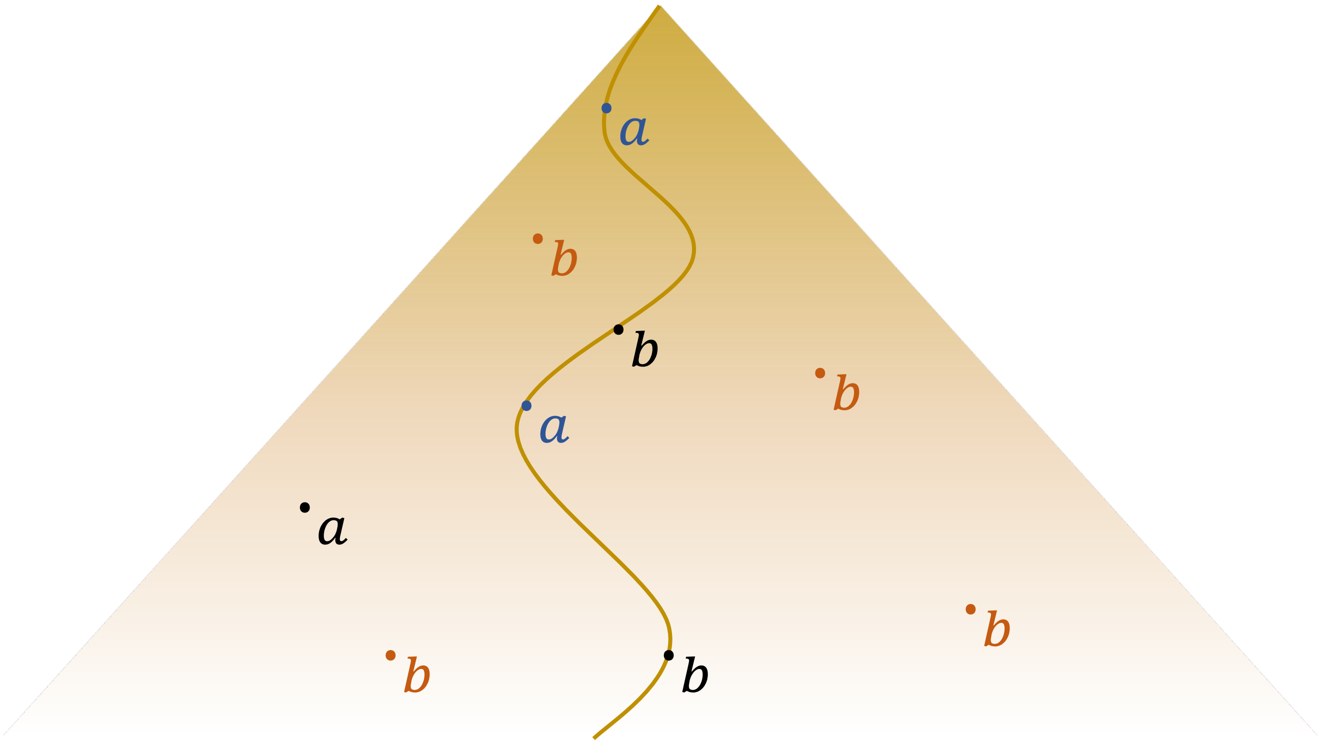
Decidable (Ac)counting with Parikh and Muller: Adding Presburger Arithmetic to Monadic Second-Order Logic over Tree-Interpretable Structures

Luisa Herrmann, Vincent Peth, and Sebastian Rudolph

Naples, CSL 2024, Feb 23

We want to compare cardinalities!

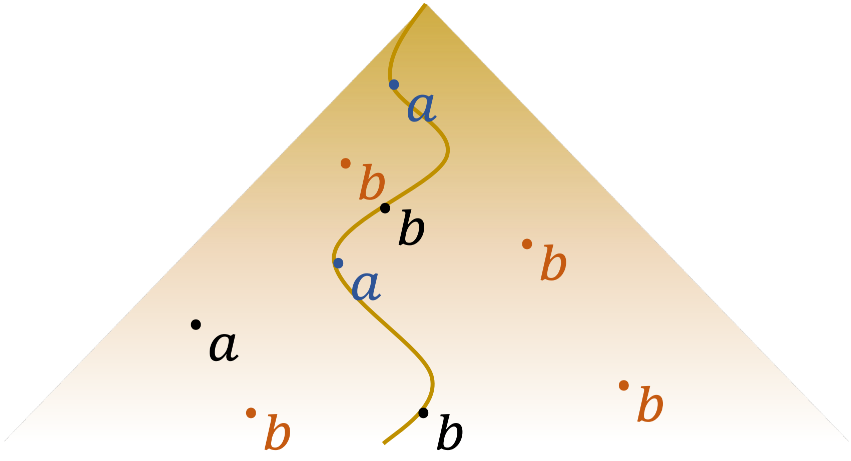
We want to compare cardinalities!



“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

We want to compare cardinalities!

- Monadic Second-Order Logic (MSO) very expressive but very limited cardinality relationships



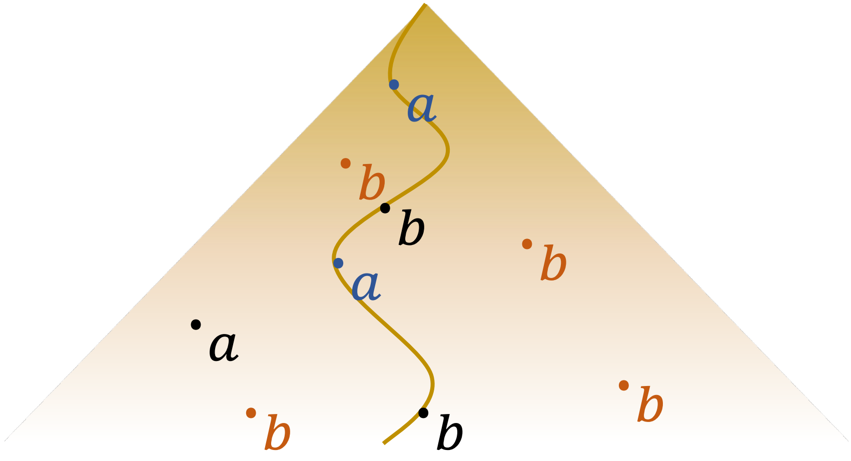
“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

We want to compare cardinalities!

- Monadic Second-Order Logic (MSO) very expressive but very limited cardinality relationships

- Counting MSO (CMSO) includes $\#X \equiv_n m$ and $\mathrm{Fin}(X)$

  → coincides on (finite) words and trees with MSO



"There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P"

We want to compare cardinalities!

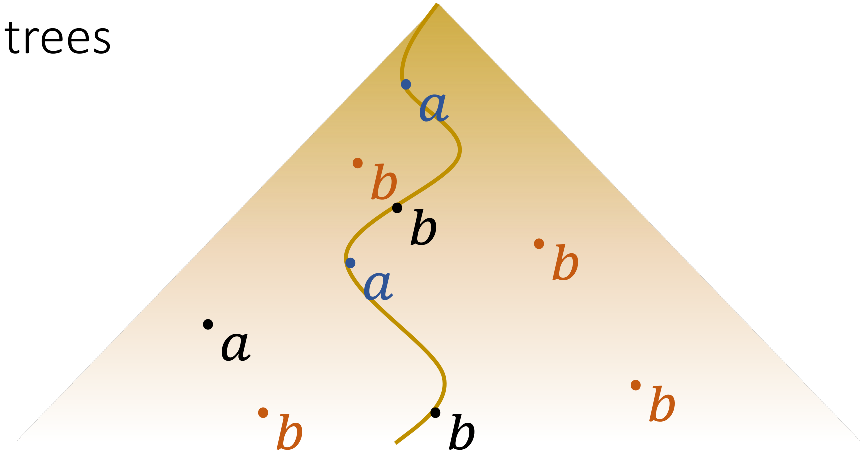
- Monadic Second-Order Logic (MSO) very expressive but very limited cardinality relationships

- Counting MSO (CMSO) includes $\#X \equiv_n m$ and $\mathrm{Fin}(X)$

  → coincides on (finite) words and trees with MSO

- MSO + Cardinality Constraints [Klaedtke, Rueß 03] includes $\#X \leq \#Y$

  → undecidable in general, decidable fragment on words, trees



“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

We want to compare cardinalities!

- Monadic Second-Order Logic (MSO) very expressive but very limited cardinality relationships

- Counting MSO (CMSO) includes $\#X \equiv_n m$ and $\mathrm{Fin}(X)$

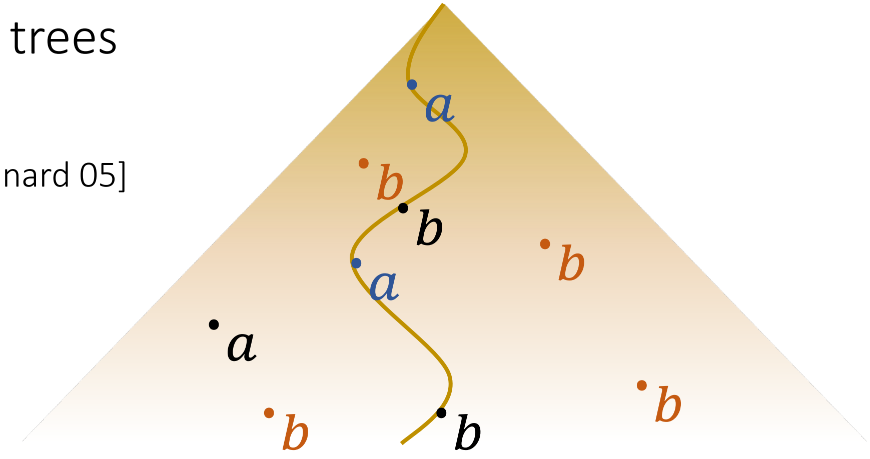
  → coincides on (finite) words and trees with MSO

- MSO + Cardinality Constraints [Klaedtke, Rueß 03] includes $\#X \leq \#Y$

  → undecidable in general, decidable fragment on words, trees

- Boolean Algebra + Presburger Arithmetic (BAPA) [Kuncak, Nguyen, Rinard 05]

  includes set operations and arithmetic
  → decidable but lacks non-unary relations



“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

We want to compare cardinalities!

- Monadic Second-Order Logic (MSO) very expressive but very limited cardinality relationships

- Counting MSO (CMSO) includes $\#X \equiv_n m$ and $\mathrm{Fin}(X)$

  → coincides on (finite) words and trees with MSO

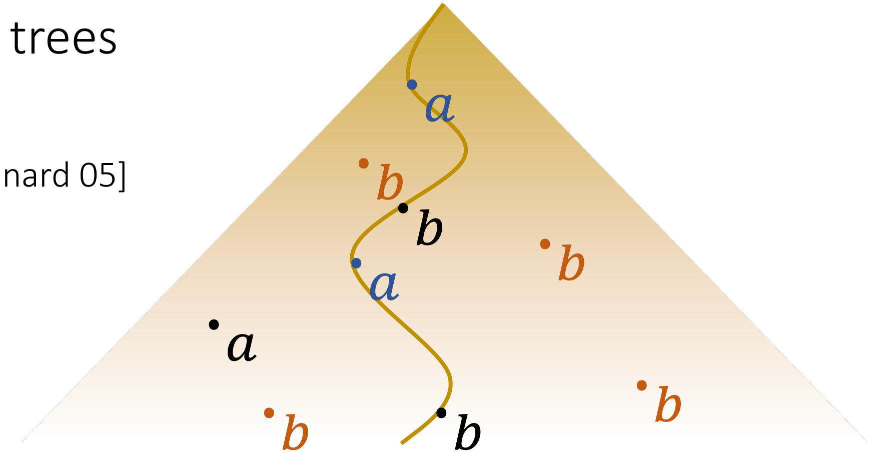
- MSO + Cardinality Constraints [Klaedtke, Rueß 03] includes $\#X \leq \#Y$

  → undecidable in general, decidable fragment on words, trees

- Boolean Algebra + Presburger Arithmetic (BAPA) [Kuncak, Nguyen, Rinard 05]

  includes set operations and arithmetic
  → decidable but lacks non-unary relations

↪ we combine these approaches



“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

Content

Content

Logic

- ωMSO·BAPA ... is undecidable
- the fragment ωMSO⋈BAPA
- ... and its normal form

Content

Logic

- ωMSO·BAPA … is undecidable
- the fragment ωMSO⋈BAPA
- … and its normal form

Automata

- Parikh-Muller Tree Automata
- … correspond to ωMSO⋈BAPA
- … have decidable emptiness problem

Content

Logic

- ωMSO·BAPA ... is undecidable
- the fragment ωMSO⋈BAPA
- ... and its normal form

Automata

- Parikh-Muller Tree Automata
- ... correspond to ωMSO⋈BAPA
- ... have decidable emptiness problem

Decidability

From CMSO to ωMSO·BAPA

signature $\mathbb{S} = \mathbb{S}_C \cup \mathbb{S}_P$

countable $\mathbb{S}$-structure $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$

From CMSO to ωMSO·BAPA

signature $\mathbb{S} = \mathbb{S}_C \cup \mathbb{S}_P$

countable $\mathbb{S}$-structure $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$

constants or variables

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \;\mid\; X(\iota) \;\mid\; \#X \equiv_n m \;\mid\; \mathrm{Fin}(X) \;\mid$$

$$\neg\varphi \;\mid\; \varphi \vee \varphi' \;\mid\; \exists x.\varphi \;\mid\; \exists X.\varphi \;\mid$$

From CMSO to ωMSO·BAPA

signature $\mathbb{S} = \mathbb{S}_C \cup \mathbb{S}_P$

countable $\mathbb{S}$-structure $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$

constants or variables

set terms $(X^c \cap Y) \cup \mathrm{P}$

$$\varphi ::= \ \mathrm{Q}(\iota_1, \ldots, \iota_n) \ \mid \ S(\iota) \ \mid \ \#S \equiv_n m \ \mid \ \mathrm{Fin}(S) \ \mid$$

$$\neg\varphi \ \mid \ \varphi \vee \varphi' \ \mid \ \exists x.\varphi \ \mid \ \exists X.\varphi \ \mid$$

From CMSO to ωMSO·BAPA

signature $\mathbb{S} = \mathbb{S}_C \cup \mathbb{S}_P$

countable $\mathbb{S}$-structure $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$

set terms $(X^c \cap Y) \cup \mathrm{P}$

constants or variables

$$\varphi ::= \mathrm{Q}(\iota_1, \ldots, \iota_n) \;\mid\; S(\iota) \;\mid\; \# S \equiv_n m \;\mid\; \mathrm{Fin}(S) \;\mid\; t_1 \le t_2 \;\mid\; t_1 \le_{\mathrm{fin}} t_2 \;\mid$$

$$\neg\varphi \;\mid\; \varphi \vee \varphi' \;\mid\; \exists x.\varphi \;\mid\; \exists X.\varphi \;\mid\; \exists k.\varphi$$

number terms $3\#S_1 + \#S_2 + k$

From CMSO to ωMSO·BAPA

signature $\mathbb{S} = \mathbb{S}_C \cup \mathbb{S}_P$

countable $\mathbb{S}$-structure $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$

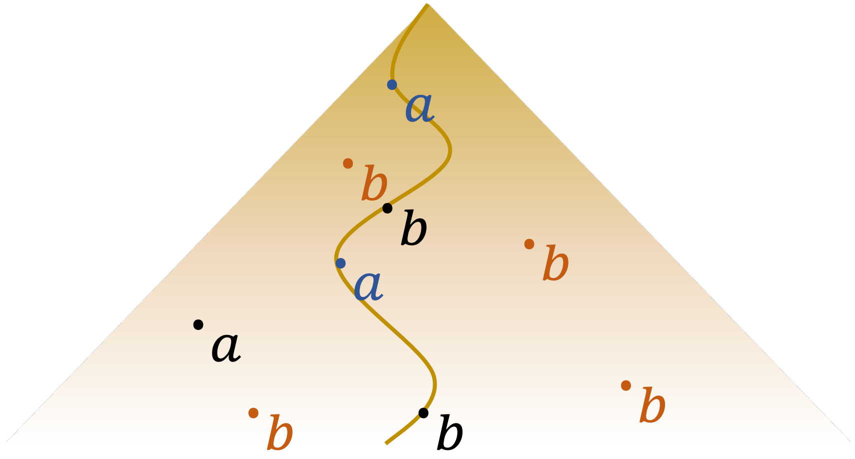
constants or variables

set terms $(X^c \cap Y) \cup \mathrm{P}$

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \ | \ S(\iota) \ | \ \# S \equiv_n m \ | \ \mathrm{Fin}(S) \ | \ t_1 \leq t_2 \ | \ t_1 \leq_{\mathrm{fin}} t_2 \ |$$

$$\neg\varphi \ | \ \varphi \vee \varphi' \ | \ \exists x.\varphi \ | \ \exists X.\varphi \ | \ \exists k.\varphi$$

number terms $3\#S_1 + \#S_2 + k$



$$\exists A.\, \mathrm{Path}(A) \ \wedge \ 2 \cdot \#(A \cap \mathrm{P}_a) \leq \#(A^c \cap \mathrm{P}_b)$$

“There exists a path P s.t. 2x the number of as on P $\leq$ the number of bs not on P”

Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\mathrm{sol}} \wedge \varphi_{\mathrm{lab}} \wedge \varphi_{\mathrm{prod}}$$

- unary predicates P_x , P_y , P_z , P_{xy} , P_{xyy} , P_{xyz}

Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{sol}} \wedge \varphi_{\text{lab}} \wedge \varphi_{\text{prod}}$$

- unary predicates P_x , P_y , P_z , P_{xy} , P_{xyy} , P_{xyz}
- $\varphi_{\text{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$

Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{sol}} \wedge \varphi_{\text{lab}} \wedge \varphi_{\text{prod}}$$

- unary predicates P_x, P_y, P_z, P_{xy}, P_{xyy}, P_{xyz}
- $\varphi_{\text{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$
- ensure that $|P_{xy}| = |P_x| \cdot |P_y|$

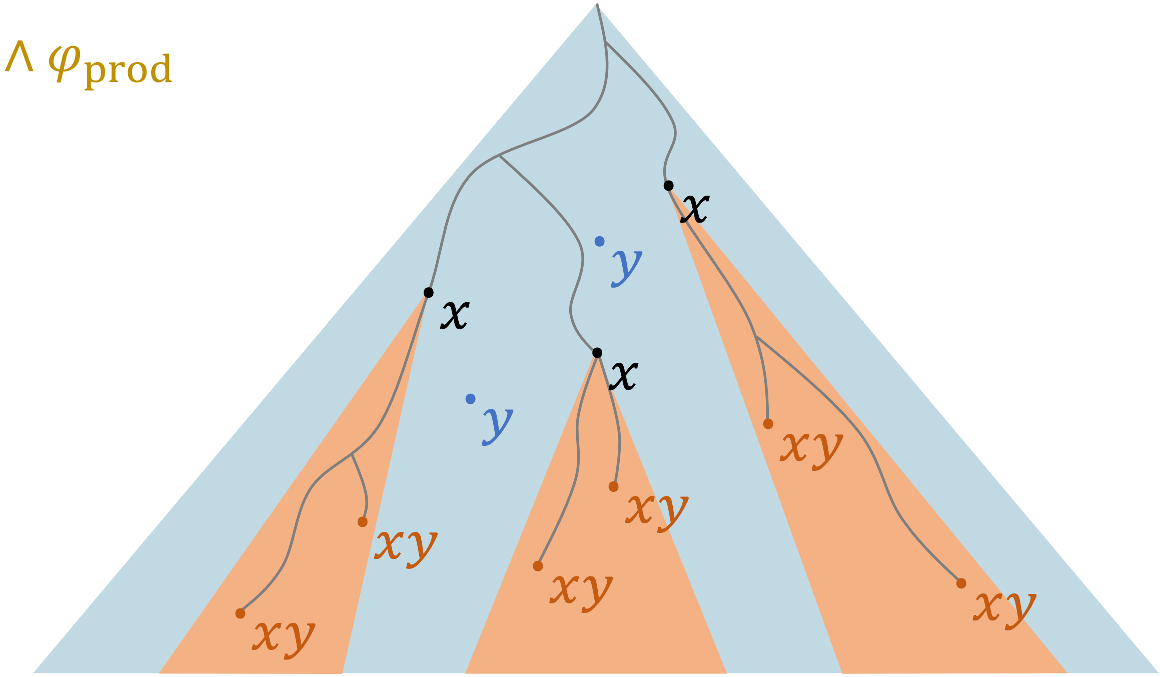
Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\mathrm{sol}} \wedge \varphi_{\mathrm{lab}} \wedge \varphi_{\mathrm{prod}}$$

- unary predicates P_x, P_y, P_z, P_{xy}, P_{xyy}, P_{xyz}
- $\varphi_{\mathrm{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$
- ensure that $|P_{xy}| = |P_x| \cdot |P_y|$

φ_{lab} for appropriate labeling

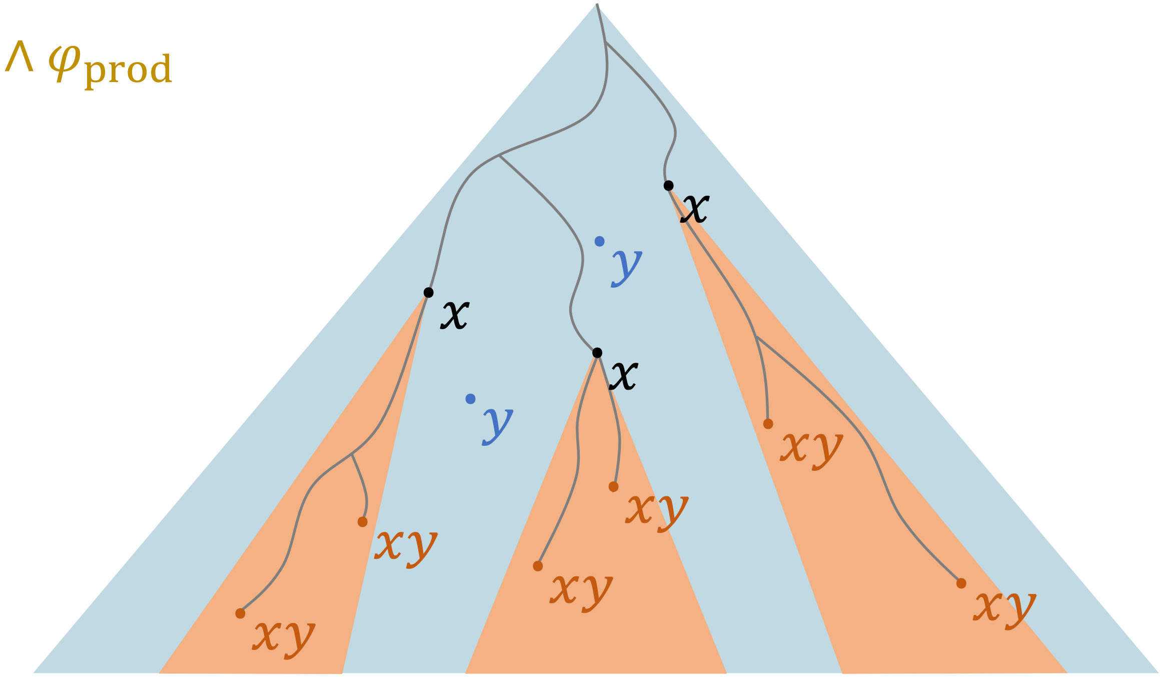


Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees

$$\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{sol}} \wedge \varphi_{\text{lab}} \wedge \varphi_{\text{prod}}$$



- unary predicates P_x , P_y , P_z , P_{xy} , P_{xyy} , P_{xyz}
- $\varphi_{\text{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$
- ensure that $|P_{xy}| = |P_x| \cdot |P_y|$

φ_{lab} for appropriate labeling

$$\varphi_{\text{prod}} \coloneqq \bigwedge_{x,xy} \forall z \in P_x. \exists Z. \operatorname{sub}(z, Z) \wedge \#\big(Z \cap P_{xy}\big) =_{\text{fin}} \#P_y$$

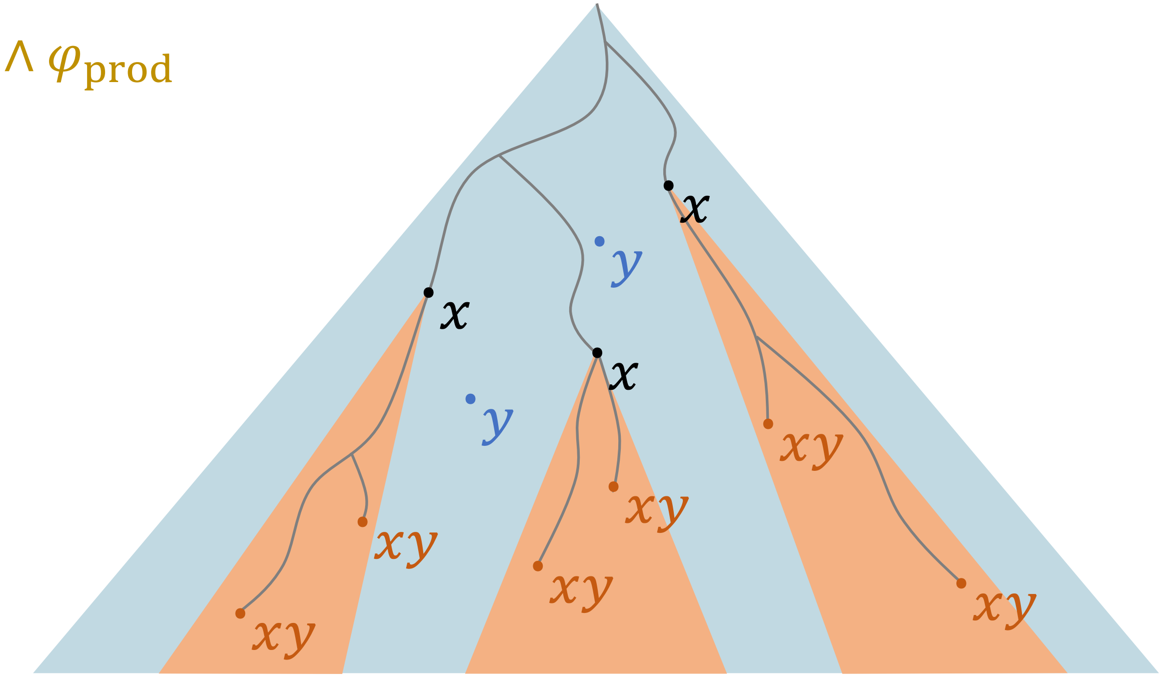
Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{sol}} \wedge \varphi_{\text{lab}} \wedge \varphi_{\text{prod}}$$

- unary predicates P_x , P_y , P_z , P_{xy} , P_{xyy} , P_{xyz}
- $\varphi_{\text{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$
- ensure that $|P_{xy}| = |P_x| \cdot |P_y|$

φ_{lab} for appropriate labeling



Proposition. For any positive Diophantine equation $\mathcal{D}$, satisfaction of $\varphi_{\mathcal{D}}$ over (finite or infinite) labeled trees coincides with solvability of $\mathcal{D}$.

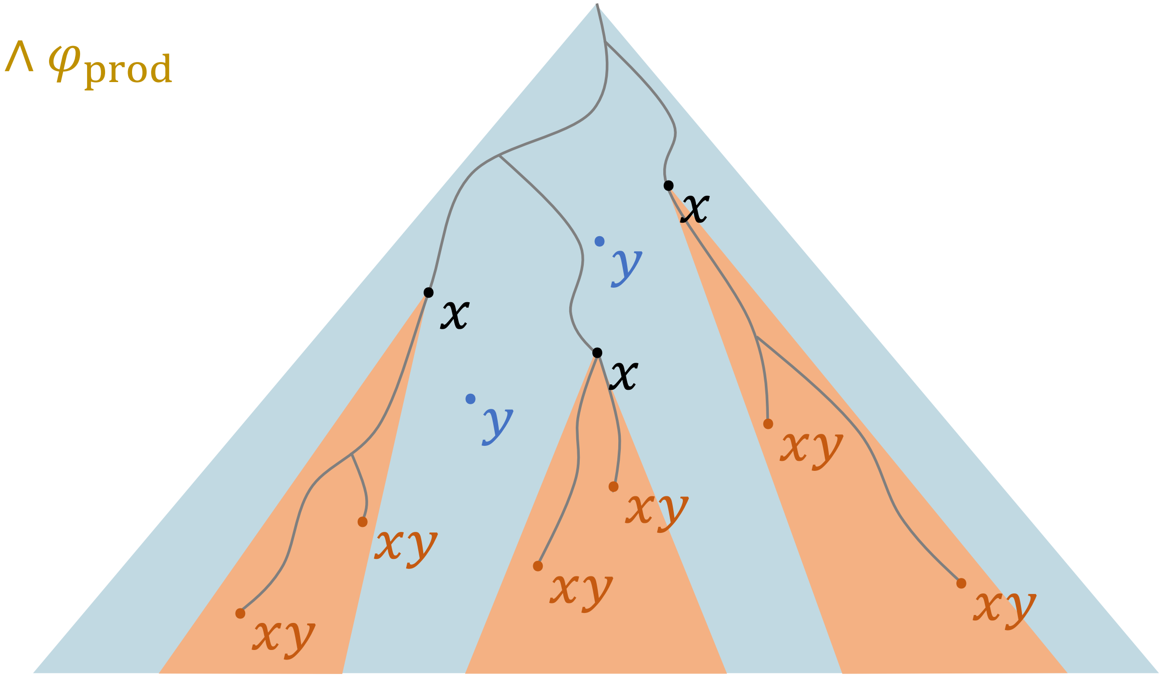
Full ωMSO·BAPA is undecidable

by encoding positive Diophantine equations on labeled trees $\mathcal{D} \coloneqq 2xy^2 + 5y + 3z = 5xyz$

$$\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{sol}} \wedge \varphi_{\text{lab}} \wedge \varphi_{\text{prod}}$$

- unary predicates P_x, P_y, P_z, P_{xy}, P_{xyy}, P_{xyz}
- $\varphi_{\text{sol}} \coloneqq 2 \cdot \#P_{xyy} + 5 \cdot \#P_y + 3 \cdot \#P_z = 5 \cdot \#P_{xyz}$
- ensure that $|P_{xy}| = |P_x| \cdot |P_y|$

φ_{lab} for appropriate labeling



Proposition. For any positive Diophantine equation $\mathcal{D}$, satisfaction of $\varphi_{\mathcal{D}}$ over (finite or infinite) labeled trees coincides with solvability of $\mathcal{D}$.

Proposition. Satisfiability of the class of ωMSO·BAPA sentences of the shape $\varphi_{\mathcal{D}}$ is undecidable.

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathrm{fin}} t_2 \mid$$

$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\,\varphi \mid \exists X.\,\varphi \mid \exists k.\,\varphi$$

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\text{fin}} t_2 \mid$$

$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \exists k.\, \varphi$$

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or "outermost" existentially quantified (not in the scope of ∀ or ¬)

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\varphi ::= \ \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathsf{fin}} t_2 \mid$$
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \exists k.\, \varphi$$

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or “outermost” existentially quantified (not in the scope of ∀ or ¬)
- delicate: non-assertive and
  - occurring in $t_1 \leq_{(\mathrm{fin})} t_2$ or
  - occurring together with a delicate variable in atom

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathsf{fin}} t_2 \mid$$
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \exists k.\, \varphi$$

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or "outermost" existentially quantified (not in the scope of ∀ or ¬)
- delicate: non-assertive and
  - occurring in $t_1 \leq_{(\mathrm{fin})} t_2$ or
  - occurring together with a delicate variable in atom

φ is an ωMSO⋈BAPA formula iff each of its predicate atoms $\mathrm{Q}(\iota_1, \dots, \iota_n)$ contains at most one delicate variable

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\exists A.\,\mathrm{Path}(A)\ \wedge\ 2 \cdot \#(A \cap \mathrm{P}_a) \leq \#(A^c \cap \mathrm{P}_b)$$

good!

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or “outermost” existentially quantified (not in the scope of ∀ or ¬)
- delicate: non-assertive and
  - occurring in $t_1 \leq_{(\mathrm{fin})} t_2$ or
  - occurring together with a delicate variable in atom

φ is an ωMSO⋈BAPA formula iff each of its predicate atoms $\mathrm{Q}(\iota_1, \ldots, \iota_n)$ contains at most one delicate variable

The Fragment ωMSO⋈BAPA [ˈoːmzo‖ˌbapa]

$$\exists X \exists V \forall Y \exists Y'.\ \mathrm{Path}(X) \wedge \varphi_{\mathrm{MSO}}(V) \wedge$$
$$\#(X \cap Y \cap V) \leq \#(Y' \cap V) \wedge (\forall z. Y'(z) \Rightarrow \mathrm{P_{red}}(z))$$

good!

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or "outermost" existentially quantified (not in the scope of ∀ or ¬)
- delicate: non-assertive and
  - occurring in $t_1 \leq_{(\mathrm{fin})} t_2$ or
  - occurring together with a delicate variable in atom

φ is an ωMSO⋈BAPA formula iff each of its predicate atoms $\mathrm{Q}(\iota_1, \ldots, \iota_n)$ contains at most one delicate variable

The Fragment ωMSO⋈BAPA ['oːmzo||ˌbapa]

$$\mathrm{sub}(z, Z) \coloneqq \forall y.\, Z(y) \Leftrightarrow \prec^{*} (z, y)$$

$$\varphi_{\mathrm{prod}} \coloneqq \bigwedge_{x,xy} \forall z \in \mathrm{P}_x.\, \exists Z.\, \mathrm{sub}(z, Z) \wedge \#\big(Z \cap \mathrm{P}_{xy}\big) =_{\mathrm{fin}} \#\mathrm{P}_y$$

bad! ☹

Individual and set variables in an ωMSO·BAPA formula φ can be

- assertive: free or "outermost" existentially quantified (not in the scope of ∀ or ¬)
- delicate: non-assertive and
  - occurring in $t_1 \leq_{(\mathrm{fin})} t_2$ or
  - occurring together with a delicate variable in atom

φ is an ωMSO⋈BAPA formula iff each of its predicate atoms $\mathrm{Q}(\iota_1, \dots, \iota_n)$ contains at most one delicate variable

Normalization of ωMSO⋈BAPA

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\text{fin}} t_2 \mid$$

$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \exists k.\, \varphi$$

Normalization of ωMSO⋈BAPA

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathrm{fin}} t_2 \mid$$
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\varphi \mid \exists X.\varphi \mid \exists k.\varphi$$

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

φ_i CMSO formulae

$\chi_{i,j}$ Parikh constraints

$$3 + \#X_2 \leq_{\mathrm{fin}} 2 \cdot \#Y + \#X_1$$

Normalization of ωMSO⋈BAPA

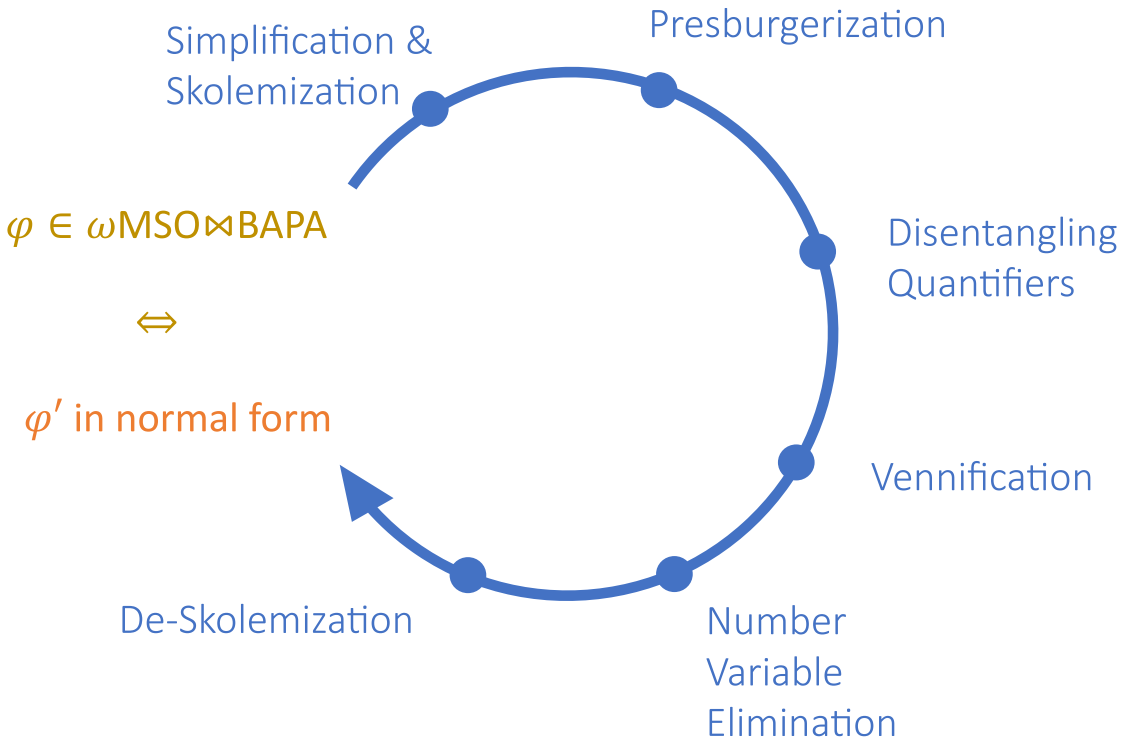
$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathrm{fin}} t_2 \mid$$
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\varphi \mid \exists X.\varphi \mid \exists k.\varphi$$

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

φ_i CMSO formulae

$\chi_{i,j}$ Parikh constraints

$$3 + \#X_2 \leq_{\mathrm{fin}} 2 \cdot \#Y + \#X_1$$



$\varphi \in \omega$MSO⋈BAPA

⇔

φ' in normal form

Simplification & Skolemization

Presburgerization

Disentangling Quantifiers

Vennification

Number Variable Elimination

De-Skolemization

Normalization of ωMSO⋈BAPA

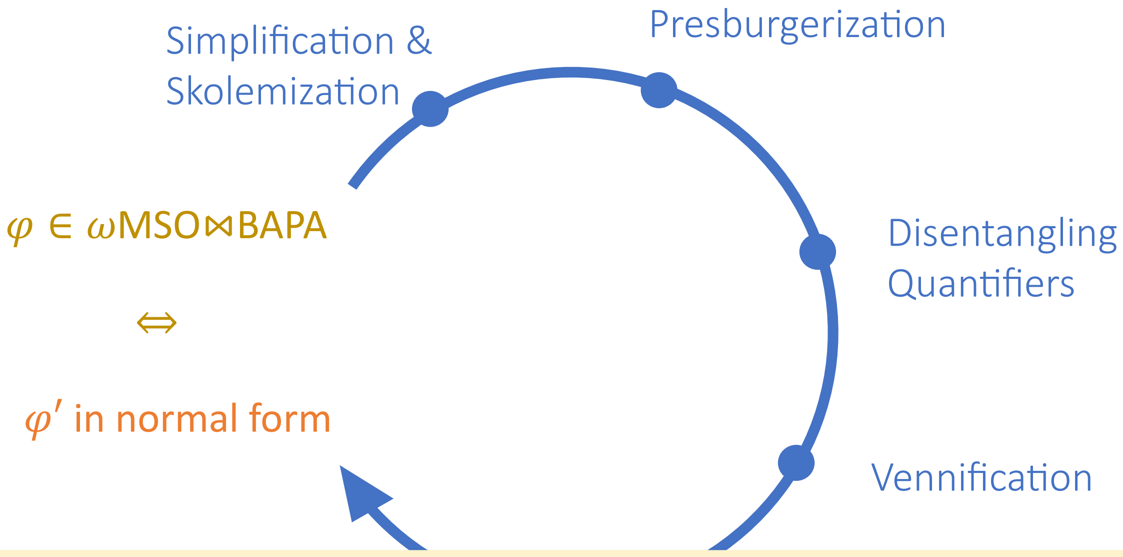
$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathrm{fin}} t_2 \mid$$
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\varphi \mid \exists X.\varphi \mid \exists k.\varphi$$

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

φ_i CMSO formulae

$\chi_{i,j}$ Parikh constraints

$$3 + \#X_2 \leq_{\mathrm{fin}} 2 \cdot \#Y + \#X_1$$


Simplification & Skolemization
Presburgerization
Disentangling Quantifiers
Vennification
$\varphi \in \omega$MSO⋈BAPA
⇔
φ' in normal form

Proposition. For each ωMSO⋈BAPA formula φ we can compute an equivalent formula φ' of the form

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

where φ_i are CMSO formulae and $\chi_{i,j}$ are Parikh constraints.

Normalization of ωMSO⋈BAPA

$$\varphi ::= \mathrm{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid \#S \equiv_n m \mid \mathrm{Fin}(S) \mid t_1 \leq t_2 \mid t_1 \leq_{\mathrm{fin}} t_2 \mid$$

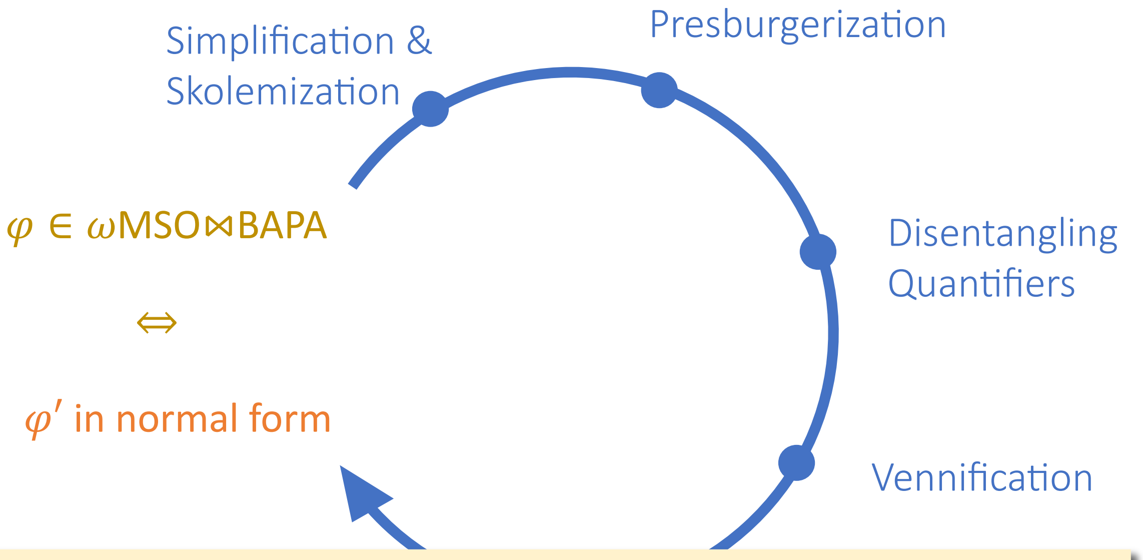
$$\neg\varphi \mid \varphi \vee \varphi' \mid \exists x.\, \varphi \mid \exists X.\, \varphi \mid \exists k.\, \varphi$$

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

φ_i CMSO formulae

$\chi_{i,j}$ Parikh constraints

$$3 + \#X_2 \leq_{\mathrm{fin}} 2 \cdot \#Y + \#X_1$$



Simplification & Skolemization
Presburgerization
$\varphi \in \omega$MSO⋈BAPA
Disentangling Quantifiers
⇔
φ' in normal form
Vennification

Proposition. For each ωMSO⋈BAPA formula φ we can compute a formula φ' of the form

$$\varphi' = \exists X_1 \dots X_n. \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

where φ_i are MSO formulae and $\chi_{i,j}$ are Parikh constraints that is equivalent to φ over labeled infinite binary trees.

Satisfiability of $\varphi = \exists X_1 \ldots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$

over infinite labeled trees decidable?

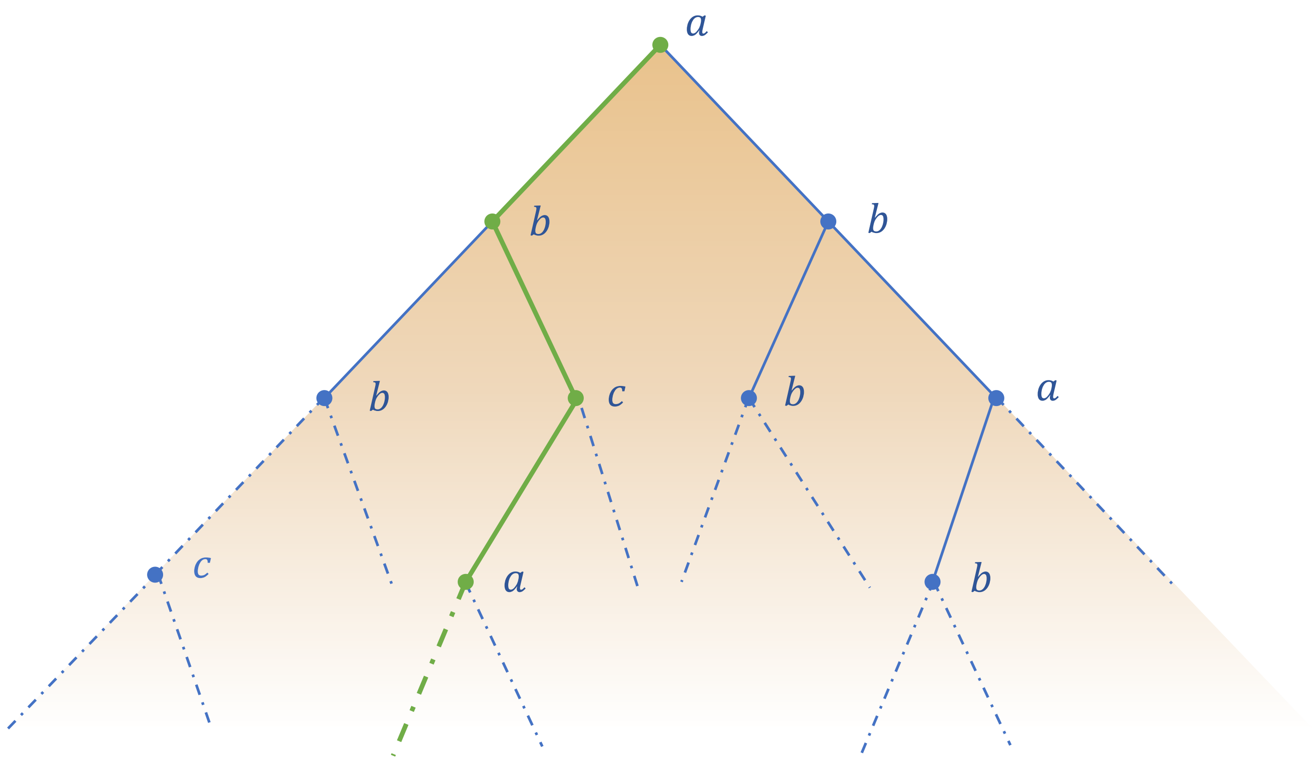
Satisfiability of $\varphi = \exists X_1 \dots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$

over infinite labeled trees decidable?

↓

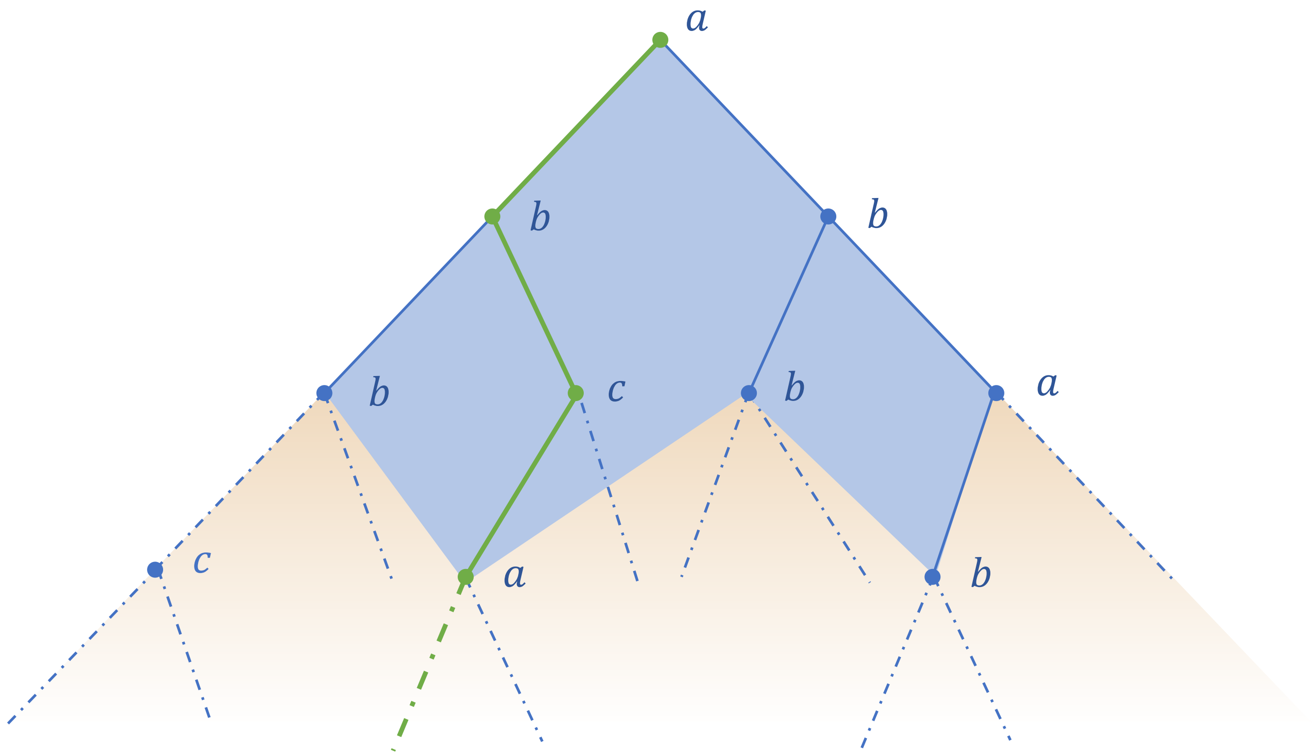
Tree Automata!

Counting with Tree Automata



“There exists a path P s.t. 2x the number of as on P $\leq_{\mathrm{fin}}$ the number of bs not on P”

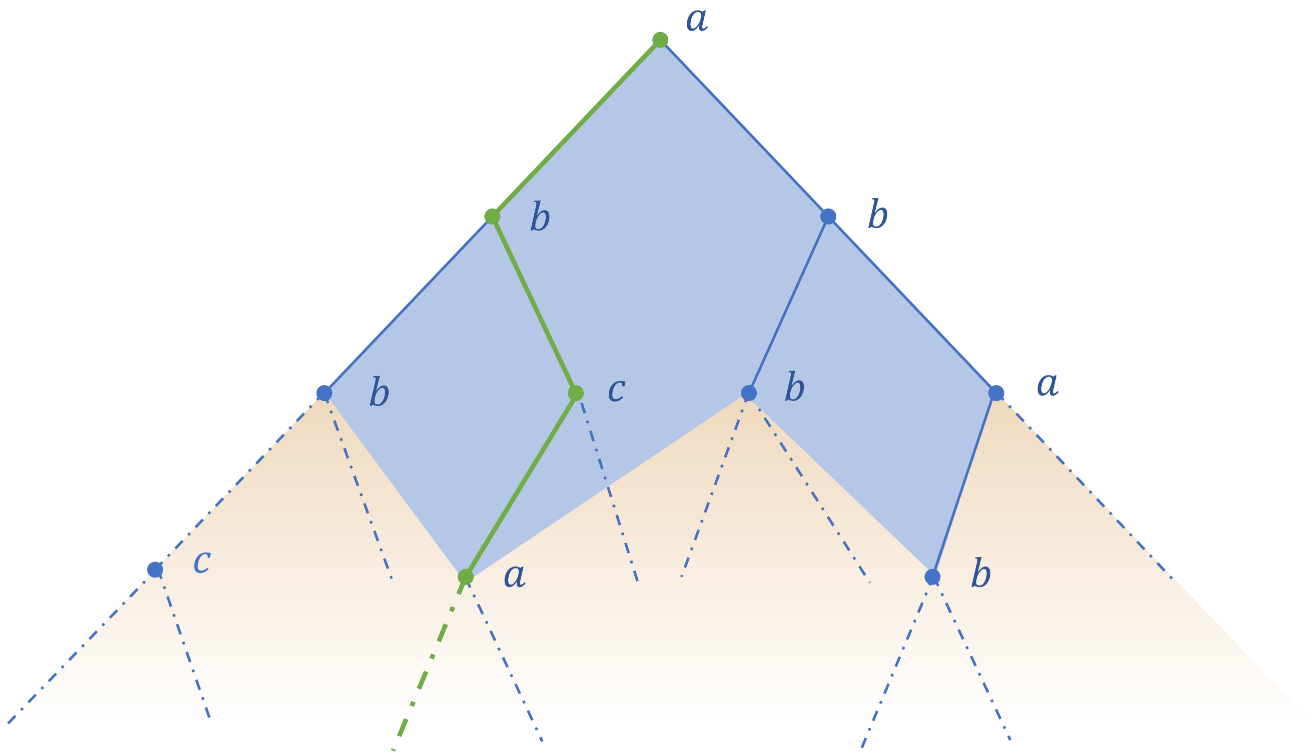
Counting with Tree Automata



"There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P"

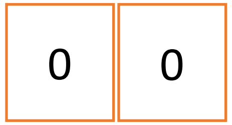
Counting with Tree Automata

Idea: generalize Parikh Automata [Klaedtke, Rueß 03]



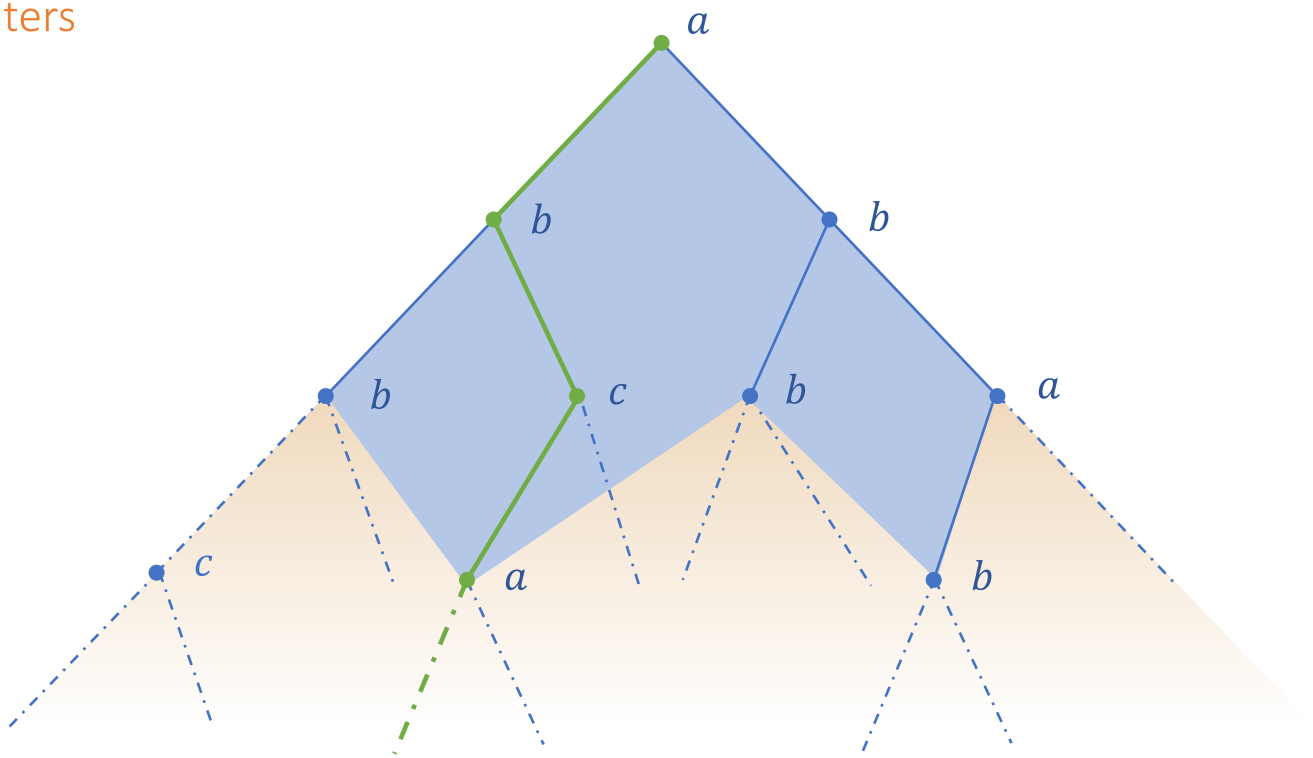
“There exists a path P s.t. 2x the number of as on P $\leq_{\mathrm{fin}}$ the number of bs not on P”

Counting with Tree Automata



Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters

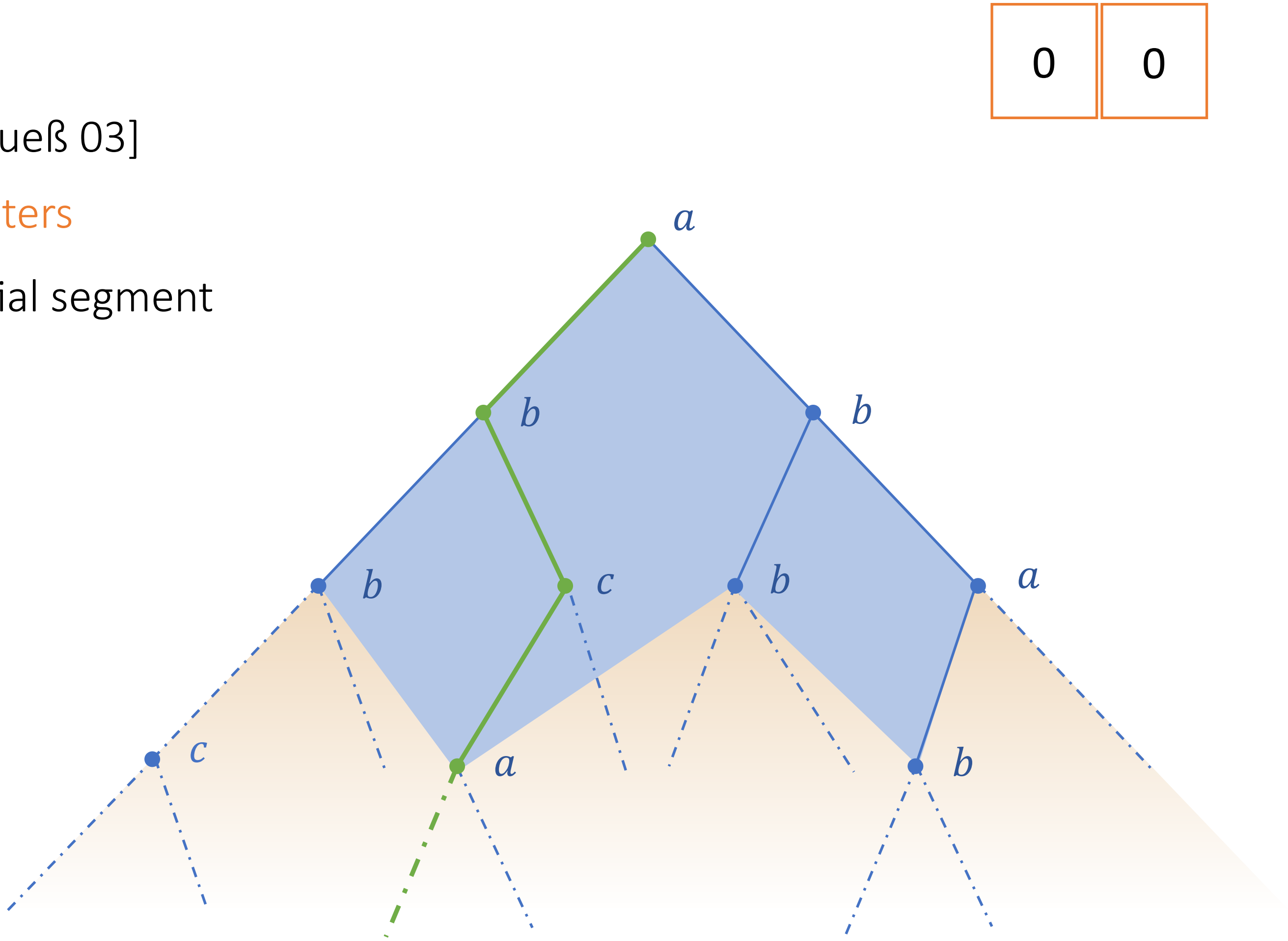


“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

Counting with Tree Automata

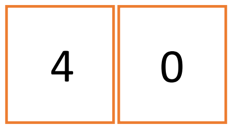
Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment



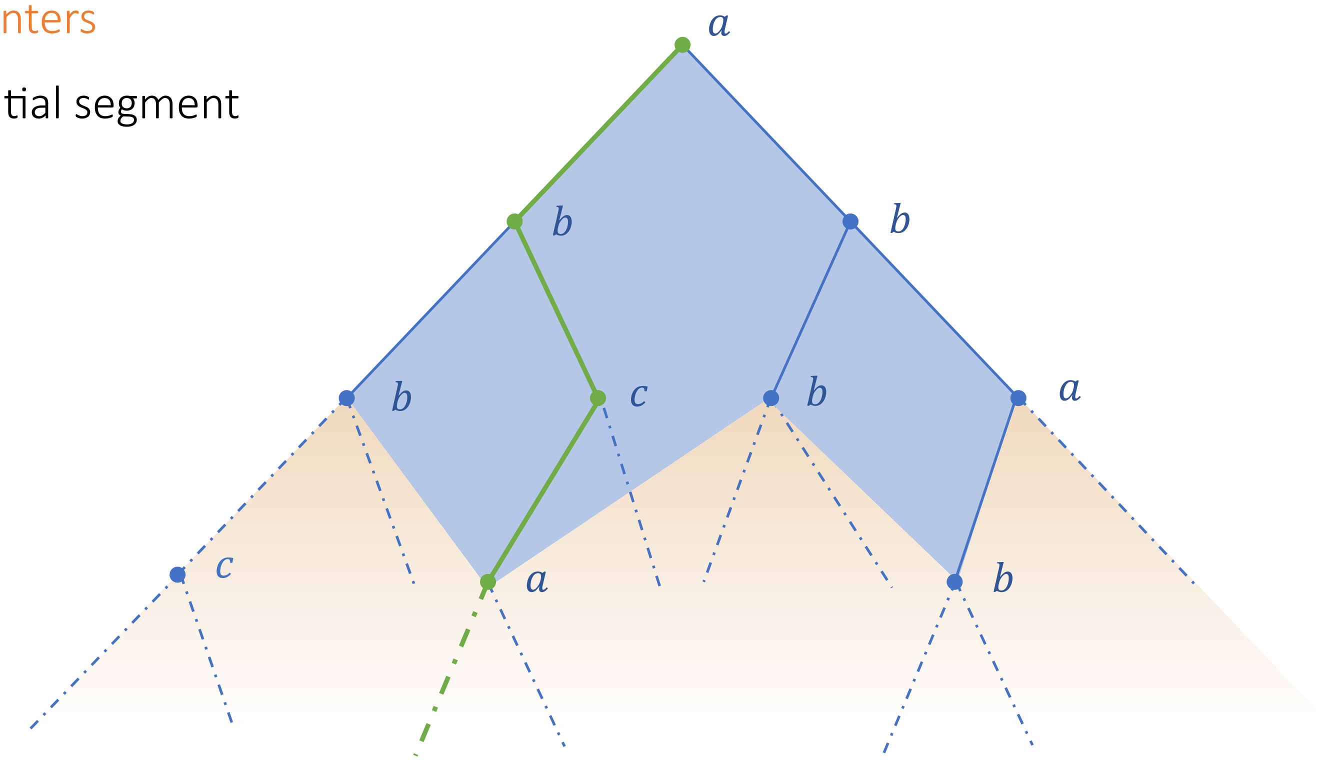
“There exists a path P s.t. 2x the number of as on P $\leq_{\mathrm{fin}}$ the number of bs not on P”

Counting with Tree Automata



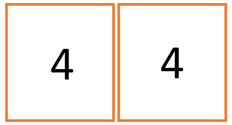
Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment



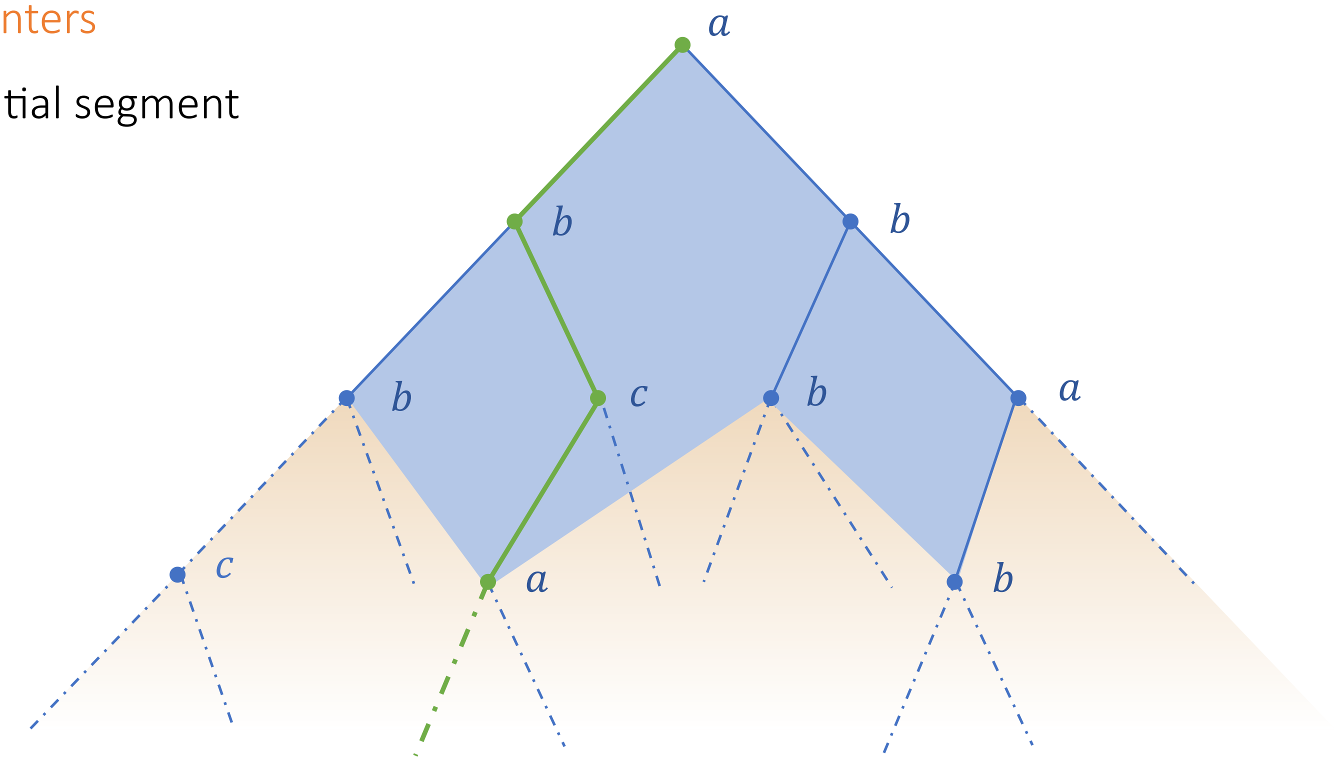
“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

Counting with Tree Automata



Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment

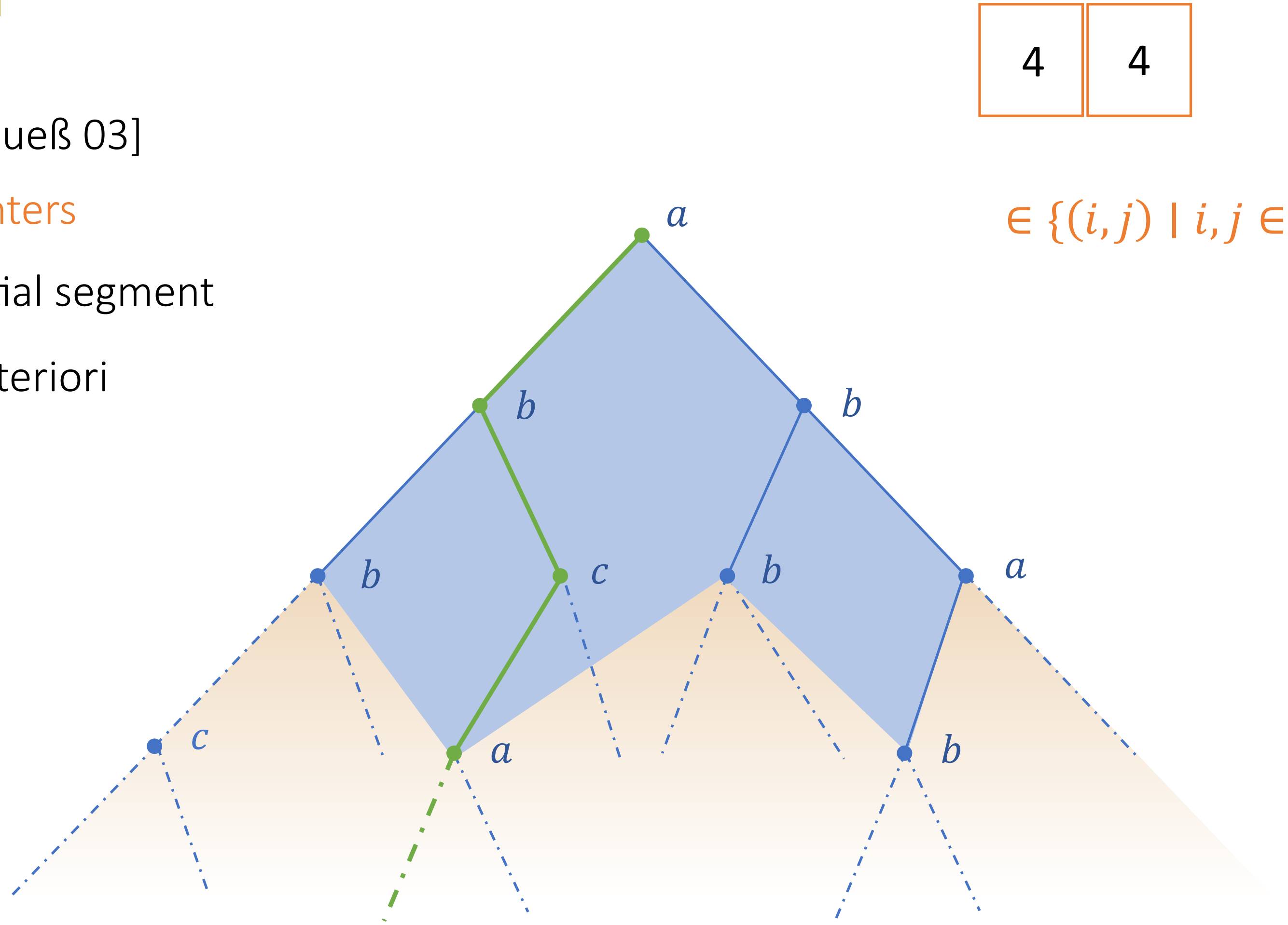


“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

Counting with Tree Automata

Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment
- test counter configuration a posteriori



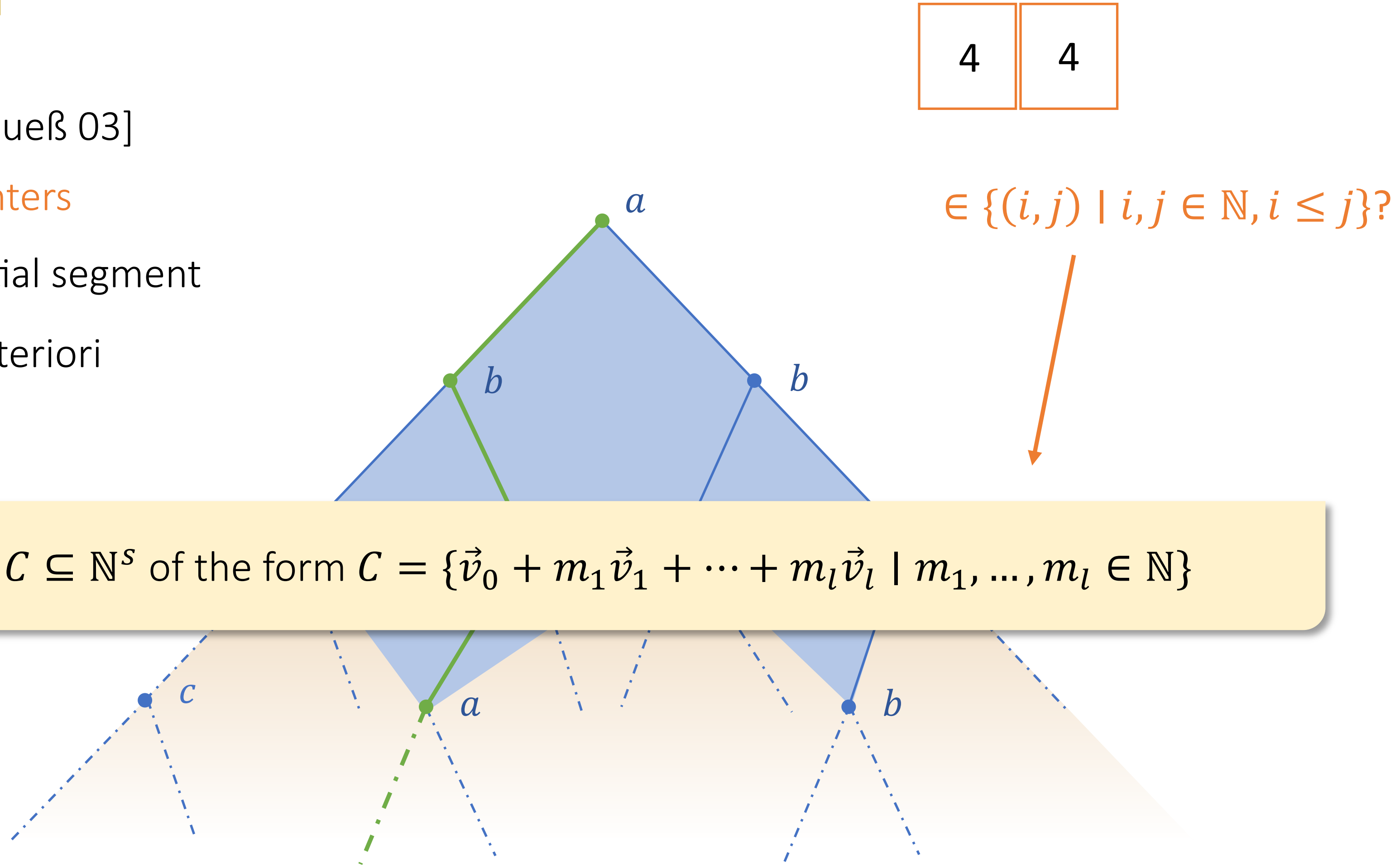
$\in \{(i,j) \mid i,j \in \mathbb{N}, i \leq j\}$?

“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

Counting with Tree Automata

Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment
- test counter configuration a posteriori



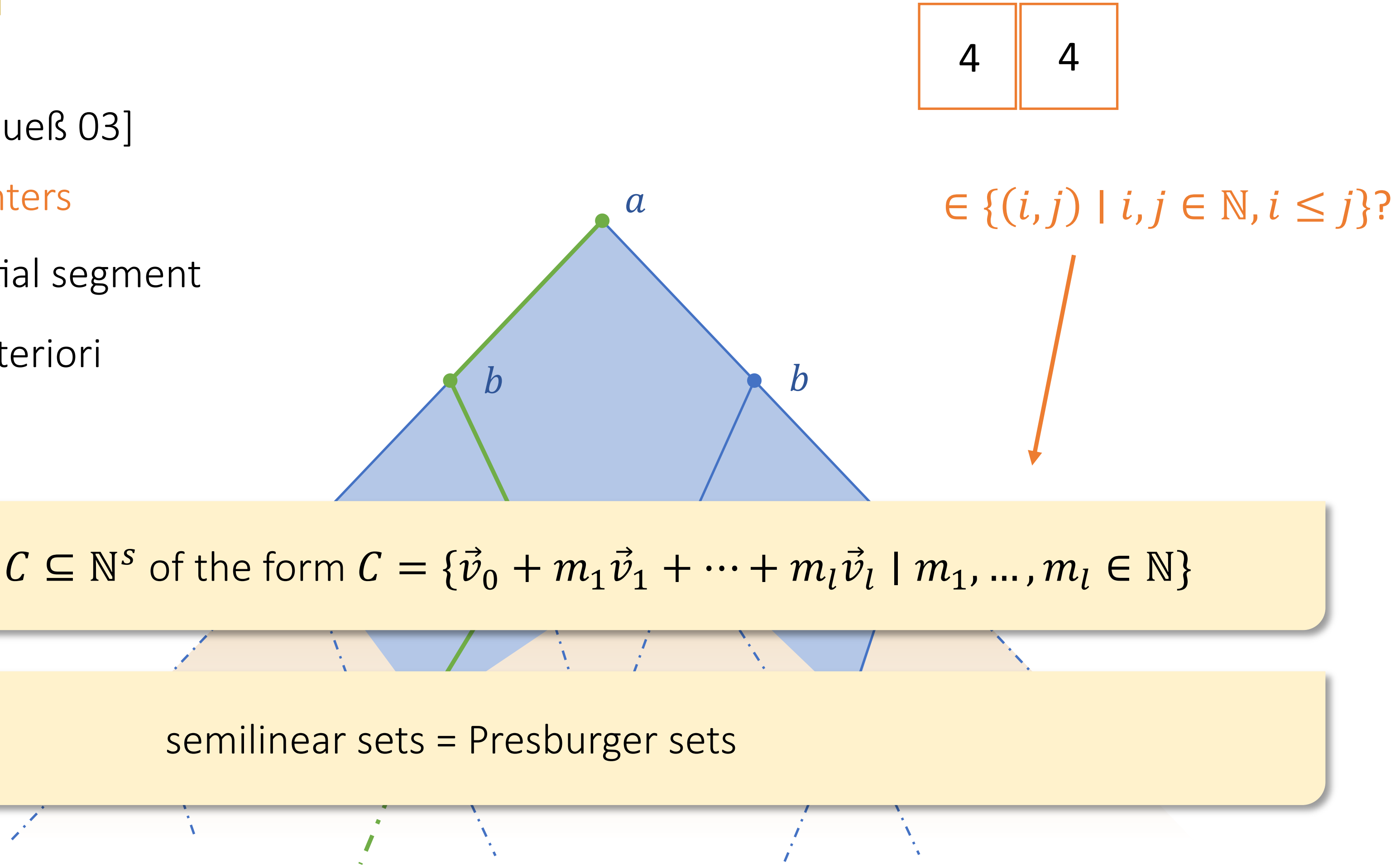
semilinear set: finite union of sets $C \subseteq \mathbb{N}^s$ of the form $C = \{\vec{v}_0 + m_1\vec{v}_1 + \cdots + m_l\vec{v}_l \mid m_1, \ldots, m_l \in \mathbb{N}\}$

"There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P"

Counting with Tree Automata

Idea: generalize Parikh Automata [Klaedtke, Rueß 03]

- use finite number of global counters
- increment counters on finite initial segment
- test counter configuration a posteriori



semilinear set: finite union of sets $C \subseteq \mathbb{N}^s$ of the form $C = \{\vec{v}_0 + m_1\vec{v}_1 + \cdots + m_l\vec{v}_l \mid m_1, \ldots, m_l \in \mathbb{N}\}$

Theorem [Ginsburg, Spanier 64]. semilinear sets = Presburger sets

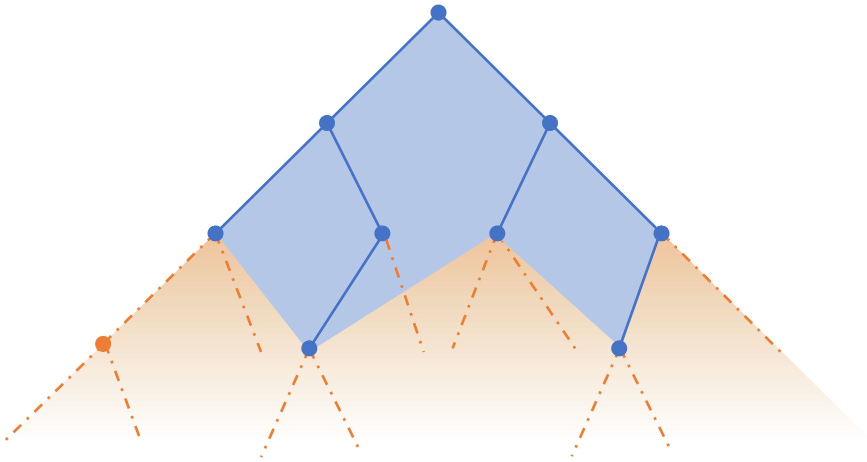
“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$



tree $\zeta \in \mathrm{T}_{\Xi}^{\omega}$

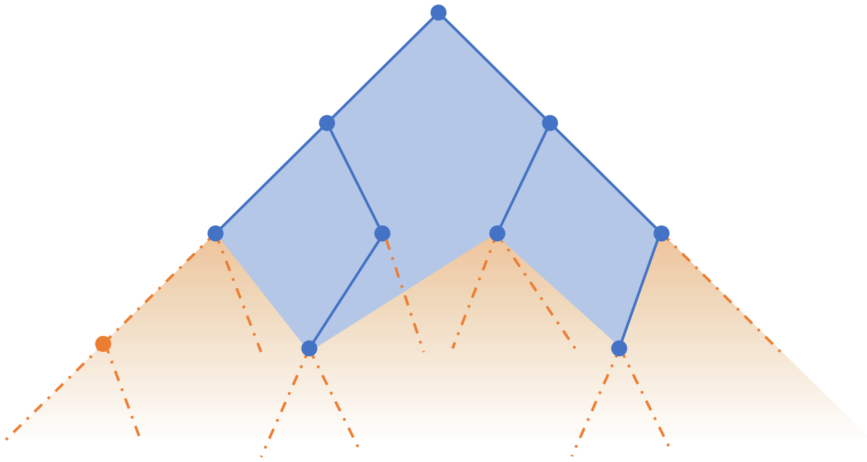
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$

- $Q = Q_P \cup Q_\omega$ finite set of states



tree $\zeta \in \mathrm{T}_\Xi^\omega$

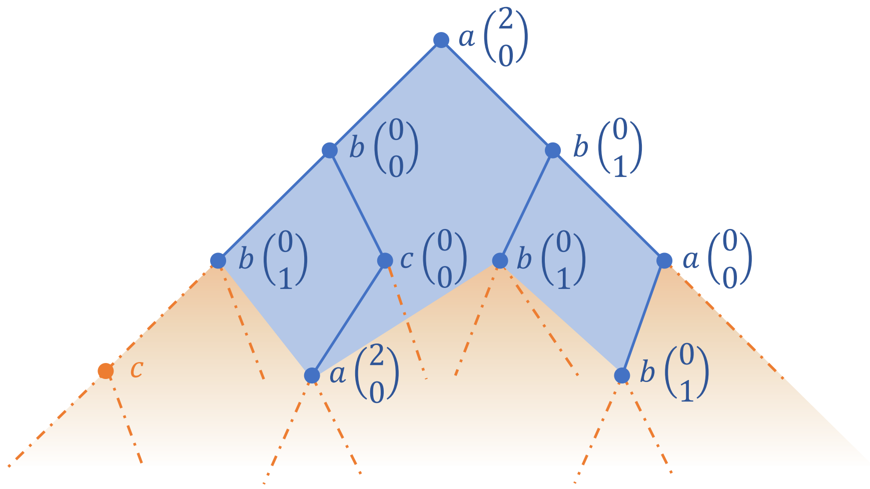
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$

- $Q = Q_P \cup Q_\omega$ finite set of states
- $\Xi = (\Sigma \times D) \cup \Sigma$ alphabet



tree $\zeta \in \mathrm{T}^\omega_\Xi$

Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

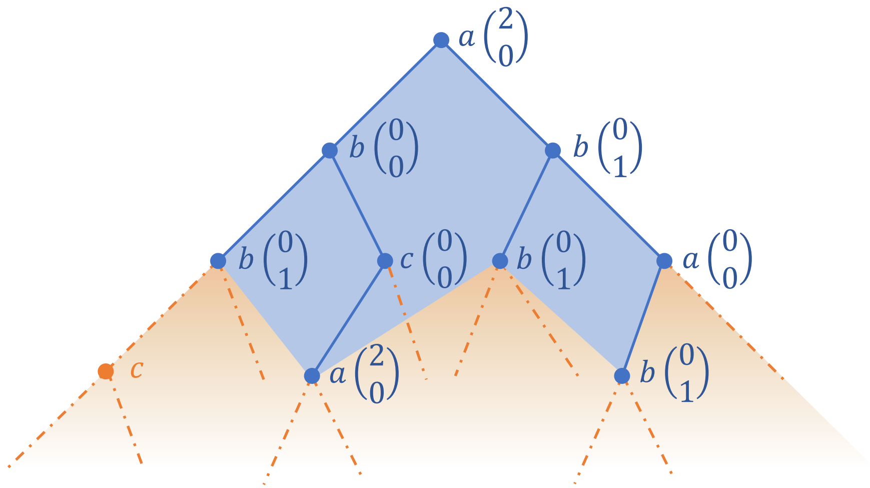
PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$

- $Q = Q_P \cup Q_\omega$ finite set of states
- $\Xi = (\Sigma \times D) \cup \Sigma$ alphabet
- $\Delta = \Delta_P \cup \Delta_\omega$ transitions

Δ_P: $p \to a \binom{2}{0} \langle p_1, p_2 \rangle$ reading the initial “counting“ segment

$p \to b \binom{0}{1} \langle q_1, p_1 \rangle$

Δ_ω: $q \to c \ \langle q_1, q_2 \rangle$ reading the remaining tree



tree $\zeta \in \mathrm{T}_\Xi^\omega$

Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$

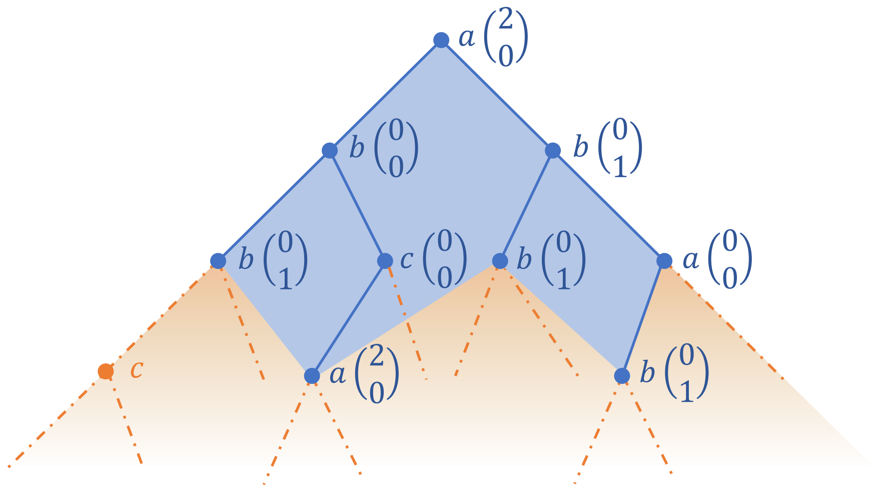
- $Q = Q_P \cup Q_\omega$ finite set of states
- $\Xi = (\Sigma \times D) \cup \Sigma$ alphabet
- $\Delta = \Delta_P \cup \Delta_\omega$ transitions

Δ_P: $p \to a\binom{2}{0}\langle p_1, p_2 \rangle$ reading the initial “counting“ segment

$p \to b\binom{0}{1}\langle q_1, p_1 \rangle$

Δ_ω: $q \to c\ \langle q_1, q_2 \rangle$ reading the remaining tree

- $\mathcal{F} \subseteq 2^{Q_\omega}$ final state sets
- $C \subseteq \mathbb{N}^s$ semilinear set

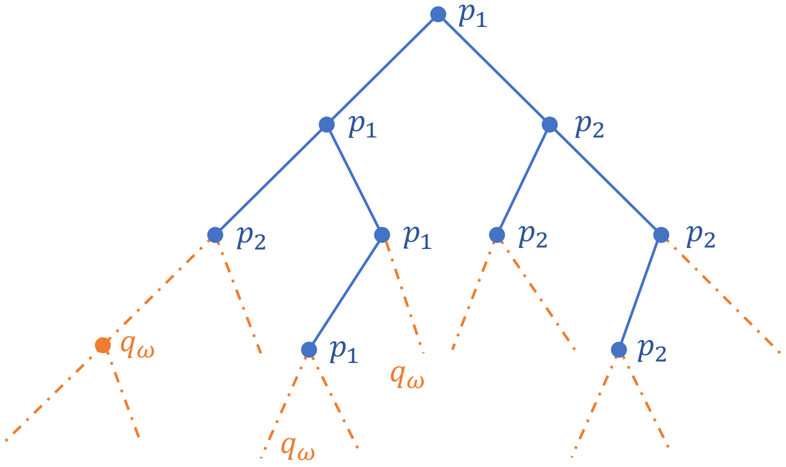


tree $\zeta \in \mathrm{T}_\Xi^\omega$

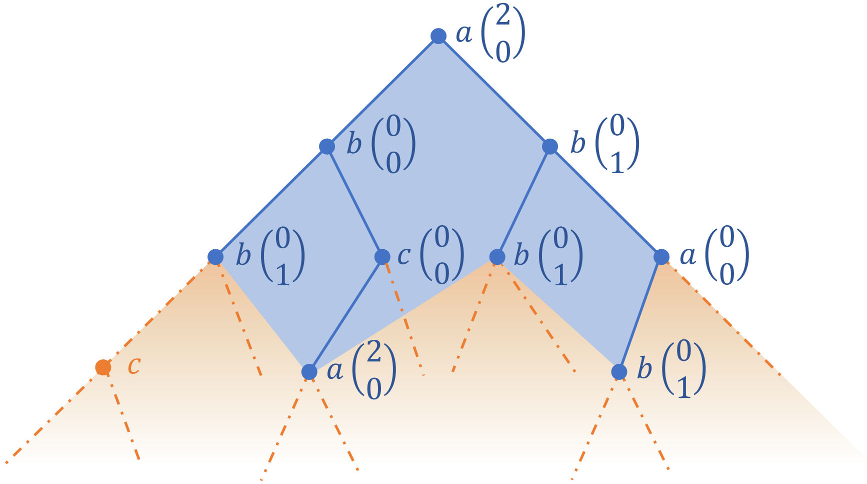
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$



run κ of ζ

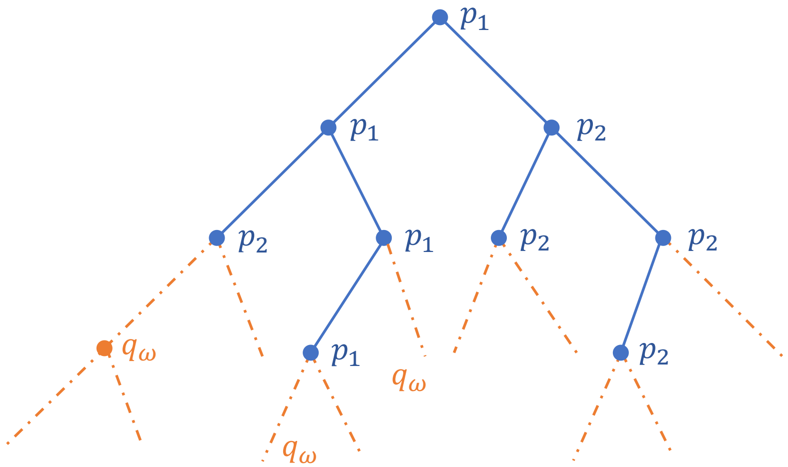


tree $\zeta \in \mathrm{T}_\Xi^\omega$

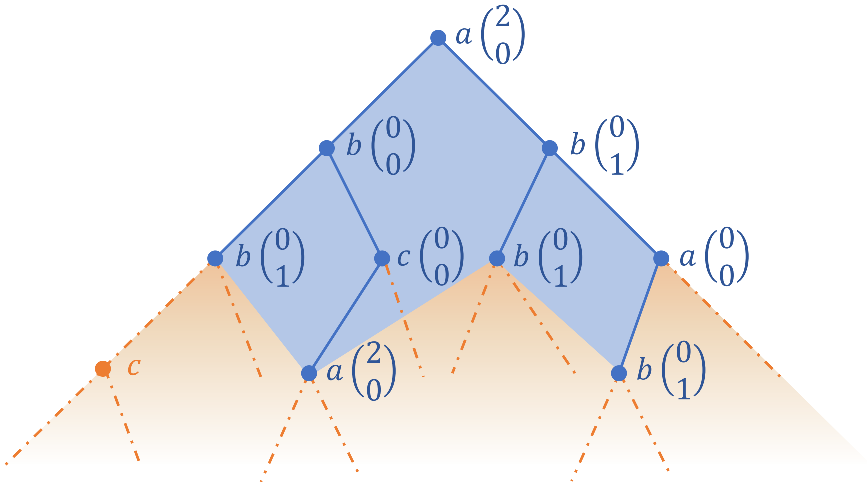
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$



run κ of ζ



tree $\zeta \in \mathrm{T}_\Xi^\omega$

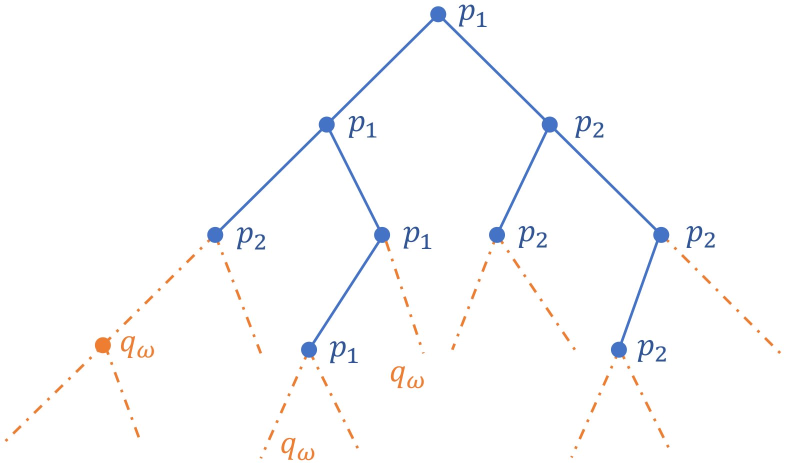
run κ of ζ is successful if

- Muller acceptance condition holds

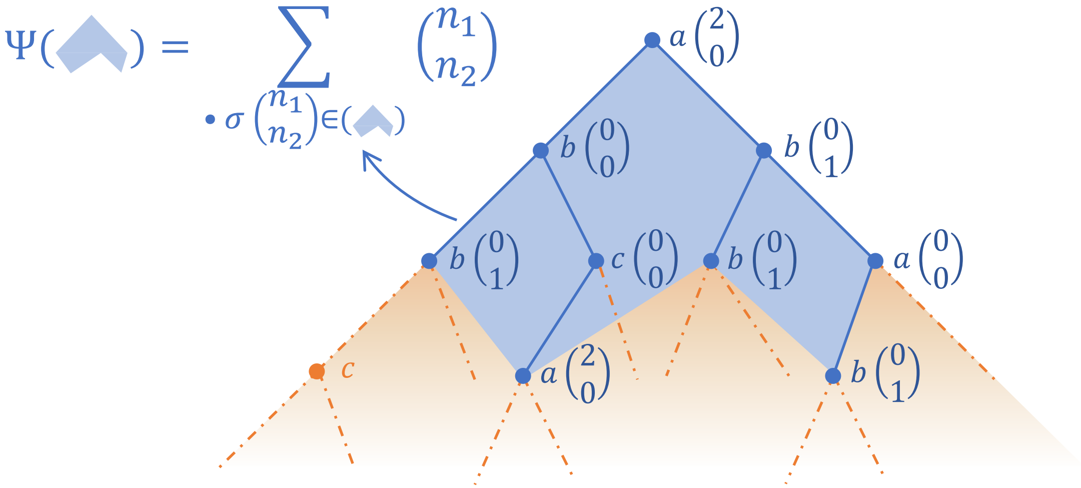
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$

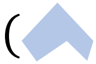


run κ of ζ



tree $\zeta \in \mathrm{T}_\Xi^\omega$

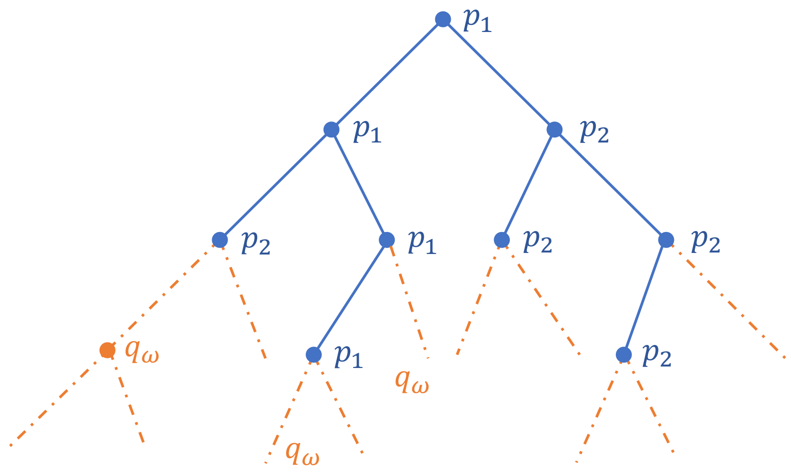
run κ of ζ is successful if

- Muller acceptance condition holds
- Parikh condition: $\Psi($$) \in C$ holds

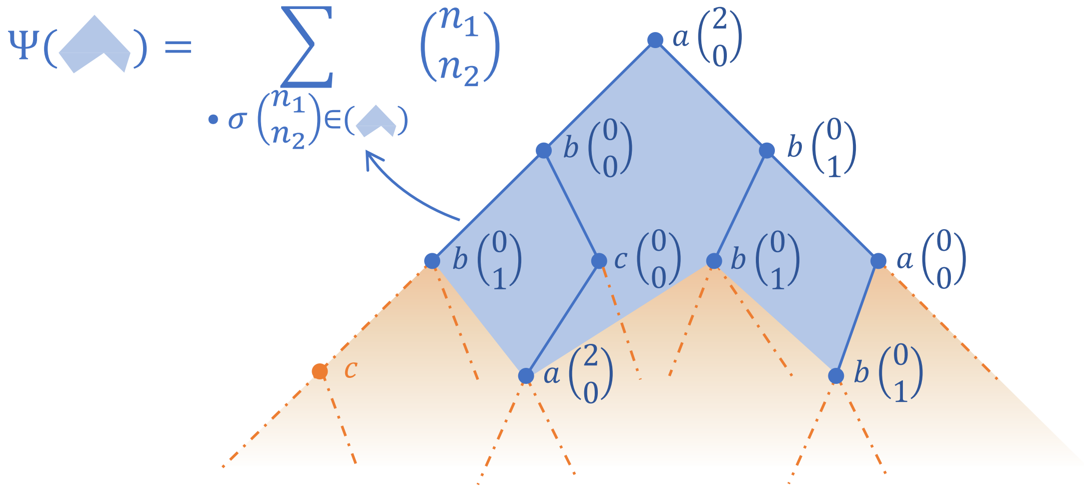
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$



run κ of ζ



tree $\zeta \in T_\Xi^\omega$

run κ of ζ is successful if

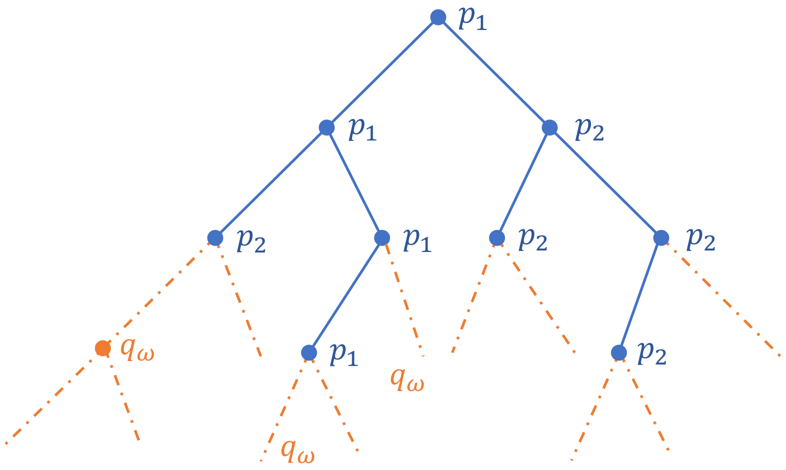
- Muller acceptance condition holds
- Parikh condition: $\Psi($▲$) \in C$ holds

$$\mathcal{L}(\mathcal{A}) = \{\xi \in T_\Sigma \mid \exists \zeta \in T_\Xi^\omega \text{ with } (\zeta)_\Sigma = \xi \text{ and } \exists \text{ successful run } \kappa \text{ on } \zeta\}$$

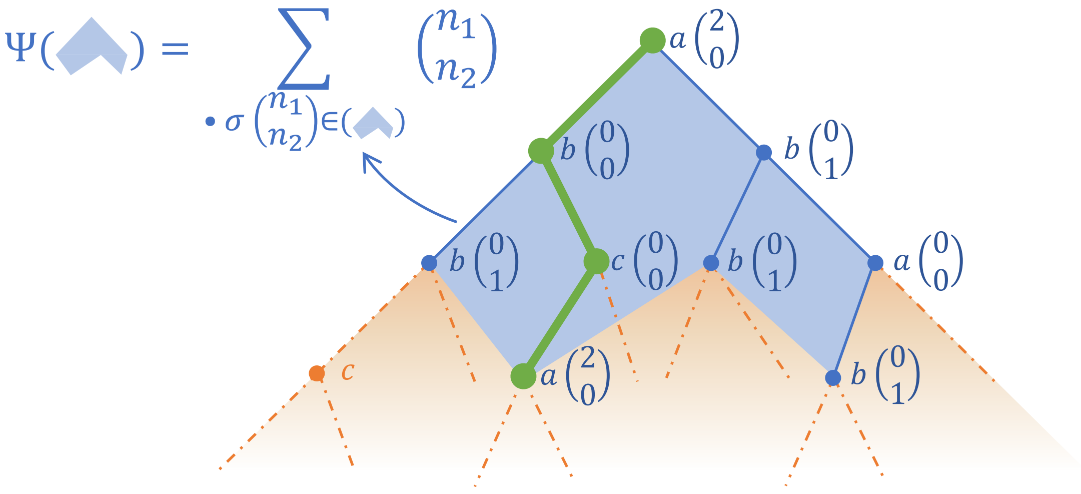
Parikh-Muller Tree Automata

dimension $s \in \mathbb{N}$

counter increments $D \subseteq_{\text{fin}} \mathbb{N}^s$



run κ of ζ



tree $\zeta \in T_\Xi^\omega$

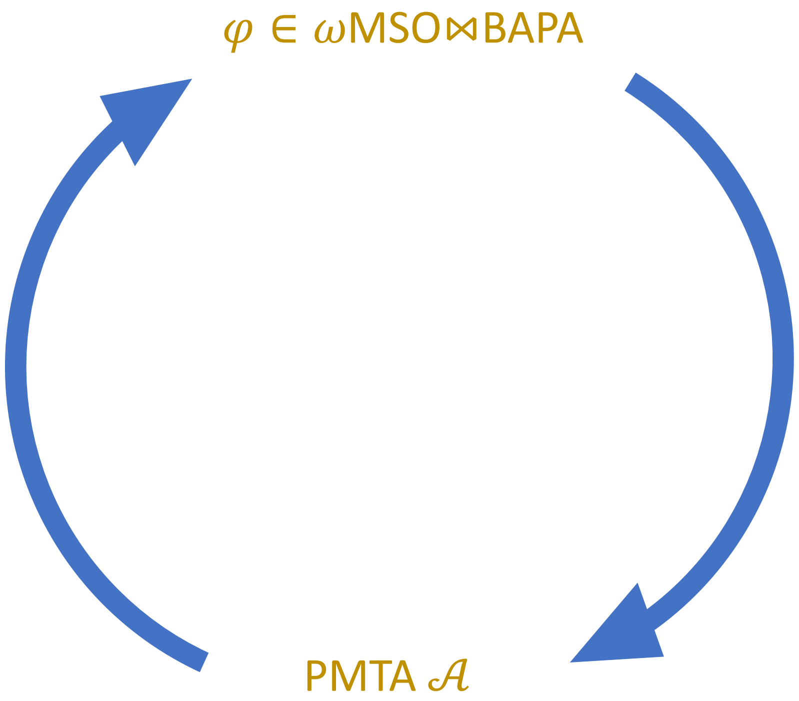
run κ of ζ is successful if

“There exists a path P s.t. 2x the number of as on P $\leq_{\text{fin}}$ the number of bs not on P”

- Muller acceptance condition holds
- Parikh condition: $\Psi(▲) \in C$ holds

$$\mathcal{L}(\mathcal{A}) = \{\xi \in T_\Sigma \mid \exists \zeta \in T_\Xi^\omega \text{ with } (\zeta)_\Sigma = \xi \text{ and } \exists \text{ successful run } \kappa \text{ on } \zeta\}$$

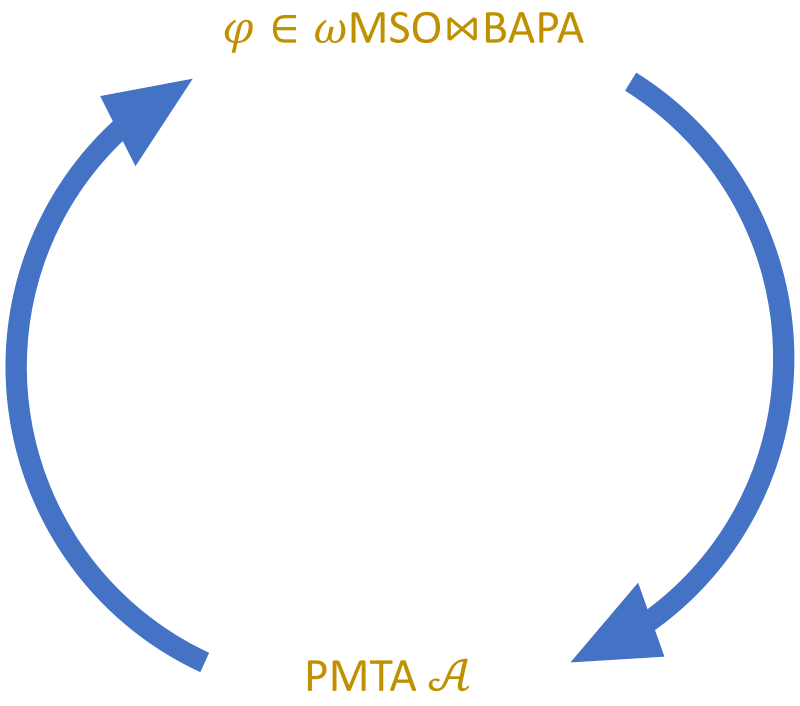
Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

PMTA $\mathcal{A}$

Correspondence of $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$ and PMTA



$\varphi \in \omega\mathrm{MSO}\bowtie\mathrm{BAPA}$

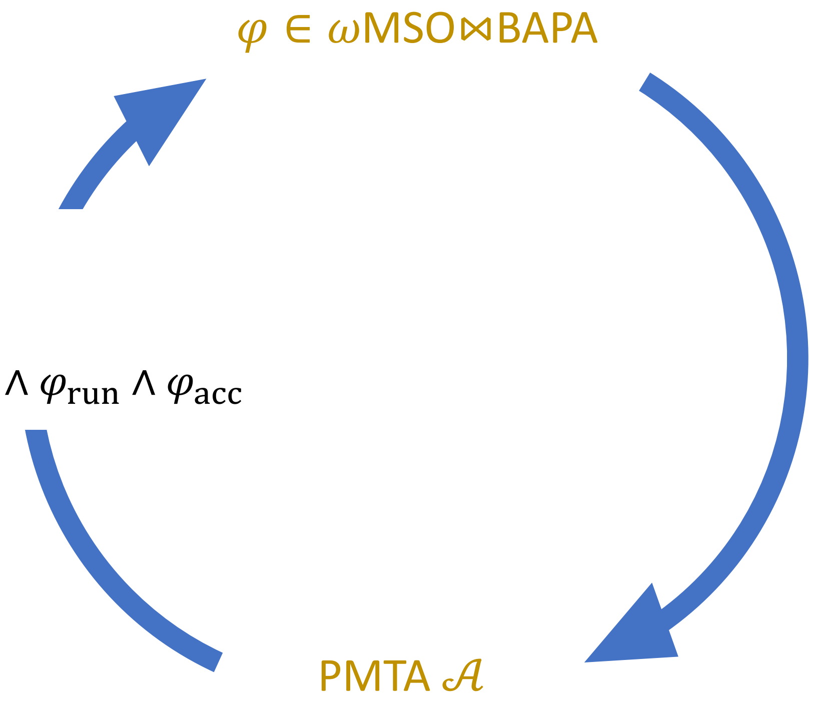
PMTA $\mathcal{A}$

Induction on structure of

$$\varphi = \exists X_1 \dots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

PMTA $\mathcal{A}$

Construction of formula φ

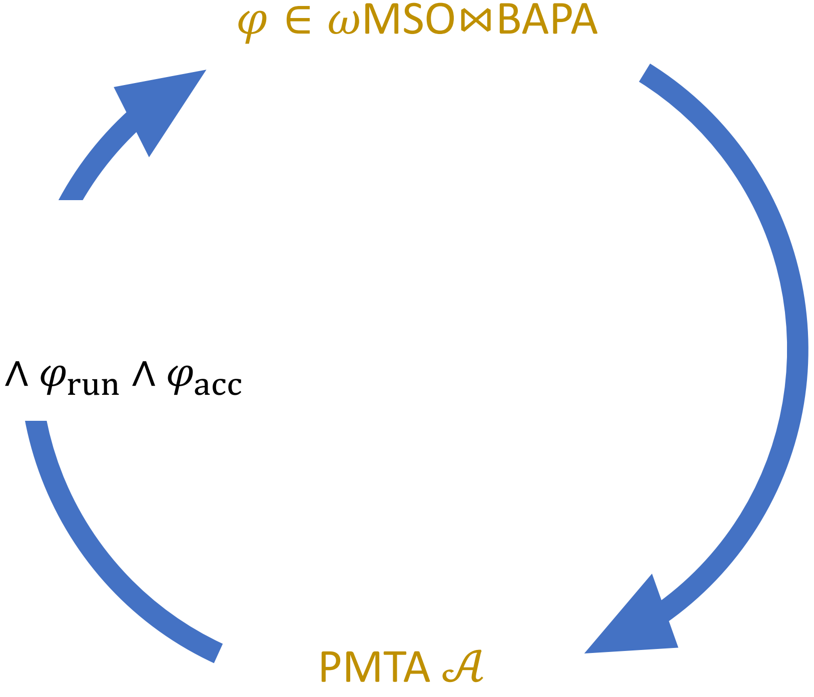
$$\exists X_1 \ldots X_{|Q|} \forall P.\, \varphi_{\text{part}} \wedge \varphi_{\text{run}} \wedge \varphi_{\text{acc}}$$

Induction on structure of

$$\varphi = \exists X_1 \ldots X_n.\, \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

PMTA $\mathcal{A}$

Construction of formula φ

$\exists Z_1^0 \dots Z_1^K \dots Z_S^0 \dots Z_S^K.\, \exists X_1 \dots X_{|Q|} \forall P.\, \varphi_{\text{part}} \land \varphi_{\text{run}} \land \varphi_{\text{acc}}$

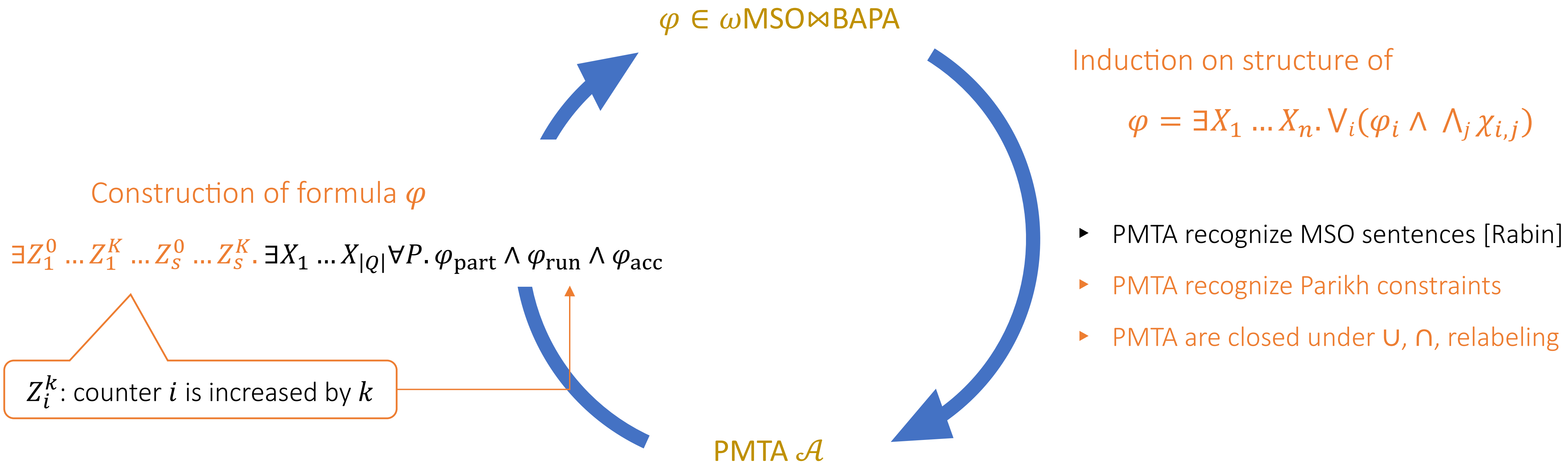
Z_i^k: counter i is increased by k

Induction on structure of

$$\varphi = \exists X_1 \dots X_n.\, \bigvee_i (\varphi_i \land \bigwedge_j \chi_{i,j})$$

- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

Induction on structure of

$$\varphi = \exists X_1 \dots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

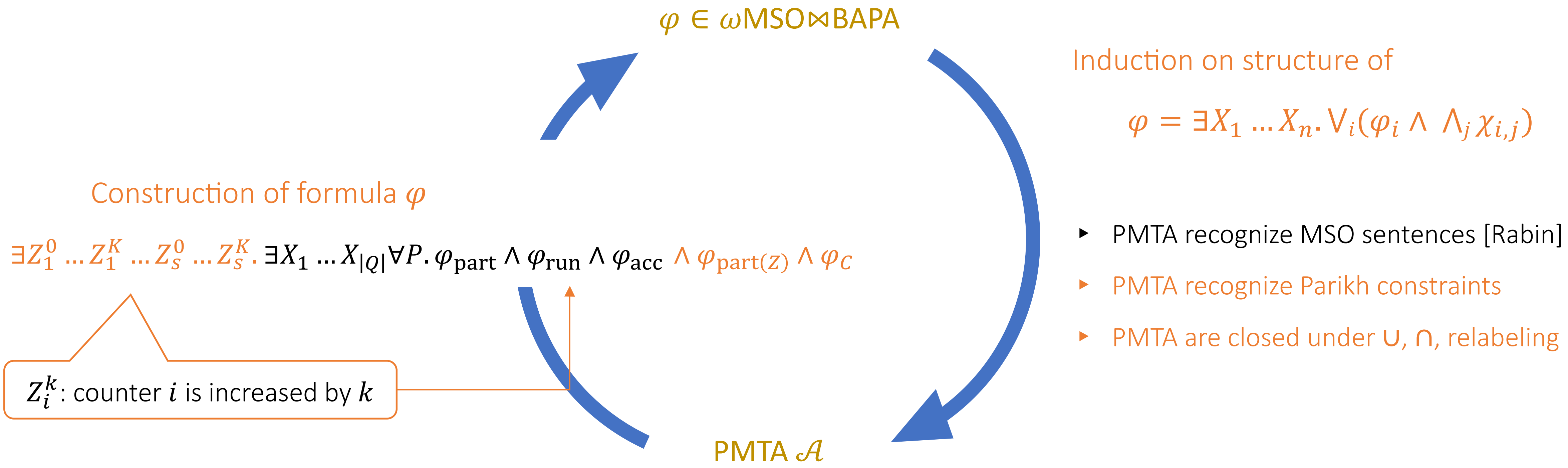
PMTA $\mathcal{A}$

Construction of formula φ

$$\exists Z_1^0 \dots Z_1^K \dots Z_s^0 \dots Z_s^K . \exists X_1 \dots X_{|Q|} \forall P . \varphi_{\text{part}} \wedge \varphi_{\text{run}} \wedge \varphi_{\text{acc}}$$

Z_i^k: counter i is increased by k

Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

Induction on structure of

$$\varphi = \exists X_1 \dots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

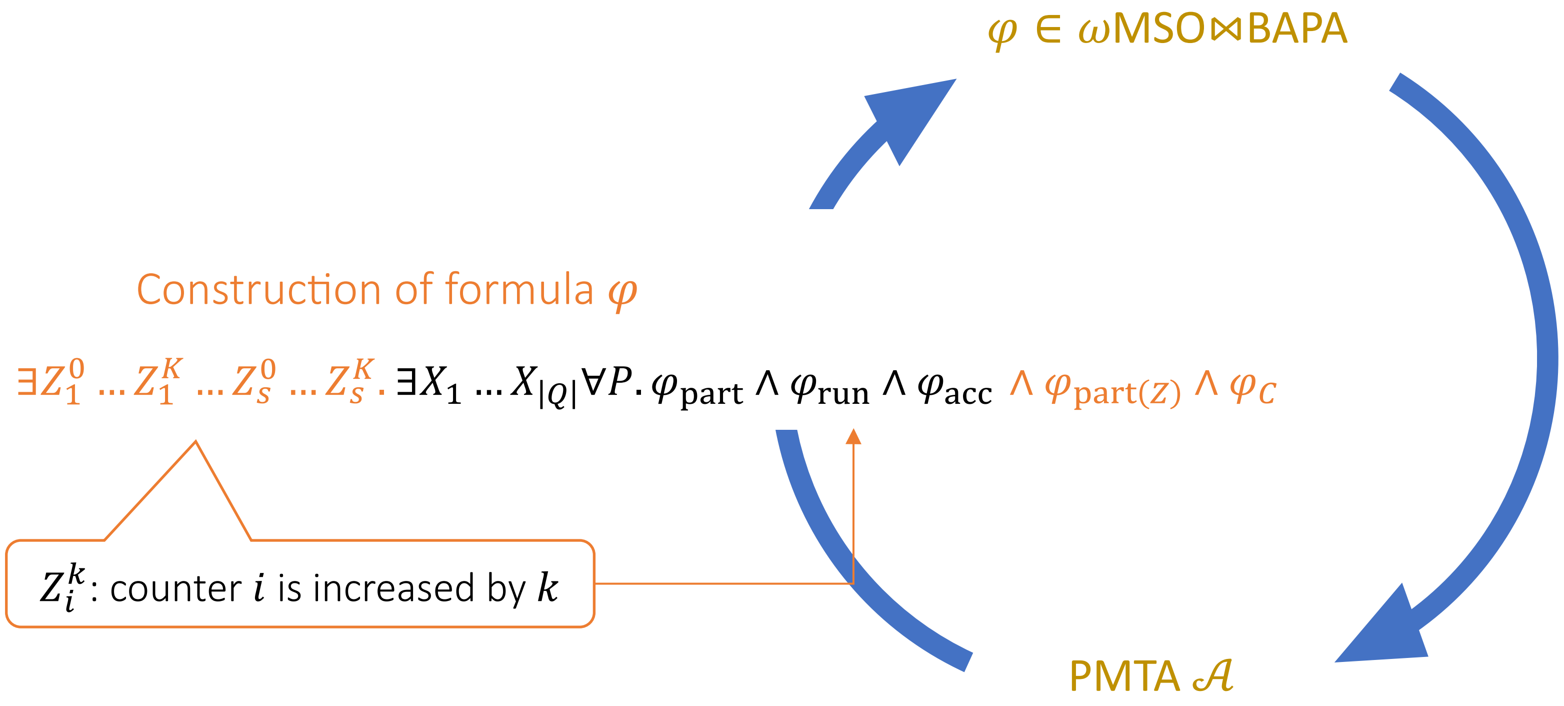
Construction of formula φ

$$\exists Z_1^0 \dots Z_1^K \dots Z_S^0 \dots Z_S^K . \exists X_1 \dots X_{|Q|} \forall P . \varphi_{\text{part}} \wedge \varphi_{\text{run}} \wedge \varphi_{\text{acc}} \wedge \varphi_{\text{part}(Z)} \wedge \varphi_C$$

Z_i^k: counter i is increased by k

PMTA $\mathcal{A}$

Correspondence of ωMSO⋈BAPA and PMTA



$\varphi \in \omega$MSO⋈BAPA

Construction of formula φ

$\exists Z_1^0 \dots Z_1^K \dots Z_S^0 \dots Z_S^K . \exists X_1 \dots X_{|Q|} \forall P. \varphi_{\mathrm{part}} \wedge \varphi_{\mathrm{run}} \wedge \varphi_{\mathrm{acc}} \wedge \varphi_{\mathrm{part}(Z)} \wedge \varphi_C$

Z_i^k: counter i is increased by k

PMTA $\mathcal{A}$

Induction on structure of

$$\varphi = \exists X_1 \dots X_n . \bigvee_i (\varphi_i \wedge \bigwedge_j \chi_{i,j})$$

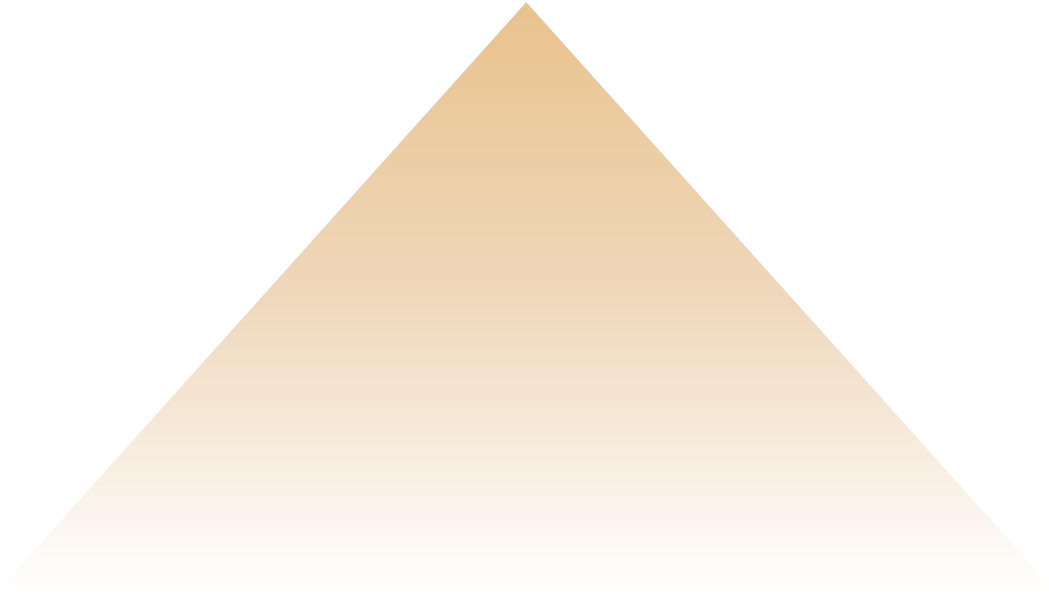
- PMTA recognize MSO sentences [Rabin]
- PMTA recognize Parikh constraints
- PMTA are closed under ∪, ∩, relabeling

Theorem. ωMSO⋈BAPA = PMTA (on infinite labeled trees)

Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

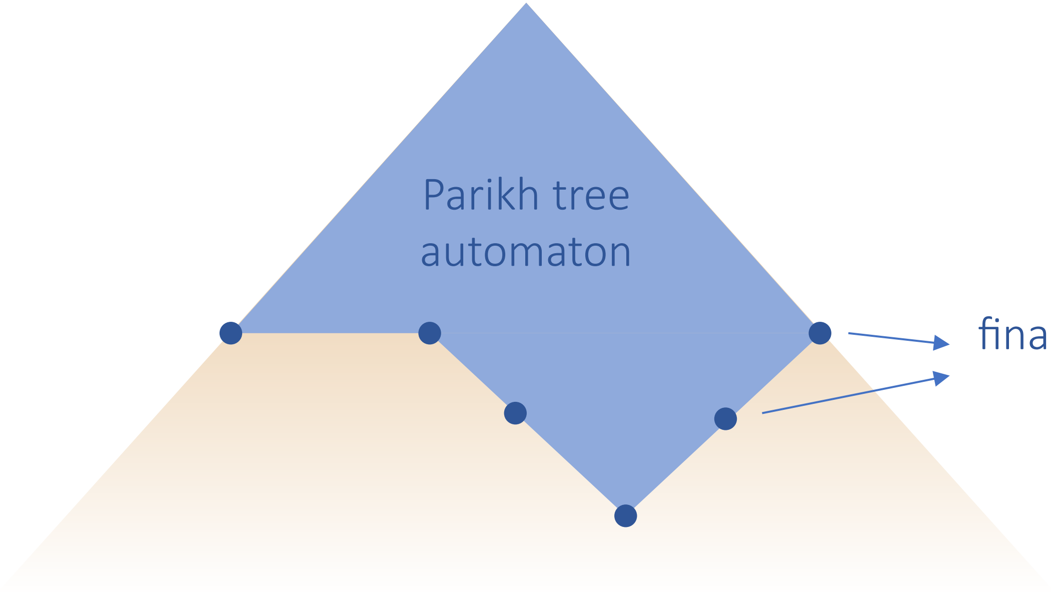
Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.



Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

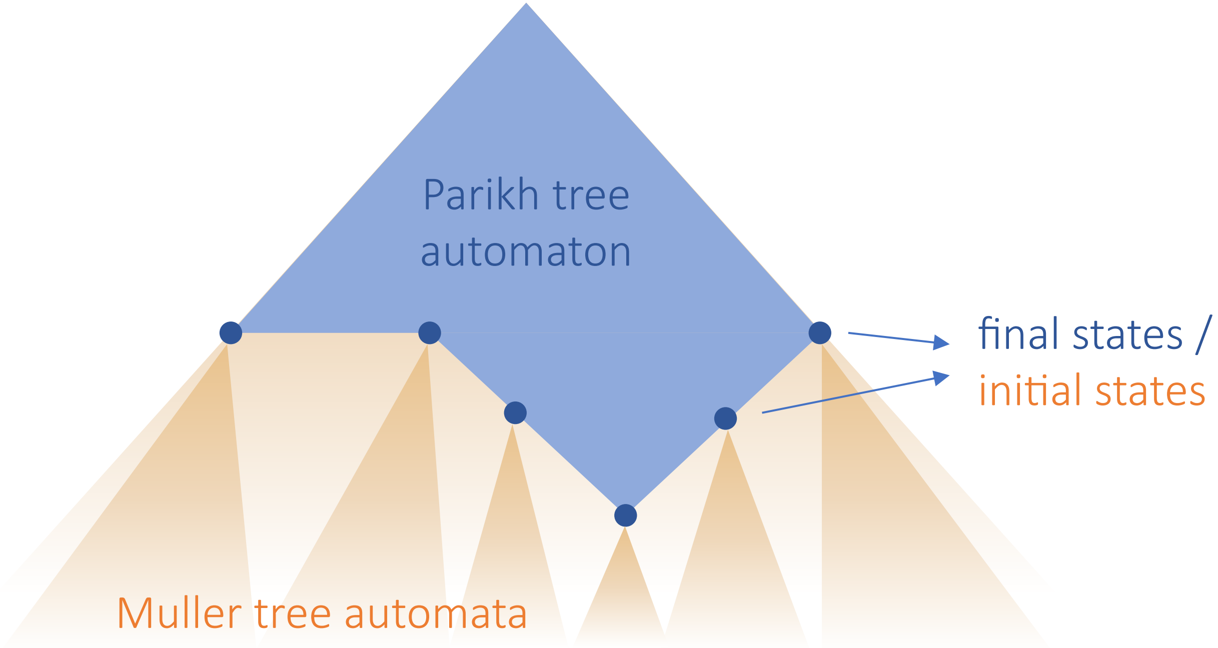


final states /

Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

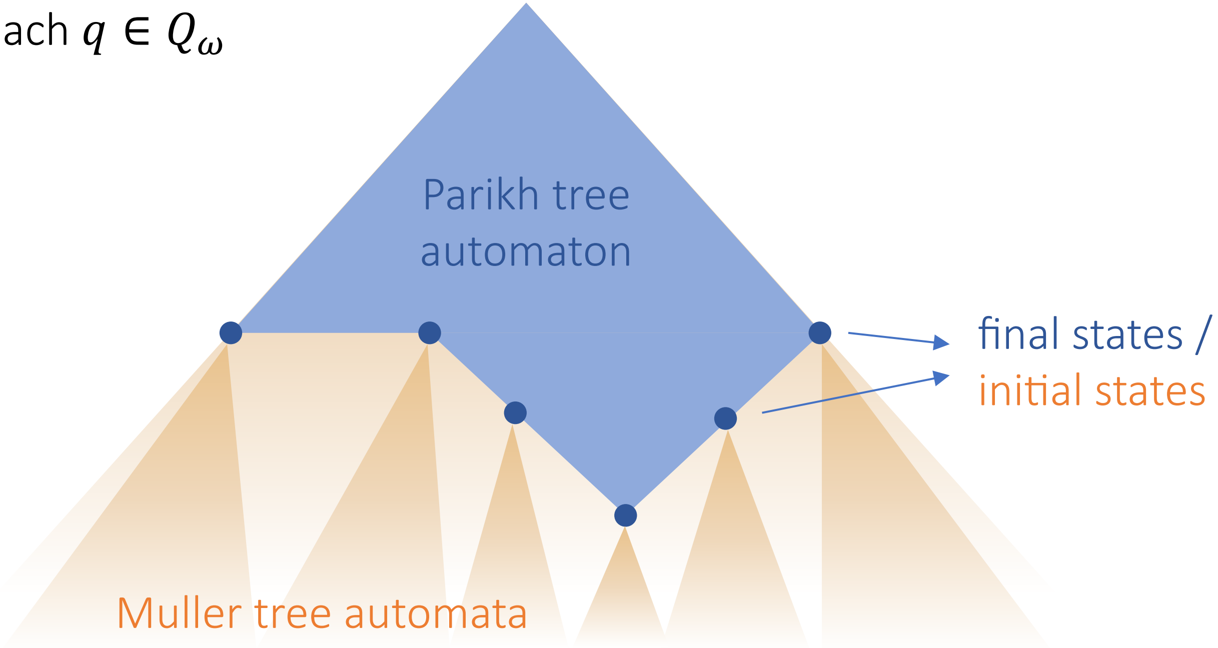


Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

- $\mathcal{A}_q = (Q_\omega, q, \Delta_\omega, \mathcal{F})$ Muller tree automaton for each $q \in Q_\omega$

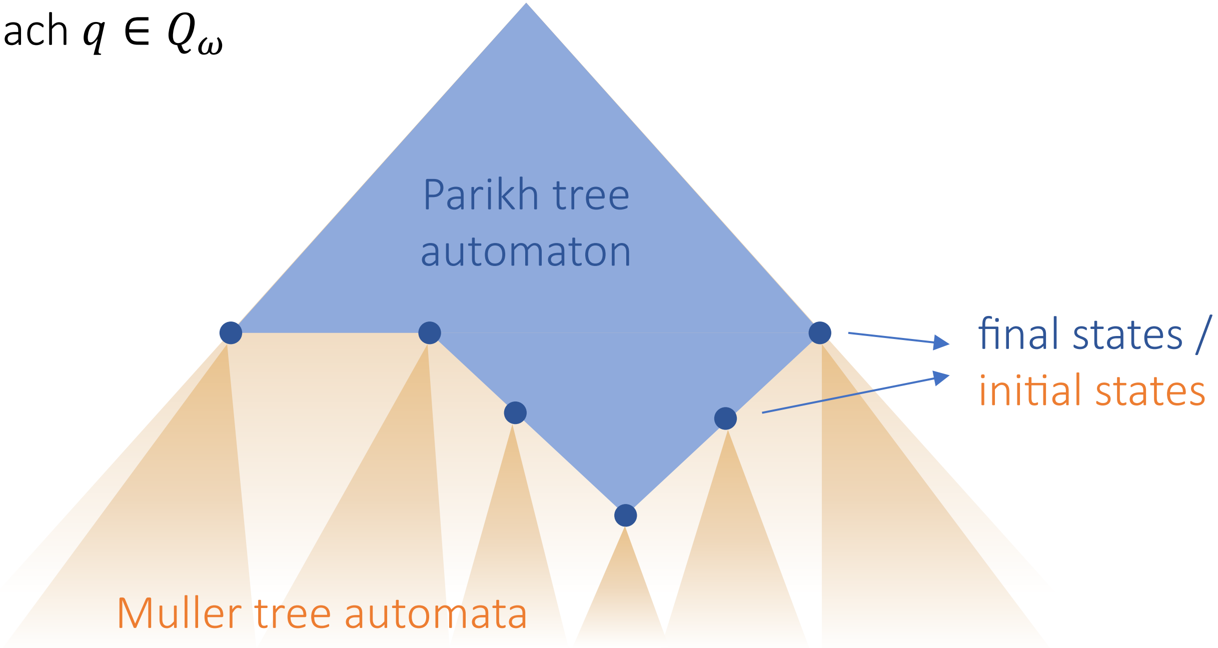


Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

- $\mathcal{A}_q = (Q_\omega, q, \Delta_\omega, \mathcal{F})$ Muller tree automaton for each $q \in Q_\omega$
- $F_P = \{q \in Q_\omega \mid \mathcal{L}(\mathcal{A}_q) \neq \emptyset\}$



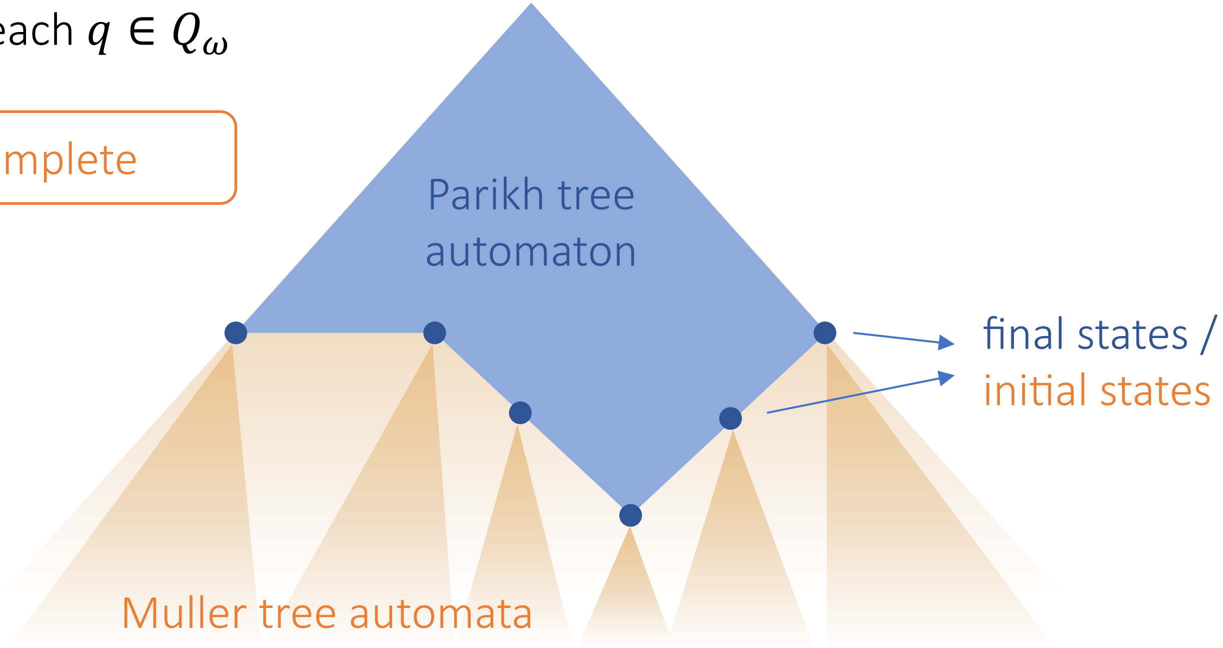
Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

- $\mathcal{A}_q = (Q_\omega, q, \Delta_\omega, \mathcal{F})$ Muller tree automaton for each $q \in Q_\omega$
- $F_P = \{q \in Q_\omega \mid \mathcal{L}(\mathcal{A}_q) \neq \emptyset\}$

PSPACE-complete



Emptiness of PMTA

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Proof. Given PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$.

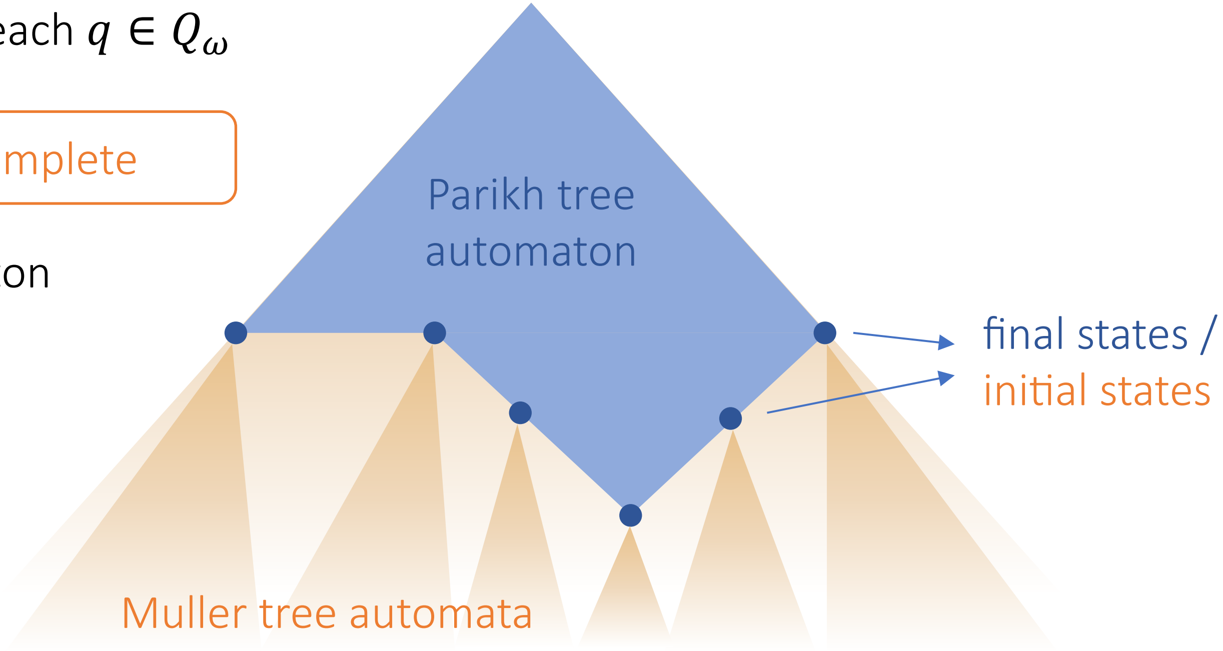
- $\mathcal{A}_q = (Q_\omega, q, \Delta_\omega, \mathcal{F})$ Muller tree automaton for each $q \in Q_\omega$
- $F_P = \{q \in Q_\omega \mid \mathcal{L}(\mathcal{A}_q) \neq \emptyset\}$

PSPACE-complete

- $\mathcal{A}_P = (Q, \Sigma{\times}D, q_I, \Delta_P, F_p, C)$ Parikh tree automaton

$\mathcal{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}_P) \neq \emptyset$

NP-complete



Satisfiability of ωMSO⋈BAPA

Theorem. ωMSO⋈BAPA = PMTA (on infinite labeled trees)

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Satisfiability of ωMSO⋈BAPA

Theorem. ωMSO⋈BAPA = PMTA (on infinite labeled trees)

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Corollary. Satisfiability of ωMSO⋈BAPA on infinite labeled trees is decidable.

Satisfiability of ωMSO$\bowtie$BAPA

Theorem. ωMSO$\bowtie$BAPA = PMTA (on infinite labeled trees)

Theorem. Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.

Corollary. Satisfiability of ωMSO$\bowtie$BAPA on infinite labeled trees is decidable.

can be lifted with MSO-interpretations
to all tree-interpretable classes

Theorem. Satisfiability of ωMSO$\bowtie$BAPA is decidable over the classes of finite or countable $\mathbb{S}$-structures of bounded treewidth, cliquewidth, and partitionswidth.

Summary

- highly expressive logic ωMSO·BAPA for cardinality relationships → undecidable in general
- fragment ωMSO⋈BAPA: still expressive and admits normal form
- Parikh-Muller tree automata correspond to ωMSO⋈BAPA on infinite trees
- … and have a decidable emptiness problem
- satisfiability of ωMSO⋈BAPA on infinite trees and tree-interpretable classes is decidable
- decidability showcases: coupling with $\mathrm{FO}^2_{\mathrm{Pres}}$, µ-calculus with global Presburger constraints

Summary

- highly expressive logic ωMSO·BAPA for cardinality relationships → undecidable in general
- fragment ωMSO⋈BAPA: still expressive and admits normal form
- Parikh-Muller tree automata correspond to ωMSO⋈BAPA on infinite trees
- … and have a decidable emptiness problem
- satisfiability of ωMSO⋈BAPA on infinite trees and tree-interpretable classes
- decidability showcases: coupling with $\mathrm{FO}^2_{\mathrm{Pres}}$, μ-calculus with global Presburger constraints

Thank you!