# Cube Terms and the Subpower Membership Problem

**Ágnes Szendrei**

CU Boulder/U Szeged
Research supported by NSF grant DMS 1500254 and OTKA grants K104251, K115518

**AAA94 + NSAC 2017**

Novi Sad, June 15–18, 2017

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

## Problem

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

## Problem

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

SMP($\mathbf{A}$):
INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}^m$
QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle = \mathbf{R}$ ?

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

## Problem

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

SMP($\mathbf{A}$):
INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}^m$
QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle = \mathbf{R}$ ?

- Hard in general:
  - 'naive algorithm' is in EXPTIME;

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

**Problem**

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

SMP($\mathbf{A}$):
INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}^m$
QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle = \mathbf{R}$ ?

- Hard in general:
  - 'naive algorithm' is in EXPTIME;
  - $\exists\, \mathbf{K}$ such that SMP($\mathbf{K}$) is EXPTIME-complete [Kozik, 2008].

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

## Problem

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

SMP($\mathbf{A}$):
INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}^m$
QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle = \mathbf{R}$ ?

- Hard in general:
  - 'naive algorithm' is in EXPTIME;
  - $\exists\, \mathbf{K}$ such that SMP($\mathbf{K}$) is EXPTIME-complete [Kozik, 2008].
- Easy (polynomial time algorithm) for
  - vector spaces, groups;
    more generally, groups with multilinear operations [Willard, 2007];

# The Subpower Membership Problem

Let $\mathbf{A}$ be a finite algebra (in a finite language).
Let $\mathbf{R}$ be an $m$-ary compatible relation of $\mathbf{A}$ (i.e., $\mathbf{R} \leq \mathbf{A}^m$).

**Problem**

*If $\mathbf{R}$ is given by a generating set and $b \in \mathbf{A}^m$, how hard is it to check if $b \in \mathbf{R}$?*

SMP($\mathbf{A}$):
INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}^m$
QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle = \mathbf{R}$ ?

- Hard in general:
  - 'naive algorithm' is in EXPTIME;
  - $\exists\, \mathbf{K}$ such that SMP($\mathbf{K}$) is EXPTIME-complete [Kozik, 2008].
- Easy (polynomial time algorithm) for
  - vector spaces, groups;
    more generally, groups with multilinear operations [Willard, 2007];
  - lattices; more generally, algebras with NU terms [Baker–Pixley, 1975].

# Classification

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

## Question

What is the relationship between the complexity of $\mathrm{SMP}(\mathbf{A})$ and the 'strength' of (term) operations of $\mathbf{A}$?

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

## Question

What is the relationship between the complexity of $\mathrm{SMP}(\mathbf{A})$ and the 'strength' of (term) operations of $\mathbf{A}$?

'strength' of op's of $\mathbf{A}$ $\quad \rightsquigarrow \quad$ Maltsev conditions satisfied by $\mathcal{V}(\mathbf{A})$ (or $\mathbf{A}$)

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

## Question

What is the relationship between the complexity of $\mathrm{SMP}(\mathbf{A})$ and the 'strength' of (term) operations of $\mathbf{A}$?

'strength' of op's of $\mathbf{A}$ $\quad \leftrightsquigarrow \quad$ Maltsev conditions satisfied by $\mathcal{V}(\mathbf{A})$ (or $\mathbf{A}$)

- A variety $\mathcal{V}$ *satisfies a strong Maltsev condition* $(\mathcal{H}, \Sigma)$ ($\mathcal{H}$, $\Sigma$ finite) if every symbol $h \in \mathcal{H}$ can be interpreted by a term of $\mathcal{V}$ such that $\mathcal{V} \models \Sigma$.

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

## Question

What is the relationship between the complexity of $\mathrm{SMP}(\mathbf{A})$ and the 'strength' of (term) operations of $\mathbf{A}$?

'strength' of op's of $\mathbf{A}$ $\quad \leftrightsquigarrow \quad$ Maltsev conditions satisfied by $\mathcal{V}(\mathbf{A})$ (or $\mathbf{A}$)

- A variety $\mathcal{V}$ *satisfies a strong Maltsev condition* $(\mathcal{H}, \Sigma)$ ($\mathcal{H}$, $\Sigma$ finite) if every symbol $h \in \mathcal{H}$ can be interpreted by a term of $\mathcal{V}$ such that $\mathcal{V} \models \Sigma$.
- **Example 1.** $\mathcal{V}$ is a variety of groups with additional operations iff $\mathcal{V}$ satisfies the Maltsev condition $(\{\cdot\}, \{\text{group laws}\})$.

# Classification

Recall: for any finite algebra $\mathbf{A}$,

$$\{\text{compatible relations of } \mathbf{A}\} \quad \longleftrightarrow \quad \mathrm{Clo}(\mathbf{A}) \quad \longrightarrow \quad \mathcal{V}(\mathbf{A}).$$

## Question

What is the relationship between the complexity of $\mathrm{SMP}(\mathbf{A})$ and the 'strength' of (term) operations of $\mathbf{A}$?

'strength' of op's of $\mathbf{A}$ $\quad \leftrightsquigarrow \quad$ Maltsev conditions satisfied by $\mathcal{V}(\mathbf{A})$ (or $\mathbf{A}$)

- A variety $\mathcal{V}$ *satisfies a strong Maltsev condition* $(\mathcal{H}, \Sigma)$ ($\mathcal{H}$, $\Sigma$ finite) if every symbol $h \in \mathcal{H}$ can be interpreted by a term of $\mathcal{V}$ such that $\mathcal{V} \models \Sigma$.
- **Example 1.** $\mathcal{V}$ is a variety of groups with additional operations iff $\mathcal{V}$ satisfies the Maltsev condition $(\{\cdot\}, \{\text{group laws}\})$.
- **Example 2.** $\mathcal{V}$ has a *symmetric binary term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{\circ\}, \{x \circ y \approx y \circ x\})$.

# Classification (Cont'd)

- **Example 3.** $\mathcal{V}$ has a *Maltsev term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{p\}, \{p(x,x,y) \approx y \approx p(y,x,x)\})$.

# Classification (Cont'd)

- **Example 3.** $\mathcal{V}$ has a *Maltsev term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{p\}, \{p(x,x,y) \approx y \approx p(y,x,x)\})$.
- **Example 4.** $\mathcal{V}$ has a *majority term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{t\}, \{t(x,y,y) \approx t(y,x,y) \approx t(y,y,x) \approx y\})$.

# Classification (Cont'd)

- **Example 3.** $\mathcal{V}$ has a *Maltsev term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{p\}, \{p(x,x,y) \approx y \approx p(y,x,x)\})$.
- **Example 4.** $\mathcal{V}$ has a *majority term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{t\}, \{t(x,y,y) \approx t(y,x,y) \approx t(y,y,x) \approx y\})$.
- **Example 5.** A *d-cube term* for $\mathcal{V}$ is a term $C$ witnessing that $\mathcal{V}$ satisfies the Maltsev condition $(\{C\}, \Sigma)$ where

$$\Sigma = \left\{ \text{rows of } C\left( \underbrace{\begin{bmatrix} x \\ y \\ \vdots \\ y \end{bmatrix}, \begin{bmatrix} y \\ x \\ \vdots \\ y \end{bmatrix}, \ldots, \begin{bmatrix} y \\ y \\ \vdots \\ x \end{bmatrix}, \begin{bmatrix} x \\ x \\ \vdots \\ y \end{bmatrix}, \ldots}_{d\text{-tuples in } x, y, \text{ with at least one } x} \right) \approx \begin{bmatrix} y \\ y \\ \vdots \\ y \end{bmatrix} \right\}.$$

# Classification (Cont'd)

- **Example 3.** $\mathcal{V}$ has a *Maltsev term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{p\}, \{p(x,x,y) \approx y \approx p(y,x,x)\})$.
- **Example 4.** $\mathcal{V}$ has a *majority term* iff $\mathcal{V}$ satisfies the Maltsev condition $(\{t\}, \{t(x,y,y) \approx t(y,x,y) \approx t(y,y,x) \approx y\})$.
- **Example 5.** A *d-cube term* for $\mathcal{V}$ is a term $C$ witnessing that $\mathcal{V}$ satisfies the Maltsev condition $(\{C\}, \Sigma)$ where

$$\Sigma = \left\{ \text{rows of } C\left( \underbrace{\begin{bmatrix} x \\ y \\ \vdots \\ y \end{bmatrix}, \begin{bmatrix} y \\ x \\ \vdots \\ y \end{bmatrix}, \ldots, \begin{bmatrix} y \\ y \\ \vdots \\ x \end{bmatrix}, \begin{bmatrix} x \\ x \\ \vdots \\ y \end{bmatrix}, \ldots}_{d\text{-tuples in } x, y, \text{ with at least one } x} \right) \approx \begin{bmatrix} y \\ y \\ \vdots \\ y \end{bmatrix} \right\}.$$

- Examples 2–5 are *linear* Maltsev conditions; Example 1 is not.

# Cube Terms and SMP

# Cube Terms and SMP

## Problem

*Is it true that if* $\mathbf{A}$ *is a finite algebra (in a finite language) such that* $\mathbf{A}$ *has a cube term, then* $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$ *?*

# Cube Terms and SMP

## Problem

*Is it true that if* $\mathbf{A}$ *is a finite algebra (in a finite language) such that* $\mathbf{A}$ *has a cube term, then* $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$ *?*

## Theorem 1 (Jeff Shriner)

*If* $\mathcal{M} = (\mathcal{H}, \Sigma)$ *is a consistent, strong linear Maltsev condition such that* $\Sigma$ *does not entail cube identities for any* $h \in \mathcal{H}$*, then there exists a finite algebra* $\mathbf{A}$ *(in a finite language) such that* $\mathbf{A}$ *satisfies* $\mathcal{M}$ *and* $\mathrm{SMP}(\mathbf{A})$ *is* EXPTIME*-complete.*

# Cube Terms and SMP

## Problem

*Is it true that if* $\mathbf{A}$ *is a finite algebra (in a finite language) such that* $\mathbf{A}$ *has a cube term, then* $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$ *?*

## Theorem 1 (Jeff Shriner)

*If* $\mathcal{M} = (\mathcal{H}, \Sigma)$ *is a consistent, strong linear Maltsev condition such that* $\Sigma$ *does not entail cube identities for any* $h \in \mathcal{H}$*, then there exists a finite algebra* $\mathbf{A}$ *(in a finite language) such that* $\mathbf{A}$ *satisfies* $\mathcal{M}$ *and* $\mathrm{SMP}(\mathbf{A})$ *is* EXPTIME*-complete.*

## Theorem 2 (Peter Mayr – ASz)

*If* $\mathbf{A}$ *is a finite algebra (in a finite language) such that*

$(*)$ *for every SI in* $\mathbb{HS}(\mathbf{A})$ *with abelian monolith* $\mu$*,* $(0 : \mu)$ *is supernilpotent,*

*then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$.

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

[Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

  [Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

For a finite algebra $\mathbf{A}$ with a cube term:

- We have a structure theorem for the critical subalgebras of $\mathbf{A}^m$ ($m \geq 1$). [Kearnes–ASz, 2012]

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

  [Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

For a finite algebra $\mathbf{A}$ with a cube term:

- We have a structure theorem for the critical subalgebras of $\mathbf{A}^m$ ($m \geq 1$). [Kearnes–ASz, 2012]
- $\mathrm{SMP}(\mathbf{A}) \in \mathsf{NP}$. [Bulatov–Mayr–ASz, 201?]

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

  [Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

For a finite algebra $\mathbf{A}$ with a cube term:

- We have a structure theorem for the critical subalgebras of $\mathbf{A}^m$ ($m \geq 1$). [Kearnes–ASz, 2012]
- $\mathrm{SMP}(\mathbf{A}) \in \mathsf{NP}$. [Bulatov–Mayr–ASz, 201?]
  - Recall: For Kozik's algebra $\mathbf{K}$, $\mathrm{SMP}(\mathbf{K})$ is EXPTIME-complete.

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

  [Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

For a finite algebra $\mathbf{A}$ with a cube term:

- We have a structure theorem for the critical subalgebras of $\mathbf{A}^m$ ($m \geq 1$). [Kearnes–ASz, 2012]
- $\mathrm{SMP}(\mathbf{A}) \in \mathsf{NP}$. [Bulatov–Mayr–ASz, 201?]
  - Recall: For Kozik's algebra $\mathbf{K}$, $\mathrm{SMP}(\mathbf{K})$ is EXPTIME-complete.
- If $\theta \in \mathrm{Con}(\mathbf{A})$, then $\mathrm{SMP}(\mathbf{A}/\theta)$ is poly-time reducible to $\mathrm{SMP}(\mathbf{A})$. [Bulatov–Mayr–ASz, 201?]

# Existence of a Cube Term

- The following conditions on a finite algebra $\mathbf{A}$ are equivalent:
  - for some polynomial $p$, $|\mathrm{Sub}(\mathbf{A}^m)| \leq 2^{p(m)}$;
  - for some polynomial $p$, every $\mathbf{R} \leq \mathbf{A}^m$ has a generating set of size $\leq p(m)$;
  - $\mathbf{A}$ has a cube term.

  [Berman–Idziak–Marković–McKenzie–Valeriote–Willard, 2010]

For a finite algebra $\mathbf{A}$ with a cube term:

- We have a structure theorem for the critical subalgebras of $\mathbf{A}^m$ ($m \geq 1$). [Kearnes–ASz, 2012]
- $\mathrm{SMP}(\mathbf{A}) \in \mathsf{NP}$. [Bulatov–Mayr–ASz, 201?]
  - Recall: For Kozik's algebra $\mathbf{K}$, $\mathrm{SMP}(\mathbf{K})$ is EXPTIME-complete.
- If $\theta \in \mathrm{Con}(\mathbf{A})$, then $\mathrm{SMP}(\mathbf{A}/\theta)$ is poly-time reducible to $\mathrm{SMP}(\mathbf{A})$. [Bulatov–Mayr–ASz, 201?]
  - There exists a 10-element band $\mathbf{S}$ and a 9-element quotient $\mathbf{S}/\theta$ such that $\mathrm{SMP}(\mathbf{S}) \in \mathsf{P}$ and $\mathrm{SMP}(\mathbf{S}/\theta)$ is NP-complete. [Steindl, 2017/8]

# An Application in AI

Learnability:

# An Application in AI

Learnability:

- Fix a finite set $A$.

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.
- A concept class $\Gamma$ is of little computational value unless it is *polynomially evaluable*, i.e., there exists a polynomial time algorithm which, given an encoding of a concept $R \in \Gamma$ and a tuple $b \in \bigcup_{m \geq 1} A^m$, determines if $b \in R$.

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.
- A concept class $\Gamma$ is of little computational value unless it is *polynomially evaluable*, i.e., there exists a polynomial time algorithm which, given an encoding of a concept $R \in \Gamma$ and a tuple $b \in \bigcup_{m \geq 1} A^m$, determines if $b \in R$.
- A learning algorithm

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.
- A concept class $\Gamma$ is of little computational value unless it is *polynomially evaluable*, i.e., there exists a polynomial time algorithm which, given an encoding of a concept $R \in \Gamma$ and a tuple $b \in \bigcup_{m \geq 1} A^m$, determines if $b \in R$.
- A learning algorithm called the *exact model with equivalence queries*:
  - the algorithm sends queries to an oracle; each query is a hypothetical encoding $h$ of the target concept $R \in \Gamma$ to be learned;

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.
- A concept class $\Gamma$ is of little computational value unless it is *polynomially evaluable*, i.e., there exists a polynomial time algorithm which, given an encoding of a concept $R \in \Gamma$ and a tuple $b \in \bigcup_{m \geq 1} A^m$, determines if $b \in R$.
- A learning algorithm called the *exact model with equivalence queries*:
  - the algorithm sends queries to an oracle; each query is a hypothetical encoding $h$ of the target concept $R \in \Gamma$ to be learned;
  - the oracle either confirms that $h$ encodes $R$, or it returns a counterexample from the symmetric difference of $R$ and the concept encoded by $h$.

# An Application in AI

Learnability:

- Fix a finite set $A$.
- A *concept* is a set $R \subseteq \bigcup_{m \geq 1} A^m$, along with an *encoding* of $R$. (Encoding is not necessarily unique.)
- A *concept class* is a set $\Gamma$ of concepts.
- A concept class $\Gamma$ is of little computational value unless it is *polynomially evaluable*, i.e., there exists a polynomial time algorithm which, given an encoding of a concept $R \in \Gamma$ and a tuple $b \in \bigcup_{m \geq 1} A^m$, determines if $b \in R$.
- A learning algorithm called the *exact model with equivalence queries*:
  - the algorithm sends queries to an oracle; each query is a hypothetical encoding $h$ of the target concept $R \in \Gamma$ to be learned;
  - the oracle either confirms that $h$ encodes $R$, or it returns a counterexample from the symmetric difference of $R$ and the concept encoded by $h$.
- The query on $h$ is *improper* if $h$ does not encode a concept in $\Gamma$.

# An Application in AI (Cont'd)

Concept classes from algebras:

# An Application in AI (Cont'd)

Concept classes from algebras:

- Let $\mathbf{A}$ be a finite algebra (in a finite language).

# An Application in AI (Cont'd)

Concept classes from algebras:

- Let $\mathbf{A}$ be a finite algebra (in a finite language).
- Concept class: $\Gamma = \bigcup_{m \geq 1} \mathrm{Sub}(\mathbf{A}^m)$,
  concepts $R \in \Gamma$ are encoded by (possibly special) generating sets.

# An Application in AI (Cont'd)

Concept classes from algebras:

- Let $\mathbf{A}$ be a finite algebra (in a finite language).
- Concept class: $\Gamma = \bigcup_{m \geq 1} \mathrm{Sub}(\mathbf{A}^m)$,
  concepts $R \in \Gamma$ are encoded by (possibly special) generating sets.
- If any generating set is allowed to encode concepts in $\Gamma$, then

  $$\Gamma \text{ is polynomially evaluable iff } \mathrm{SMP}(\mathbf{A}) \in \mathsf{P}.$$

# An Application in AI (Cont'd)

Concept classes from algebras:

- Let $\mathbf{A}$ be a finite algebra (in a finite language).
- Concept class: $\Gamma = \bigcup_{m \geq 1} \mathrm{Sub}(\mathbf{A}^m)$,
  concepts $R \in \Gamma$ are encoded by (possibly special) generating sets.
- If any generating set is allowed to encode concepts in $\Gamma$, then

$$\Gamma \text{ is polynomially evaluable iff } \mathrm{SMP}(\mathbf{A}) \in \mathsf{P}.$$

If $\mathbf{A}$ has a cube term, then:

- $\Gamma$ *is polynomially exactly learnable with improper equivalence queries.*
  [Idziak–Marković–McKenzie–Valeriote–Willard, 2010]
  - Generalizes [Dalmau–Jeavons, 2003] and [Bulatov–Chen–Dalmau, 2007].

# An Application in AI (Cont'd)

Concept classes from algebras:

- Let $\mathbf{A}$ be a finite algebra (in a finite language).
- Concept class: $\Gamma = \bigcup_{m \geq 1} \mathrm{Sub}(\mathbf{A}^m)$,
  concepts $R \in \Gamma$ are encoded by (possibly special) generating sets.
- If any generating set is allowed to encode concepts in $\Gamma$, then

$$\Gamma \text{ is polynomially evaluable iff } \mathrm{SMP}(\mathbf{A}) \in \mathsf{P}.$$

If $\mathbf{A}$ has a cube term, then:

- $\Gamma$ *is polynomially exactly learnable with improper equivalence queries.*
  [Idziak–Marković–McKenzie–Valeriote–Willard, 2010]
  - Generalizes [Dalmau–Jeavons, 2003] and [Bulatov–Chen–Dalmau, 2007].
- $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$ would imply that $\Gamma$ is polynomially exactly learnable with proper equivalence queries.

# Jeff's Theorem

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.
Wlog: $\mathcal{F} \cap \mathcal{H} = \emptyset$.

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.
Wlog: $\mathcal{F} \cap \mathcal{H} = \emptyset$.

## Theorem 1 (Jeff Shriner)

*There exists a finite algebra $\mathbf{A}_\mathcal{M}$ (in the language $\mathcal{F} \cup \mathcal{H}$) such that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$, and $\mathrm{SMP}(\mathbf{A}_\mathcal{M})$ is at least as hard as $\mathrm{SMP}(\mathbf{A})$.*

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.
Wlog: $\mathcal{F} \cap \mathcal{H} = \emptyset$.

**Theorem 1 (Jeff Shriner)**

*There exists a finite algebra $\mathbf{A}_\mathcal{M}$ (in the language $\mathcal{F} \cup \mathcal{H}$) such that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$, and* $\mathrm{SMP}(\mathbf{A}_\mathcal{M})$ *is at least as hard as* $\mathrm{SMP}(\mathbf{A})$.

- For Kozik's algebra $\mathbf{K}$,
  $\mathbf{K}_\mathcal{M}$ satisfies $\mathcal{M}$ and $\mathrm{SMP}(\mathbf{K}_\mathcal{M})$ is EXPTIME-complete.

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.
Wlog: $\mathcal{F} \cap \mathcal{H} = \emptyset$.

**Theorem 1 (Jeff Shriner)**

*There exists a finite algebra $\mathbf{A}_\mathcal{M}$ (in the language $\mathcal{F} \cup \mathcal{H}$) such that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$, and $\mathrm{SMP}(\mathbf{A}_\mathcal{M})$ is at least as hard as $\mathrm{SMP}(\mathbf{A})$.*

- For Kozik's algebra $\mathbf{K}$,
  $\mathbf{K}_\mathcal{M}$ satisfies $\mathcal{M}$ and $\mathrm{SMP}(\mathbf{K}_\mathcal{M})$ is EXPTIME-complete.

**Proof.**

**Step 1:** Definition of $\mathbf{A}_\mathcal{M}$; checking that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$.

# Jeff's Theorem

Let $\mathbf{A}$ be any finite algebra (in a finite language $\mathcal{F}$).
Let $\mathcal{M} = (\mathcal{H}, \Sigma)$ be a consistent, strong linear Maltsev condition such that $\Sigma$ does not entail cube identities for any $h \in \mathcal{H}$.
Wlog: $\mathcal{F} \cap \mathcal{H} = \emptyset$.

**Theorem 1 (Jeff Shriner)**

*There exists a finite algebra $\mathbf{A}_\mathcal{M}$ (in the language $\mathcal{F} \cup \mathcal{H}$) such that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$, and $\mathrm{SMP}(\mathbf{A}_\mathcal{M})$ is at least as hard as $\mathrm{SMP}(\mathbf{A})$.*

- For Kozik's algebra $\mathbf{K}$,
  $\mathbf{K}_\mathcal{M}$ satisfies $\mathcal{M}$ and $\mathrm{SMP}(\mathbf{K}_\mathcal{M})$ is EXPTIME-complete.

**Proof.**

**Step 1:** Definition of $\mathbf{A}_\mathcal{M}$; checking that $\mathbf{A}_\mathcal{M}$ satisfies $\mathcal{M}$.

**Step 2:** Polynomial time reduction of $\mathrm{SMP}(\mathbf{A})$ to $\mathrm{SMP}(\mathbf{A}_\mathcal{M})$. □

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows:

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;
- for each $h \in \mathcal{H}$ (say $h$ is $k$-ary) the operation $h$ on $A \cup \{0\}$ is defined by

$$h(b_1, \ldots, b_k) := \begin{cases} b_i & \text{if an identity entailed by } \Sigma \text{ forces this,} \end{cases}$$

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;
- for each $h \in \mathcal{H}$ (say $h$ is $k$-ary) the operation $h$ on $A \cup \{0\}$ is defined by

$$h(b_1, \ldots, b_k) := \begin{cases} b_i & \text{if an identity entailed by } \Sigma \text{ forces this,} \\ & \text{i.e., if } \Sigma \vdash h(x_1, \ldots, x_k) \approx x_i \\ & \quad \text{for some variables } x_1, \ldots, x_k \text{ such that} \\ & \quad b_p = b_q \text{ whenever } x_p = x_q, \end{cases}$$

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;
- for each $h \in \mathcal{H}$ (say $h$ is $k$-ary) the operation $h$ on $A \cup \{0\}$ is defined by

$$h(b_1, \ldots, b_k) := \begin{cases} b_i & \text{if an identity entailed by } \Sigma \text{ forces this,} \\ & \text{i.e., if } \Sigma \vdash h(x_1, \ldots, x_k) \approx x_i \\ & \quad \text{for some variables } x_1, \ldots, x_k \text{ such that} \\ & \quad b_p = b_q \text{ whenever } x_p = x_q, \\ 0 & \text{otherwise.} \end{cases}$$

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;
- for each $h \in \mathcal{H}$ (say $h$ is $k$-ary) the operation $h$ on $A \cup \{0\}$ is defined by

$$h(b_1, \ldots, b_k) := \begin{cases} b_i & \text{if an identity entailed by } \Sigma \text{ forces this,} \\ & \text{i.e., if } \Sigma \vdash h(x_1, \ldots, x_k) \approx x_i \\ & \quad \text{for some variables } x_1, \ldots, x_k \text{ such that} \\ & \quad b_p = b_q \text{ whenever } x_p = x_q, \\ 0 & \text{otherwise.} \end{cases}$$

- **Note:** $x_i \in \{x_1, \ldots, x_k\}$, since $\mathcal{M}$ is consistent.
$h$ is well-defined.

# Step 1: Constructing $\mathbf{A}_{\mathcal{M}}$

Given: $\mathbf{A} = (A; \mathcal{F})$ and $\mathcal{M} = (\mathcal{H}, \Sigma)$.
Define $\mathbf{A}_{\mathcal{M}} = (A \cup \{0\}; \mathcal{F} \cup \mathcal{H})$ as follows: $0 \notin A$;

- for each $f \in \mathcal{F}$, the operation $f$ on $A$ is extended to $A \cup \{0\}$ so that 0 is an absorbing element;
- for each $h \in \mathcal{H}$ (say $h$ is $k$-ary) the operation $h$ on $A \cup \{0\}$ is defined by

$$h(b_1, \ldots, b_k) := \begin{cases} b_i & \text{if an identity entailed by } \Sigma \text{ forces this,} \\ & \text{i.e., if } \Sigma \vdash h(x_1, \ldots, x_k) \approx x_i \\ & \quad \text{for some variables } x_1, \ldots, x_k \text{ such that} \\ & \quad b_p = b_q \text{ whenever } x_p = x_q, \\ 0 & \text{otherwise.} \end{cases}$$

- **Note:** $x_i \in \{x_1, \ldots, x_k\}$, since $\mathcal{M}$ is consistent.
  $h$ is well-defined.

$\mathbf{A}_{\mathcal{M}}$ satisfies the Maltsev condition $\mathcal{M}$.

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

Input for SMP($\mathbf{A}$):
$a_1, \ldots, a_n, b \in \mathbf{A}^m$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

Input for SMP($\mathbf{A}$):
$a_1, \dots, a_n, b \in \mathbf{A}^m$

$\overset{\text{const time}}{\longmapsto}$

Input for SMP($\mathbf{A}_{\mathcal{M}}$):
$a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

| | Input for SMP($\mathbf{A}$): | | Input for SMP($\mathbf{A}_{\mathcal{M}}$): |
|---|---|---|---|
| | $a_1, \dots, a_n, b \in \mathbf{A}^m$ | $\overset{\text{const time}}{\longmapsto}$ | $a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$ |
| **Goal:** | $b \in \langle a_1 \dots, a_n \rangle_{\mathbf{A}^m}$ | $\iff$ | $b \in \langle a_1, \dots, a_n \rangle_{\mathbf{A}_{\mathcal{M}}^m}$ |

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

| | Input for SMP($\mathbf{A}$): | | Input for SMP($\mathbf{A}_{\mathcal{M}}$): |
|---|---|---|---|
| | $a_1, \dots, a_n, b \in \mathbf{A}^m$ | $\overset{\text{const time}}{\longmapsto}$ | $a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$ |
| **Goal:** | $b \in \langle a_1 \dots, a_n \rangle_{\mathbf{A}^m}$ | $\iff$ | $b \in \langle a_1, \dots, a_n \rangle_{\mathbf{A}_{\mathcal{M}}^m}$ |
| | | $\overset{\text{easy}}{\Longrightarrow}$ | |

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

| | Input for SMP($\mathbf{A}$): | | Input for SMP($\mathbf{A}_{\mathcal{M}}$): |
|---|---|---|---|
| | $a_1, \dots, a_n, b \in \mathbf{A}^m$ | const time $\longmapsto$ | $a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$ |
| **Goal:** | $b \in \langle a_1 \dots, a_n \rangle_{\mathbf{A}^m}$ | $\Longleftrightarrow$ | $b \in \langle a_1, \dots, a_n \rangle_{\mathbf{A}_{\mathcal{M}}^m}$ |

easy $\Longrightarrow$

!!! $\Longleftarrow$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

| | Input for SMP($\mathbf{A}$): | | Input for SMP($\mathbf{A}_{\mathcal{M}}$): |
|---|---|---|---|
| | $a_1, \dots, a_n, b \in \mathbf{A}^m$ | $\overset{\text{const time}}{\longmapsto}$ | $a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$ |
| **Goal:** | $b \in \langle a_1 \dots, a_n \rangle_{\mathbf{A}^m}$ | $\iff$ | $b \in \langle a_1, \dots, a_n \rangle_{\mathbf{A}_{\mathcal{M}}^m}$ |
| | | $\overset{\text{easy}}{\Longrightarrow}$ | |
| | | $\overset{!!!}{\Longleftarrow}$ | |

Need to show: If $b = t(a_1 \dots, a_n)$ for some term $t$ in the language $\mathcal{F} \cup \mathcal{H}$, then $b = t'(a_1 \dots, a_n)$ for some term $t'$ in the language $\mathcal{F}$.

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$)

| | Input for SMP($\mathbf{A}$): | | Input for SMP($\mathbf{A}_{\mathcal{M}}$): |
|---|---|---|---|
| | $a_1, \dots, a_n, b \in \mathbf{A}^m$ | $\overset{\text{const time}}{\longmapsto}$ | $a_1, \dots, a_n, b \in \mathbf{A}_{\mathcal{M}}^m$ |
| **Goal:** | $b \in \langle a_1 \dots, a_n \rangle_{\mathbf{A}^m}$ | $\iff$ | $b \in \langle a_1, \dots, a_n \rangle_{\mathbf{A}_{\mathcal{M}}^m}$ |

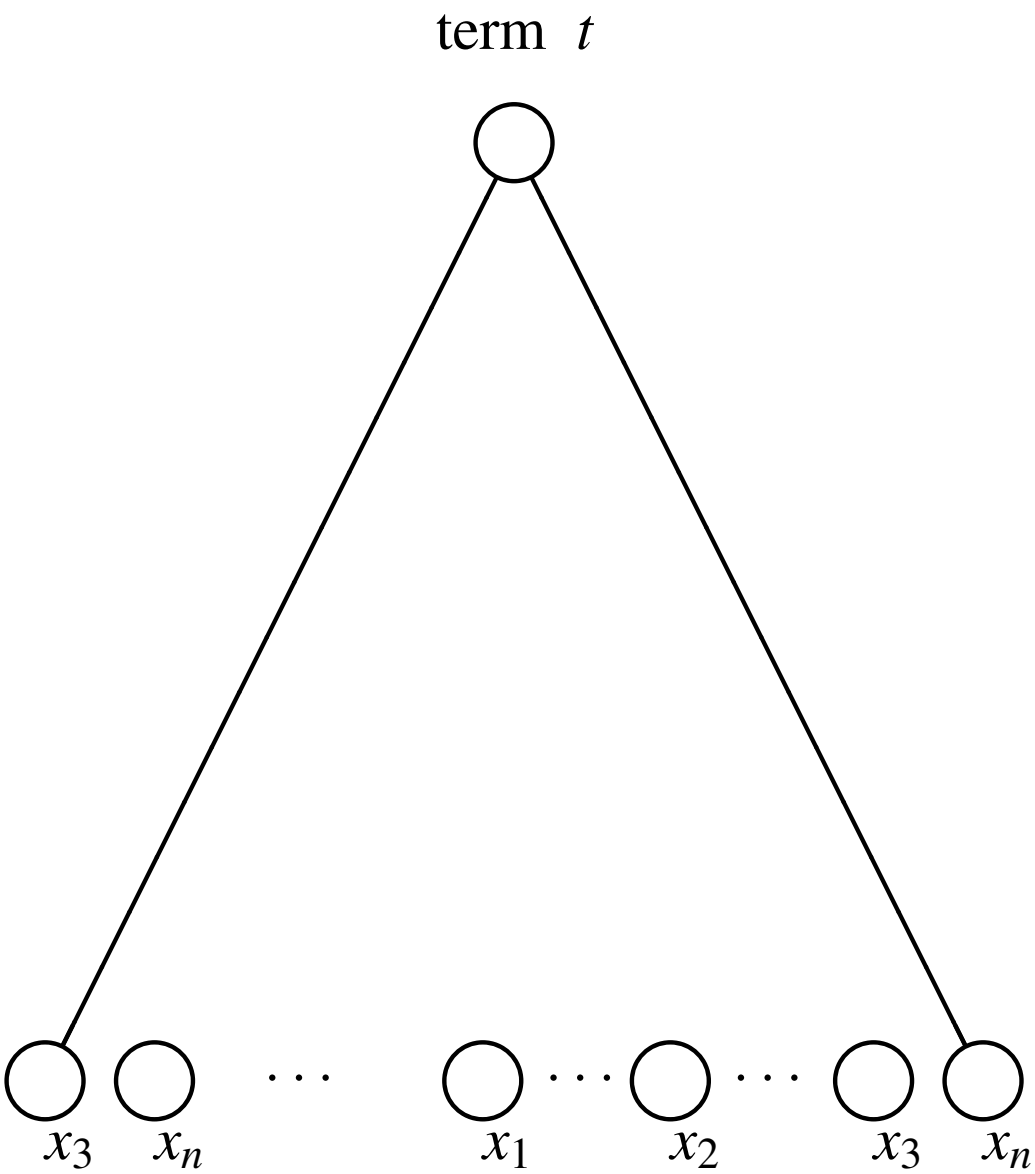
$\overset{\text{easy}}{\Longrightarrow}$

$\overset{!!!}{\Longleftarrow}$

Need to show: If $b = t(a_1 \dots, a_n)$ for some term $t$ in the language $\mathcal{F} \cup \mathcal{H}$, then $b = t'(a_1 \dots, a_n)$ for some term $t'$ in the language $\mathcal{F}$.

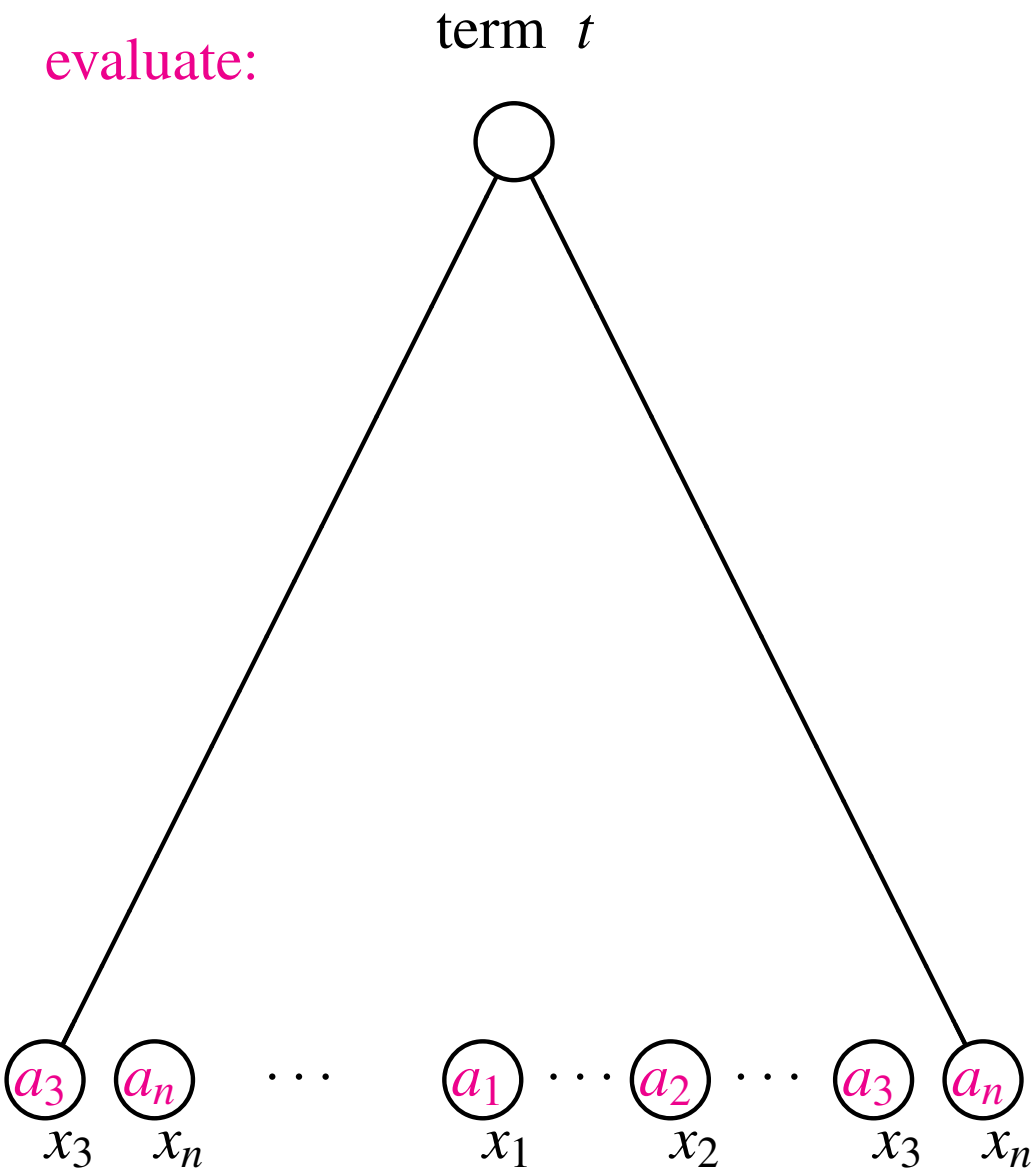
Enough: to eliminate one occurrence of a symbol $h \in \mathcal{H}$ from $t$ at a time.

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

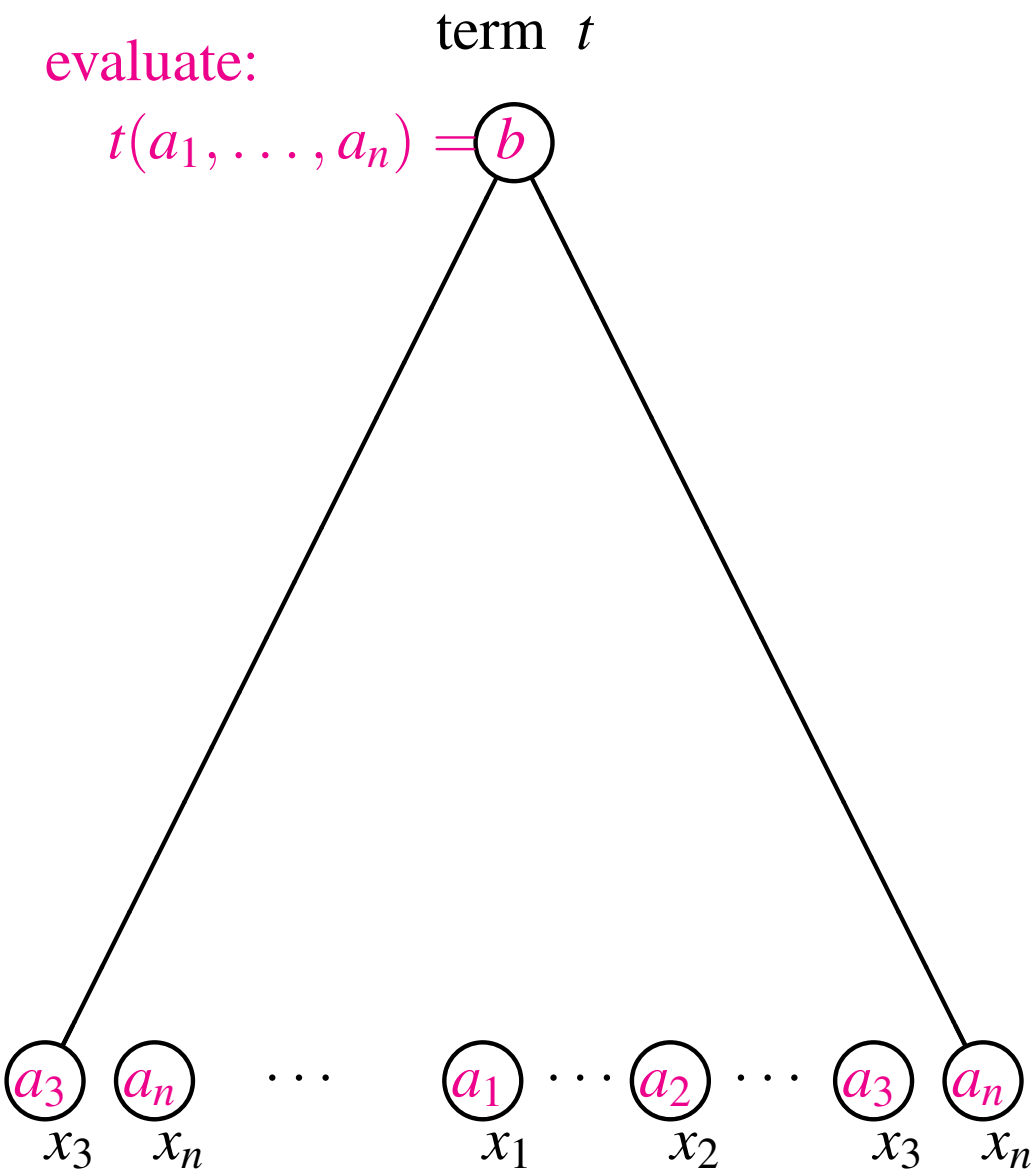


# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

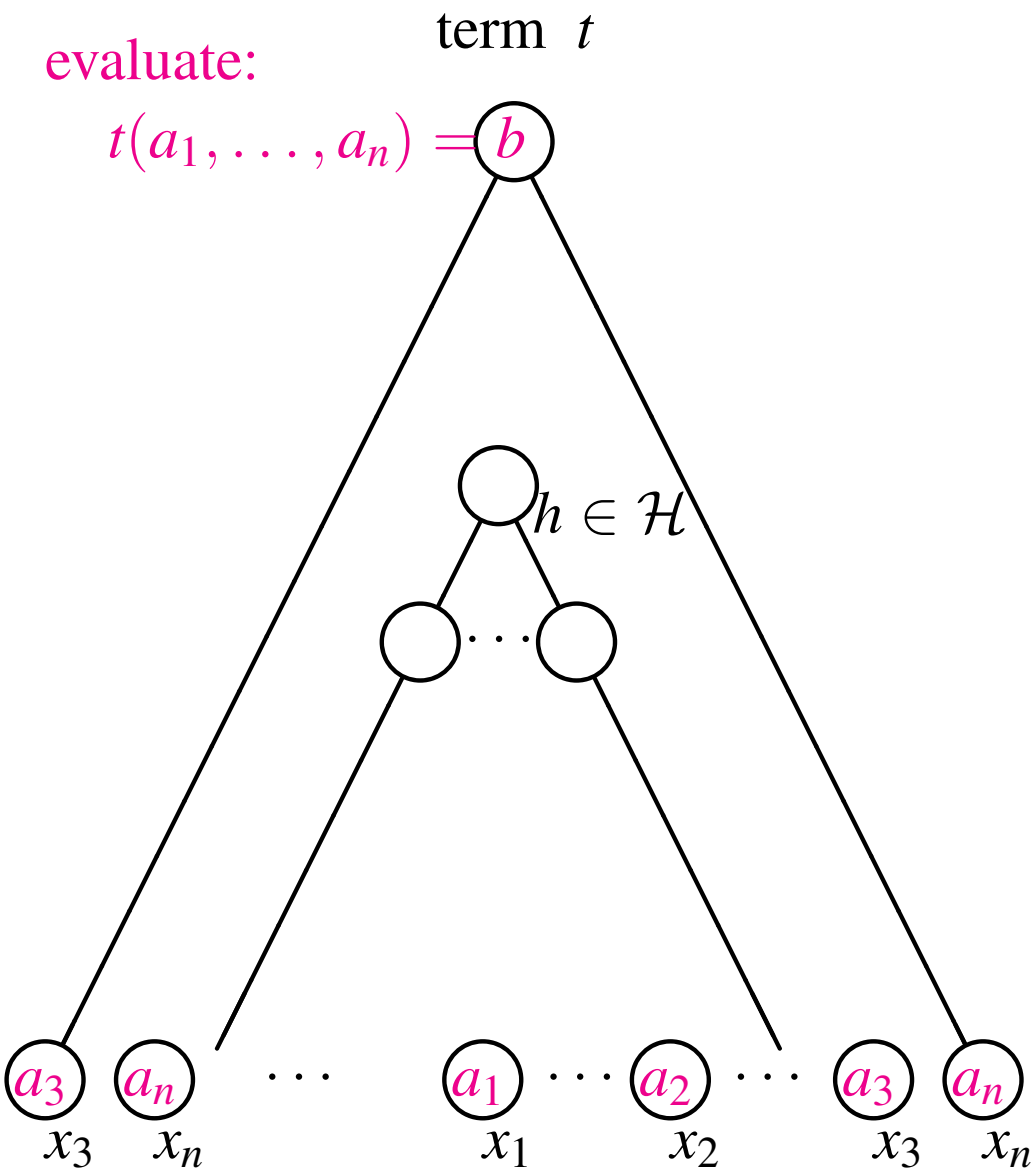
evaluate:



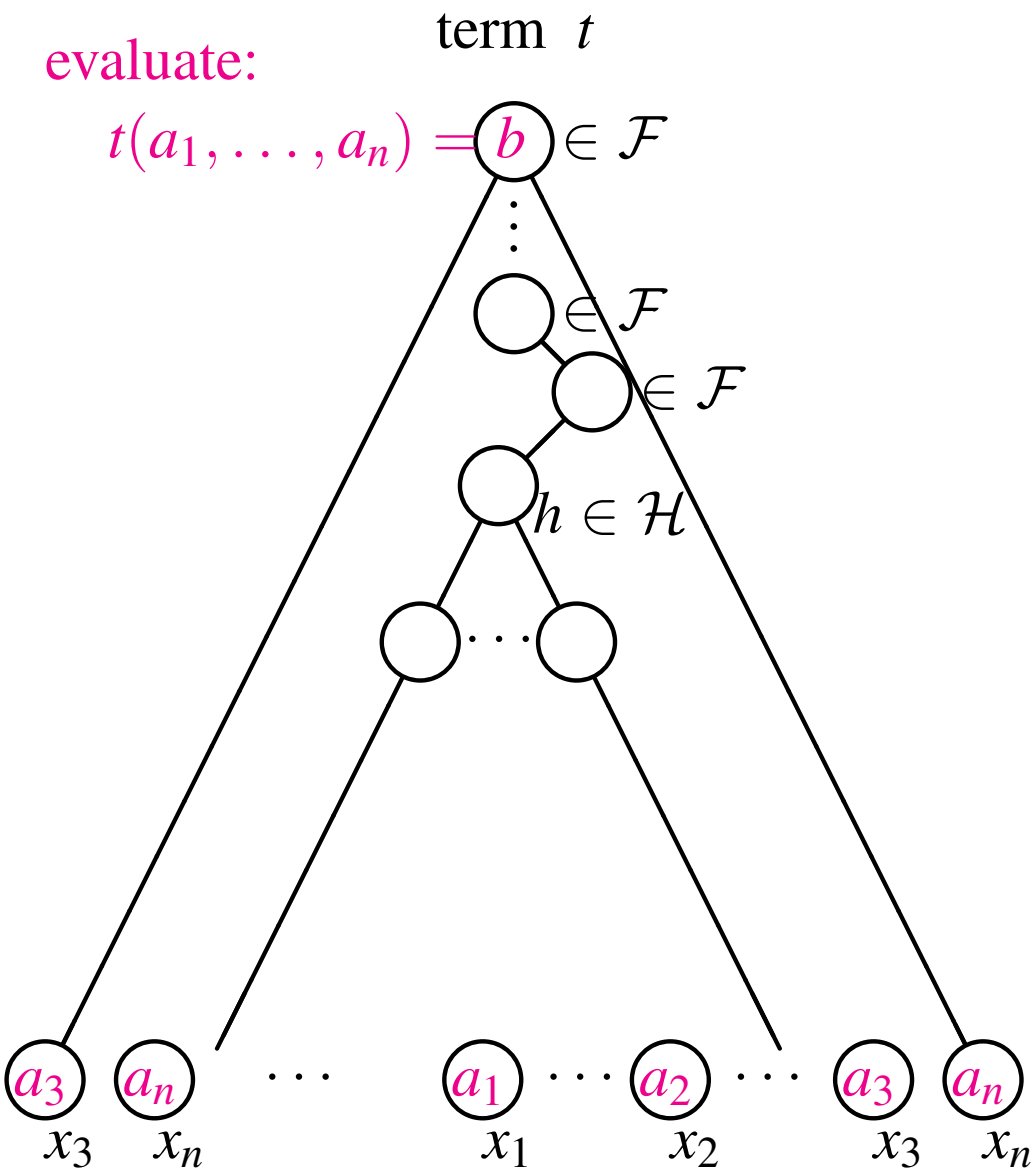
# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)



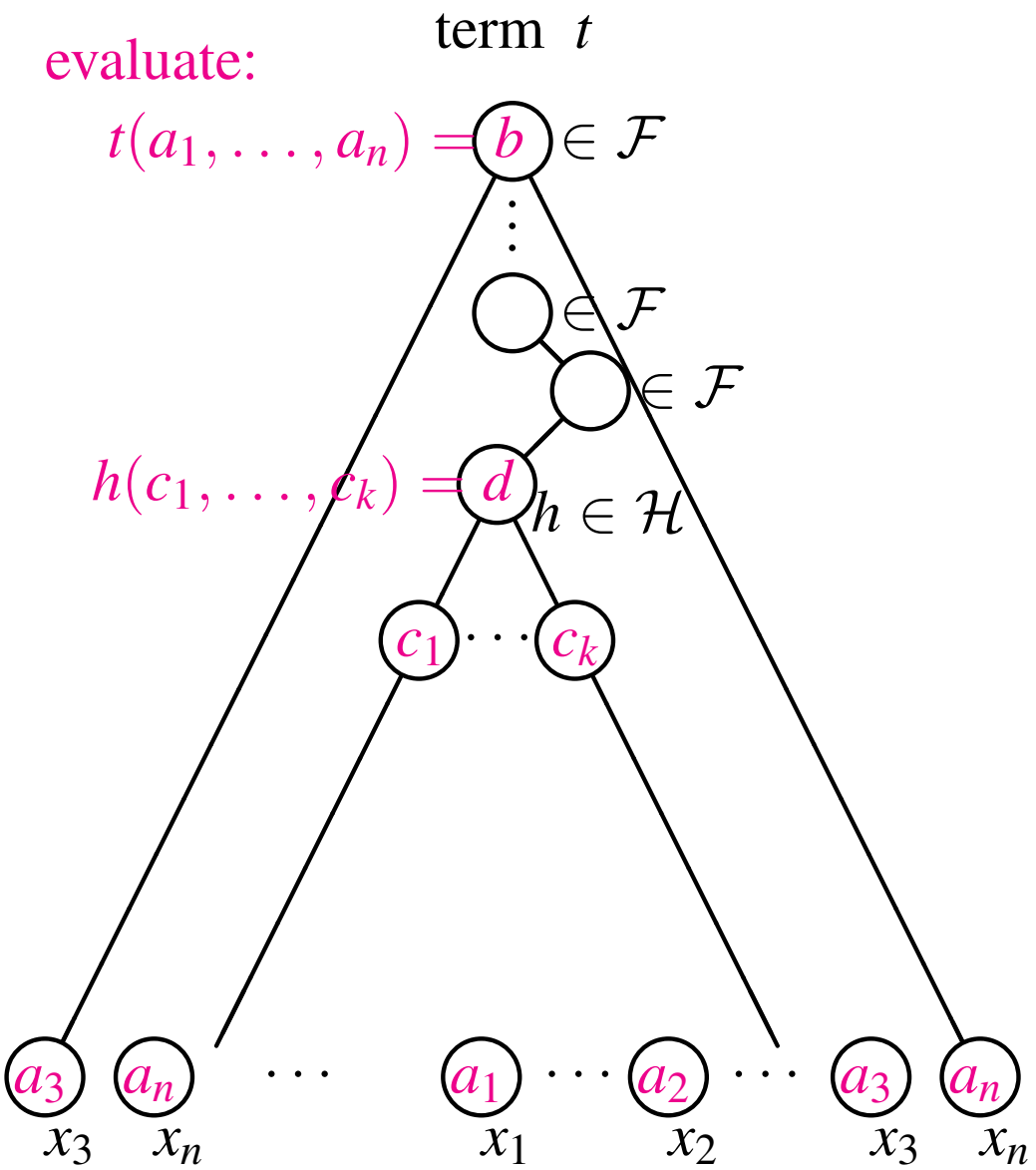
# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)



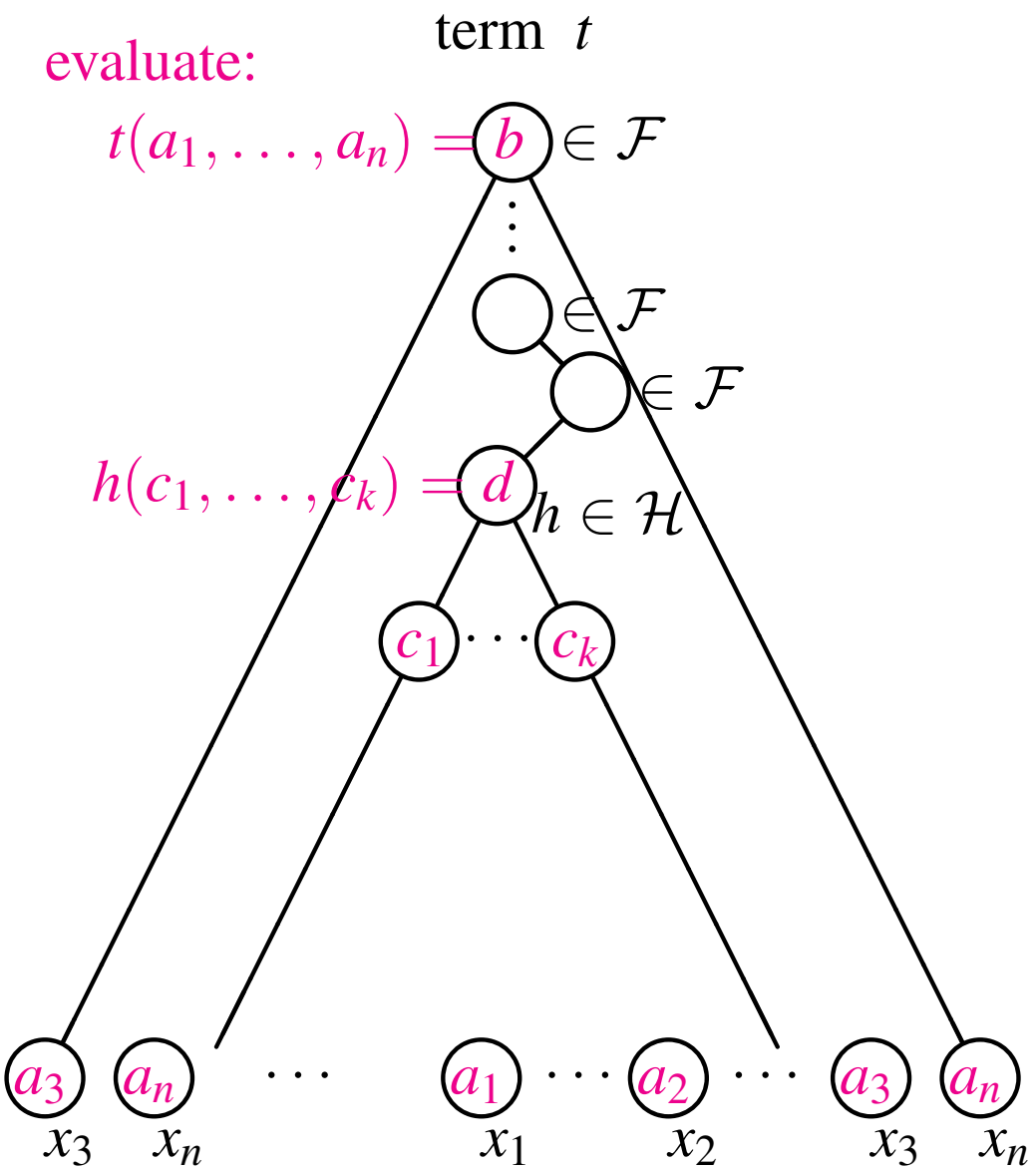
# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)



# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

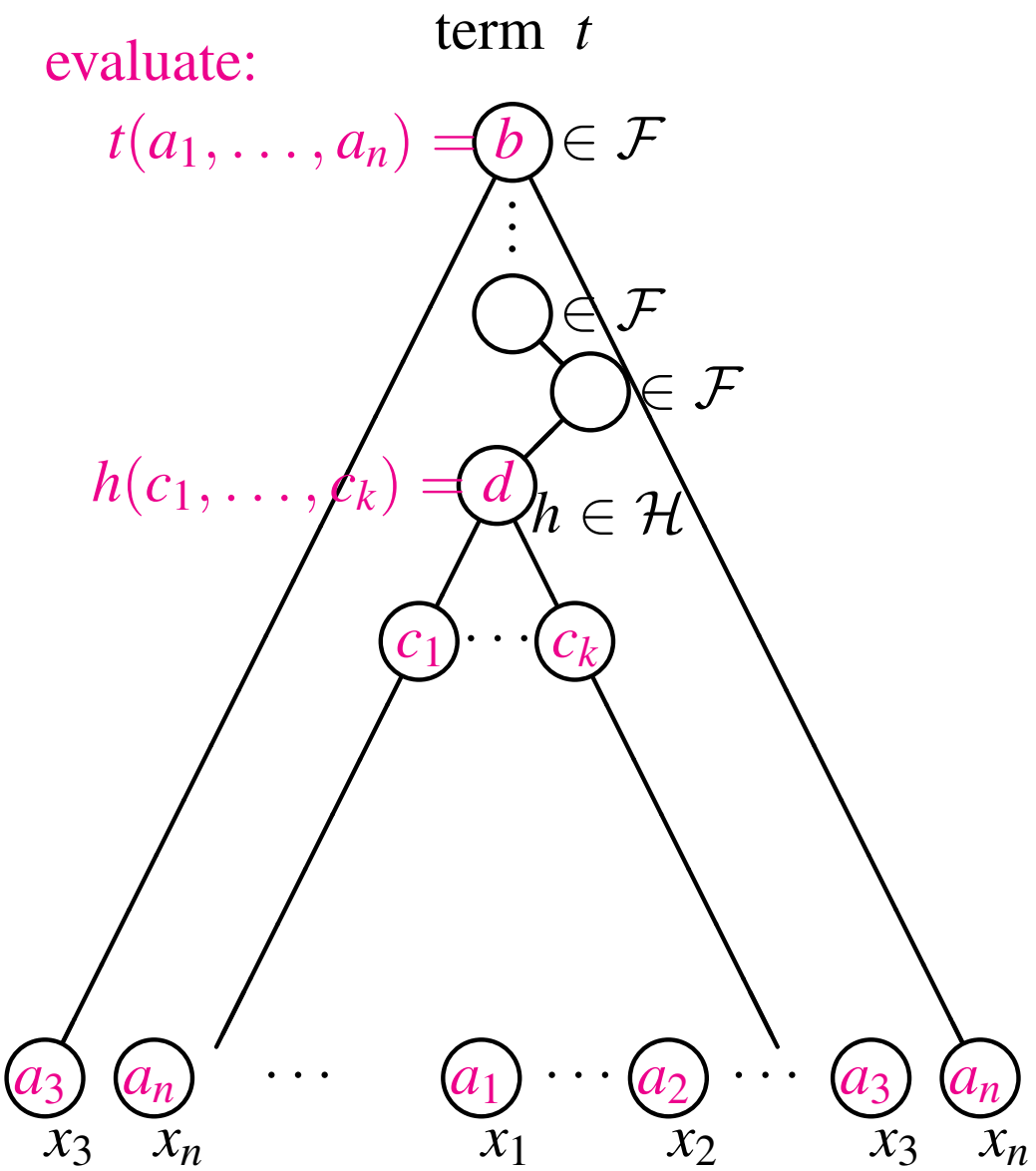


# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

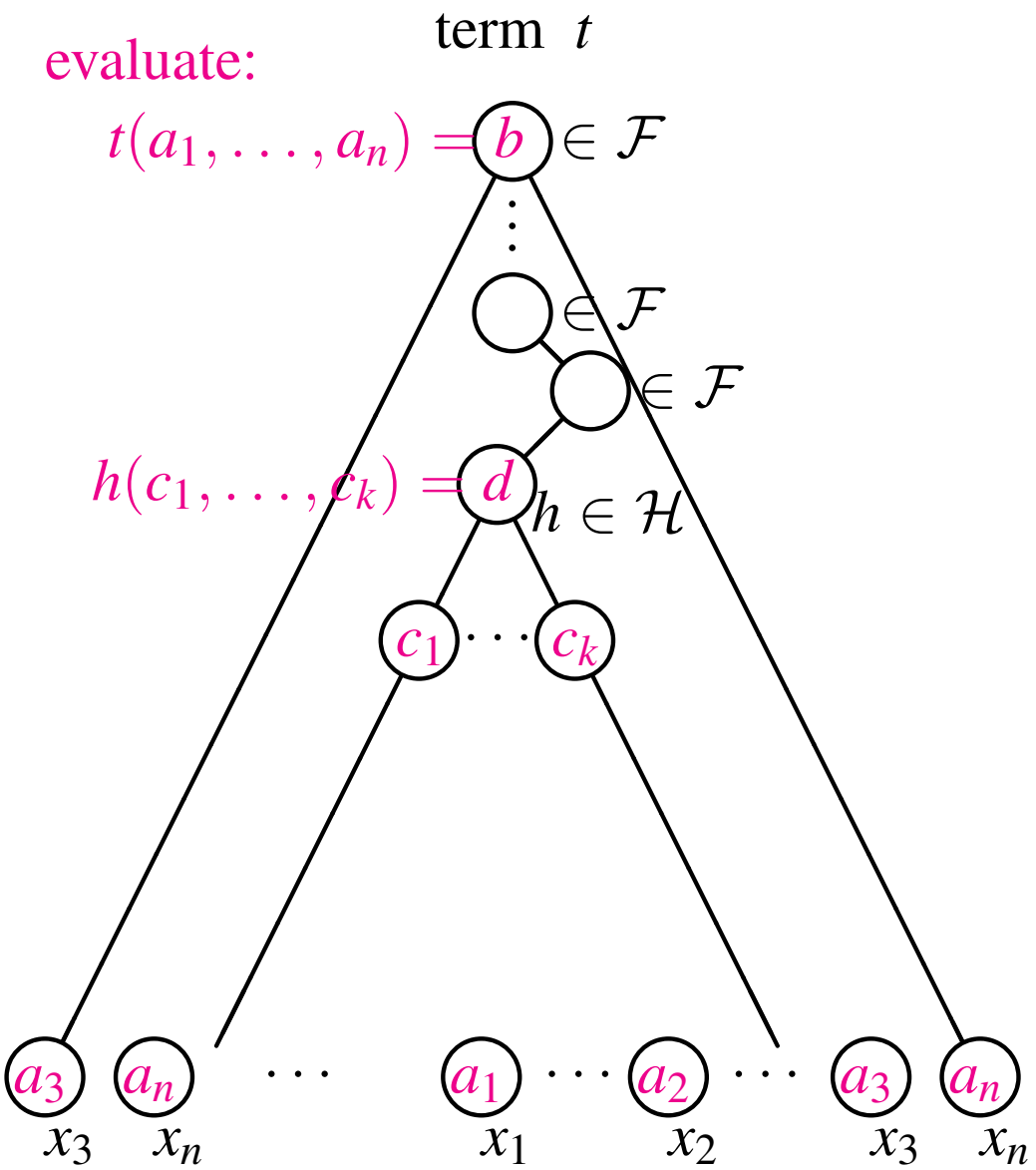
# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

**Case 1:** $d = c_i$ for some $i$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

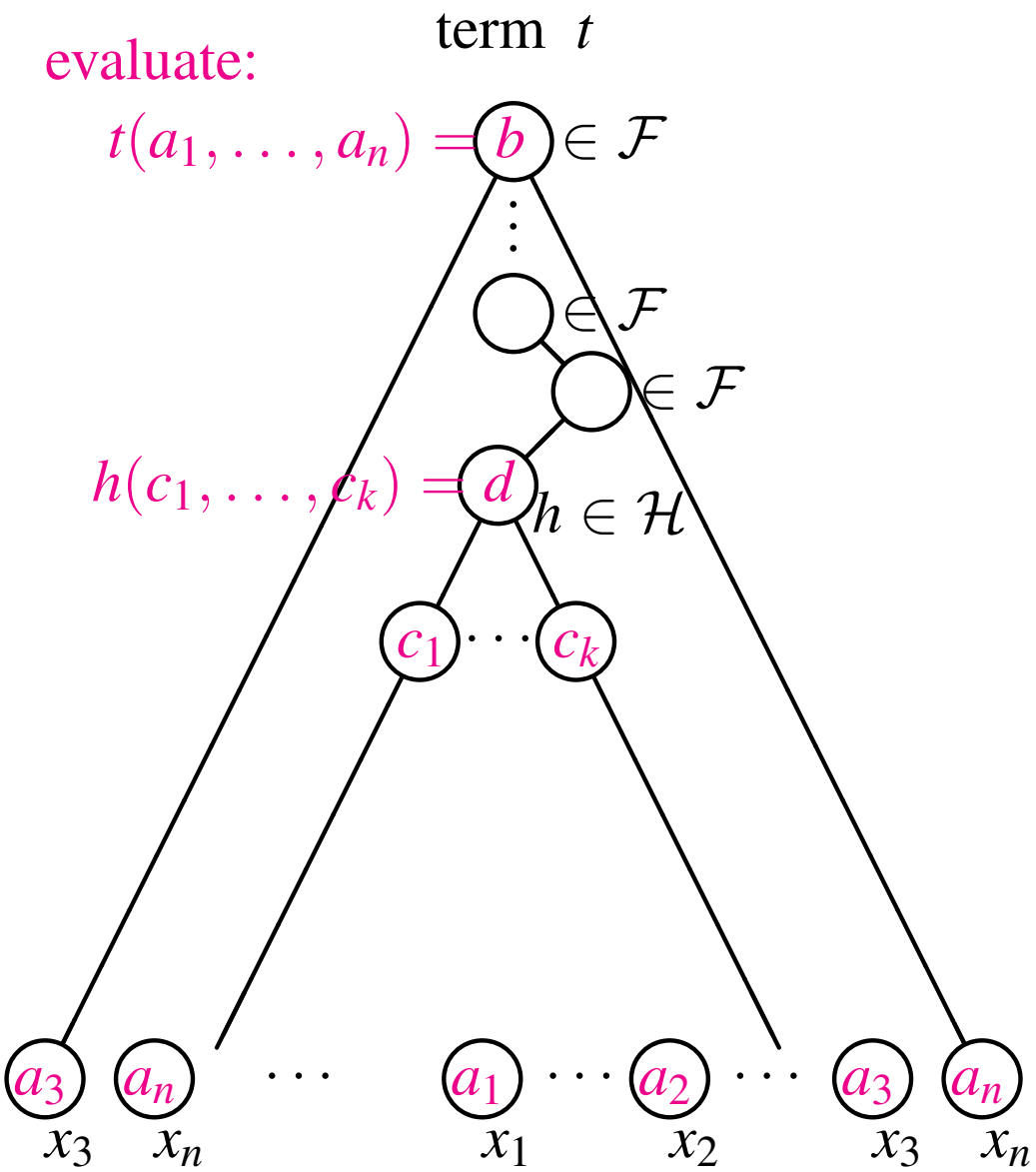



$d$ has no 0 coord's

**Case 1:** $d = c_i$ for some $i$

Eliminate $h$.

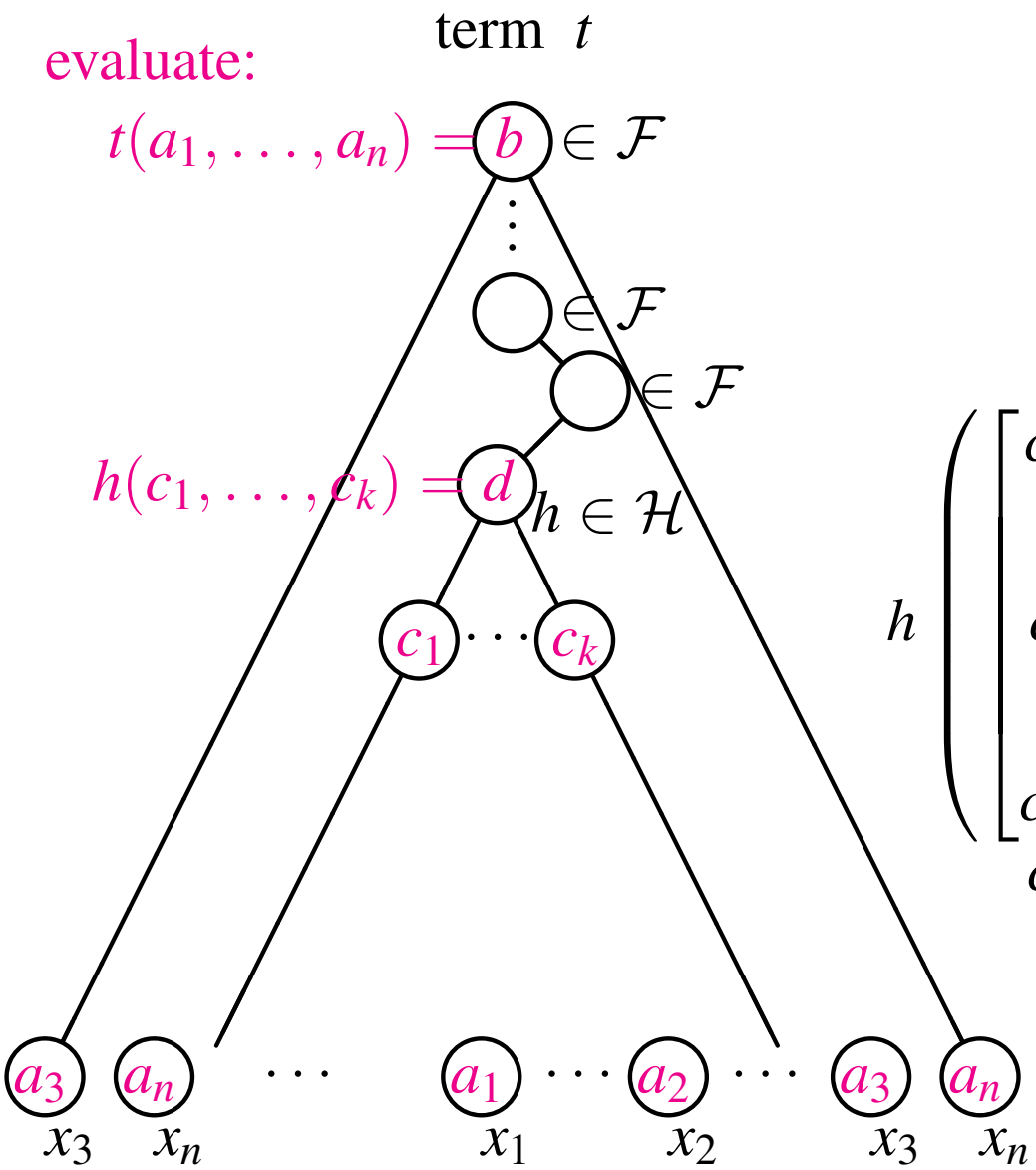
# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

**Case 2:** $d \neq c_i$ for every $i$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

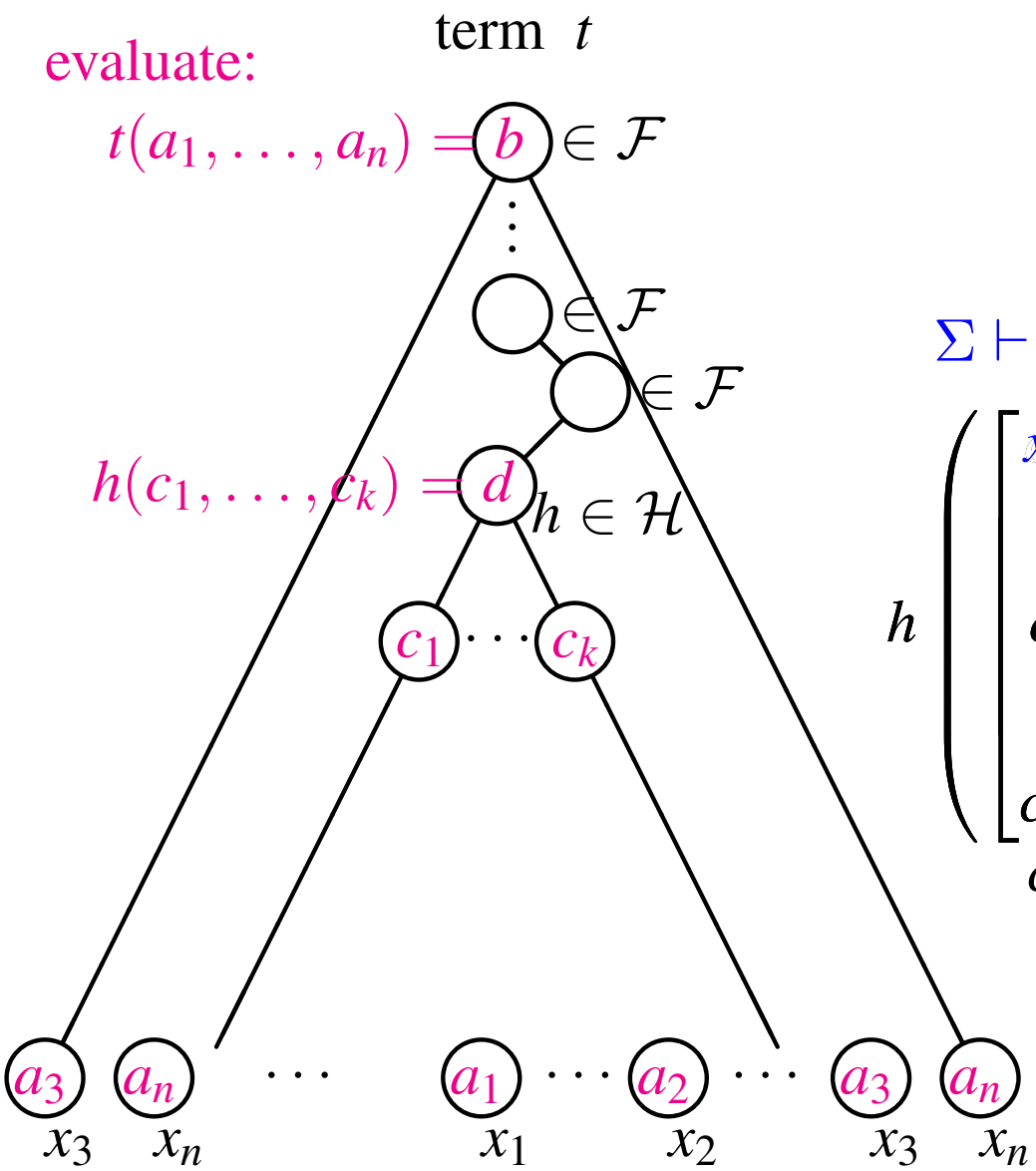



$d$ has no 0 coord's

**Case 2:** $d \neq c_i$ for every $i$

$$h\left(\underset{c_1}{\begin{bmatrix} c_{11} \\ \vdots \\ c_{j1} \\ \vdots \\ c_{m1} \end{bmatrix}} \cdots \underset{c_i}{\begin{bmatrix} c_{1i} \\ \vdots \\ c_{ji} \\ \vdots \\ c_{mi} \end{bmatrix}} \cdots \underset{c_k}{\begin{bmatrix} c_{1k} \\ \vdots \\ c_{jk} \\ \vdots \\ c_{mk} \end{bmatrix}}\right) = \underset{d}{\begin{bmatrix} d_1 \\ \vdots \\ d_j \\ \vdots \\ d_m \end{bmatrix}}$$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

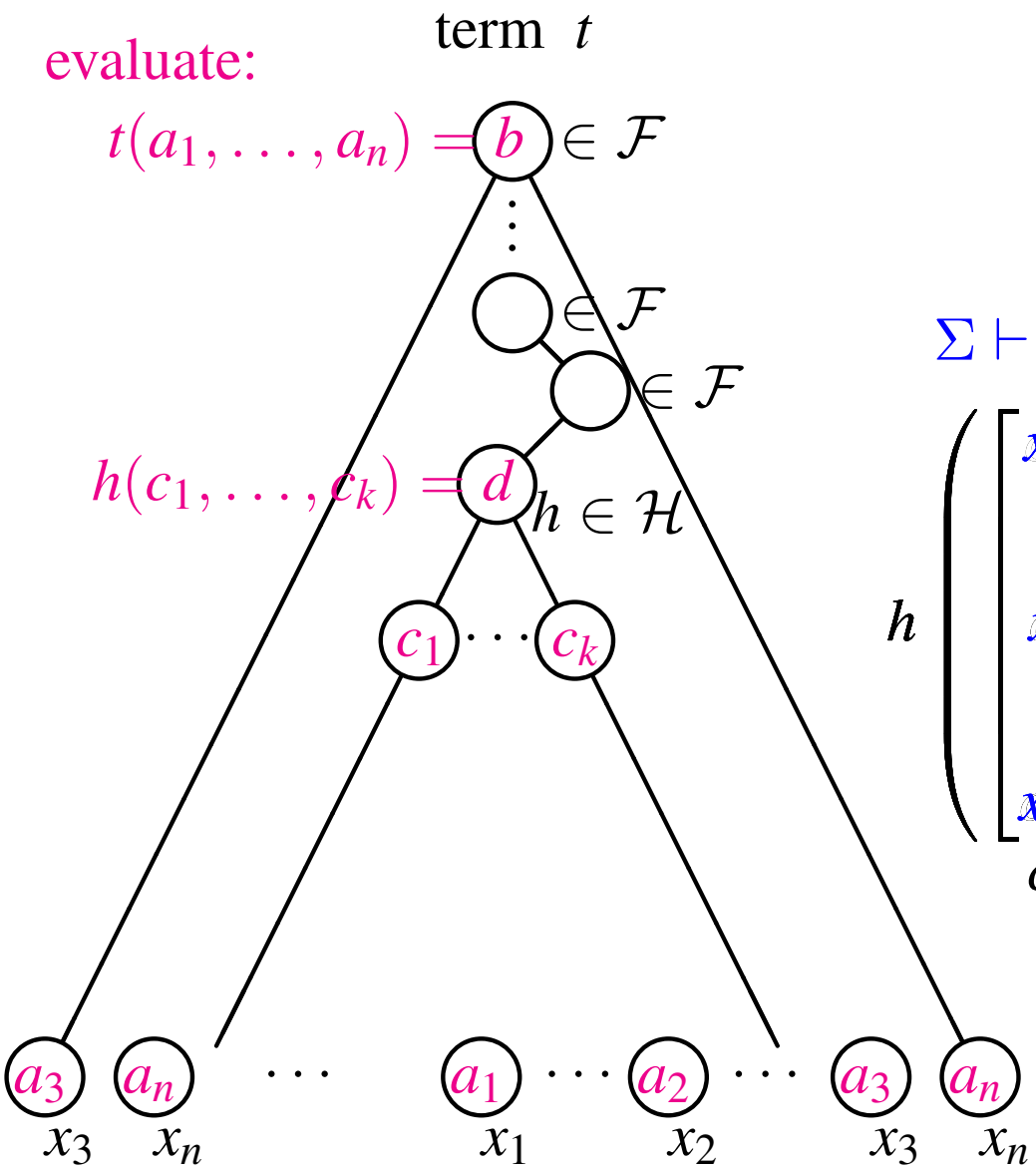



$d$ has no 0 coord's

**Case 2:** $d \neq c_i$ for every $i$

$$\Sigma \vdash h\left(\begin{bmatrix} x_{11} \\ \vdots \\ c_{j1} \\ \vdots \\ c_{m1} \end{bmatrix}_{c_1} \cdots \begin{bmatrix} x_{1i} \\ \vdots \\ c_{ji} \\ \vdots \\ c_{mi} \end{bmatrix}_{c_i} \cdots \begin{bmatrix} x_{1k} \\ \vdots \\ c_{jk} \\ \vdots \\ c_{mk} \end{bmatrix}_{c_k}\right) = \begin{bmatrix} y \\ \vdots \\ d_j \\ \vdots \\ d_m \end{bmatrix}_{d}$$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)

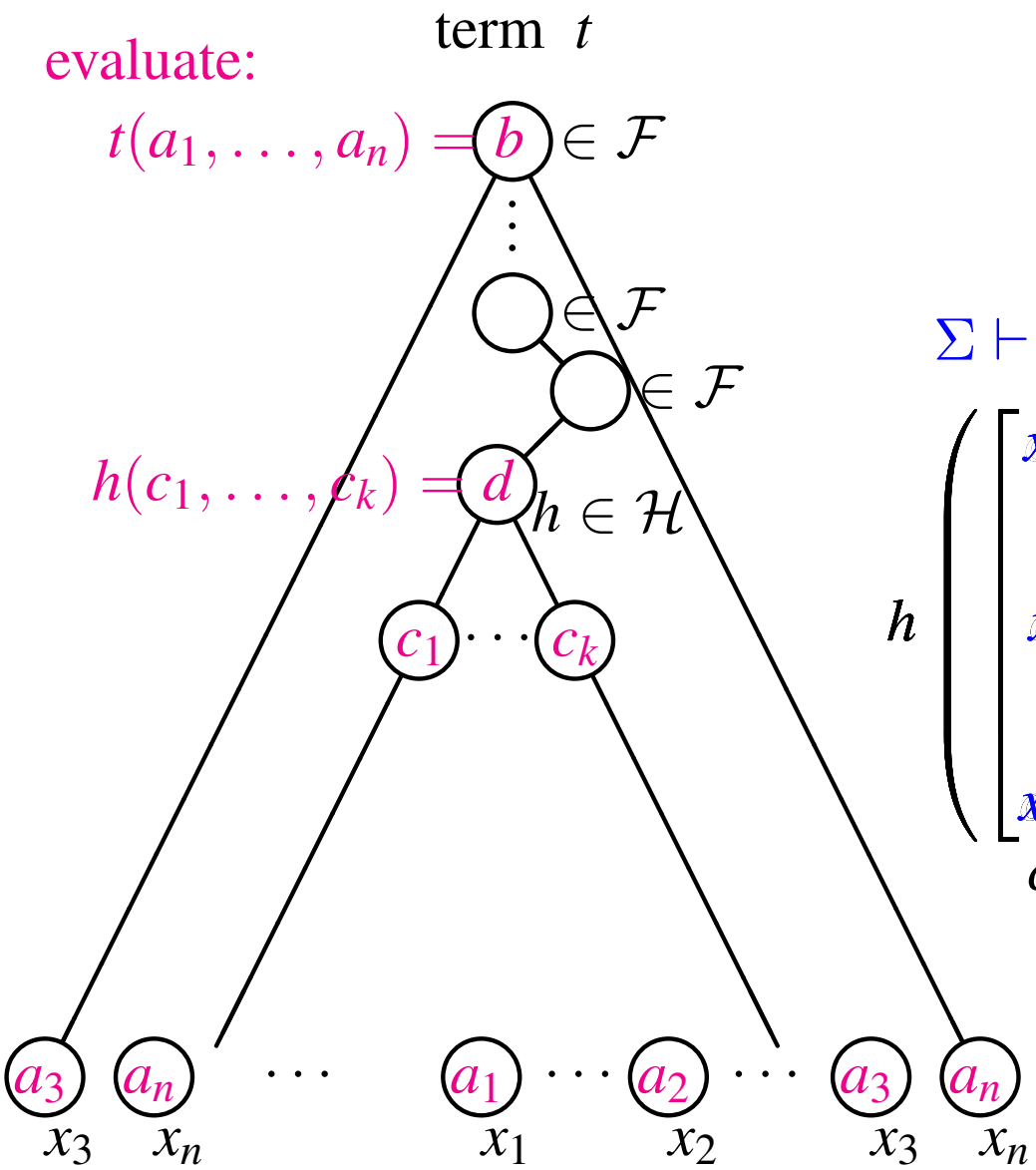



$d$ has no 0 coord's

**Case 2:** $d \neq c_i$ for every $i$

$$\Sigma \vdash h\left(\begin{bmatrix} x_{11} \\ \vdots \\ x_{j1} \\ \vdots \\ x_{m1} \end{bmatrix}_{c_1} \cdots \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{mi} \end{bmatrix}_{c_i} \cdots \begin{bmatrix} x_{1k} \\ \vdots \\ x_{jk} \\ \vdots \\ x_{mk} \end{bmatrix}_{c_k}\right) \approx \begin{bmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_m \end{bmatrix}_{d}$$

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

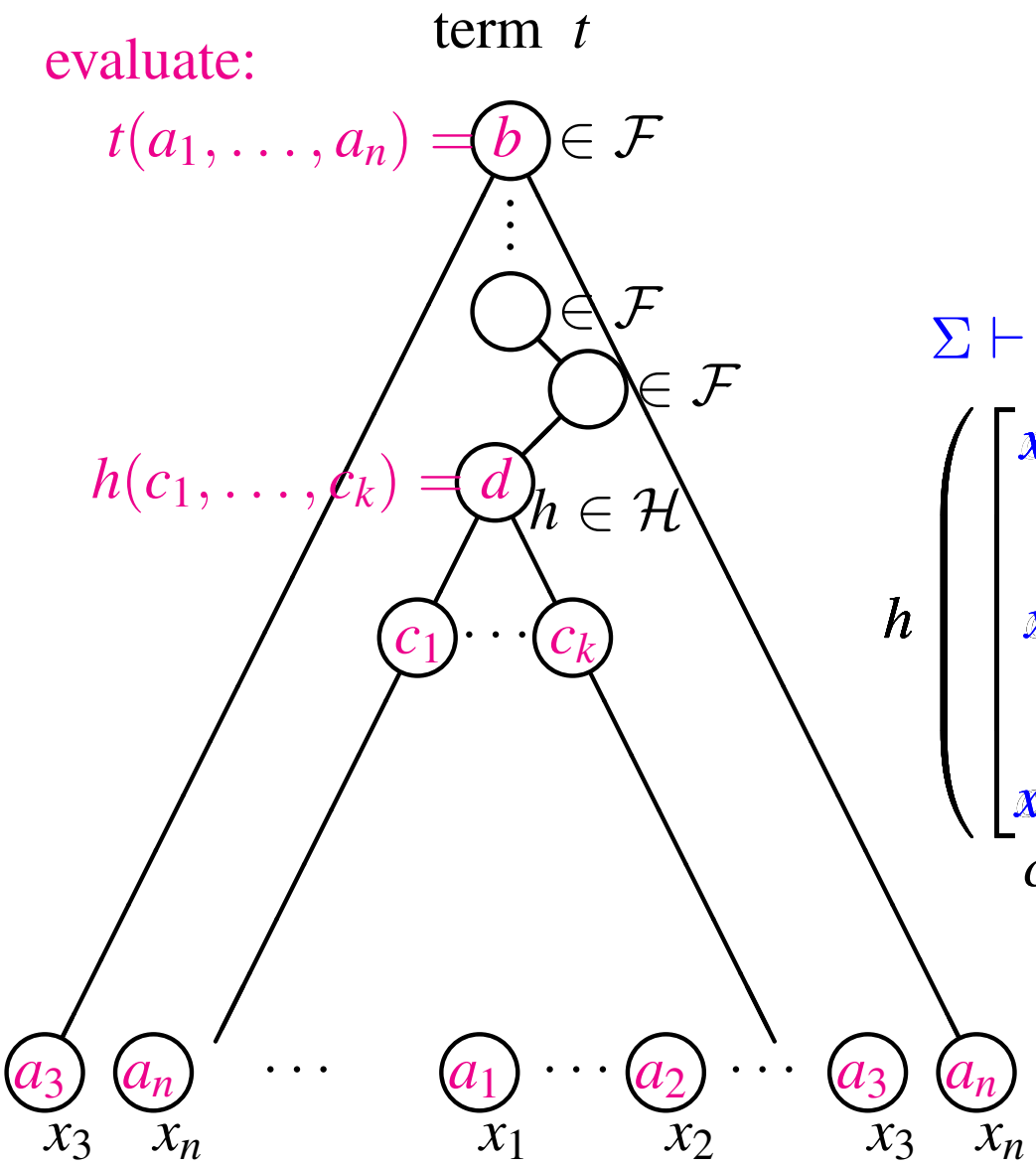
**Case 2:** $d \neq c_i$ for every $i$

$$\Sigma \vdash h\left(\begin{bmatrix} x_{11} \\ \vdots \\ x_{j1} \\ \vdots \\ x_{m1} \end{bmatrix} \cdots \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{mi} \end{bmatrix} \cdots \begin{bmatrix} x_{1k} \\ \vdots \\ x_{jk} \\ \vdots \\ x_{mk} \end{bmatrix}\right) \approx \begin{bmatrix} y \\ \vdots \\ y \\ \vdots \\ y \end{bmatrix}$$

$c_1 \quad c_i \quad c_k \quad d$

Replace each $x_{qr} \neq y$ by $x$.

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

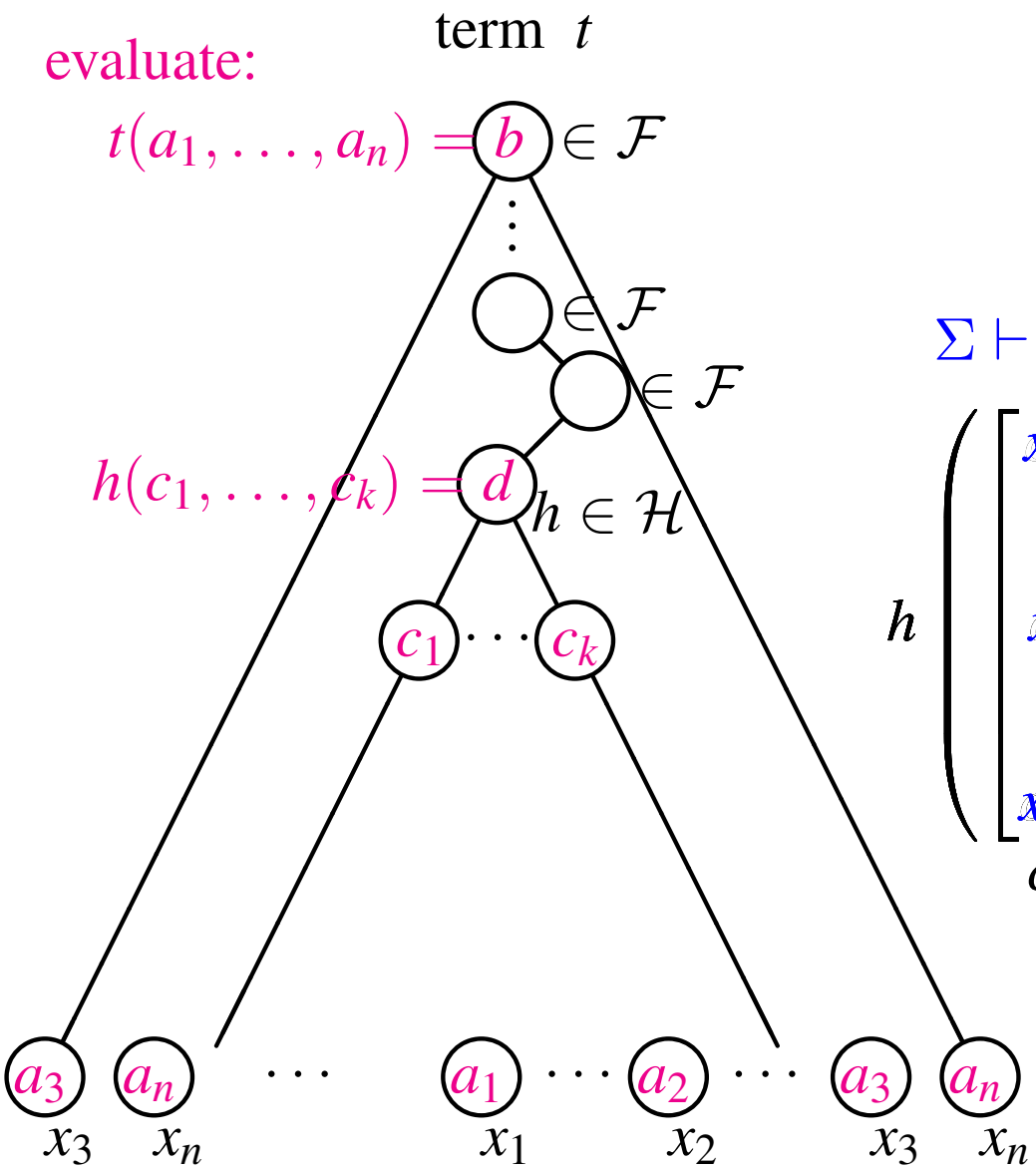
**Case 2:** $d \neq c_i$ for every $i$

$$\Sigma \vdash h\left(\begin{bmatrix} x_{11} \\ \vdots \\ x_{j1} \\ \vdots \\ x_{m1} \end{bmatrix} \cdots \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{mi} \end{bmatrix} \cdots \begin{bmatrix} x_{1k} \\ \vdots \\ x_{jk} \\ \vdots \\ x_{mk} \end{bmatrix}\right) \approx \begin{bmatrix} y \\ \vdots \\ y \\ \vdots \\ y \end{bmatrix}$$

$c_1 \quad c_i \quad c_k \quad d$

Replace each $x_{qr} \neq y$ by $x$.
We get cube identities for $h$.

# Step 2: P-time reduction of SMP($\mathbf{A}$) to SMP($\mathbf{A}_{\mathcal{M}}$) (Cont'd)




$d$ has no 0 coord's

**Case 2:** $d \neq c_i$ for every $i$

$$\Sigma \vdash h\left(\begin{bmatrix} x_{11} \\ \vdots \\ x_{j1} \\ \vdots \\ x_{m1} \end{bmatrix} \cdots \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{mi} \end{bmatrix} \cdots \begin{bmatrix} x_{1k} \\ \vdots \\ x_{jk} \\ \vdots \\ x_{mk} \end{bmatrix}\right) \approx \begin{bmatrix} y \\ \vdots \\ y \\ \vdots \\ y \end{bmatrix}$$

$c_1$ $c_i$ $c_k$ $d$

Replace each $x_{qr} \neq y$ by $x$.
We get cube identities for $h$.
Impossible!

# Sufficient Condition for $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$

Throughout:

$\mathcal{V}$: a variety in a finite language, $\mathcal{K}$: a finite set of finite algebras in $\mathcal{V}$

# Sufficient Condition for SMP($\mathbf{A}$) $\in$ P

Throughout:

$\mathcal{V}$: a variety in a finite language, $\mathcal{K}$: a finite set of finite algebras in $\mathcal{V}$

SMP($\mathcal{K}$):

INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$.

QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle$ ?

# Sufficient Condition for $\mathrm{SMP}(\mathbf{A}) \in \mathsf{P}$

Throughout:

$\mathcal{V}$: a variety in a finite language, $\mathcal{K}$: a finite set of finite algebras in $\mathcal{V}$

$\mathrm{SMP}(\mathcal{K})$:

INPUT: $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$.

QUESTION: Is $b \in \langle a_1, \ldots, a_n \rangle$ ?

**Theorem 2 (Peter Mayr – ASz)**

*If $\mathcal{V}$ has a cube term, then $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ provided*

$(*)$ *for every SI in $\mathbb{HSK}$ with abelian monolith $\mu$, $(0 : \mu)$ is supernilpotent.*

# Sufficient Condition for SMP($\mathbf{A}$) $\in$ P

Throughout:

$\mathcal{V}$: a variety in a finite language, $\mathcal{K}$: a finite set of finite algebras in $\mathcal{V}$

SMP($\mathcal{K}$):
INPUT: $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$.
QUESTION: Is $b \in \langle a_1, \dots, a_n \rangle$ ?

**Theorem 2 (Peter Mayr – ASz)**

*If $\mathcal{V}$ has a cube term, then* SMP($\mathcal{K}$) $\in$ P *provided*

$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0 : \mu)$ is supernilpotent.*

Extends the earlier result [Bulatov–Mayr–ASz, 201?] that

- same conclusion is true under the stronger assumption that

  $(**)$ for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0 : \mu)$ is abelian (equivalently: $\mathcal{V}(\mathcal{K})$ is residually small).

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$;

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \ldots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \ldots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;
- $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$;

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \ldots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \ldots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;
- $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$; in fact,
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \ldots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \ldots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;
- $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$; in fact,
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$

(3) $\nu_i$'s abelian $\Longrightarrow$ $\mathbf{A}_i/\nu_i$'s 'reduce' to abelian Maltsev alg's ($\sim$ modules).

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;
- $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$; in fact,
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$

(3) $\nu_i$'s abelian $\Longrightarrow$ $\mathbf{A}_i/\nu_i$'s 'reduce' to abelian Maltsev alg's ($\sim$ modules).

Idea of proof in the supernilpotent case: Use (1)–(2) as before, 'reduce' the $\mathbf{A}_i/\nu_i$'s to supernilpotent Maltsev algebras, and apply

# Idea of Proof

Proof in the abelian case relies on:

(1) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\mathbb{HS}\,\mathcal{K}) \in \mathsf{P} \Leftrightarrow \mathrm{SMP}(\prod_{\mathbf{A}\in\mathbb{HS}\,\mathcal{K}} \mathbf{A}) \in \mathsf{P}$; may assume $\mathcal{K} = \mathbb{HS}\,\mathcal{K}$.

(2) $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P} \Leftrightarrow$ the problem restricted to these special inputs is in $\mathsf{P}$:

- $a_1, \ldots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ with $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{K}$;
- $\mathbf{B} = \langle a_1, \ldots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \ldots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0 : \mu_i)$;
- $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$; in fact,
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$

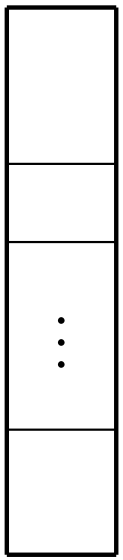
(3) $\nu_i$'s abelian $\Longrightarrow$ $\mathbf{A}_i/\nu_i$'s 'reduce' to abelian Maltsev alg's ($\sim$ modules).

Idea of proof in the supernilpotent case: Use (1)–(2) as before, 'reduce' the $\mathbf{A}_i/\nu_i$'s to supernilpotent Maltsev algebras, and apply

- $\mathrm{SMP}(\mathbf{M}) \in \mathsf{P}$ if $\mathbf{M}$ is a nilpotent Maltsev algebra of prime power order in a finite language. [Mayr, 2012]

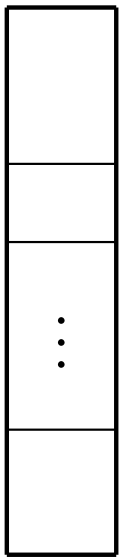
# The Reduction: Algebras from Congruences

$$\mathbf{A}, \alpha$$
$$(|\mathbf{A}/\alpha| = n)$$

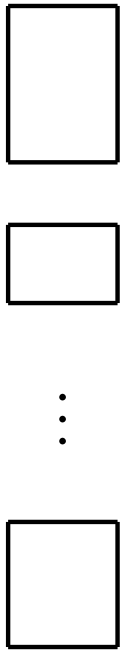


# The Reduction: Algebras from Congruences

$\mathbf{A}, \alpha$
$(|\mathbf{A}/\alpha| = n)$



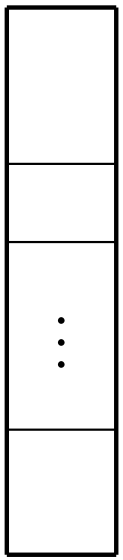
multisorted alg
$\mathrm{MS}(\mathbf{A}, \alpha)$



# The Reduction: Algebras from Congruences

$\mathbf{A}, \alpha$

$(|\mathbf{A}/\alpha| = n)$



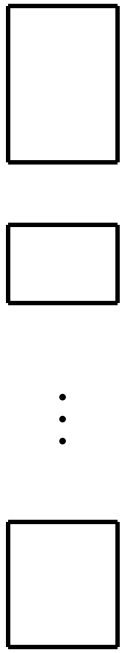
$f$

(say, binary)

operations

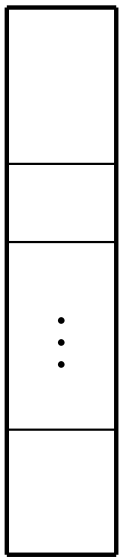
multisorted alg

$\mathrm{MS}(\mathbf{A}, \alpha)$



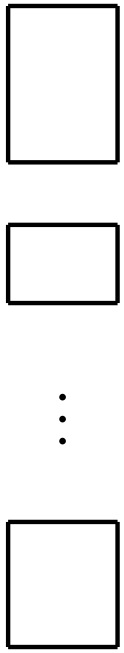
# The Reduction: Algebras from Congruences

$\mathbf{A}, \alpha$
$(|\mathbf{A}/\alpha| = n)$



$f$
(say, binary)

multisorted alg
$\mathrm{MS}(\mathbf{A}, \alpha)$



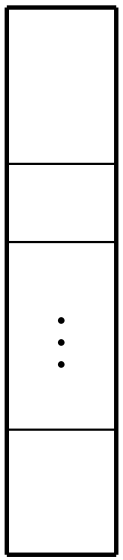
$f_{B,C;D}$
($B$, $C$, $D$ $\alpha$-classes,
$D \supseteq f[B, C]$)

operations

# The Reduction: Algebras from Congruences

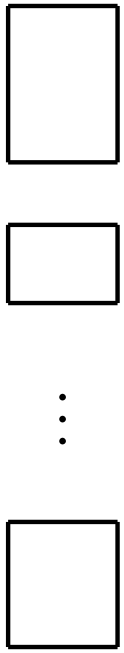
operations

$\mathbf{A}, \alpha$
$(|\mathbf{A}/\alpha| = n)$



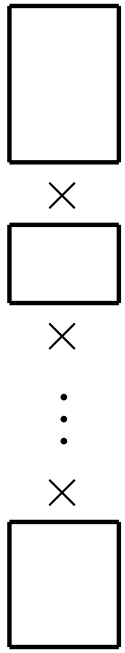
$f$
(say, binary)

multisorted alg
$\mathrm{MS}(\mathbf{A}, \alpha)$



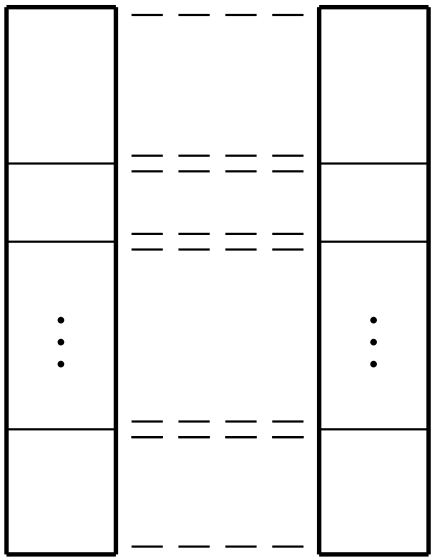
$f_{B,C;D}$
($B$, $C$, $D$ $\alpha$-classes,
$D \supseteq f[B, C]$)

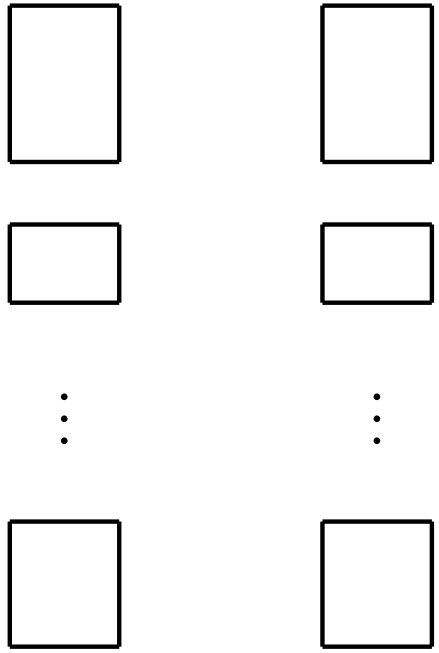
homogenization
$\mathrm{OS}(\mathbf{A}, \alpha)$



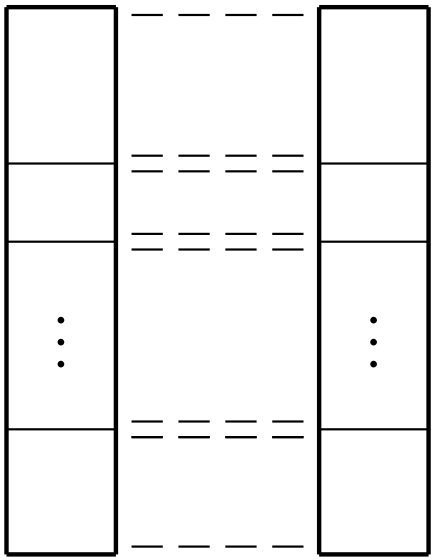
$d$ ($n$-ary diag op),
$\hat{f}_{B,C;D}$
$=$ (proj's, $f_{B,C;D}$, proj's)
$\uparrow_D$

# Subpowers

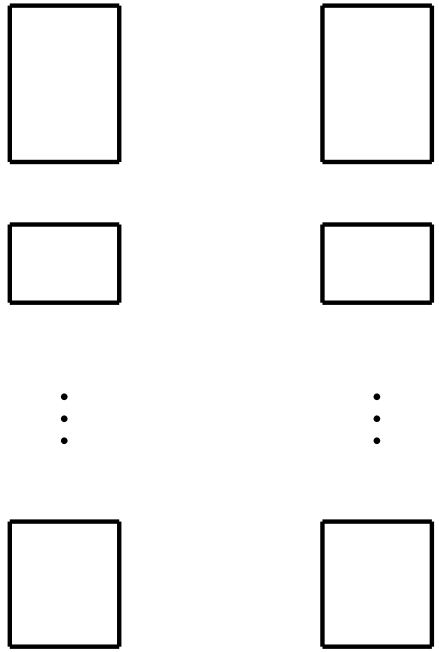




# Subpowers

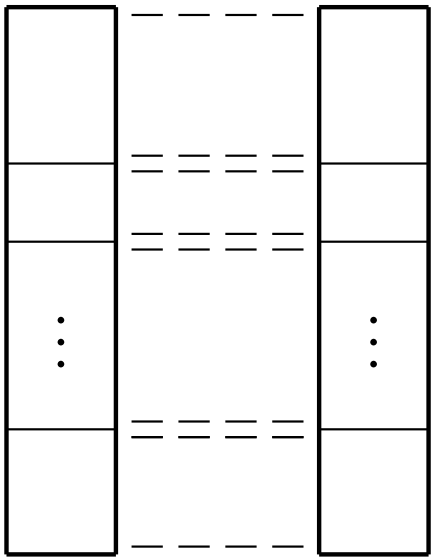


$\mathbf{A}^m[\alpha]$

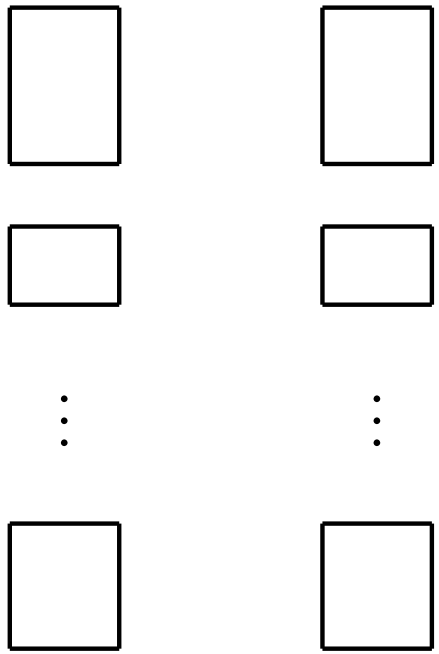


$\mathbf{MS}(\mathbf{A}, \alpha)^m$

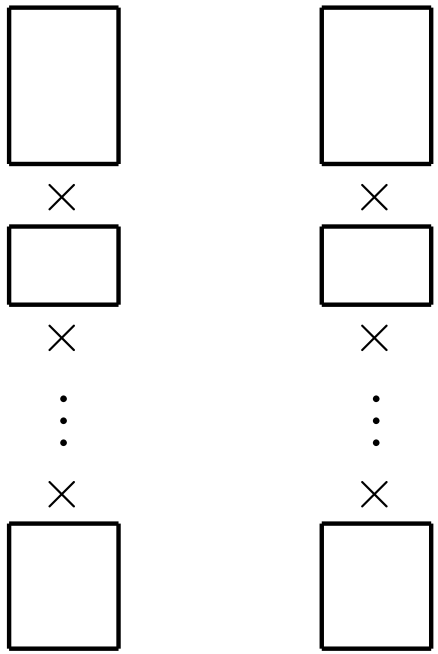
# Subpowers



$\mathbf{A}^m[\alpha]$

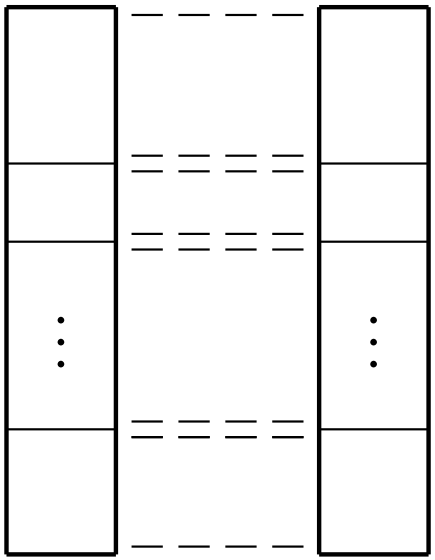


$\mathbf{MS}(\mathbf{A},\alpha)^m$

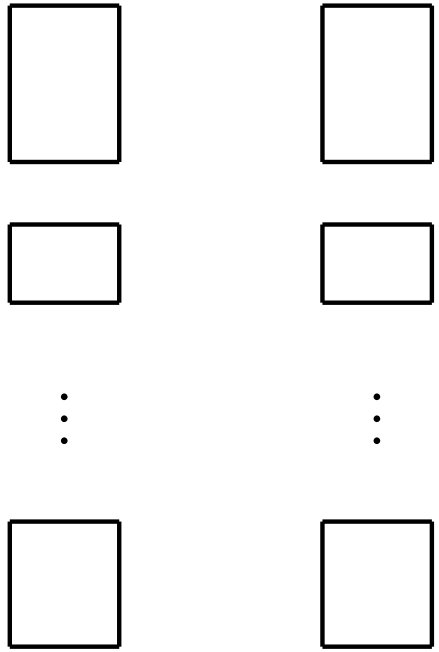


$\mathbf{OS}(\mathbf{A},\alpha)^m$

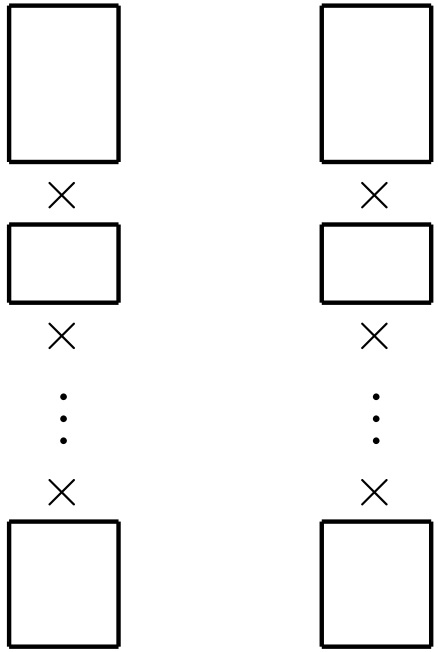
# Subpowers



$\mathbf{A}^m[\alpha]$



$\mathrm{MS}(\mathbf{A}, \alpha)^m$



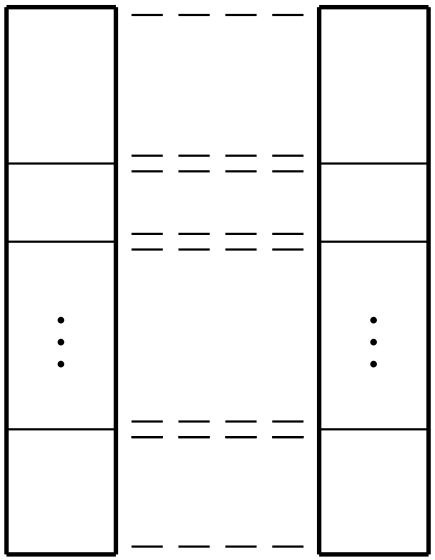
$\mathrm{OS}(\mathbf{A}, \alpha)^m$

subalgebras

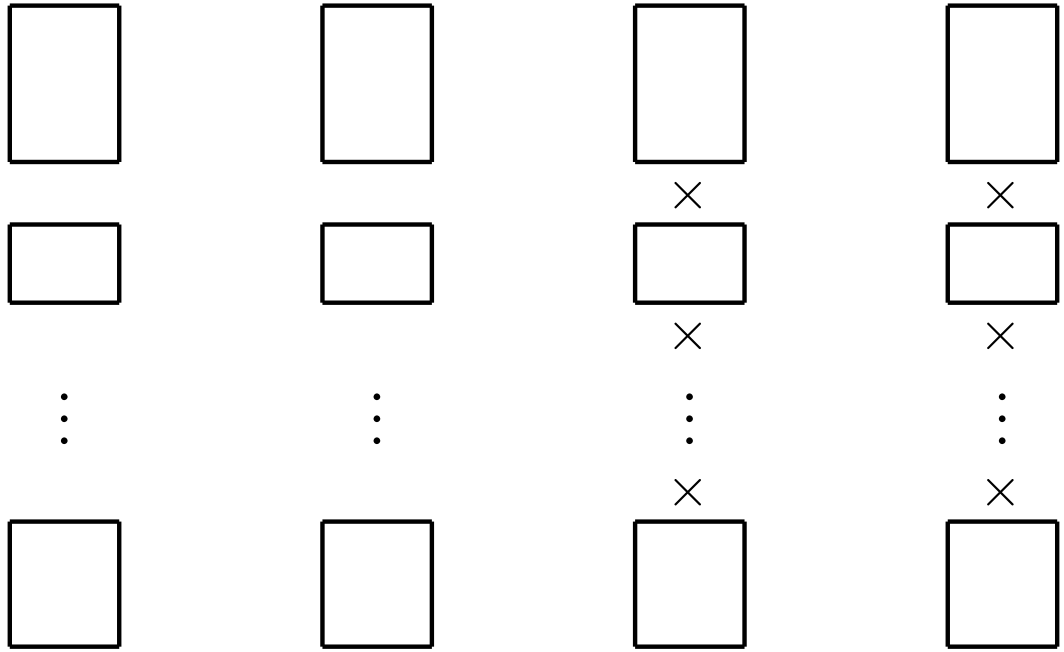
$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B}, \alpha^m)$

($\mathbf{B}$ split into $n$ $\alpha^m$-classes)

# Subpowers



$\mathbf{A}^m[\alpha]$



$\mathrm{MS}(\mathbf{A},\alpha)^m$



$\mathrm{OS}(\mathbf{A},\alpha)^m$

subalgebras

$\mathbf{B}$ $\longleftrightarrow$ $\mathrm{MS}(\mathbf{B},\alpha^m)$ $\xleftarrow{\text{if } \mathbf{B} \text{ meets all } n \ \alpha^m\text{-classes}}$ $\mathrm{OS}(\mathbf{B},\alpha^m)$

($\mathbf{B}$ split into $n$ $\alpha^m$-classes)

# Congruences

$$\mathbf{A}^m[\alpha] \qquad \mathbf{MS}(\mathbf{A},\alpha)^m \qquad \mathbf{OS}(\mathbf{A},\alpha)^m$$

$$\mathbf{B} \longleftrightarrow \mathbf{MS}(\mathbf{B},\alpha^m) \xleftrightarrow[\alpha^m\text{-classes}]{\text{if } \mathbf{B} \text{ meets all } n} \mathbf{OS}(\mathbf{B},\alpha^m)$$

# Congruences

$$\mathbf{A}^m[\alpha] \qquad \mathbf{MS}(\mathbf{A},\alpha)^m \qquad \mathbf{OS}(\mathbf{A},\alpha)^m$$

$$\mathbf{B} \longleftrightarrow \mathbf{MS}(\mathbf{B},\alpha^m) \xleftrightarrow{\text{if } \mathbf{B} \text{ meets all } n \ \alpha^m\text{-classes}} \mathbf{OS}(\mathbf{B},\alpha^m)$$

---

$$\{\gamma \in \mathrm{Con}(\mathbf{A}) : \gamma \leq \alpha\} \qquad \mathrm{Con}\big(\mathbf{MS}(\mathbf{A},\alpha)\big) \qquad \mathrm{Con}\big(\mathbf{OS}(\mathbf{A},\alpha)\big)$$

$$\gamma \longleftrightarrow \gamma \text{ (split into sorts)} \longleftrightarrow \begin{bmatrix} \gamma \\ \vdots \\ \gamma \end{bmatrix}$$

# Congruences

$$\mathbf{A}^m[\alpha] \qquad \mathrm{MS}(\mathbf{A},\alpha)^m \qquad \mathrm{OS}(\mathbf{A},\alpha)^m$$

$$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B},\alpha^m) \xleftrightarrow[\alpha^m\text{-classes}]{\text{if } \mathbf{B} \text{ meets all } n} \mathrm{OS}(\mathbf{B},\alpha^m)$$

---

$$\{\gamma \in \mathrm{Con}(\mathbf{A}) : \gamma \leq \alpha\} \qquad \mathrm{Con}\big(\mathrm{MS}(\mathbf{A},\alpha)\big) \qquad \mathrm{Con}\big(\mathrm{OS}(\mathbf{A},\alpha)\big)$$

$$\gamma \longleftrightarrow \gamma \text{ (split into sorts)} \longleftrightarrow \begin{bmatrix}\gamma\\ \vdots \\ \gamma\end{bmatrix}$$

in particular: $\alpha \longleftrightarrow 1 \longleftrightarrow 1$

# Congruences

$$\mathbf{A}^m[\alpha] \qquad \mathrm{MS}(\mathbf{A},\alpha)^m \qquad \mathrm{OS}(\mathbf{A},\alpha)^m$$

$$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B},\alpha^m) \xleftrightarrow{\text{if } \mathbf{B} \text{ meets all } n \ \alpha^m\text{-classes}} \mathrm{OS}(\mathbf{B},\alpha^m)$$

---

$$\{\gamma \in \mathrm{Con}(\mathbf{A}) : \gamma \leq \alpha\} \qquad \mathrm{Con}\big(\mathrm{MS}(\mathbf{A},\alpha)\big) \qquad \mathrm{Con}\big(\mathrm{OS}(\mathbf{A},\alpha)\big)$$

$$\gamma \longleftrightarrow \gamma \text{ (split into sorts)} \longleftrightarrow \begin{bmatrix}\gamma\\ \vdots \\ \gamma\end{bmatrix}$$

in particular: $\alpha \longleftrightarrow 1 \longleftrightarrow 1$

centrality: $[\alpha,\alpha] = 0 \longleftrightarrow [1,1] = 0$

# Congruences

$$\mathbf{A}^m[\alpha] \qquad \mathrm{MS}(\mathbf{A},\alpha)^m \qquad \mathrm{OS}(\mathbf{A},\alpha)^m$$

$$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B},\alpha^m) \xleftrightarrow[\alpha^m\text{-classes}]{\text{if } \mathbf{B} \text{ meets all } n} \mathrm{OS}(\mathbf{B},\alpha^m)$$

---

$$\{\gamma \in \mathrm{Con}(\mathbf{A}) : \gamma \leq \alpha\} \qquad \mathrm{Con}\big(\mathrm{MS}(\mathbf{A},\alpha)\big) \qquad \mathrm{Con}\big(\mathrm{OS}(\mathbf{A},\alpha)\big)$$

$$\gamma \longleftrightarrow \gamma \text{ (split into sorts)} \longleftrightarrow \begin{bmatrix}\gamma\\ \vdots \\ \gamma\end{bmatrix}$$

in particular: $\alpha \longleftrightarrow 1 \longleftrightarrow 1$

centrality: $[\alpha,\alpha] = 0 \longleftrightarrow [1,1] = 0$

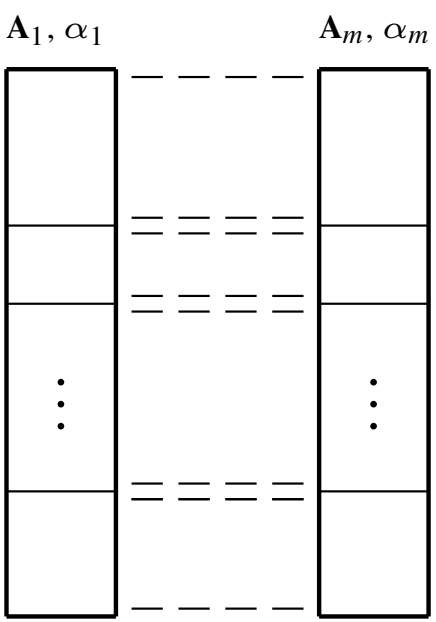
$$[\underbrace{\alpha,\ldots,\alpha}_{k}] = 0 \longleftrightarrow [\underbrace{1,\ldots,1}_{k}] = 0$$

# Subpowers Generalized

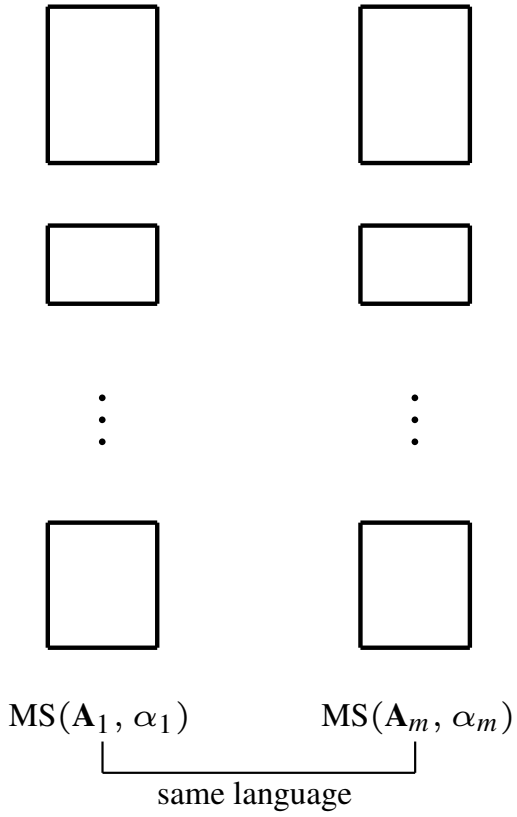
Assume: $\mathbf{A}_1/\alpha_1 \cong \ldots \cong \mathbf{A}_m/\alpha_m$; fix iso's: $\mathbf{A}_1/\alpha_1 \xrightarrow{f_i} \mathbf{A}_i/\alpha_i$ $(f_1 = \mathrm{id})$

# Subpowers Generalized

Assume: $\mathbf{A}_1/\alpha_1 \cong \ldots \cong \mathbf{A}_m/\alpha_m$; fix iso's: $\mathbf{A}_1/\alpha_1 \xrightarrow{f_i} \mathbf{A}_i/\alpha_i$ $(f_1 = \mathrm{id})$

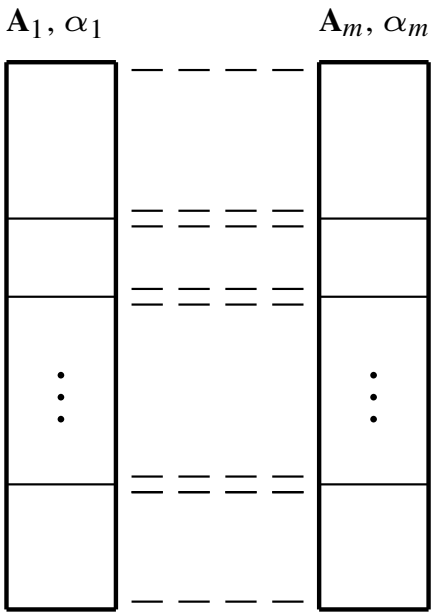



$\{(a_1, \ldots, a_m) : a_i \, \alpha_i f_i(a_1)\}$

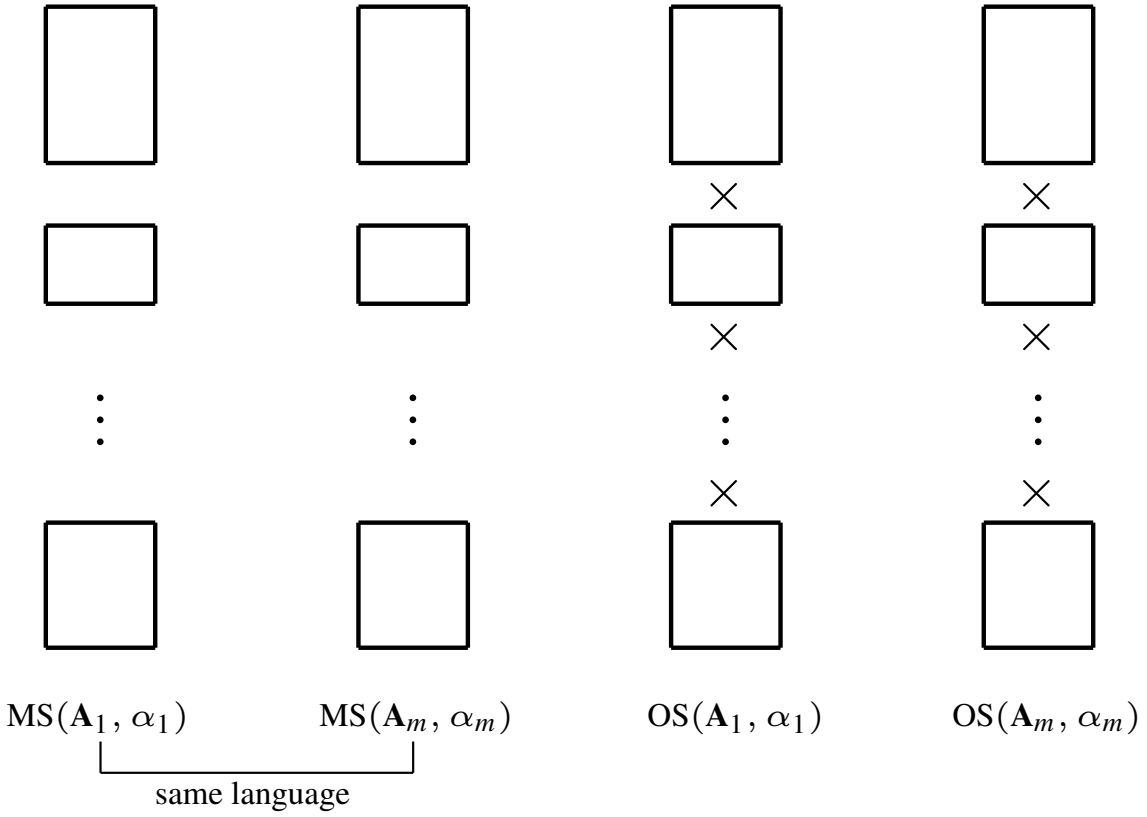


# Subpowers Generalized

Assume: $\mathbf{A}_1/\alpha_1 \cong \ldots \cong \mathbf{A}_m/\alpha_m$; fix iso's: $\mathbf{A}_1/\alpha_1 \xrightarrow{f_i} \mathbf{A}_i/\alpha_i$ $(f_1 = \mathrm{id})$

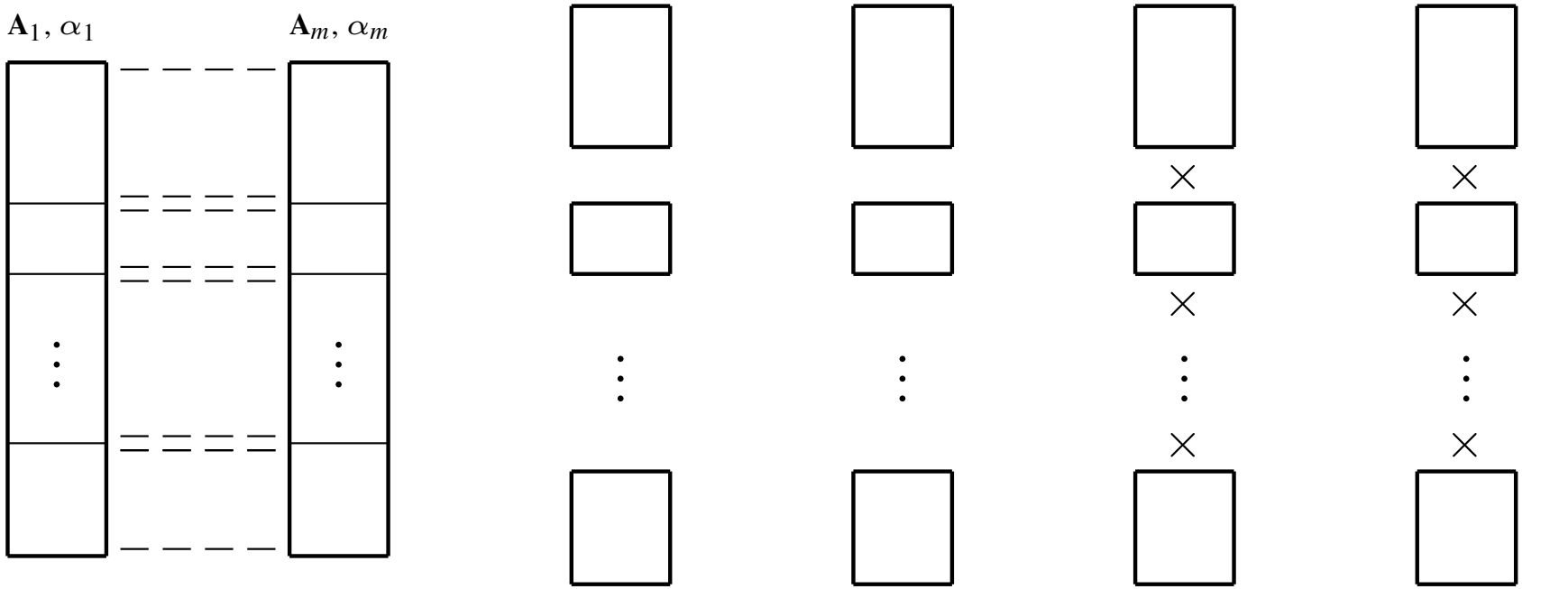



$\{(a_1, \ldots, a_m) : a_i \, \alpha_i f_i(a_1)\}$



# Subpowers Generalized

Assume: $\mathbf{A}_1/\alpha_1 \cong \ldots \cong \mathbf{A}_m/\alpha_m$; fix iso's: $\mathbf{A}_1/\alpha_1 \xrightarrow{f_i} \mathbf{A}_i/\alpha_i$ $(f_1 = \mathrm{id})$

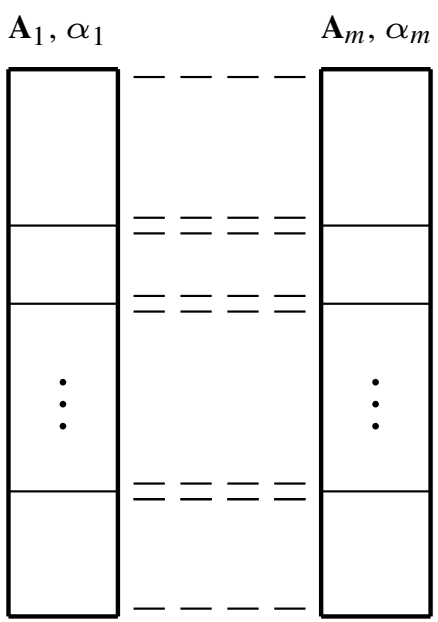


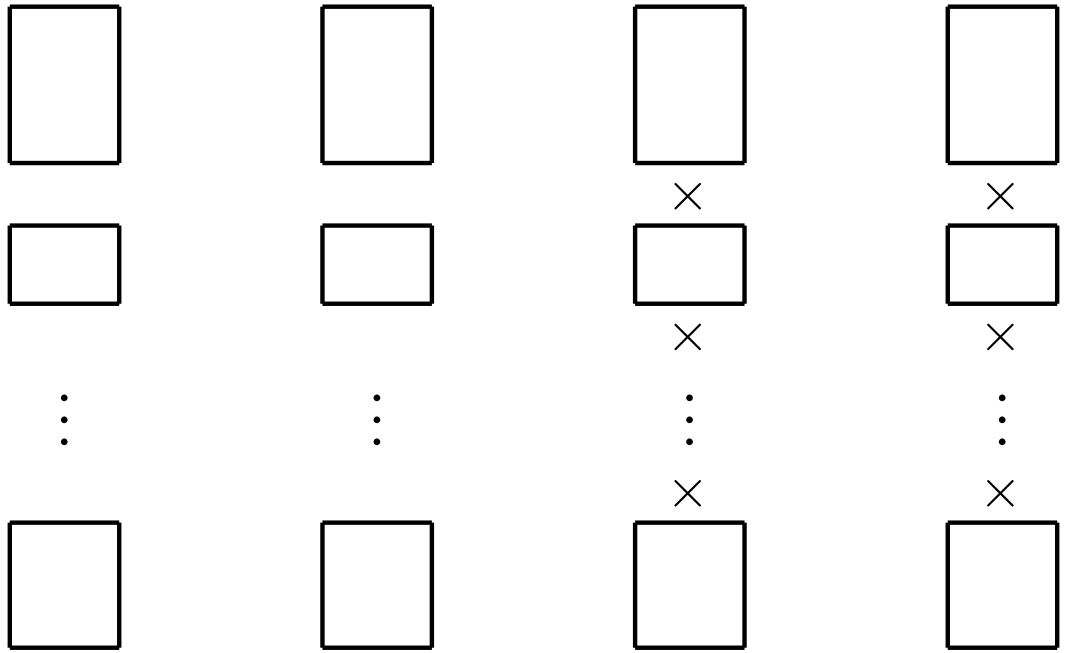

subalgs

$$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B}, \Pi\alpha_i)$$

# Subpowers Generalized

Assume: $\mathbf{A}_1/\alpha_1 \cong \ldots \cong \mathbf{A}_m/\alpha_m$; fix iso's: $\mathbf{A}_1/\alpha_1 \xrightarrow{f_i} \mathbf{A}_i/\alpha_i$ $(f_1 = \mathrm{id})$



$\{(a_1, \ldots, a_m) : a_i \, \alpha_i f_i(a_1)\}$

$\mathrm{MS}(\mathbf{A}_1, \alpha_1)$ $\mathrm{MS}(\mathbf{A}_m, \alpha_m)$ $\mathrm{OS}(\mathbf{A}_1, \alpha_1)$ $\mathrm{OS}(\mathbf{A}_m, \alpha_m)$

same language

subalgs

$$\mathbf{B} \longleftrightarrow \mathrm{MS}(\mathbf{B}, \Pi\alpha_i) \xleftarrow{\text{if } \mathbf{B} \text{ meets all } n \text{ } \Pi\alpha_i\text{-classes}} \mathrm{OS}(\mathbf{B}, \Pi\alpha_i)$$

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0 : \mu)$ is supernilpotent.*

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

Replace $\mathbf{B}, \mathbf{B}' \leq \prod_i \mathbf{A}_i$ by $\mathrm{OS}(\mathbf{B}, \prod \nu_i), \mathrm{OS}(\mathbf{B}', \prod \nu_i) \leq \prod_i \mathrm{OS}(\mathbf{A}_i, \nu_i)$, resp.

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

Replace $\mathbf{B}, \mathbf{B}' \leq \prod_i \mathbf{A}_i$ by $\mathrm{OS}(\mathbf{B}, \prod \nu_i), \mathrm{OS}(\mathbf{B}', \prod \nu_i) \leq \prod_i \mathrm{OS}(\mathbf{A}_i, \nu_i)$, resp.

- $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ can be computed in constant time. Generators for $\mathrm{OS}(\mathbf{B}, \prod \nu_i)$ and $\mathrm{OS}(\mathbf{B}', \prod \nu_i)$ can be computed from those of $\mathbf{B}, \mathbf{B}'$ in polynomial time.

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided*
$(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

Replace $\mathbf{B}, \mathbf{B}' \leq \prod_i \mathbf{A}_i$ by $\mathrm{OS}(\mathbf{B}, \prod \nu_i), \mathrm{OS}(\mathbf{B}', \prod \nu_i) \leq \prod_i \mathrm{OS}(\mathbf{A}_i, \nu_i)$, resp.

- $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ can be computed in constant time. Generators for $\mathrm{OS}(\mathbf{B}, \prod \nu_i)$ and $\mathrm{OS}(\mathbf{B}', \prod \nu_i)$ can be computed from those of $\mathbf{B}, \mathbf{B}'$ in polynomial time.
- This yields an equivalent instance of SMP, but now:

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided* $(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

Replace $\mathbf{B}, \mathbf{B}' \leq \prod_i \mathbf{A}_i$ by $\mathrm{OS}(\mathbf{B}, \prod \nu_i)$, $\mathrm{OS}(\mathbf{B}', \prod \nu_i) \leq \prod_i \mathrm{OS}(\mathbf{A}_i, \nu_i)$, resp.

- $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ can be computed in constant time. Generators for $\mathrm{OS}(\mathbf{B}, \prod \nu_i)$ and $\mathrm{OS}(\mathbf{B}', \prod \nu_i)$ can be computed from those of $\mathbf{B}$, $\mathbf{B}'$ in polynomial time.
- This yields an equivalent instance of SMP, but now:
  - assumption $(*)$ & $\mathcal{V}$ CM $\Rightarrow$ the coordinate algebras $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ are supernilpotent & lie in a CP variety; hence, they are direct products of nilpotent algebras of prime power order.

# Proof of Theorem 2

**Theorem 2.** [Mayr–ASz] *If $\mathcal{V}$ has a cube term, then* $\mathrm{SMP}(\mathcal{K}) \in \mathsf{P}$ *provided* $(*)$ *for every SI in $\mathbb{HS}\,\mathcal{K}$ with abelian monolith $\mu$, $(0:\mu)$ is supernilpotent.*

---

Consider a special input $a_1, \dots, a_n, b \in \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$ ($\mathbf{A}_1, \dots, \mathbf{A}_m \in \mathcal{K}$); i.e.,

- $\mathbf{B} = \langle a_1, \dots, a_n \rangle$, $\mathbf{B}' = \langle a_1, \dots, a_n, b \rangle \leq_{\mathrm{sd}} \mathbf{A}_1 \times \cdots \times \mathbf{A}_m$;
- each $\mathbf{A}_i$ is SI with abelian monolith $\mu_i$; let $\nu_i = (0:\mu_i)$;
- $\mathbf{B}/\prod \nu_i = \mathbf{B}'/\prod \nu_i$ is a 'string of iso's' $\mathbf{A}_1/\nu_1 \cong \ldots \cong \mathbf{A}_m/\nu_m$.

Replace $\mathbf{B}, \mathbf{B}' \leq \prod_i \mathbf{A}_i$ by $\mathrm{OS}(\mathbf{B}, \prod \nu_i), \mathrm{OS}(\mathbf{B}', \prod \nu_i) \leq \prod_i \mathrm{OS}(\mathbf{A}_i, \nu_i)$, resp.

- $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ can be computed in constant time. Generators for $\mathrm{OS}(\mathbf{B}, \prod \nu_i)$ and $\mathrm{OS}(\mathbf{B}', \prod \nu_i)$ can be computed from those of $\mathbf{B}, \mathbf{B}'$ in polynomial time.
- This yields an equivalent instance of SMP, but now:
  - assumption $(*)$ & $\mathcal{V}$ CM $\Rightarrow$ the coordinate algebras $\mathrm{OS}(\mathbf{A}_i, \nu_i)$ are supernilpotent & lie in a CP variety; hence, they are direct products of nilpotent algebras of prime power order.
- Therefore, P. Mayr's result (combined with (1)) yields a P-time algorithm.