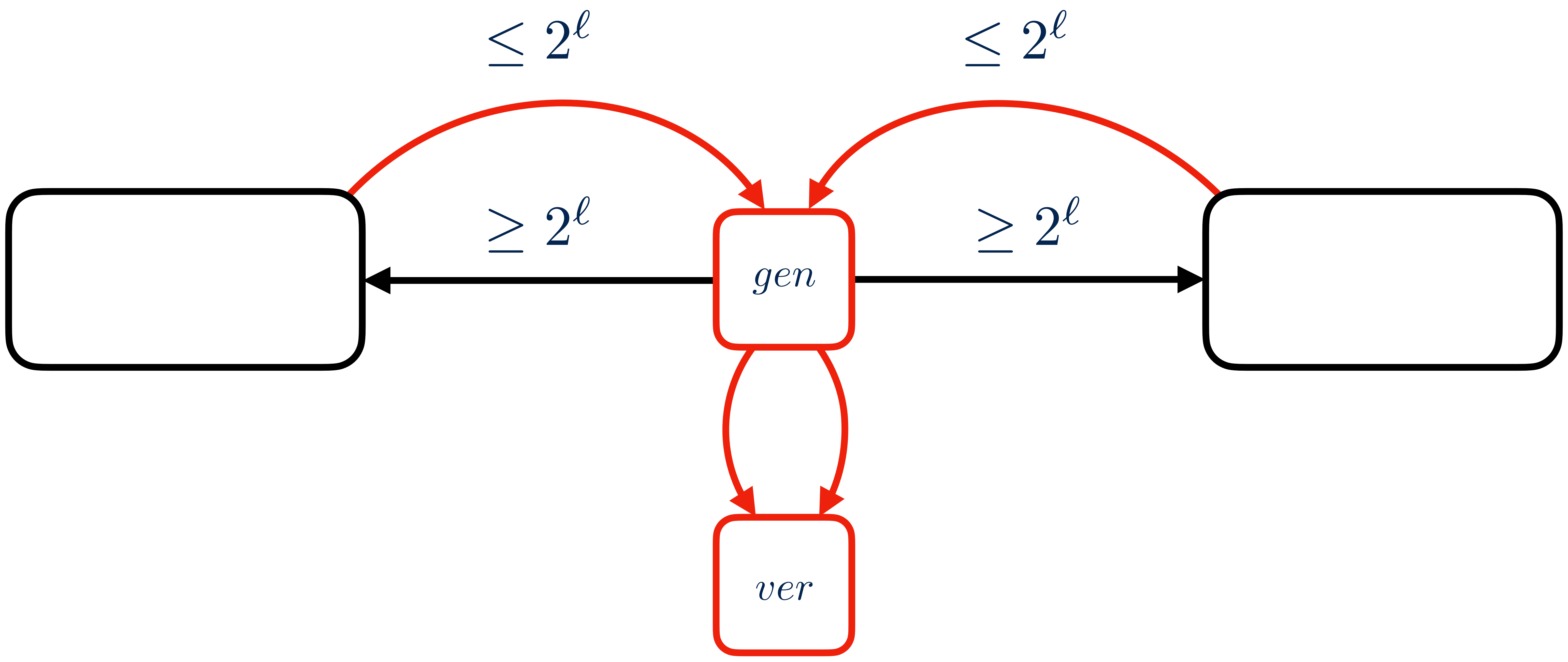
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

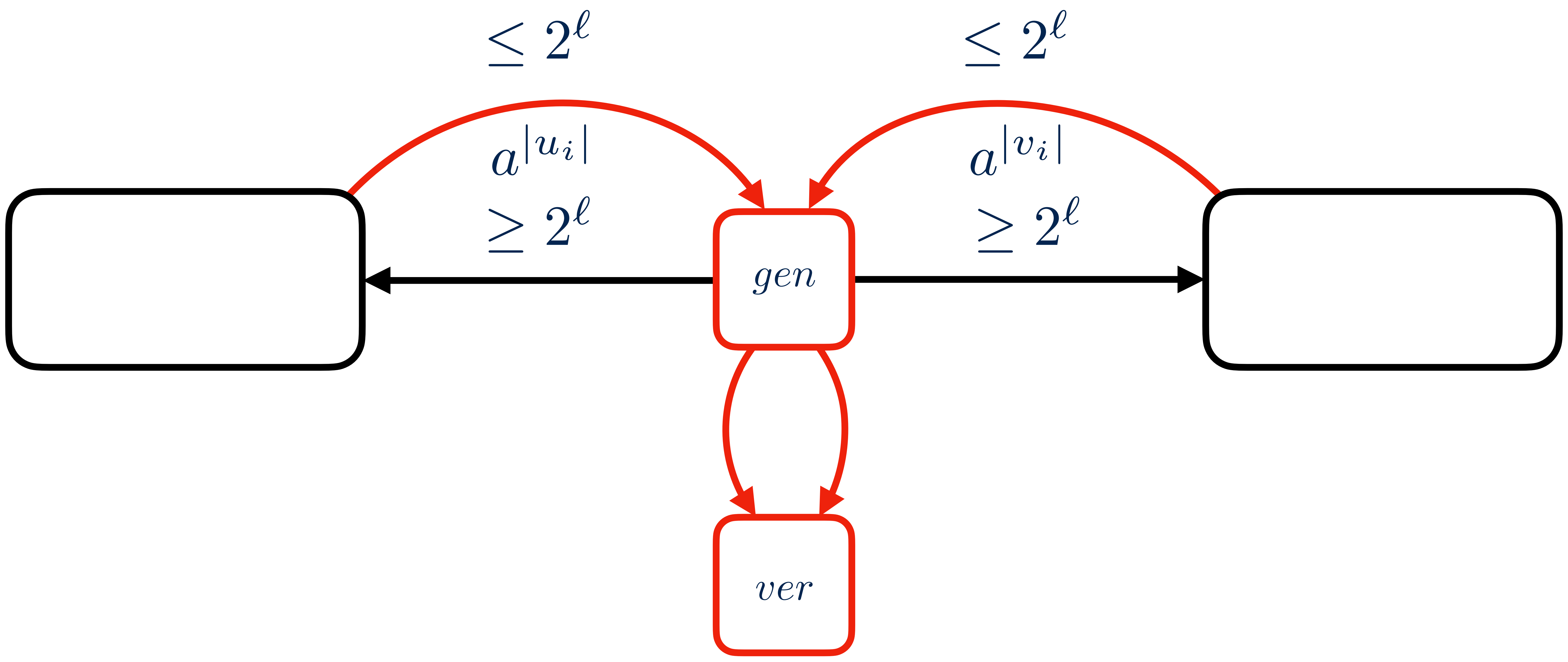
sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k$

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

AND

$$\mathtt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \mathtt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell \textbf{ ?}$$



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

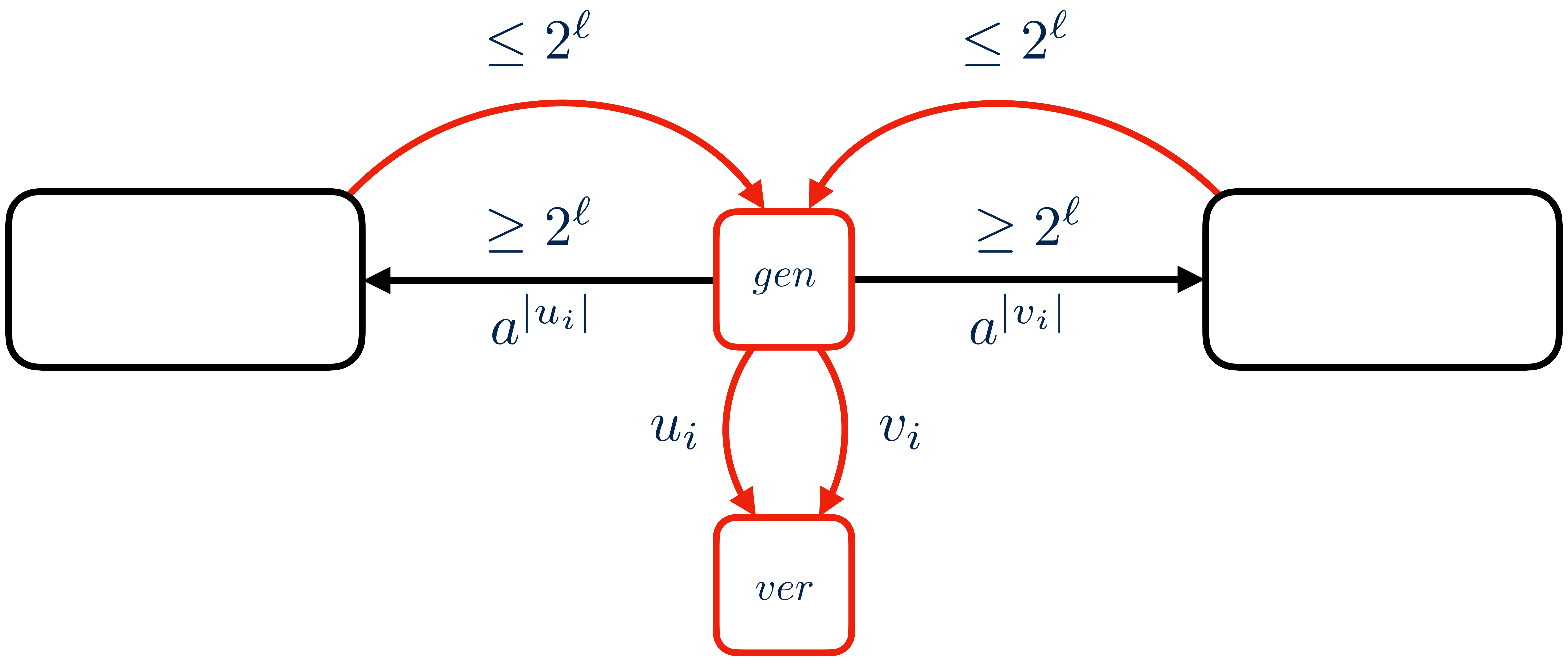
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists$ $i_1, i_2, \ldots, i_k$ $\quad u_{i_1} \cdot u_{i_2} \cdots u_{i_k}$ = $v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

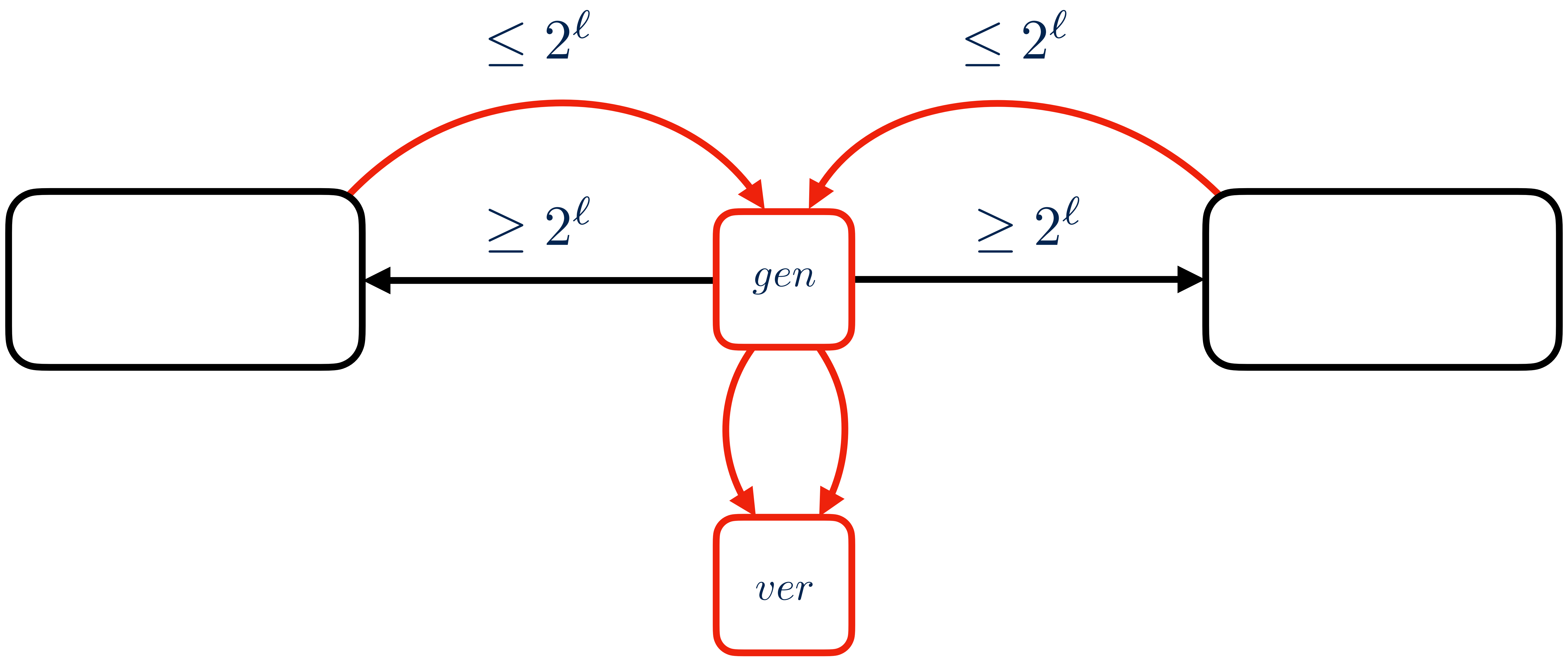
sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k$

$$u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$$

AND

$\mathtt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \mathtt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

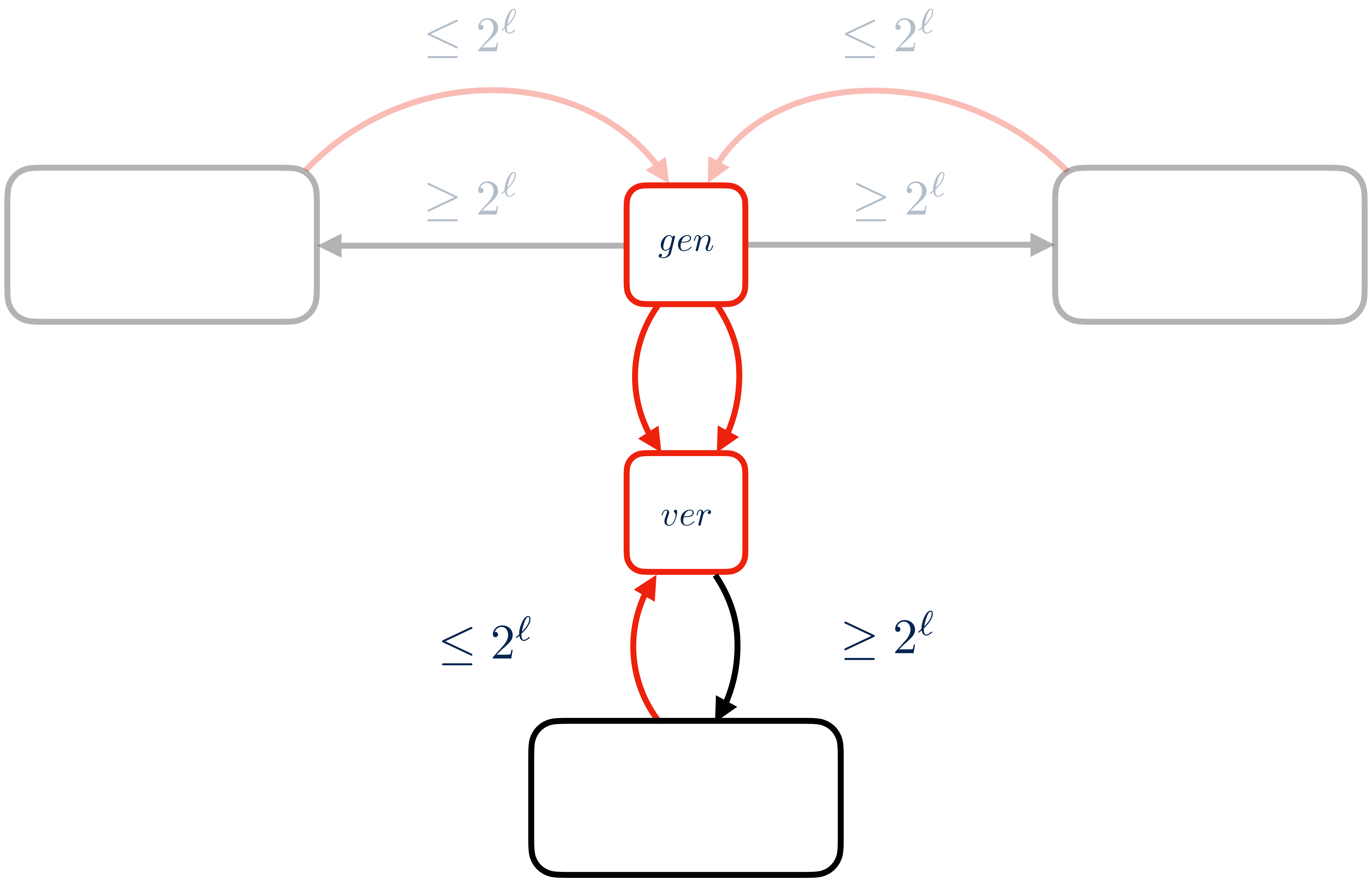
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\mathtt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \mathtt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

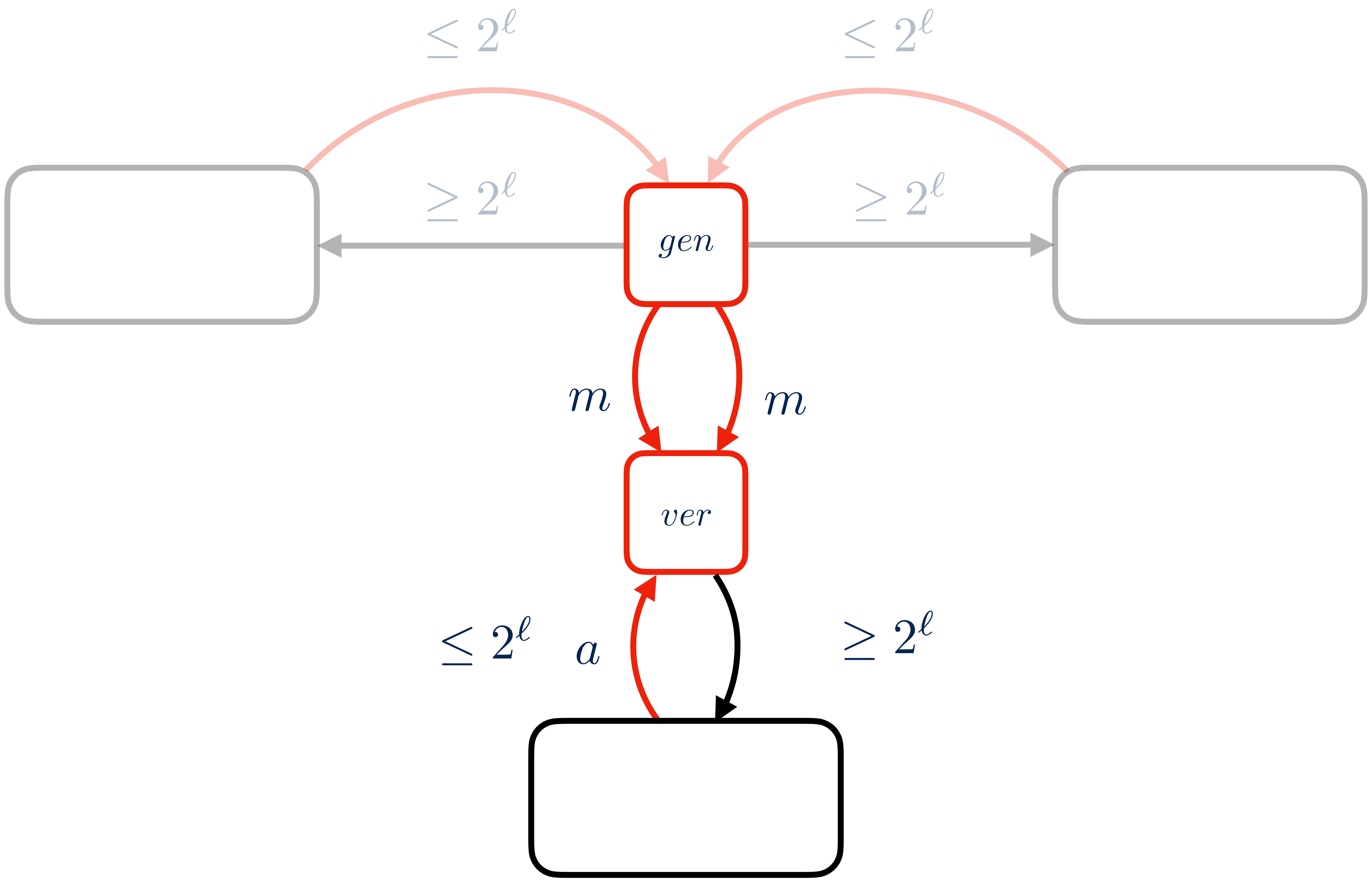
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

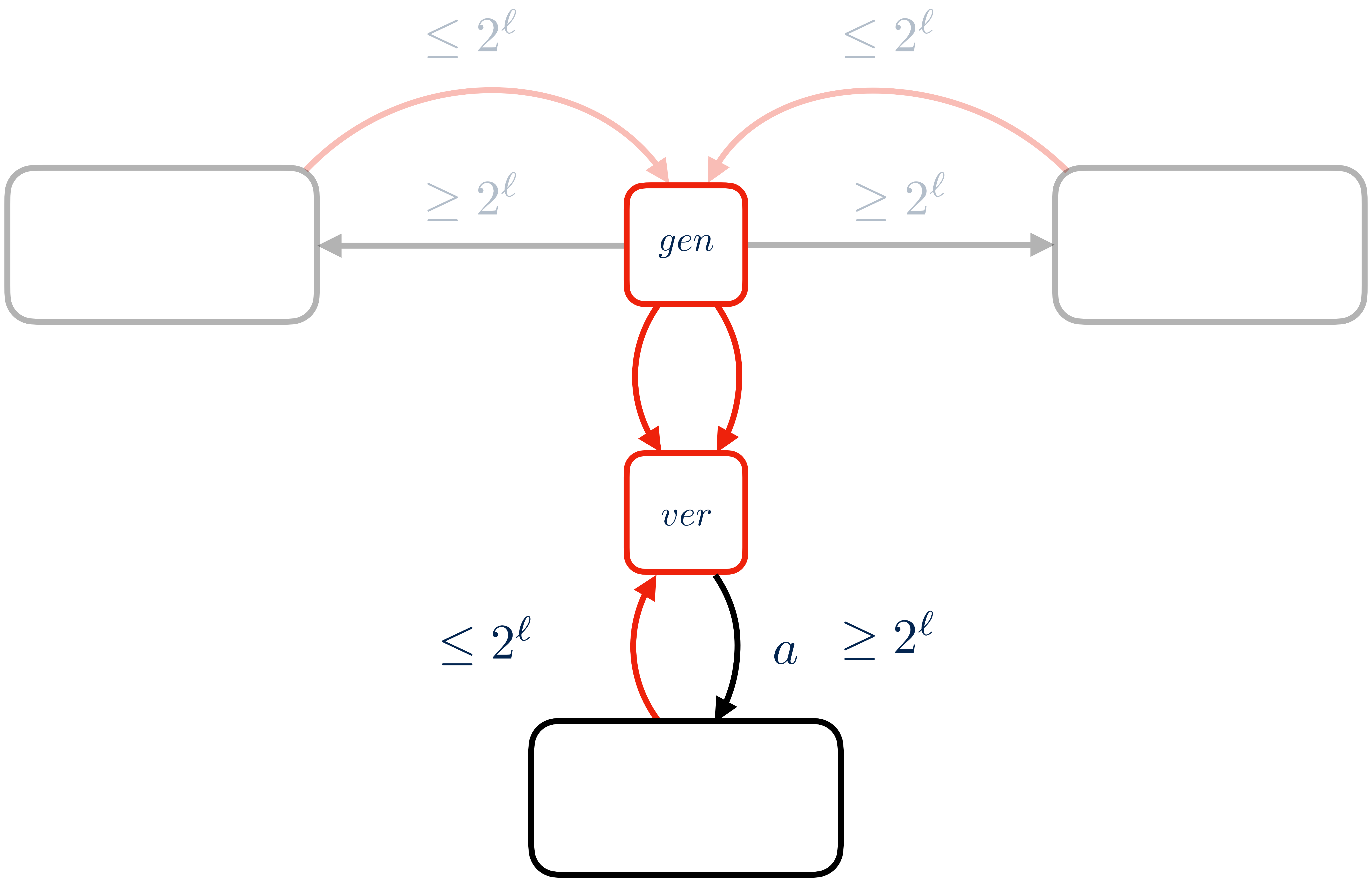
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

NEXPTIME lower bound

PCP instance over alphabet Σ:

Given two vector of strings U and V, each having n elements:

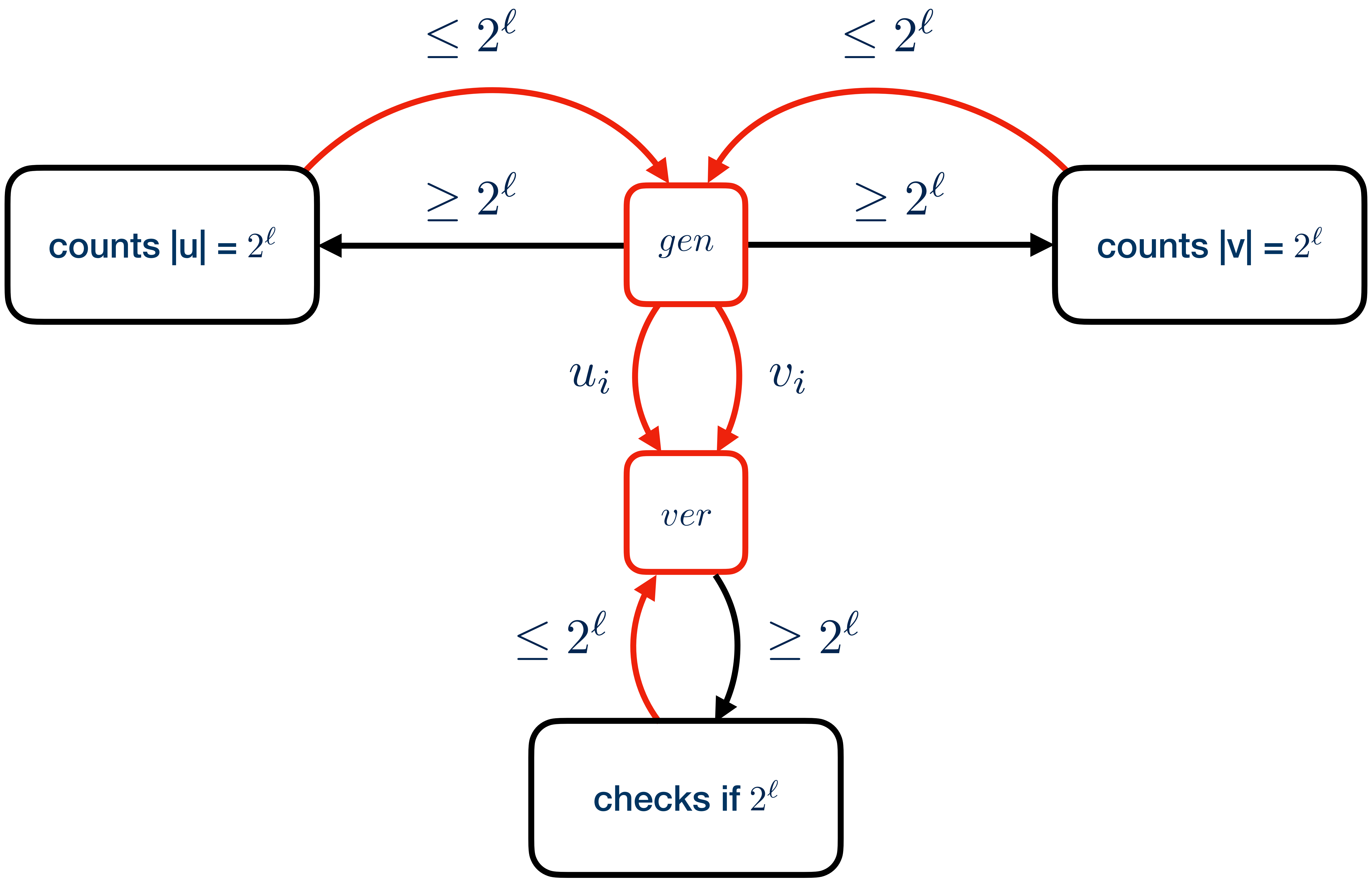
	1	2	...	n
U	u_1	u_2	...	u_n
V	v_1	v_2	...	v_n

sequence of indices

$\exists \quad i_1, i_2, \ldots, i_k \qquad u_{i_1} \cdot u_{i_2} \cdots u_{i_k} = v_{i_1} \cdot v_{i_2} \cdots v_{i_k}$

AND

$\texttt{len}(u_{i_1} \cdot u_{i_2} \cdots u_{i_k}) = \texttt{len}(v_{i_1} \cdot v_{i_2} \cdots v_{i_k}) = 2^\ell$ **?**



Topology for the PCP simulation

Summary

Summary

1. String constraints

Summary

1. String constraints

2. Context-free String Constraints with Subword Order — Satisfiability is Undecidable!

Summary

1. String constraints

2. Context-free String Constraints with Subword Order — Satisfiability is Undecidable!

3. Acyclic Fragment — Satisfiability has NEXPTIME Upper bound

Summary

1. String constraints

2. Context-free String Constraints with Subword Order

   Satisfiability is Undecidable!

3. Acyclic Fragment

   Satisfiability has NEXPTIME Upper bound

4. Acyclic Lossy Channel Pushdown Systems

Summary

1. String constraints

2. Context-free String Constraints with Subword Order — Satisfiability is Undecidable!

3. Acyclic Fragment — Satisfiability has NEXPTIME Upper bound

4. Acyclic Lossy Channel Pushdown Systems

5. From Acyclic Lossy Channel Pushdown Systems to Acyclic String constraints — Reachability has NEXPTIME Upper bound

Summary

1.	String constraints	
2.	Context-free String Constraints with Subword Order	Satisfiability is Undecidable!
3.	Acyclic Fragment	Satisfiability has NEXPTIME Upper bound
4.	Acyclic Lossy Channel Pushdown Systems	
5.	From Acyclic Lossy Channel Pushdown Systems to Acyclic String constraints	Reachability has NEXPTIME Upper bound
6.	From Bounded PCP to Acyclic Lossy Channel Pushdown Systems	$\Longrightarrow$ Both Reachability for Acyclic LCS and Satisfiability for Acyclic String constraints are NEXPTIME complete!

Thank You!