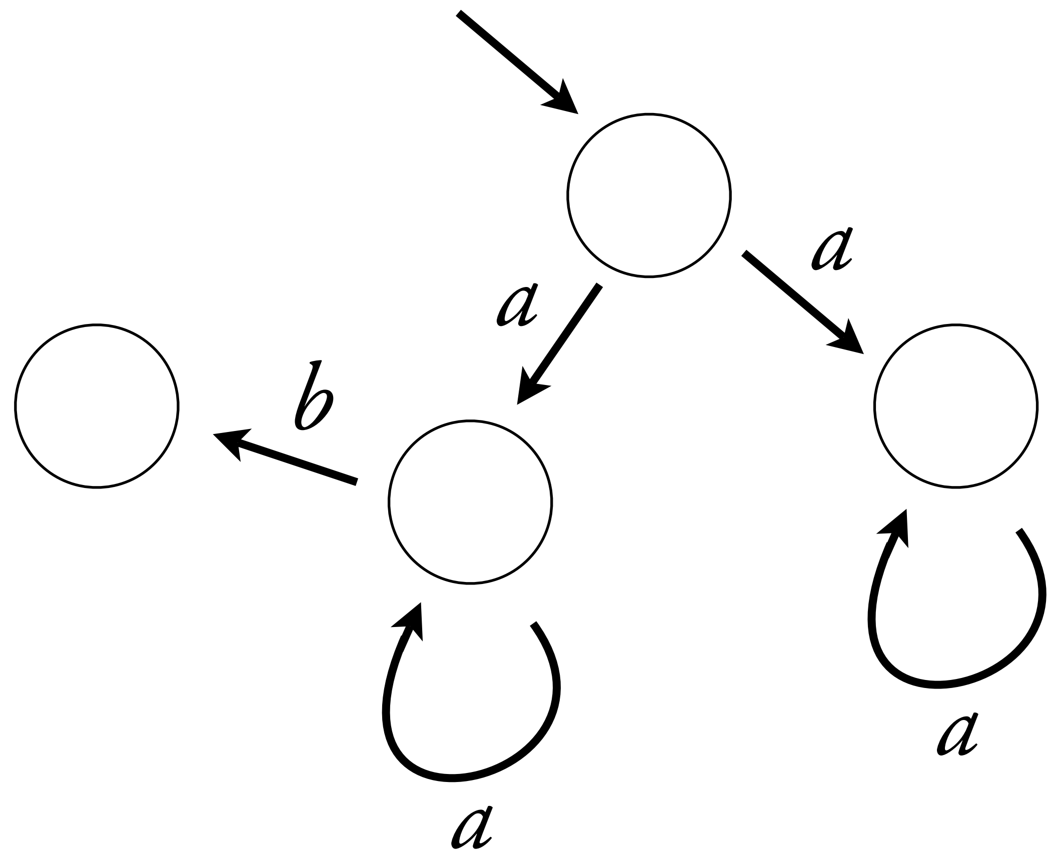
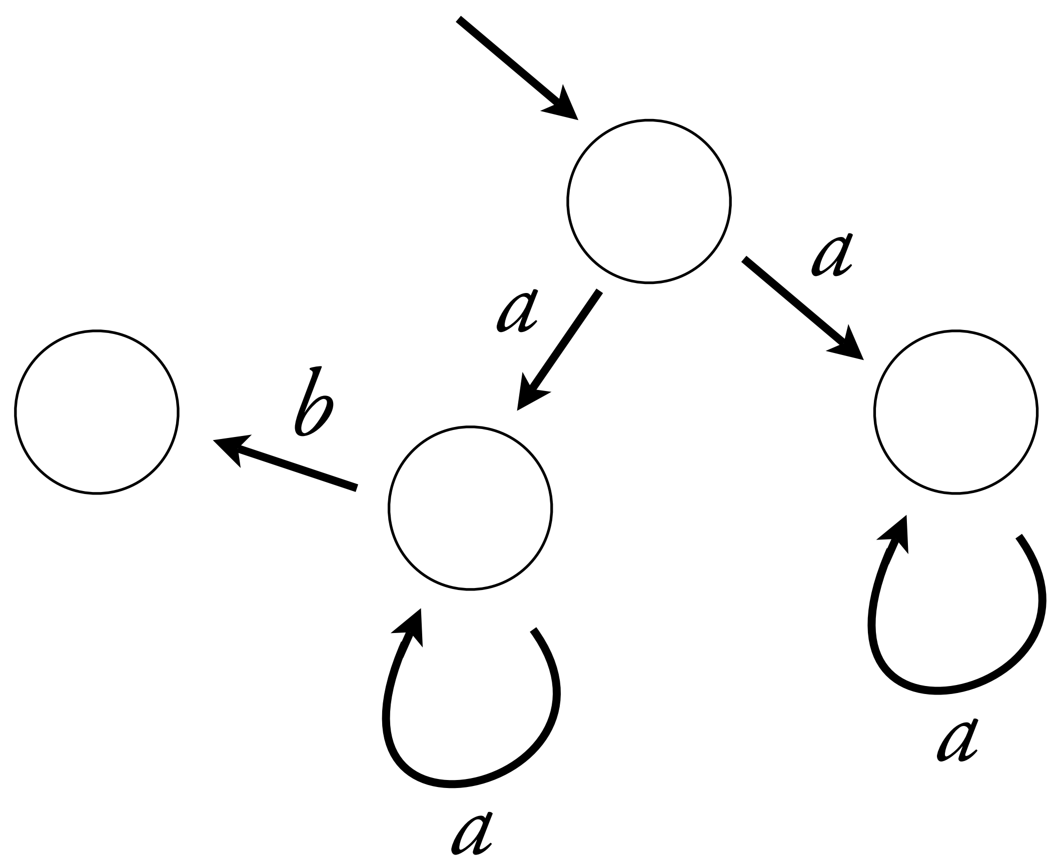
The common fragment of ACTL and CTL

Mikołaj Bojańczyk
Warsaw University

Logics that talk about transition systems



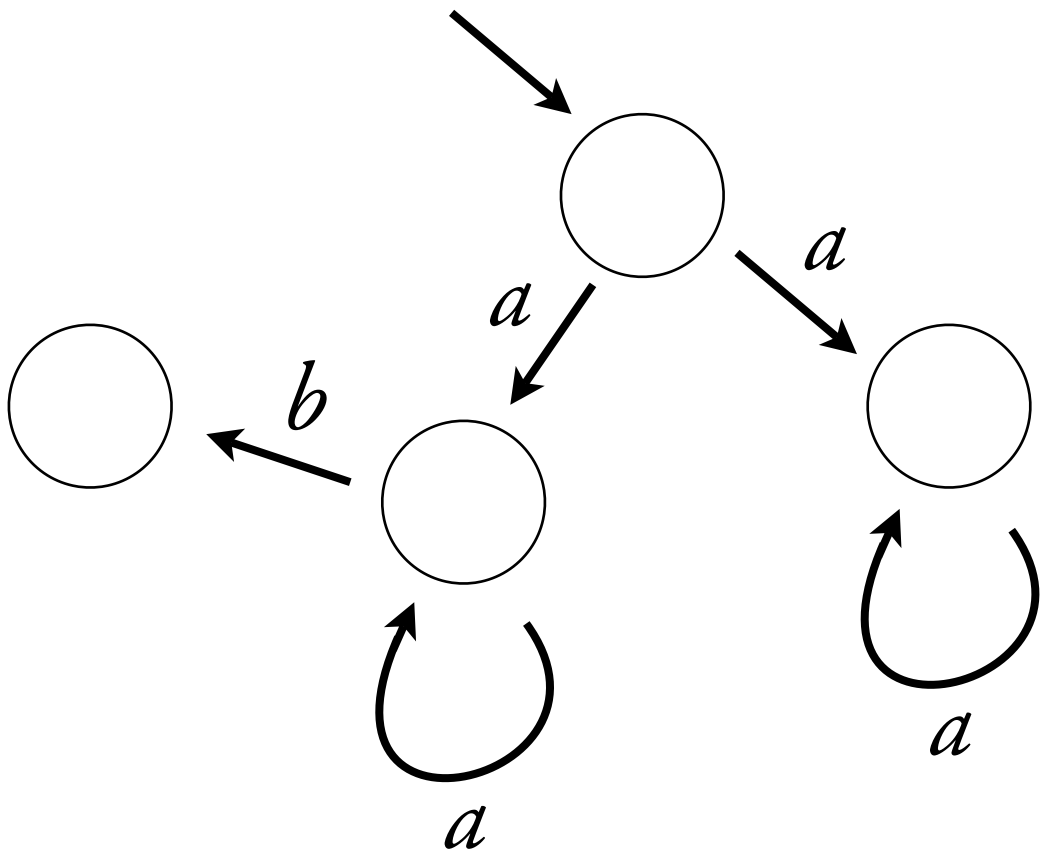
Logics that talk about transition systems



Linear time logic.

every finite path in $a^* + a^*b$

Logics that talk about transition systems



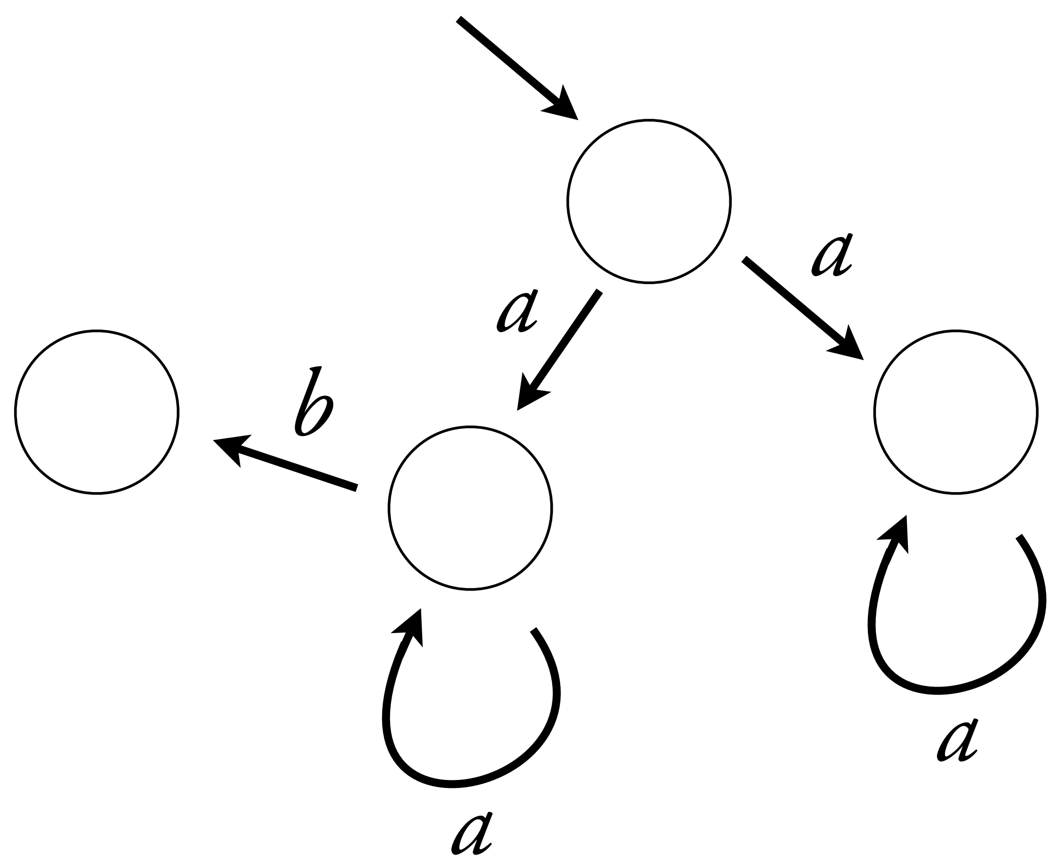
Linear time logic.

every finite path in $a^* + a^*b$

Branching time logic.

after every a^* prefix, b is possible

Logics that talk about transition systems



Linear time logic.

every finite path in $a^* + a^*b$

has the same paths as

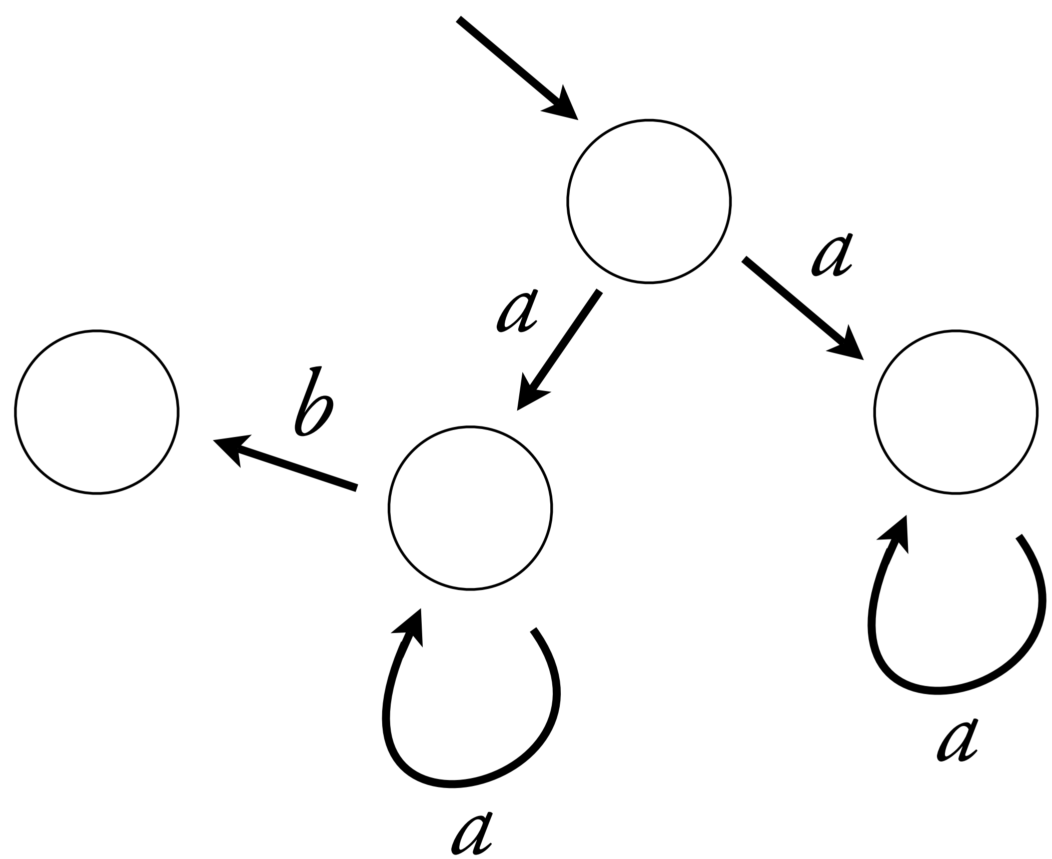
Branching time logic.

after every a^* prefix, b is possible

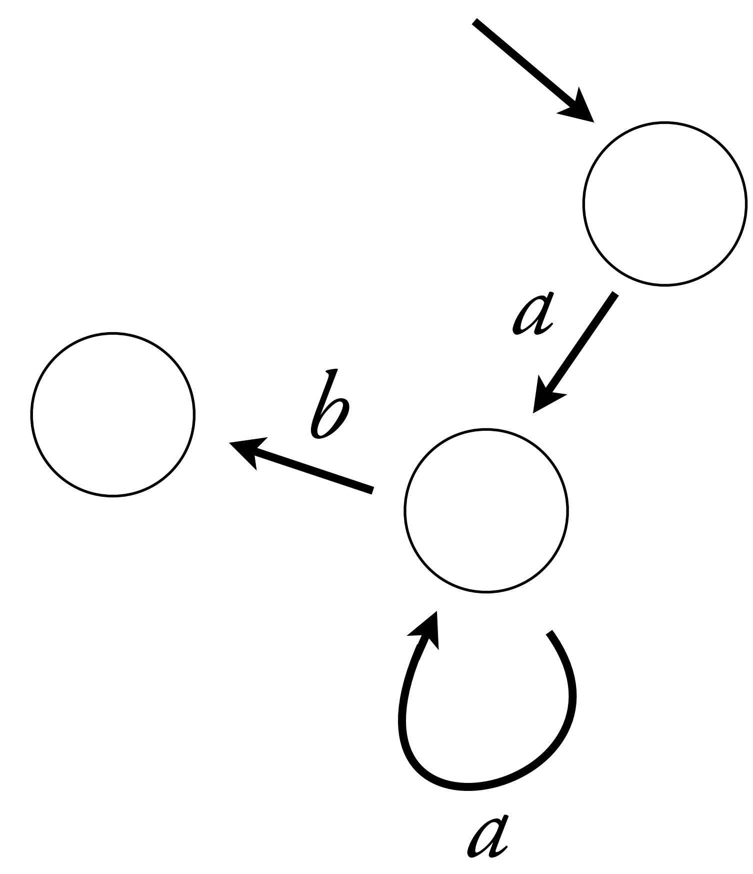
Logics that talk about transition systems

Linear time logic.

every finite path in $a^* + a^*b$



has the same paths as

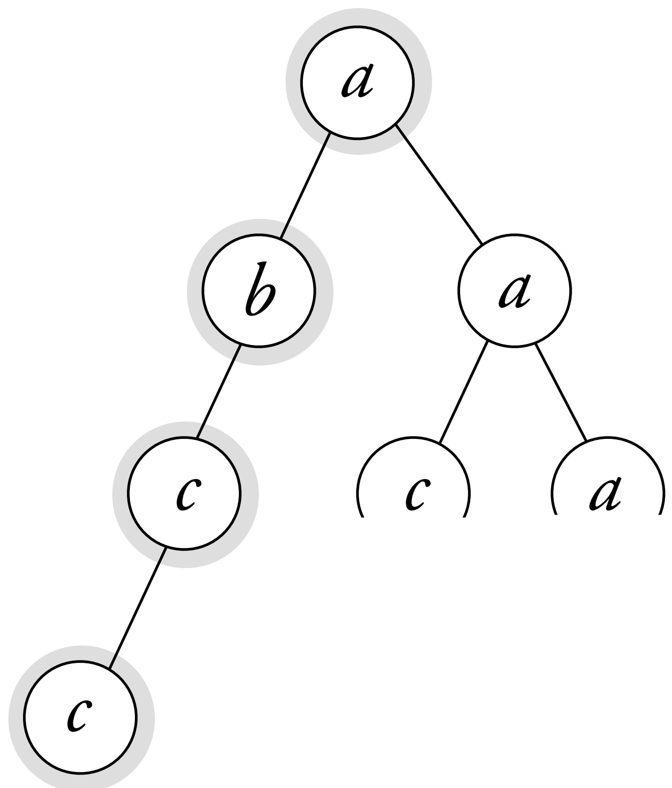


Branching time logic.

after every a^* prefix, b is possible

Instead of transition systems, we use labeled trees.
The results generalize to transition systems.

Instead of transition systems, we use labeled trees.
The results generalize to transition systems.



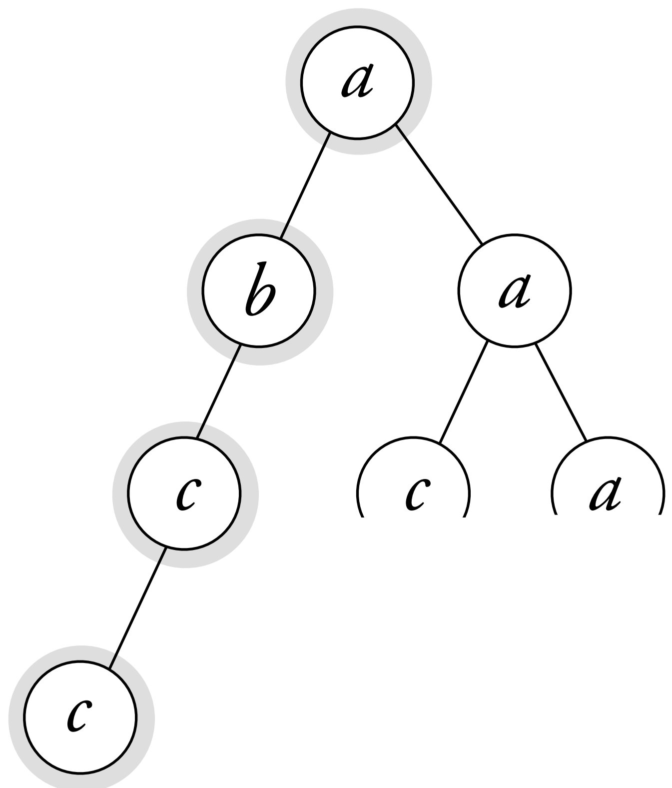
LTL

Languages of the form:
every path satisfies φ

φ a word property defined in LTL, e.g. $\mathsf{GF}\ a$

Here, instead of LTL, we use regular expressions for the word languages, e.g. $((b+c)^*a)^\omega$

Instead of transition systems, we use labeled trees.
The results generalize to transition systems.

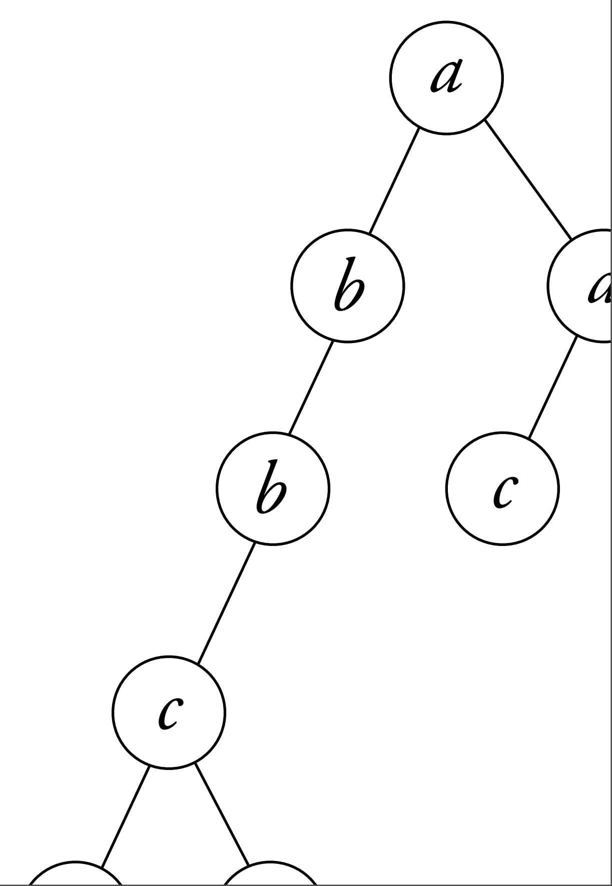


LTL

Languages of the form:
every path satisfies φ

φ a word property defined in LTL, e.g. $\mathsf{GF}\ a$

Here, instead of LTL, we use regular expressions for the word languages, e.g. $((b+c)^*a)^\omega$



CTL

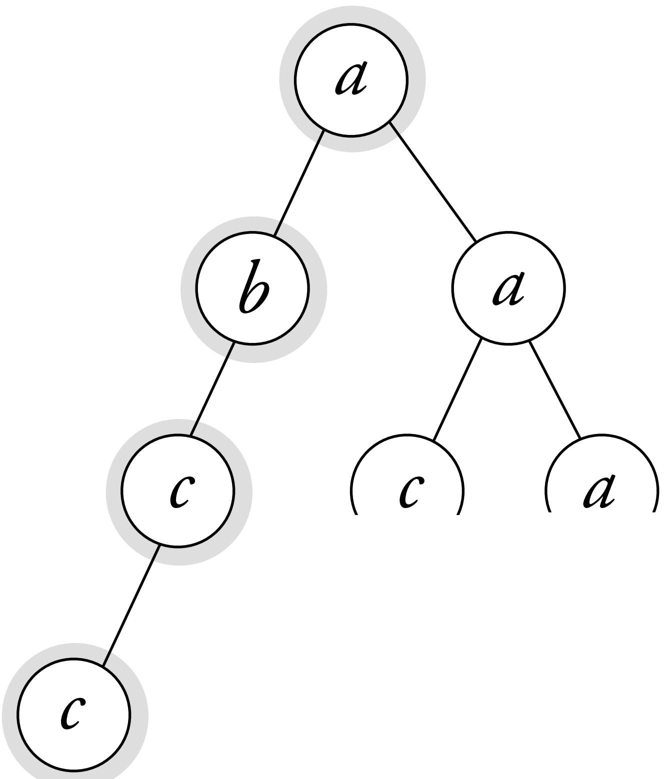
$\mathsf{A}\,\varphi\,\mathsf{U}\,\psi$: one every path φ holds until ψ holds.

$\mathsf{AG}\,\varphi$: on every path, φ holds globally.

$\mathsf{AX}\,\varphi$: on every path, φ holds in the next position.

label tests and boolean operations.

Instead of transition systems, we use labeled trees.
The results generalize to transition systems.



LTL

Languages of the form:
every path satisfies φ

φ a word property defined in LTL, e.g. **GF** a

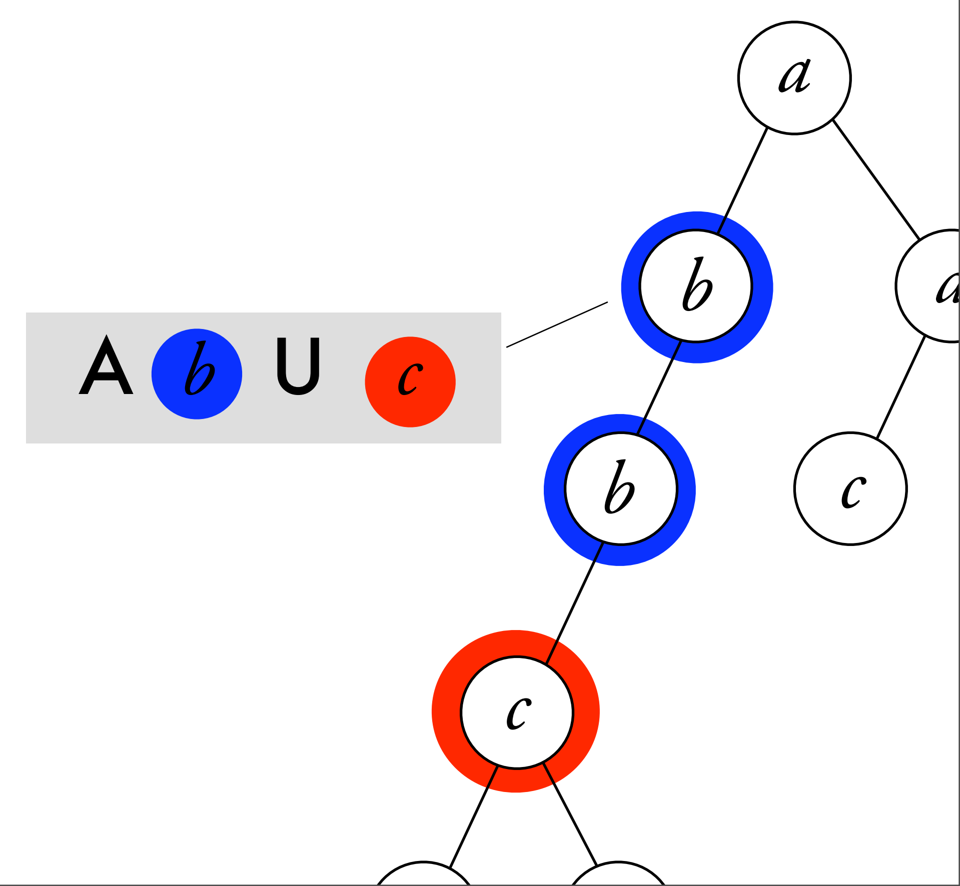
Here, instead of LTL, we use regular expressions for the word languages, e.g. $((b+c)^*a)^\omega$

CTL

A φ **U** ψ : one every path φ holds until ψ holds.

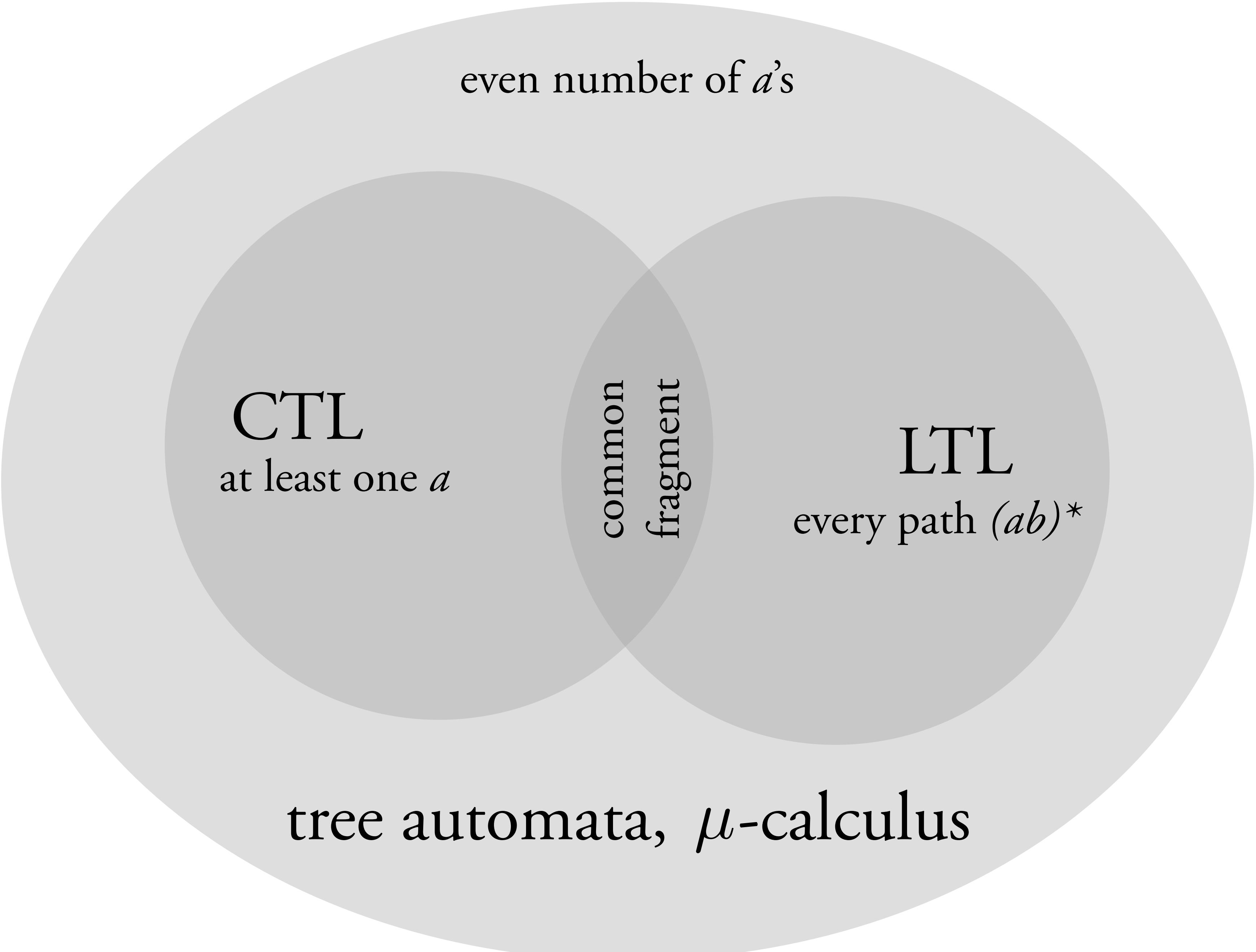
AG φ : on every path, φ holds globally.

AX φ : on every path, φ holds in the next position.



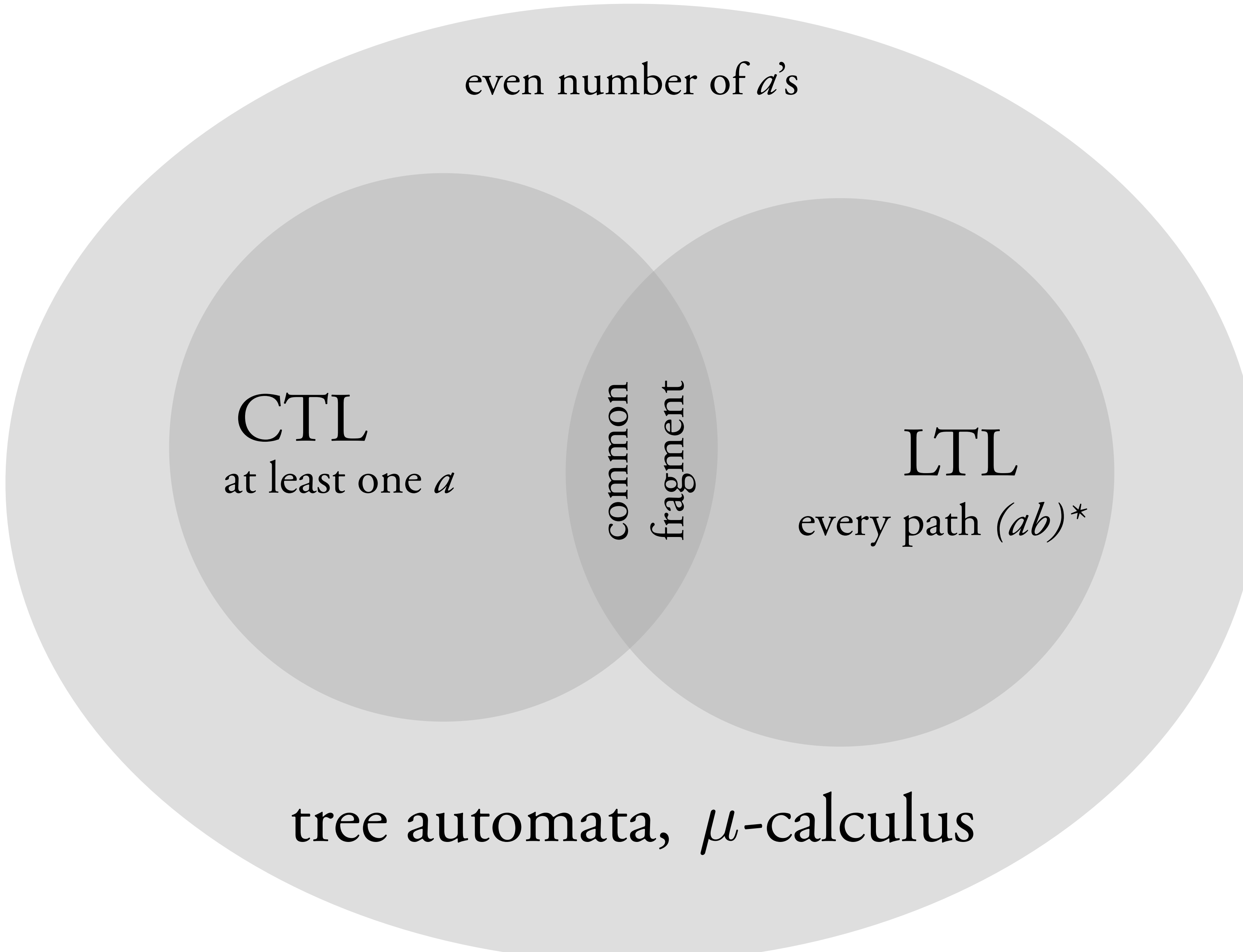
label tests and boolean operations.

What is the common fragment?



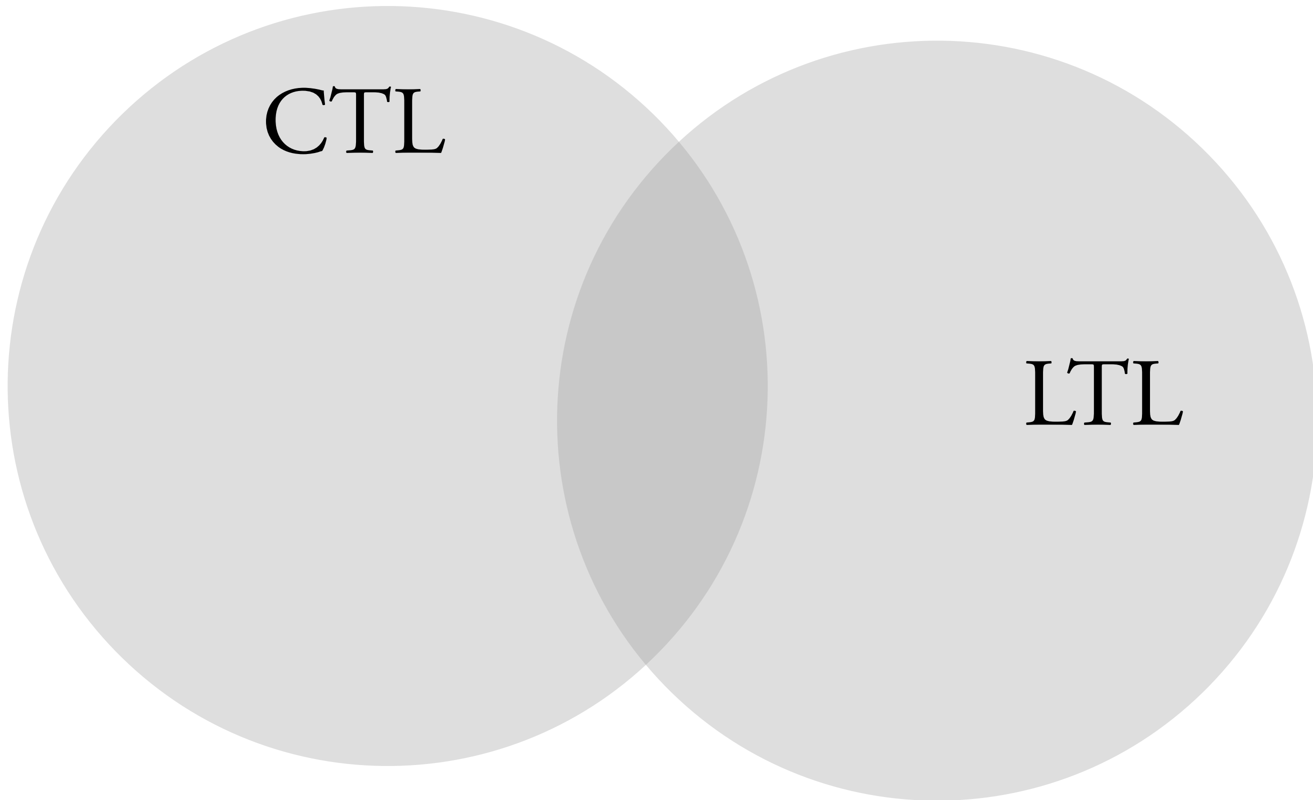
What is the common fragment?

Input:
CTL formula.
Question:
Is it in LTL?



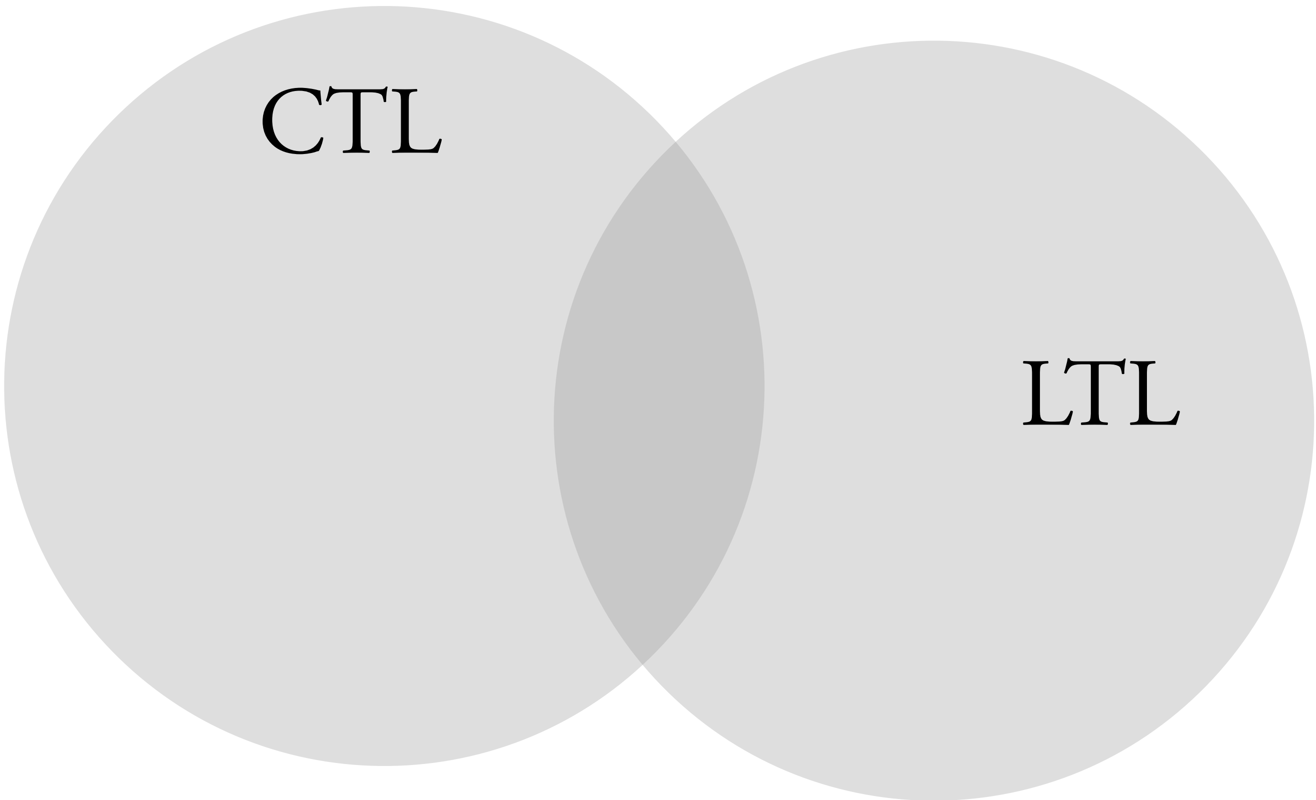
Input:
LTL formula.
Question:
Is it in CTL?

Input:
Tree automaton.
Question:
Is it in the common fragment?


CTL
LTL

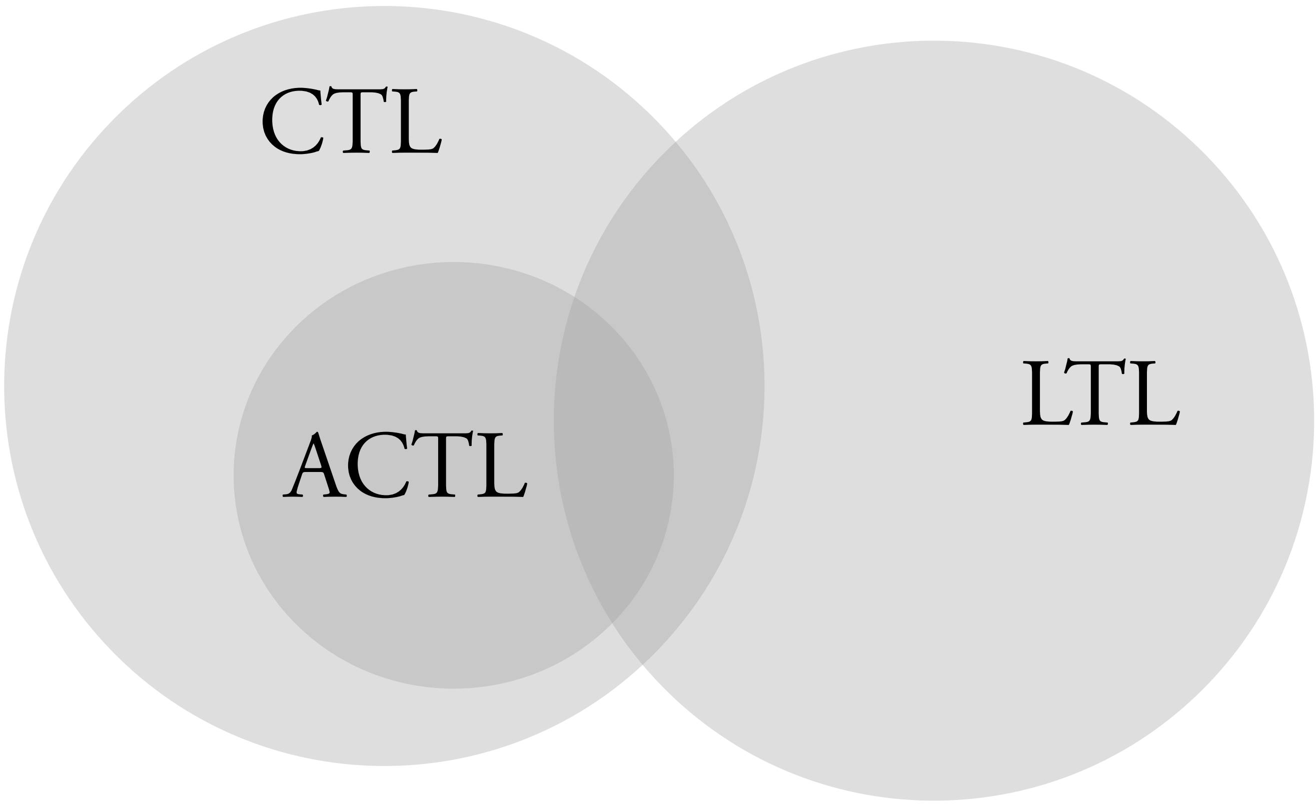
Thm.
It is decidable if a given regular language is in LTL.

Thm. [Maidl 00]
Complexity is PSPACE-complete if input is given in CTL.



Thm.
It is decidable if a given regular language is in LTL.

Thm. [Maidl 00]
Complexity is PSPACE-complete if input is given in CTL.



ACTL
CTL without negation,
only “on all paths...”

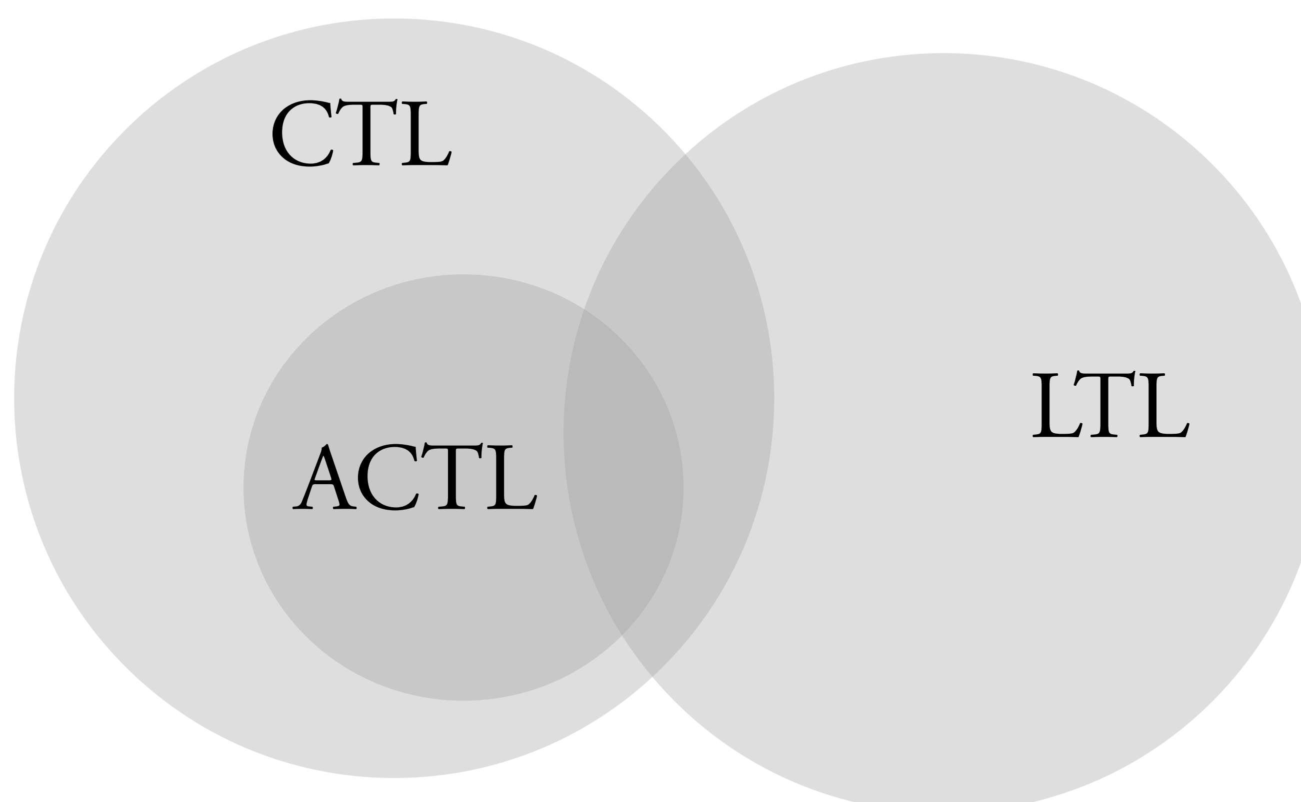
Thm.
It is decidable if a given regular language is in LTL.

Thm. [Maidl 00]
Complexity is PSPACE-complete if input is given in CTL.

Main contribution.

$\mathrm{ACTL} \cap \mathrm{LTL} \quad \subsetneq \quad \mathrm{CTL} \cap \mathrm{LTL}$

decidable membership



ACTL
CTL without negation,
only "on all paths..."

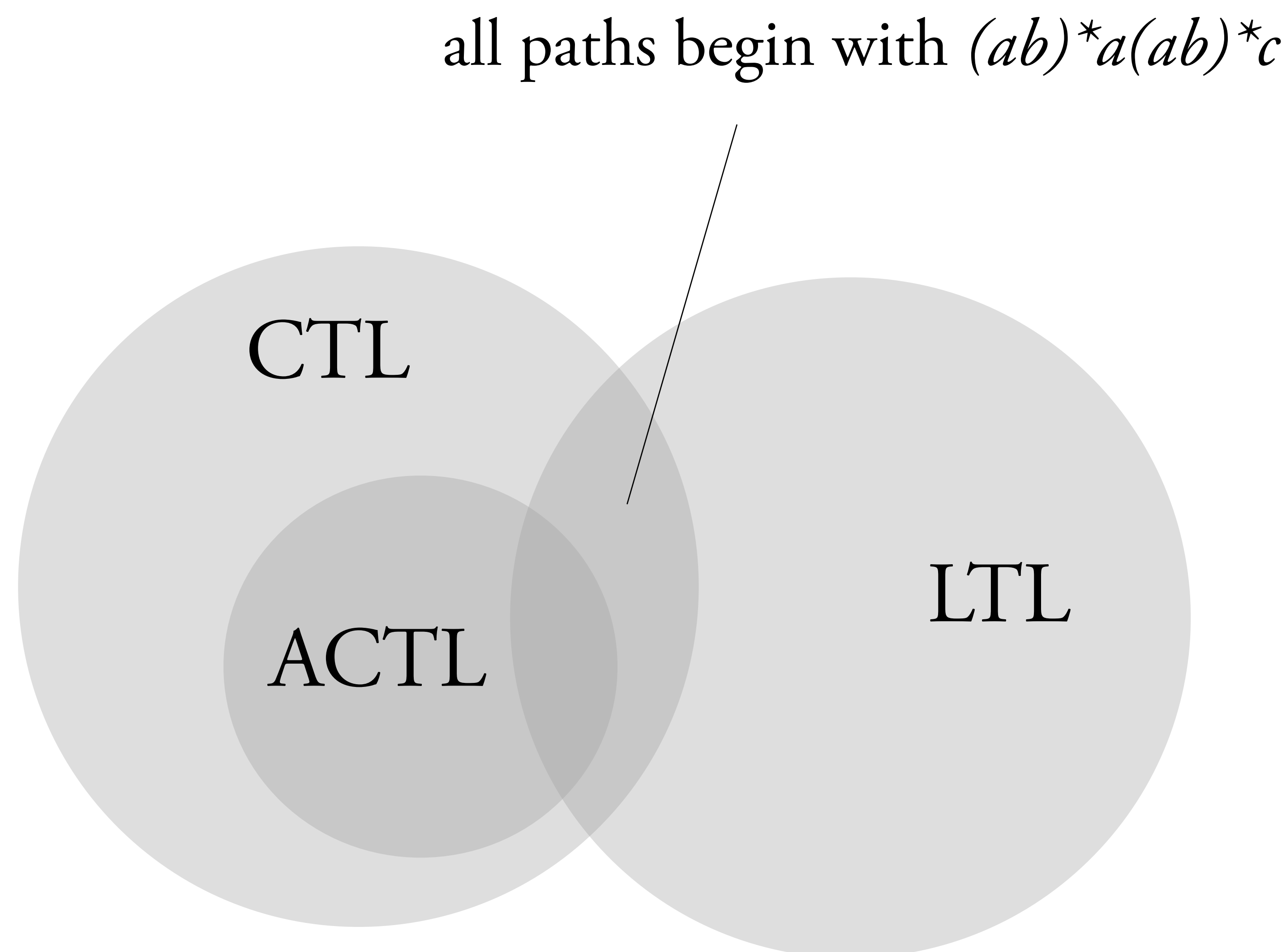
Thm.
It is decidable if a given regular language is in LTL.

Thm. [Maidl 00]
Complexity is PSPACE-complete if input is given in CTL.

Main contribution.

$$\mathrm{ACTL} \cap \mathrm{LTL} \subsetneq \mathrm{CTL} \cap \mathrm{LTL}$$

decidable membership



ACTL

CTL without negation,
only “on all paths...”

Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

$$\forall x_1 \cdots \forall x_n \quad \exists y_1 \cdots \exists y_m \quad \overbrace{\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)}^{\text{quantifier-free, using labels and } <}$$

On every path, every *a* is followed by some *b*.

$$\forall x \exists y \quad a(x) \Rightarrow \big(x < y \ \wedge \ b(y)\big)$$

$$\mathsf{AG}\big(\mathsf{a} \Rightarrow \mathsf{A}\ \mathsf{a}\ \mathsf{U}\ \mathsf{b}\big)$$

Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

$$\forall x_1 \cdots \forall x_n \quad \exists y_1 \cdots \exists y_m \quad \overbrace{\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)}^{\text{quantifier-free, using labels and } <}$$

Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

Contribution 1. Not the same thing as CTL $\cap$ LTL.

Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

Contribution 2. Effective criterion.

$\in$ CTL

ACTL $\not\ni$ all paths begin with *(ab)*a(ab)*c*

$\in$ LTL

words that begin with *(ab)*a(ab)*c* $\notin \Pi_2(<)$

Obvious

$\in$ CTL

ACTL $\not\ni$ all paths begin with $(ab)^*a(ab)^*c$

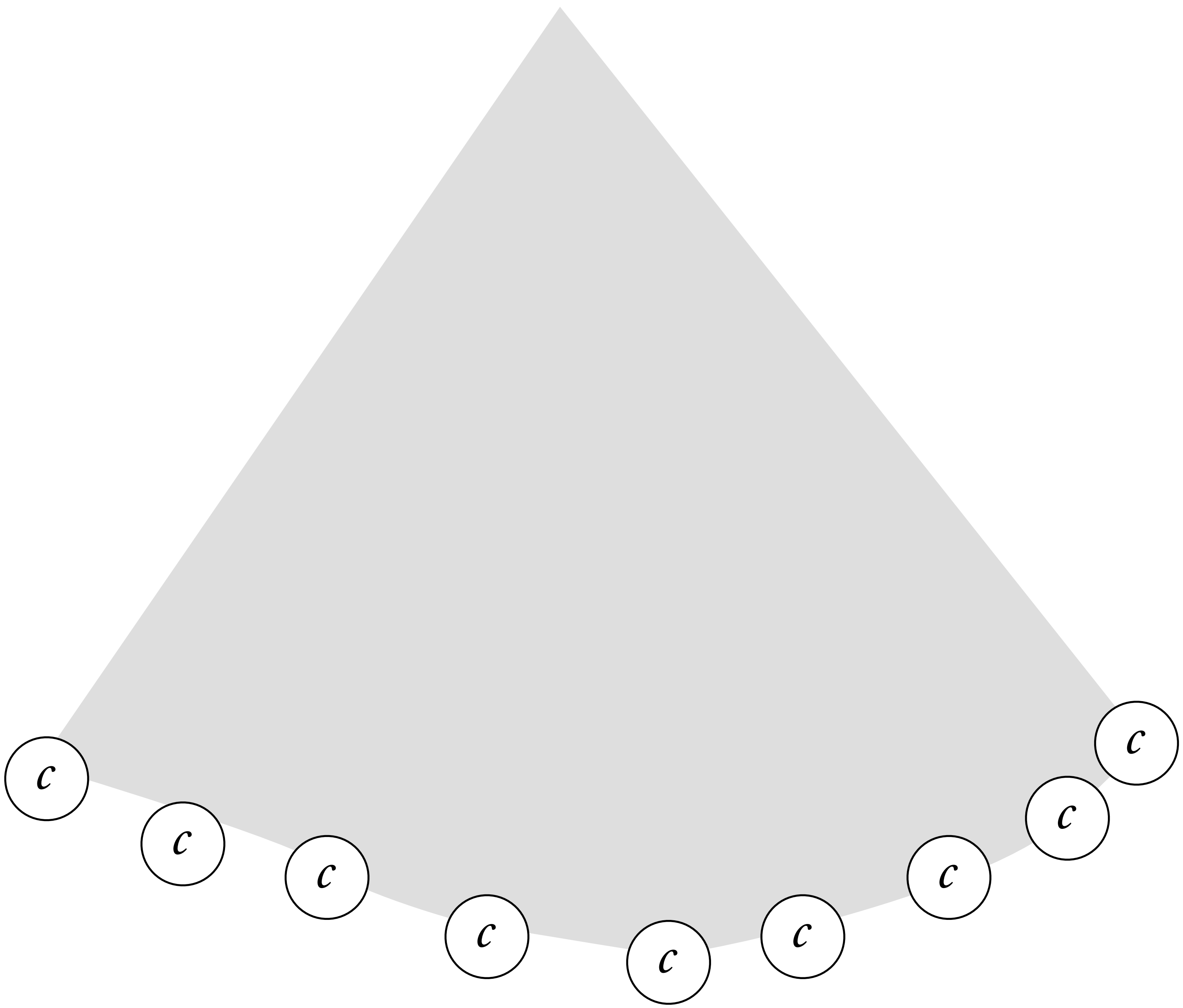
$\in$ LTL

words that begin with $(ab)^*a(ab)^*c$ $\notin \Pi_2(<)$

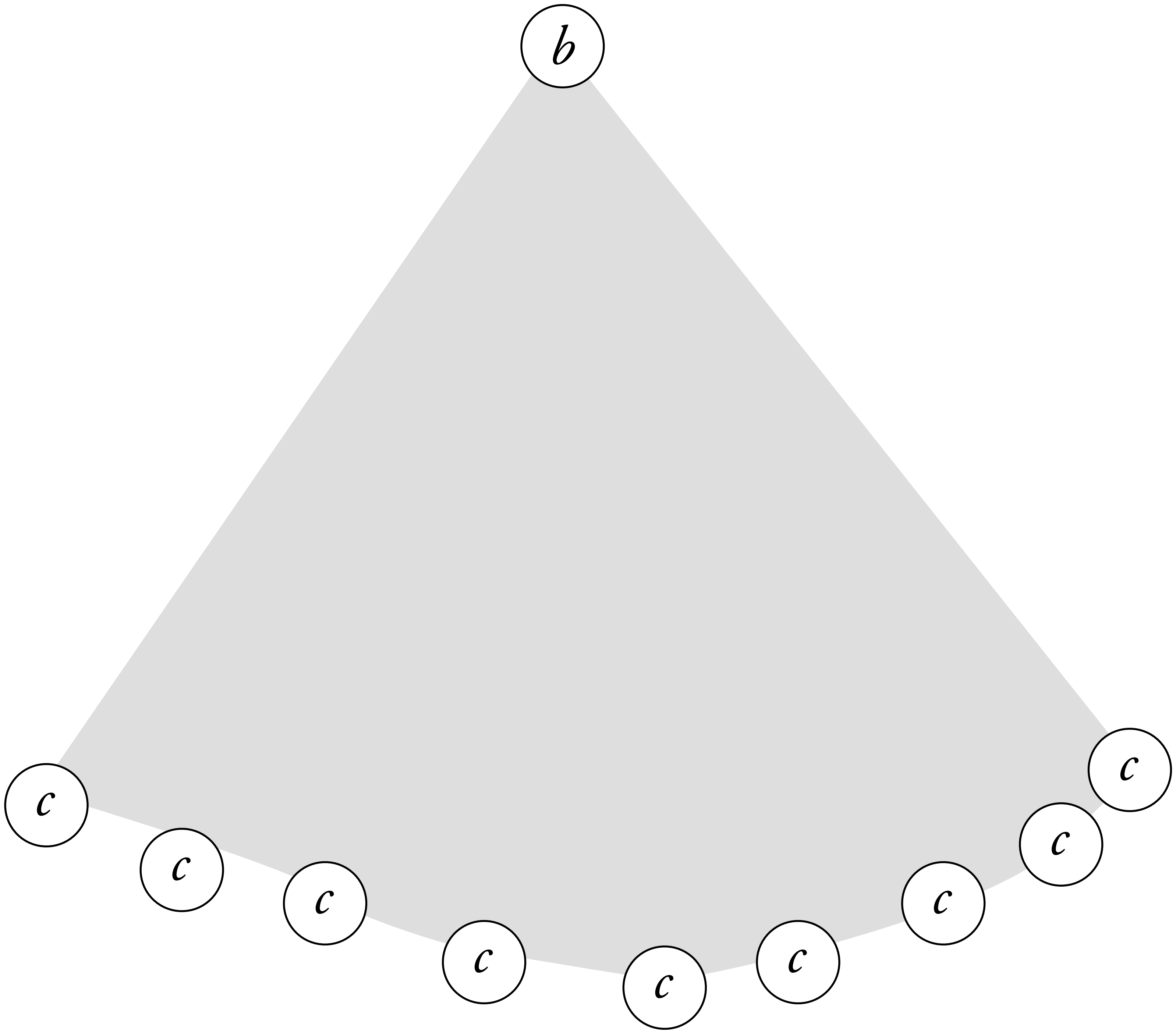
Obvious

1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$

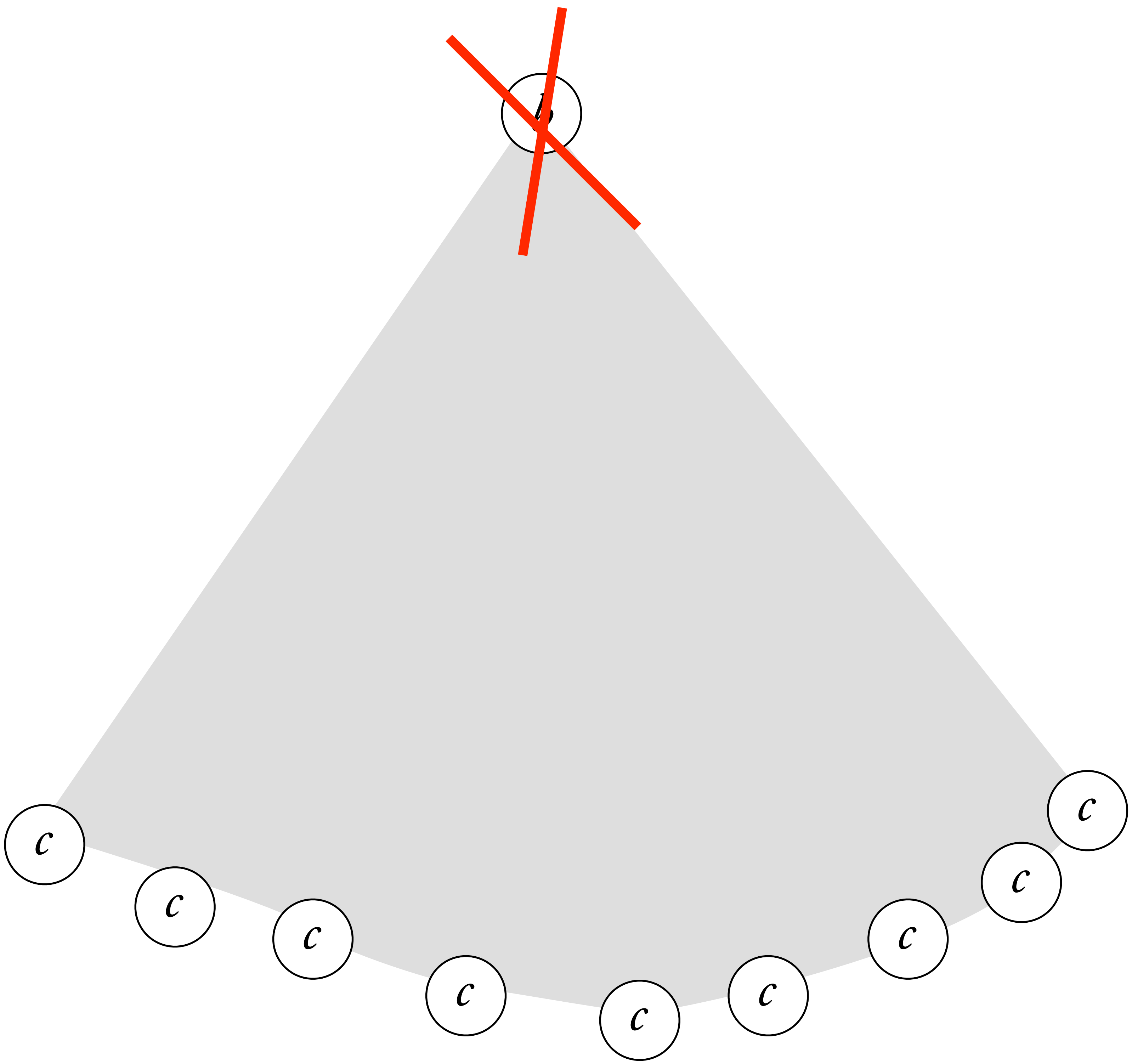
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



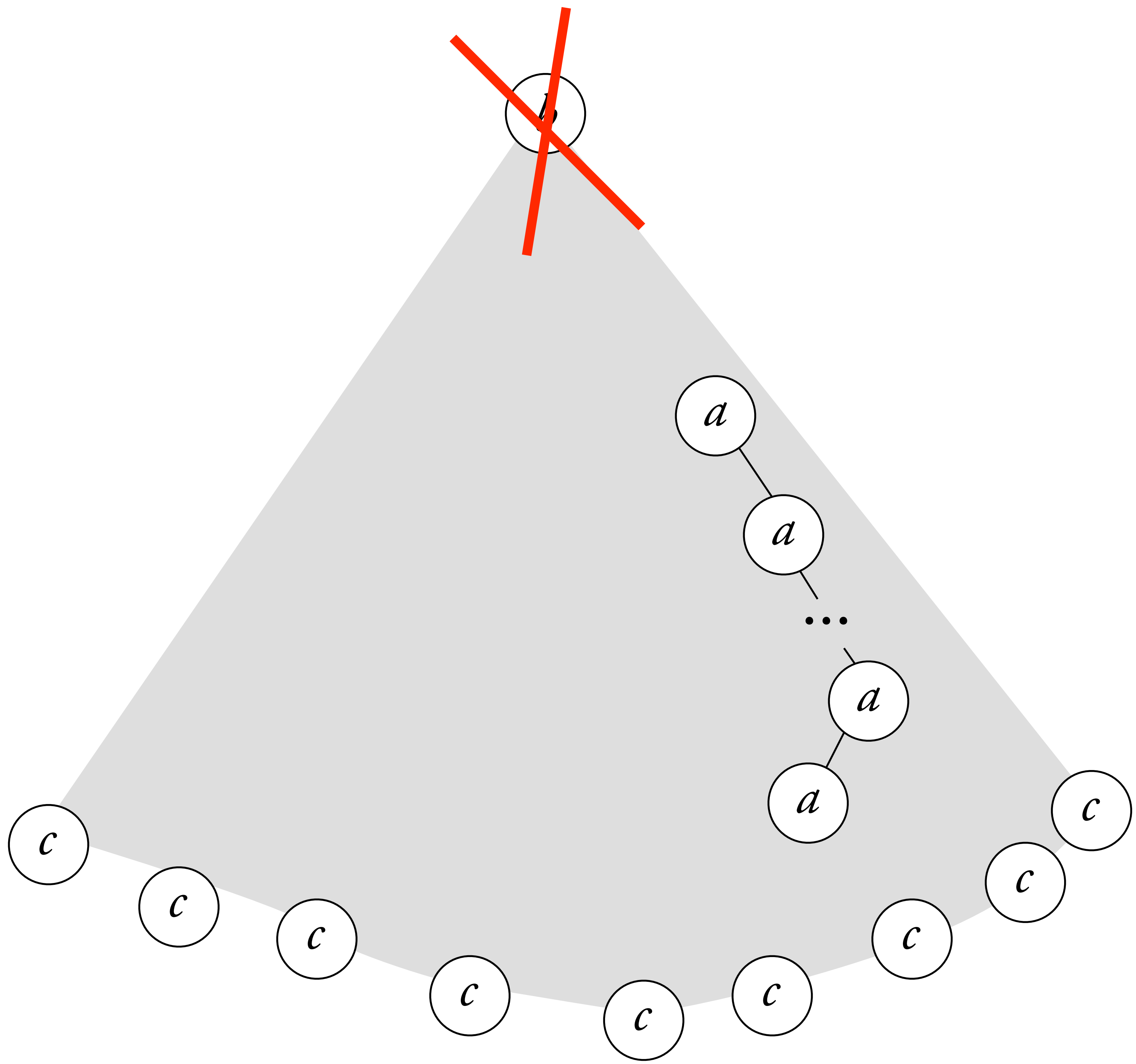
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



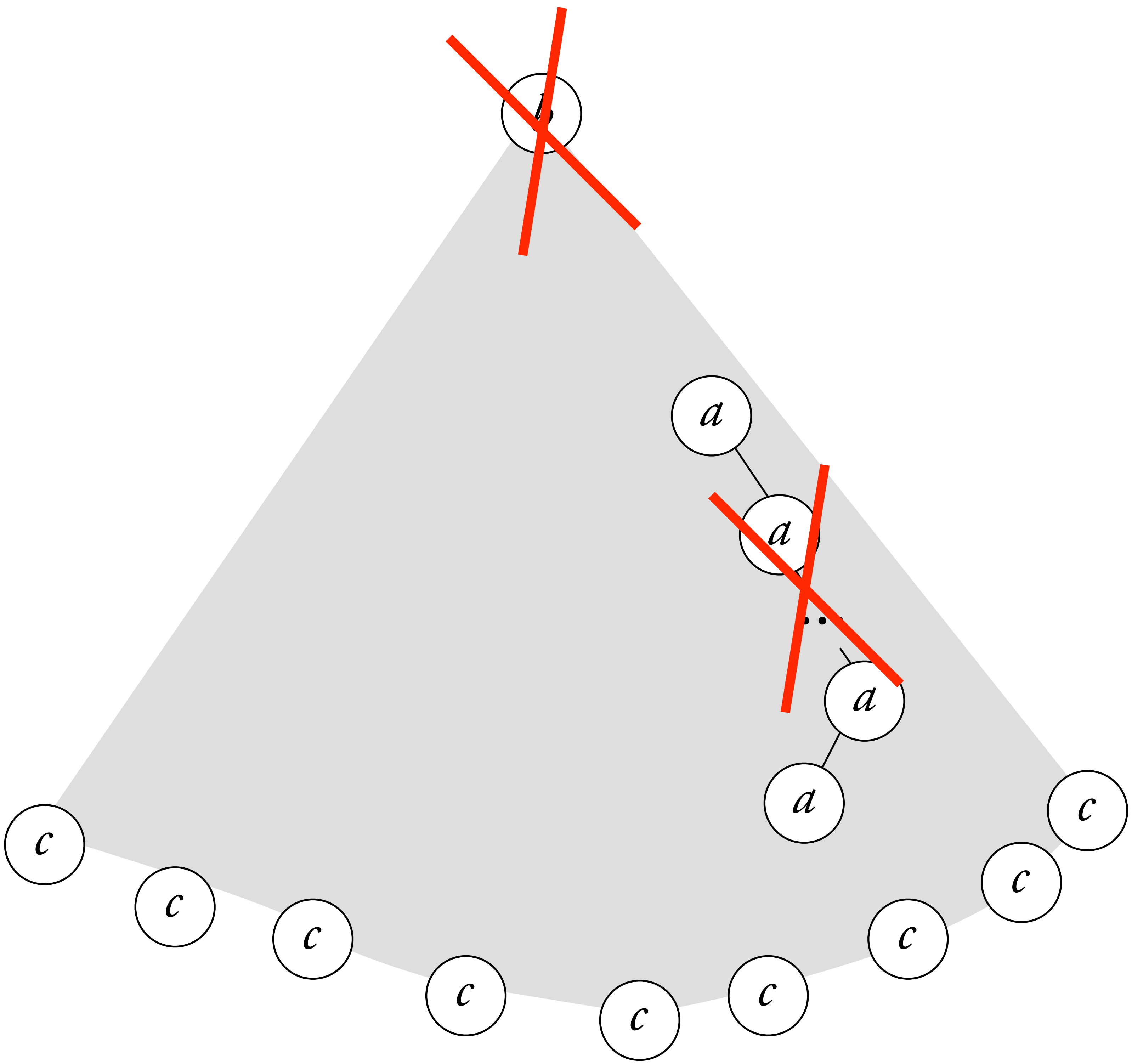
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



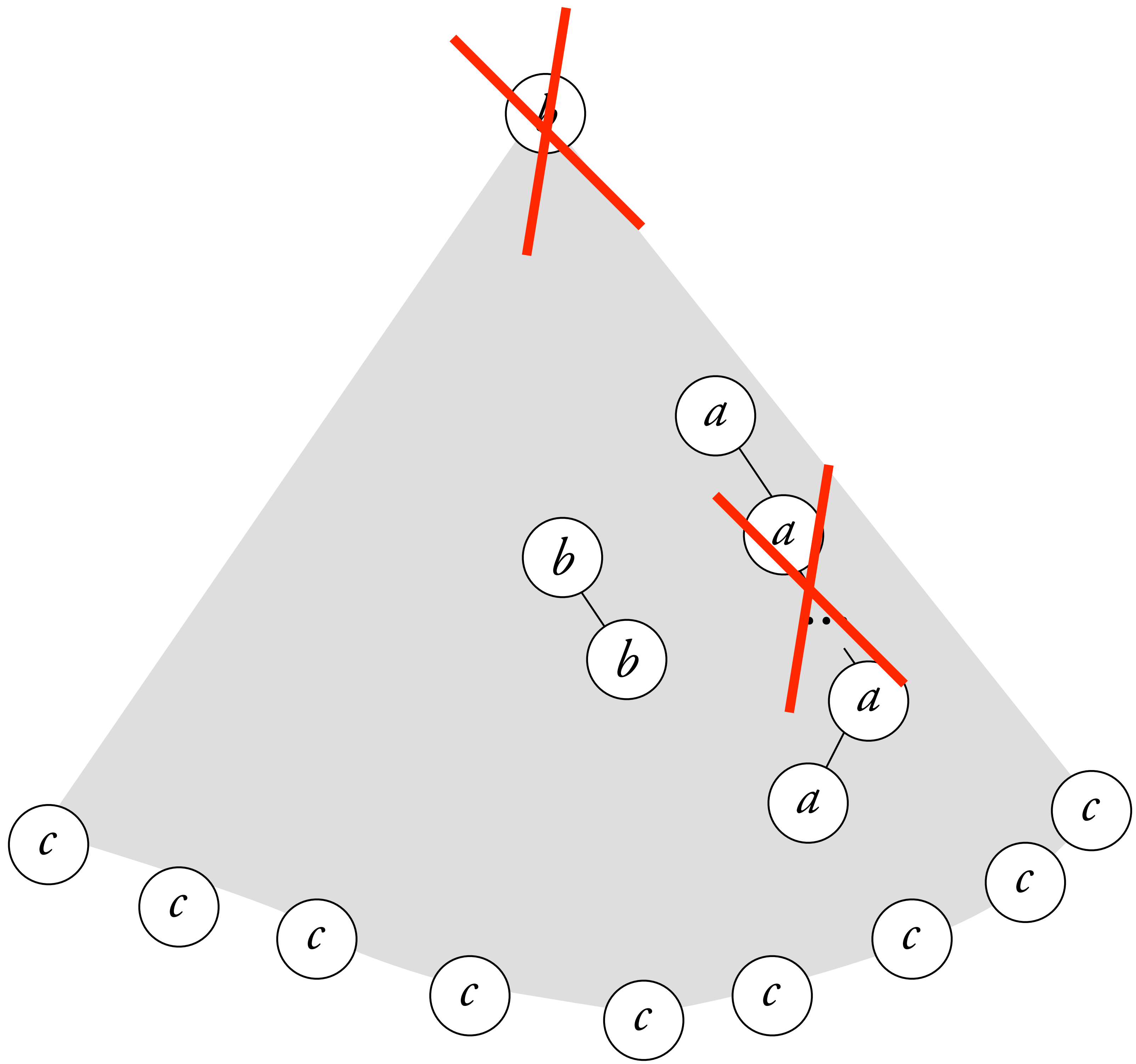
1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*



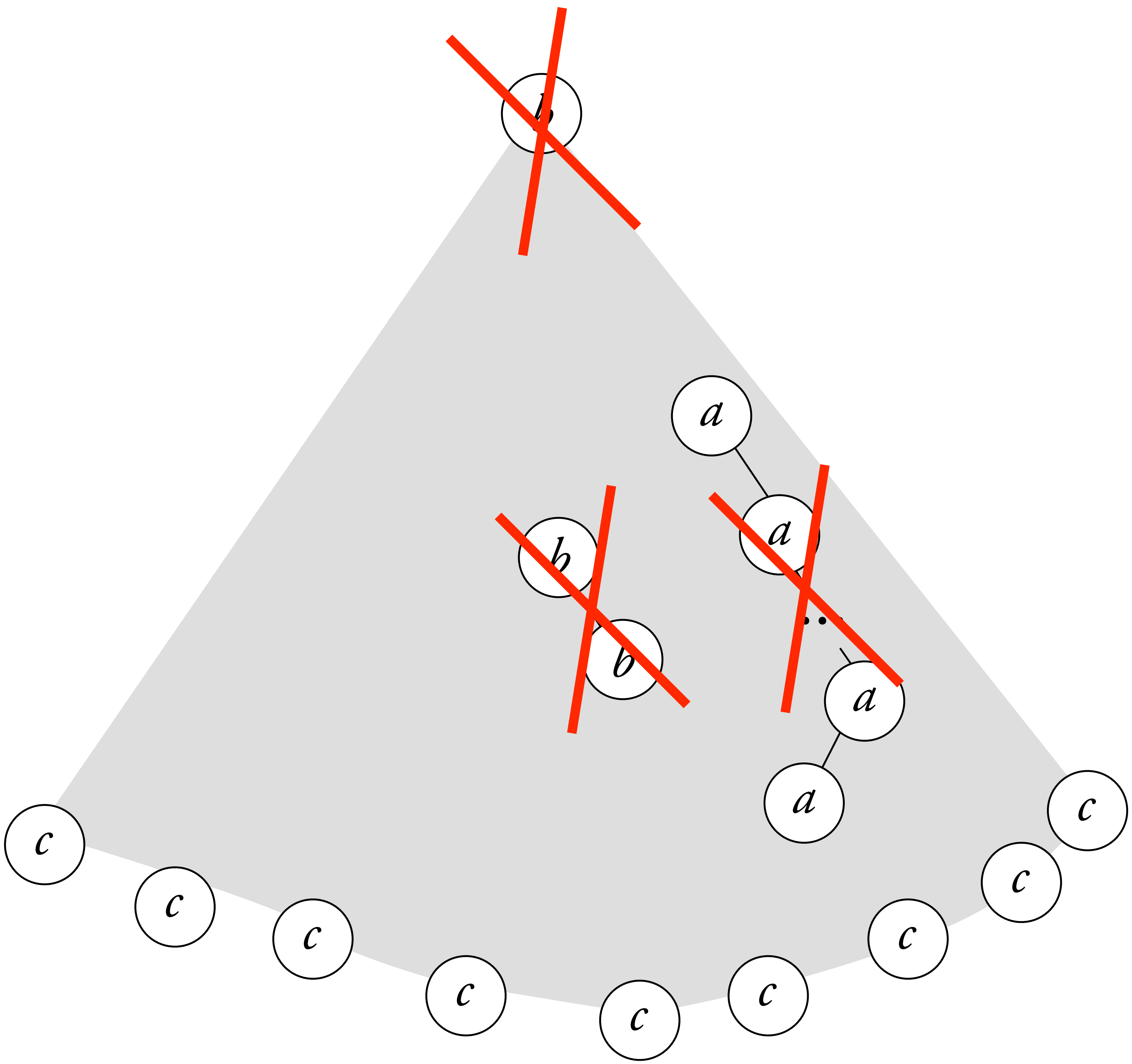
1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*



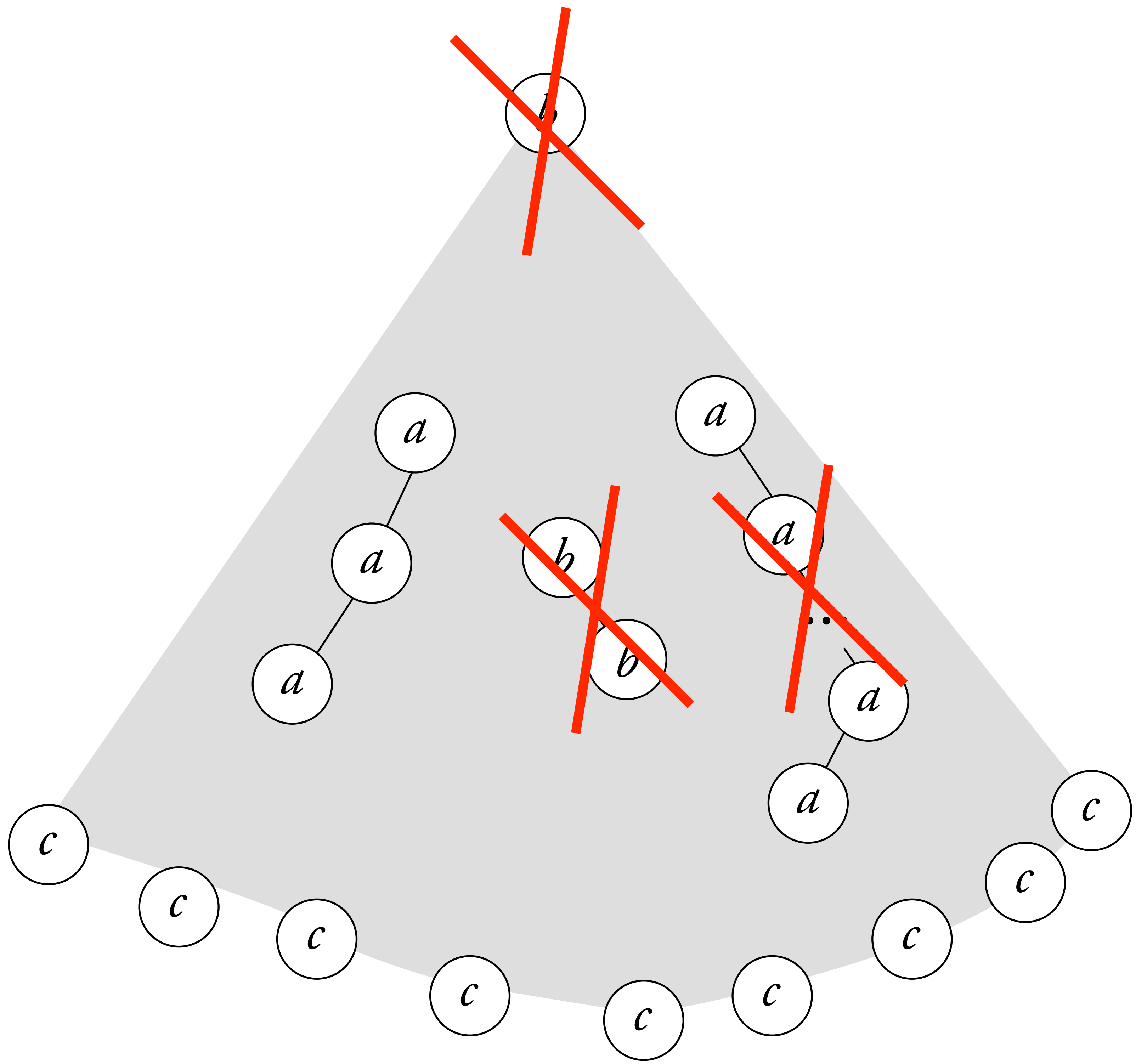
1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*



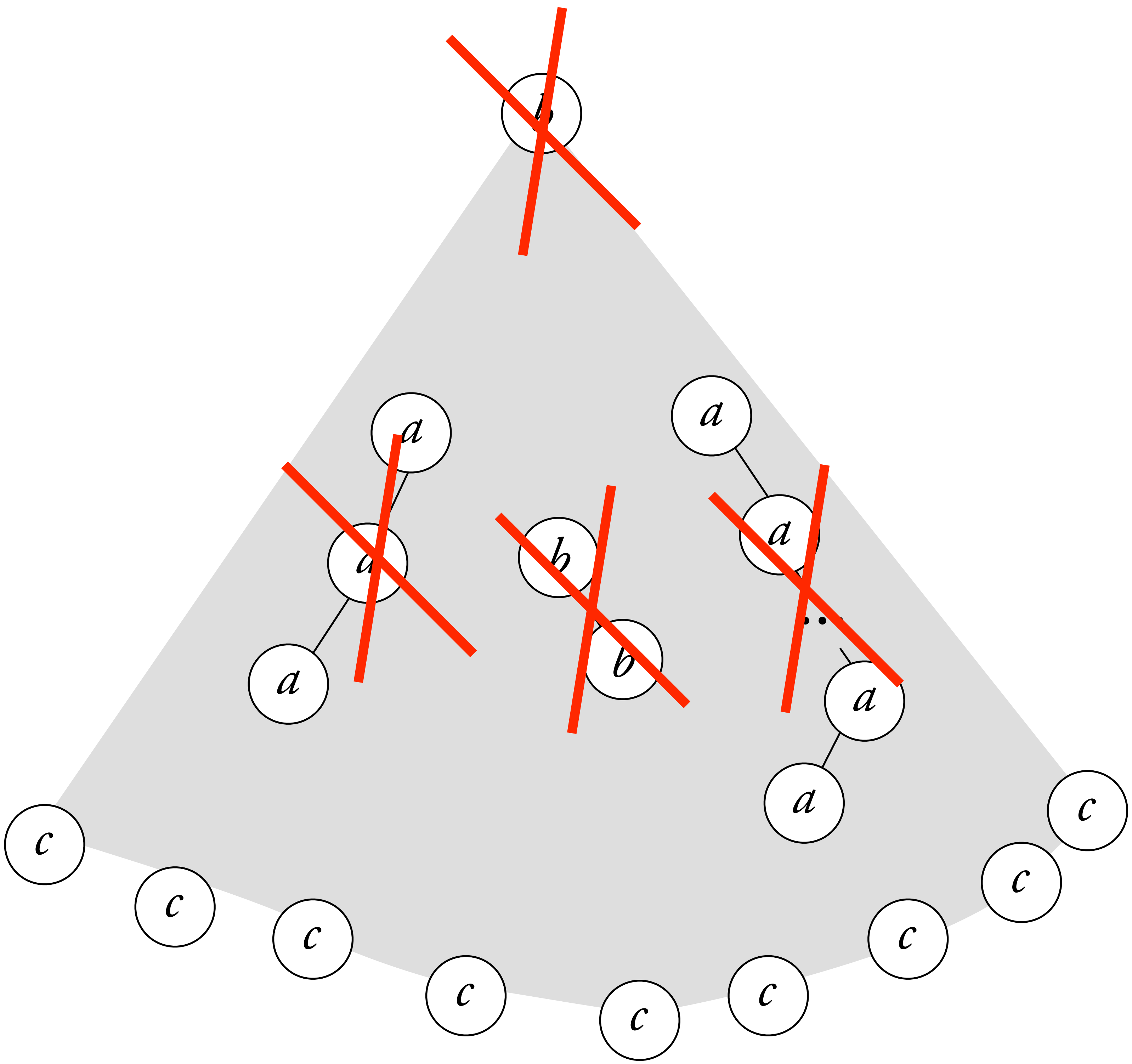
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



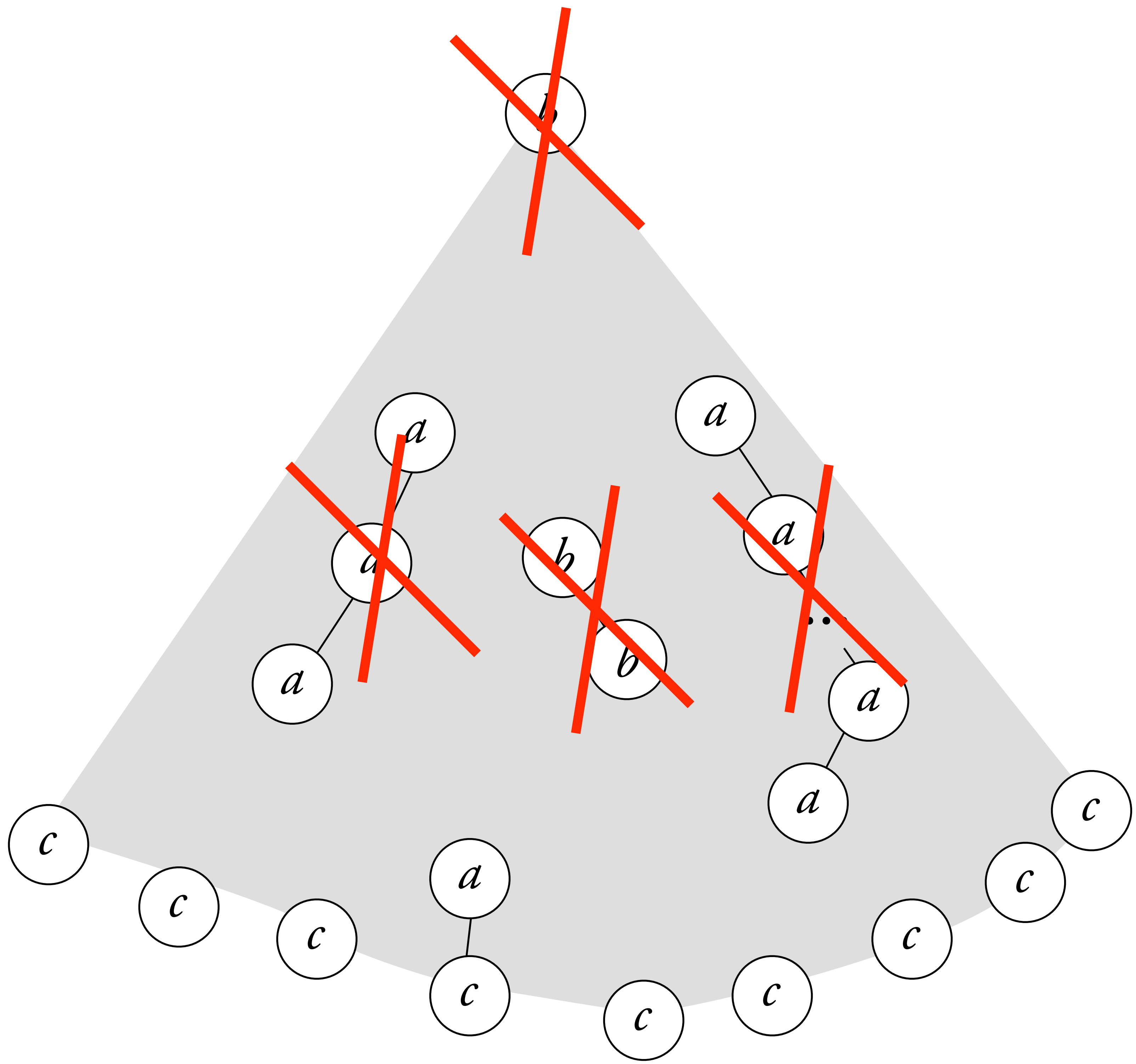
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



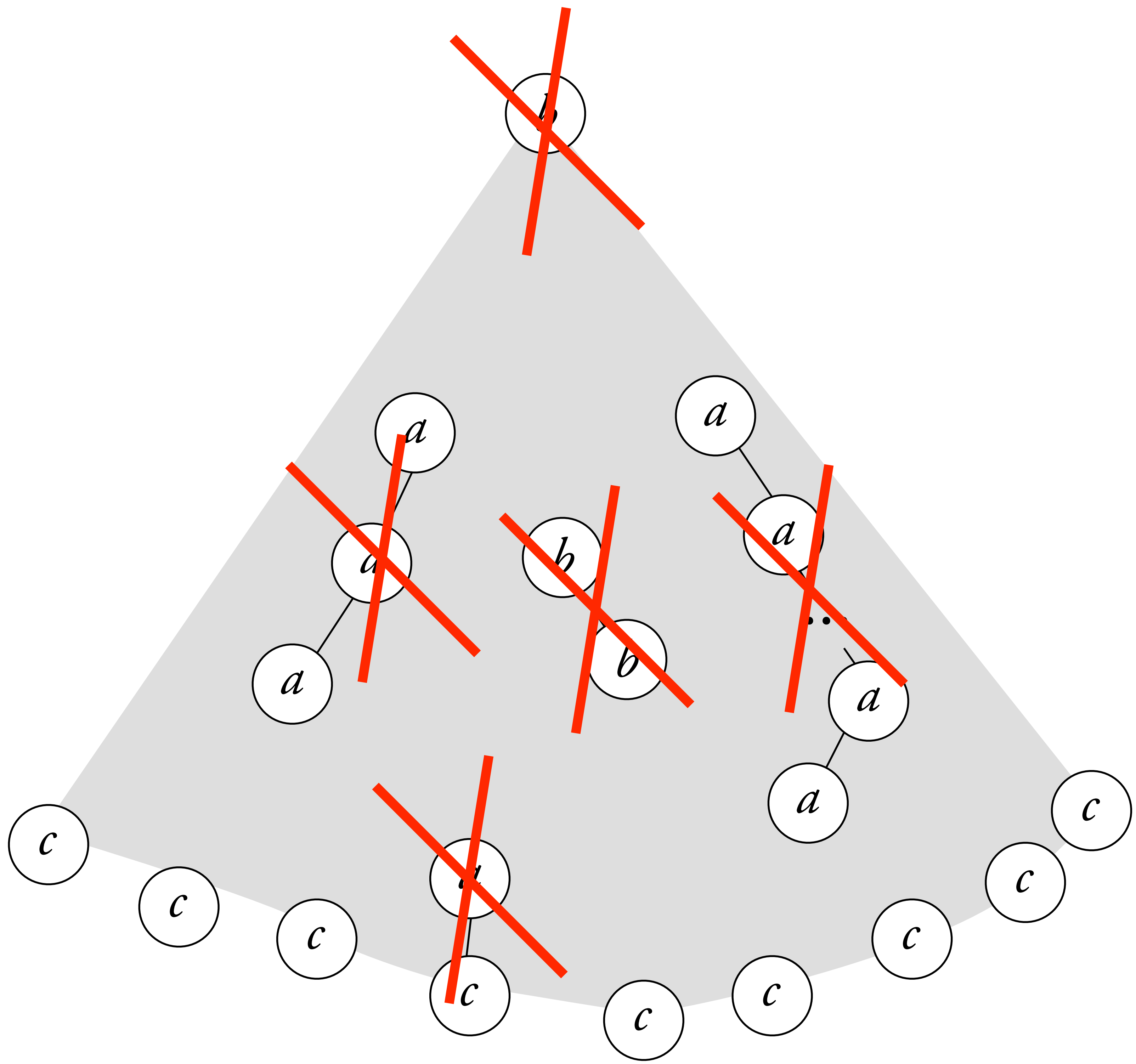
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



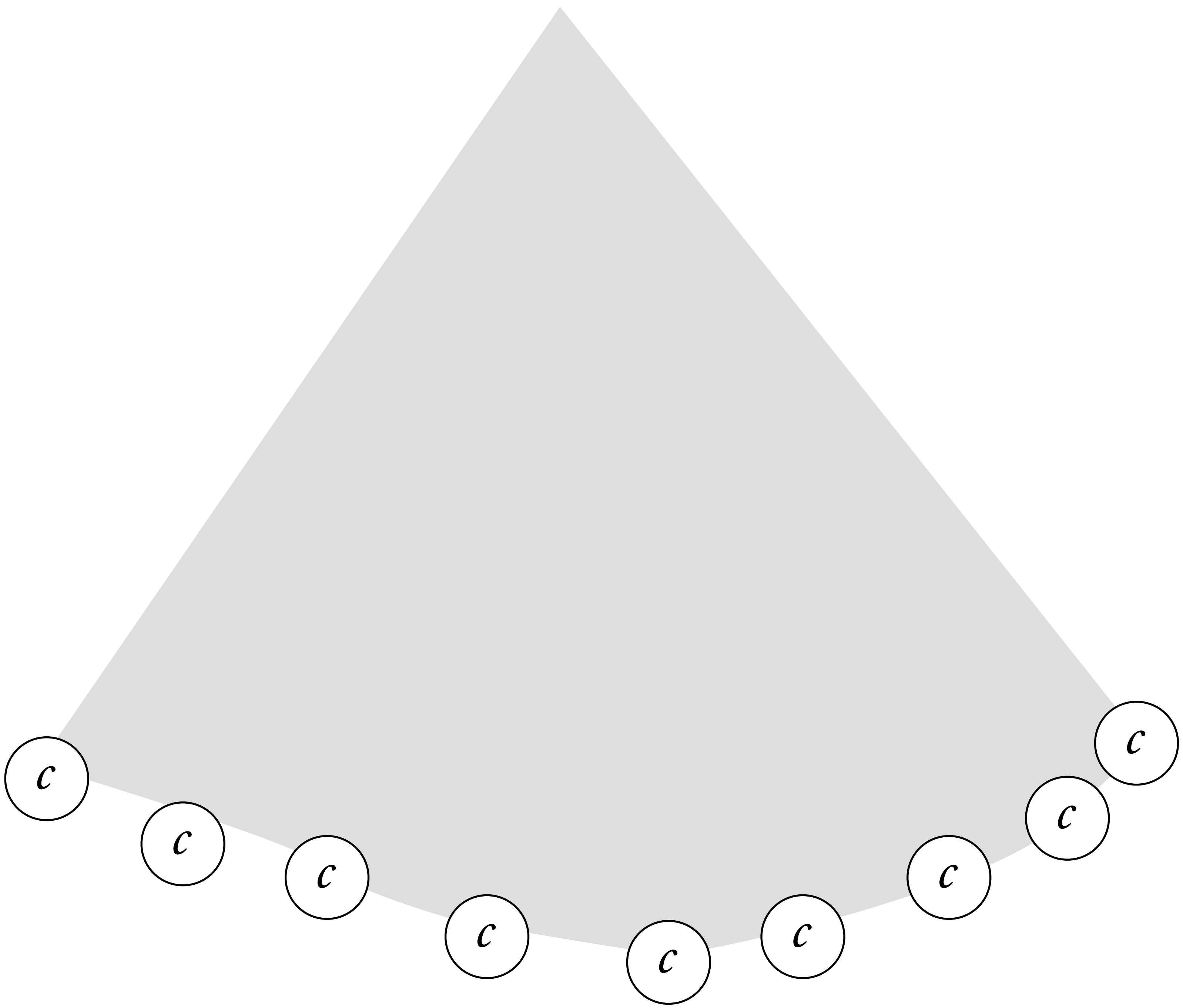
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



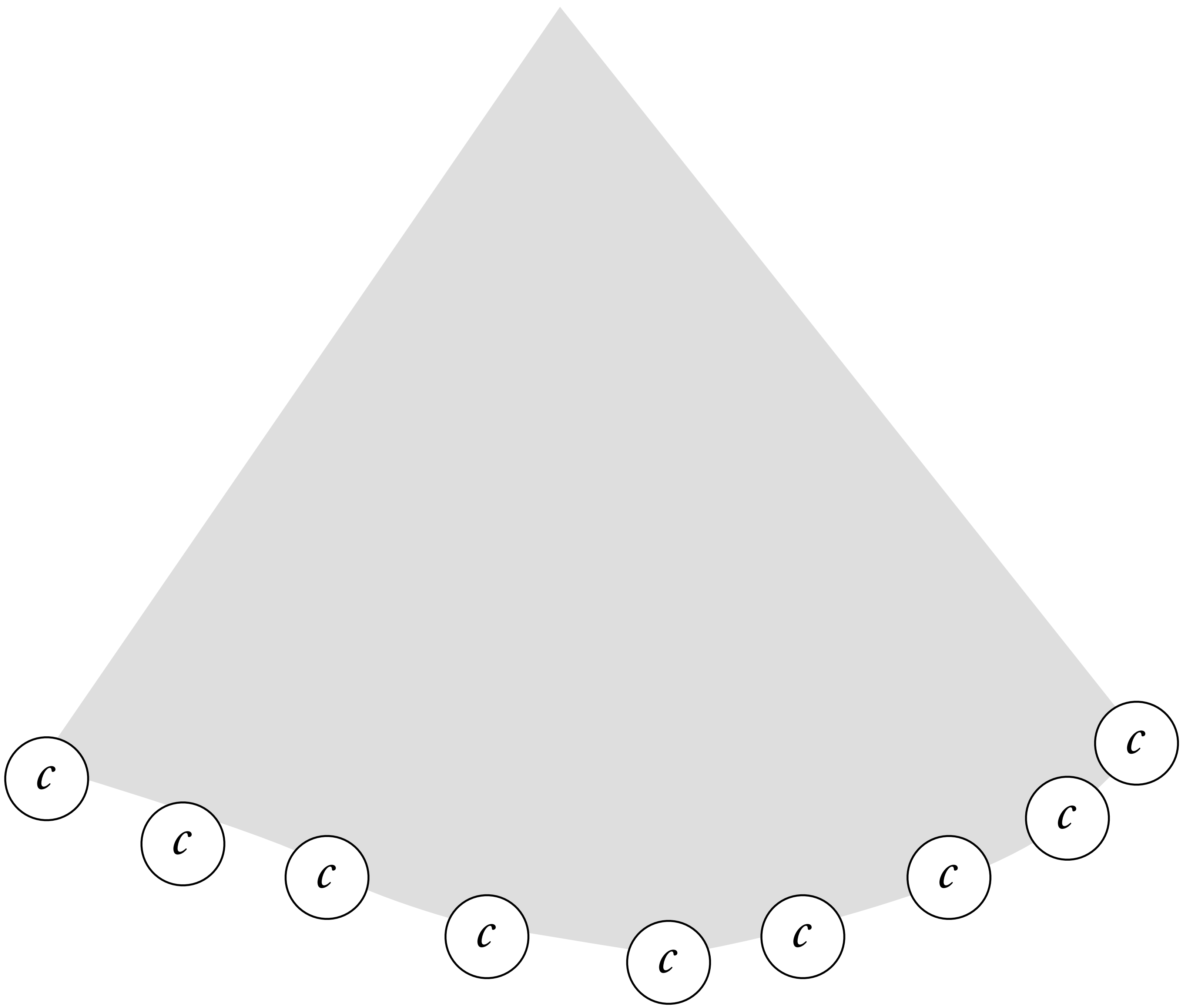
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



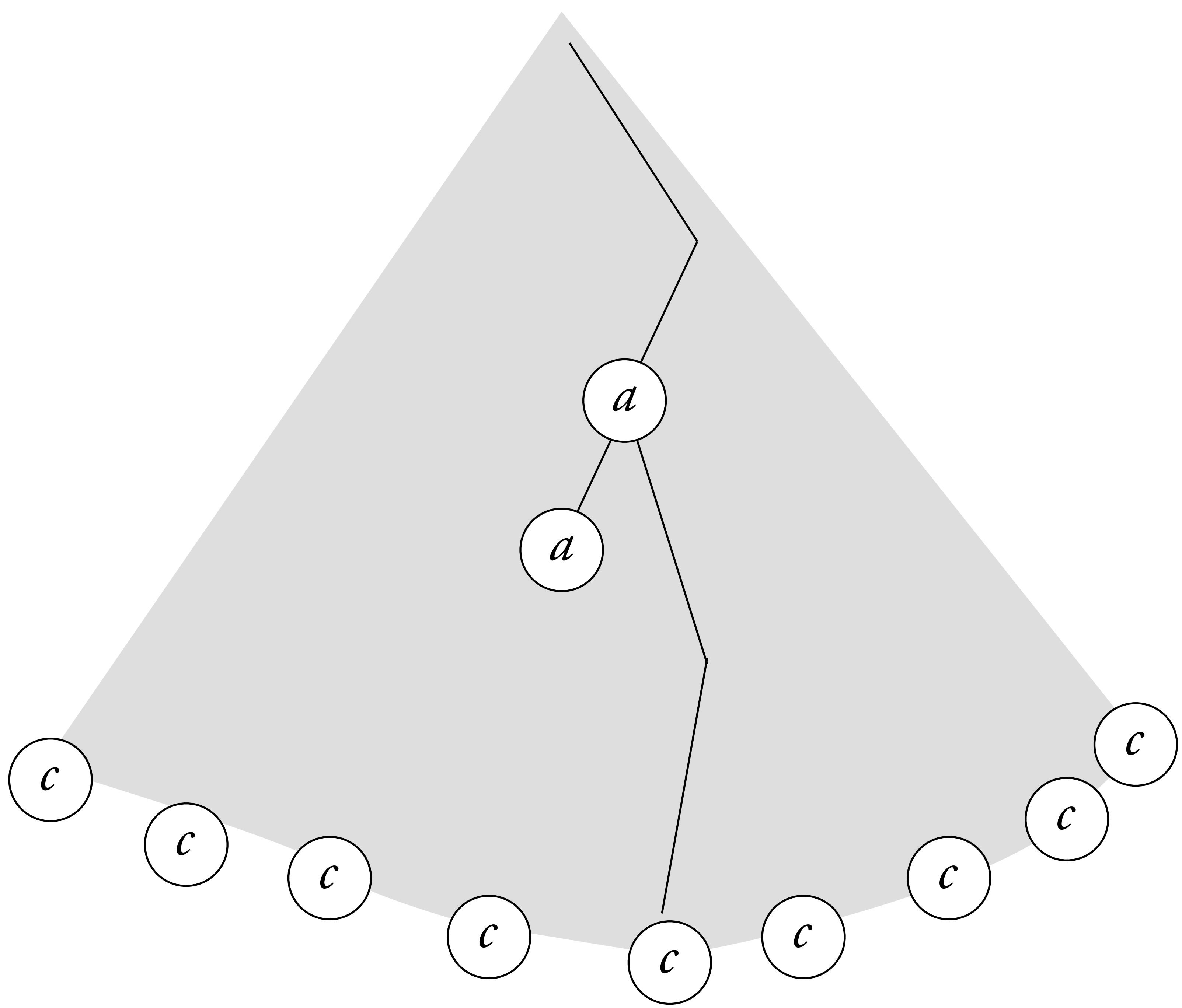
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$



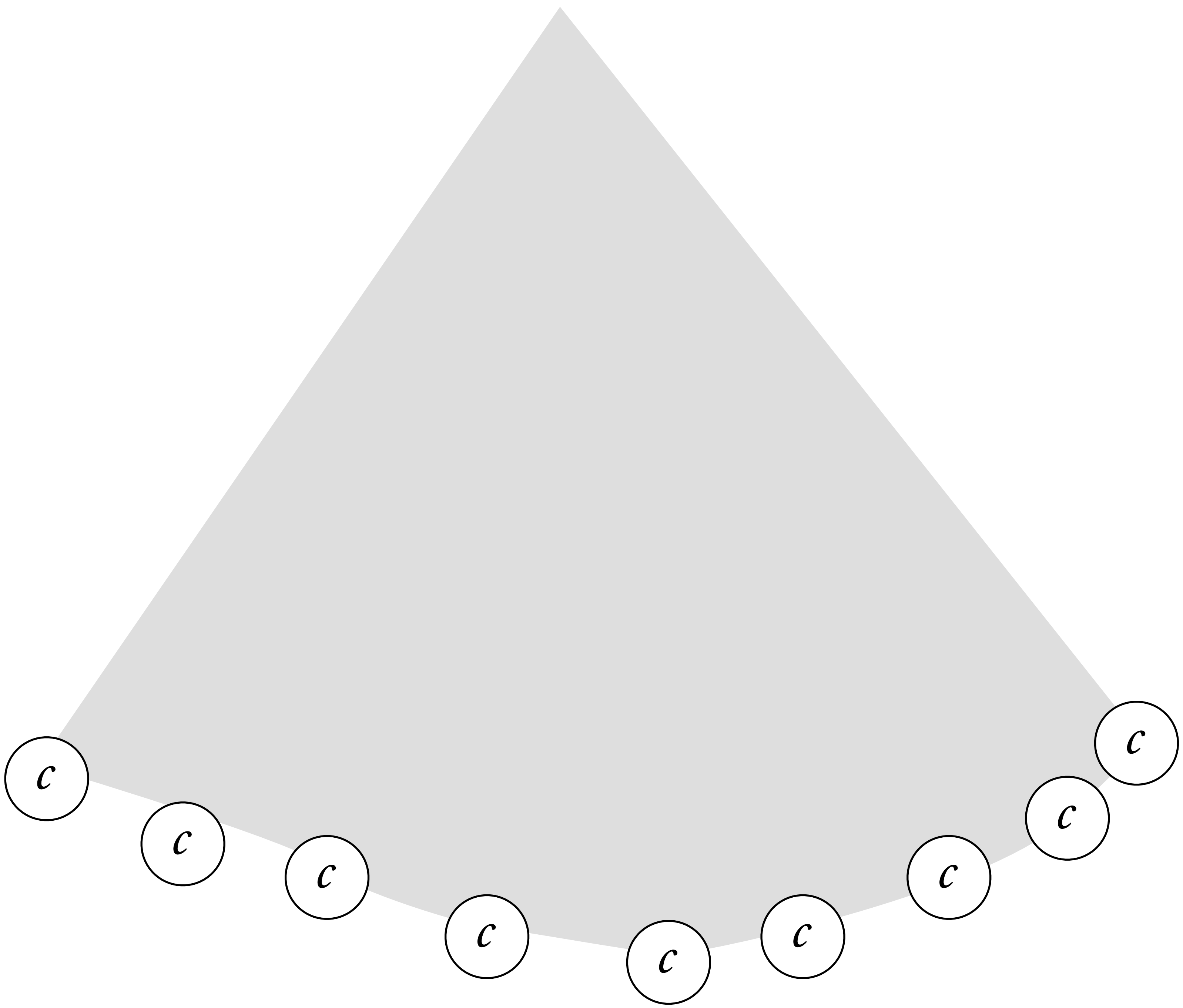
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$
2. on every path, some *a* before *c* has an *a* child.



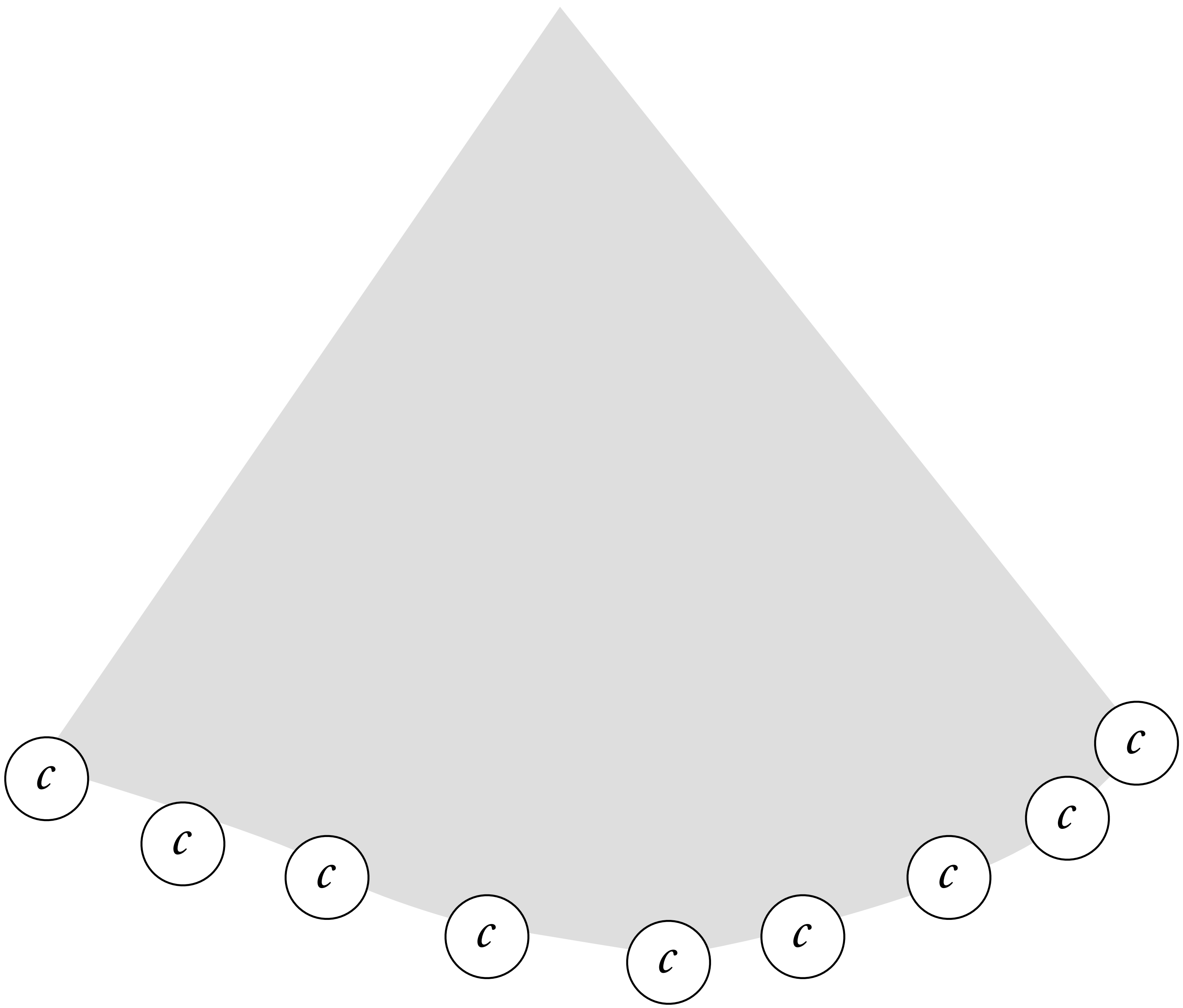
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$
2. on every path, some a before c has an a child.



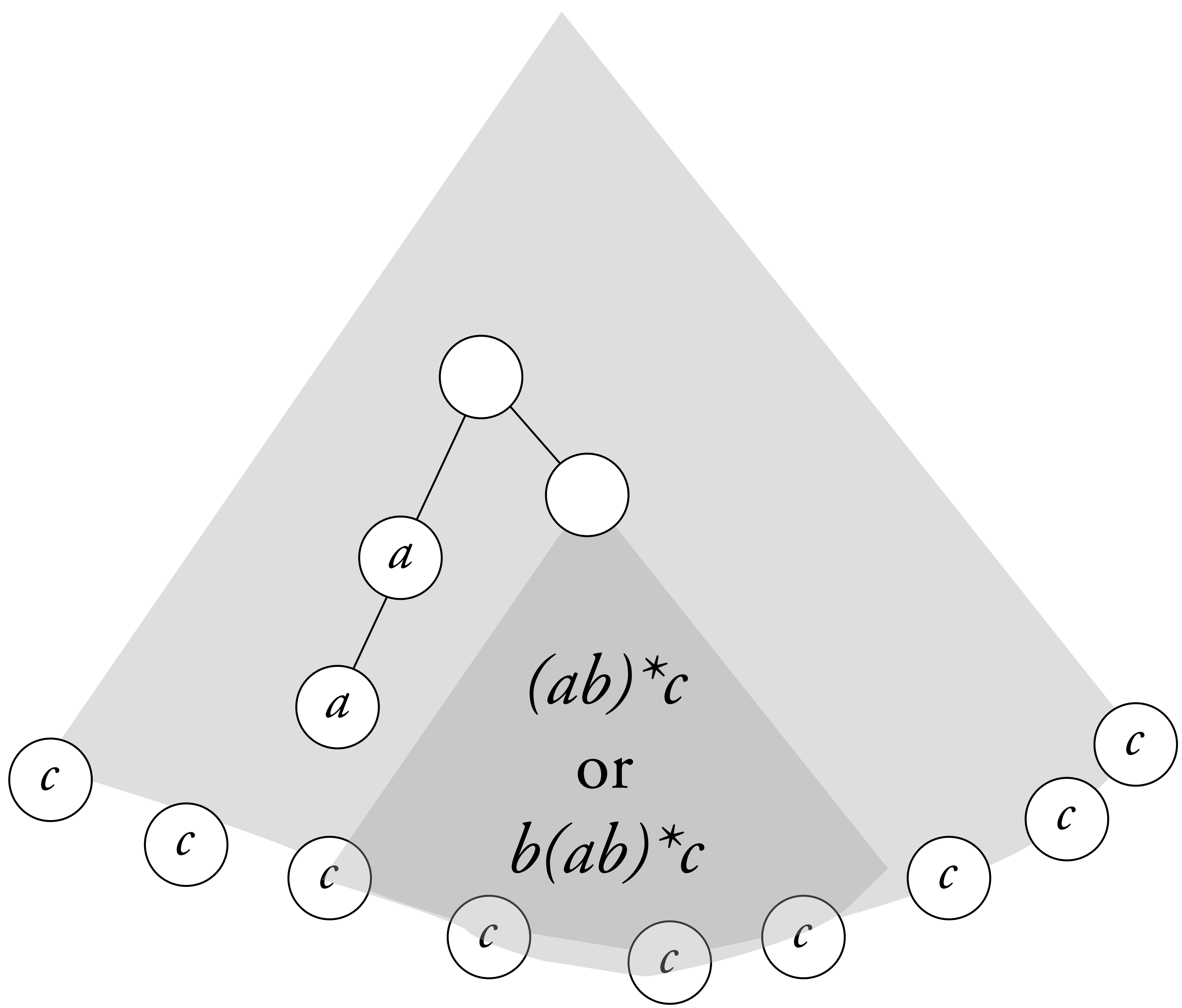
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$
2. on every path, some *a* before *c* has an *a* child.



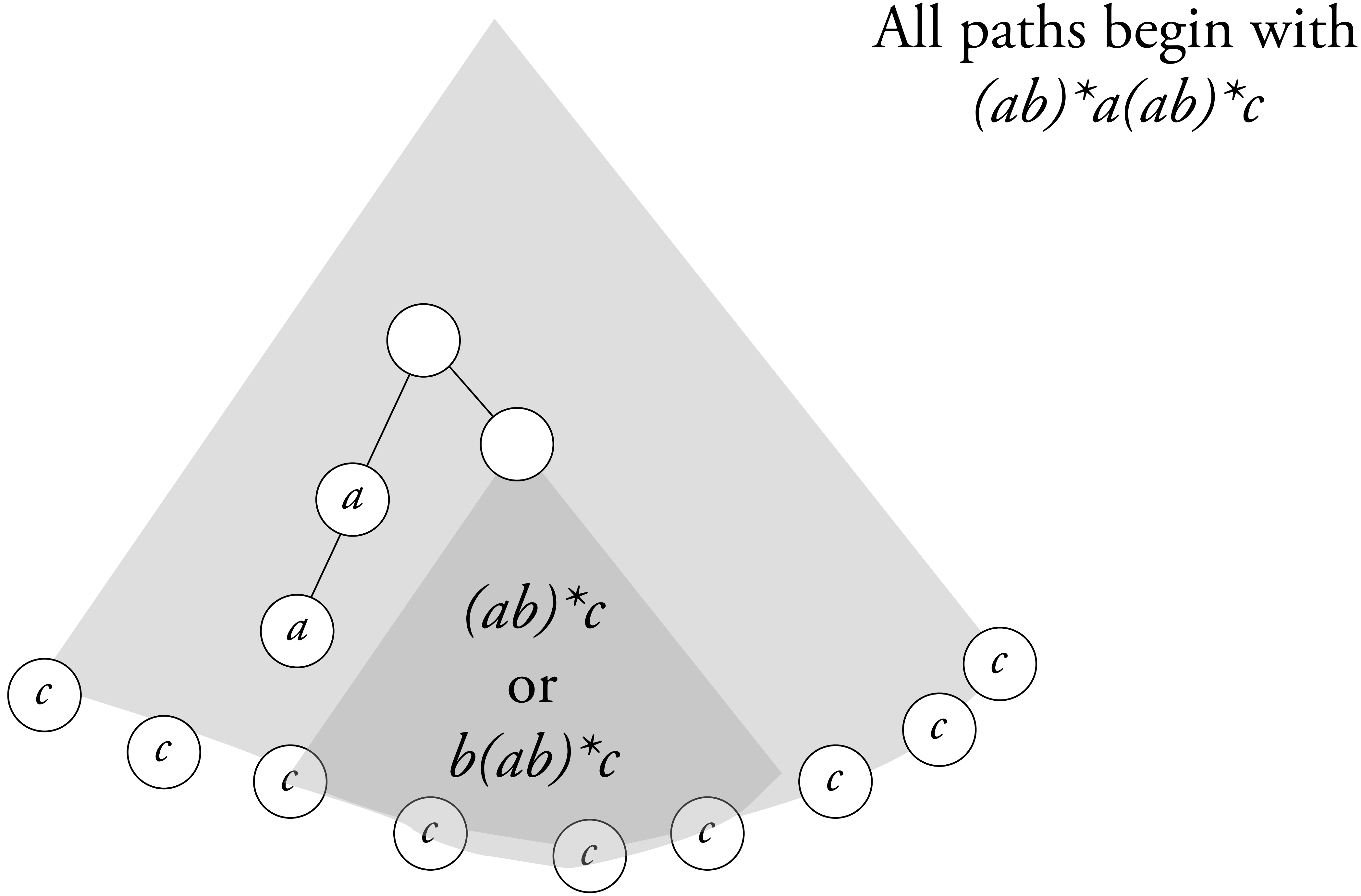
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.



1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.



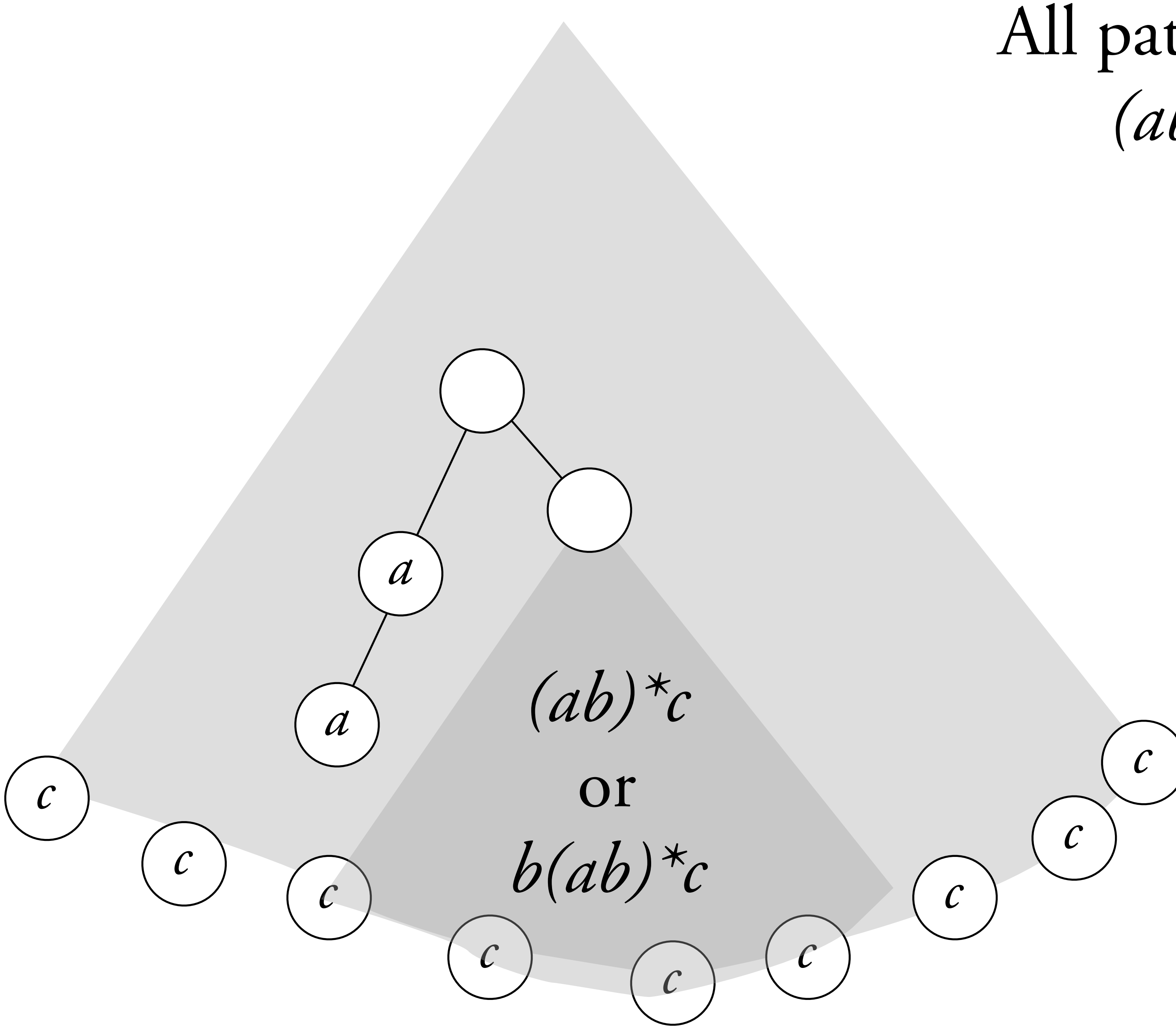
1. all paths begin with $(ab)^*a(ab)^*c$ or $(ab)^*c$
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.



1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.

$\Uparrow$

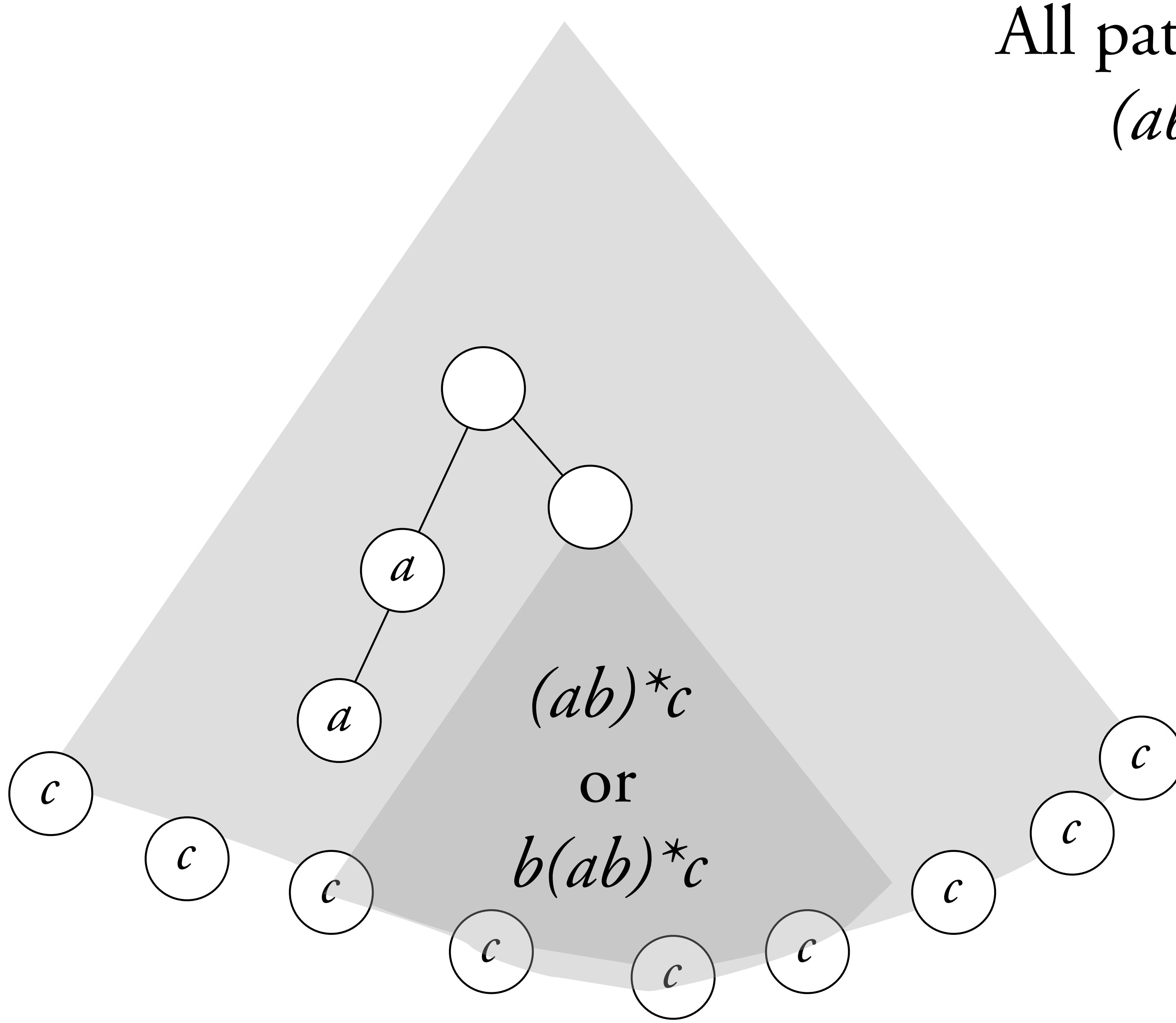
All paths begin with *(ab)*a(ab)*c*



1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.

$\Uparrow \Downarrow$

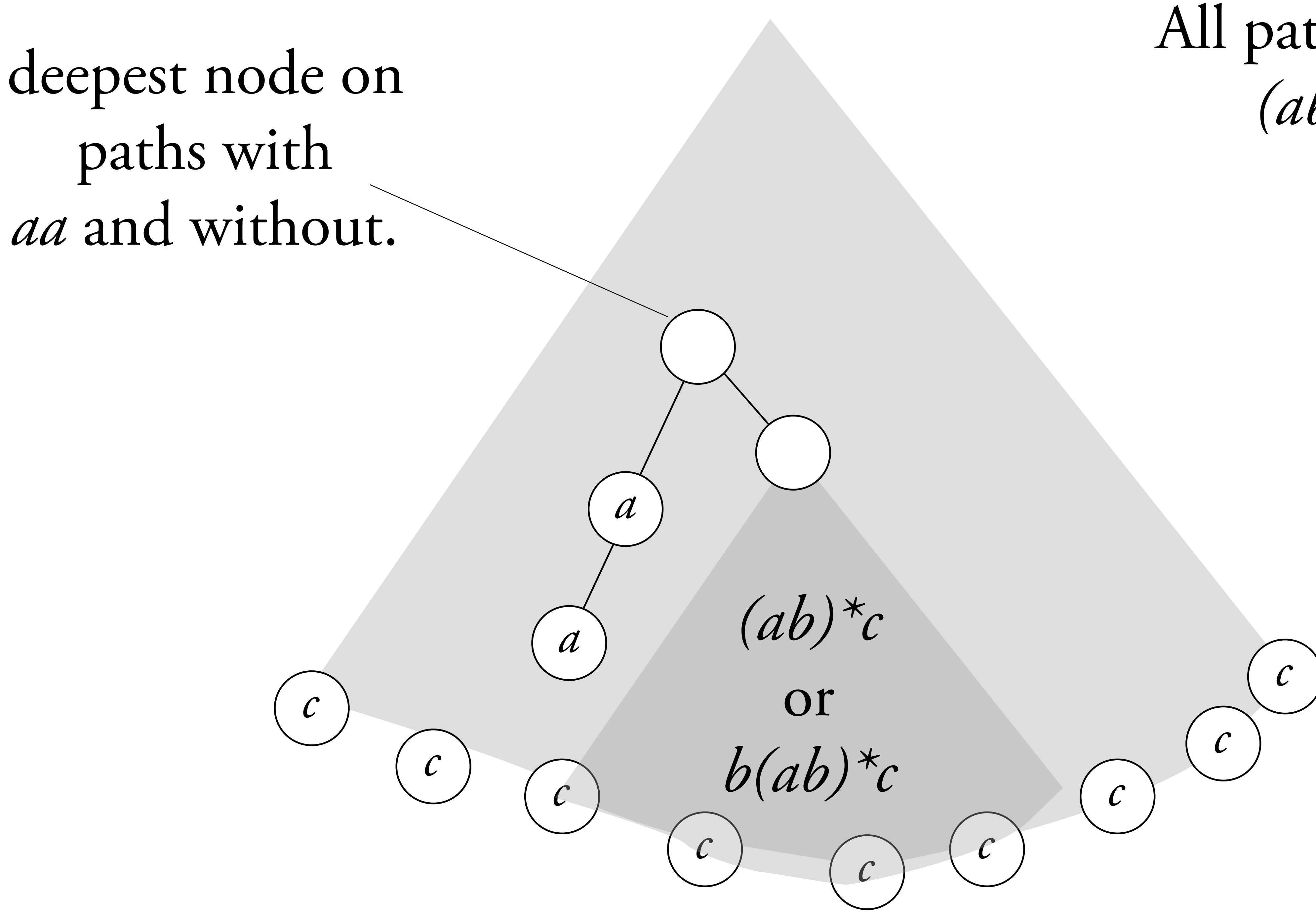
All paths begin with
*(ab)*a(ab)*c*



1. all paths begin with *(ab)*a(ab)*c* or *(ab)*c*
2. on every path, some *a* before *c* has an *a* child.
3. forbidden: siblings, one with *aa* in subtree, one without.

$\Updownarrow$

All paths begin with *(ab)*a(ab)*c*



Thm. [Maidl 00] Let L be a set of infinite words.

The tree language "all paths in L" is definable in ACTL

iff

L is definable in $\Pi_2(<)$

Thm. It is decidable if a language is definable in $\Pi_2(<)$.

generalization to infinite words
of a result of Arfi '91.

$$\exists x_1 \cdots \exists x_n \quad \forall y_1 \cdots \forall y_m \quad \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

A language definable in $\Sigma_2(<)$ is closed under the following rewriting rules.

$$\exists x_1 \cdots \exists x_n \quad \forall y_1 \cdots \forall y_m \quad \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

A language definable in $\Sigma_2(<)$ is closed under the following rewriting rules.

$$w^k \longrightarrow w^k\, v\, w^k$$

if k is large, and v is a subword of w.

$$\exists x_1 \cdots \exists x_n \quad \forall y_1 \cdots \forall y_m \quad \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

A language definable in $\Sigma_2(<)$ is closed under the following rewriting rules.

$$w^k \longrightarrow w^k\, v\, w^k$$

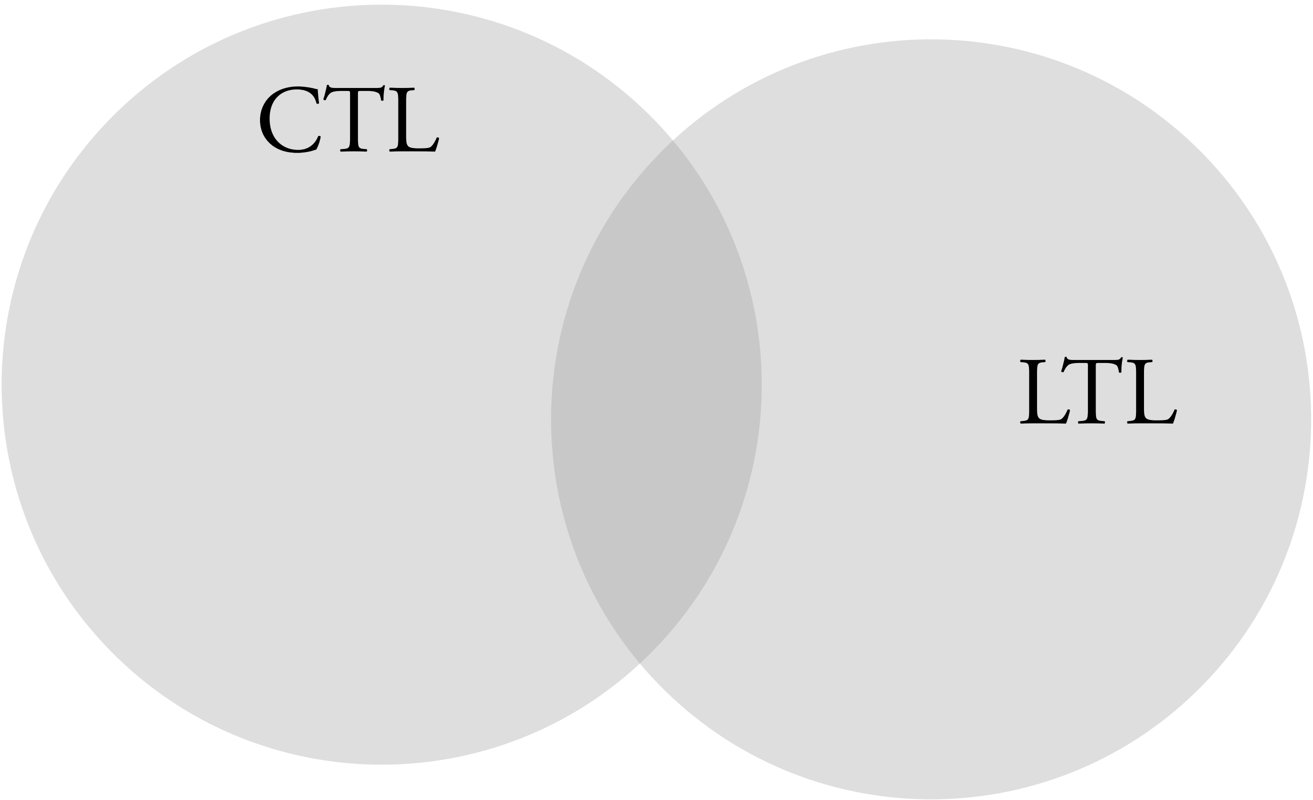
if k is large, and v is a subword of w.

$$w^\omega \longrightarrow w^k\, v\, u^\omega$$

if k is large, and v, u are subwords of w.

Contribution

- ACTL $\cap$ LTL is a proper subset of CTL $\cap$ LTL.
- An effective characterization of $\Pi_2(<)$ for infinite words.
- A simpler characterization of $\Pi_2(<)$ for finite words.



Future work

- Effective characterization of CTL...
- ...or at least the common fragment of CTL and LTL.