Monadic
Monadic Second Order Logic

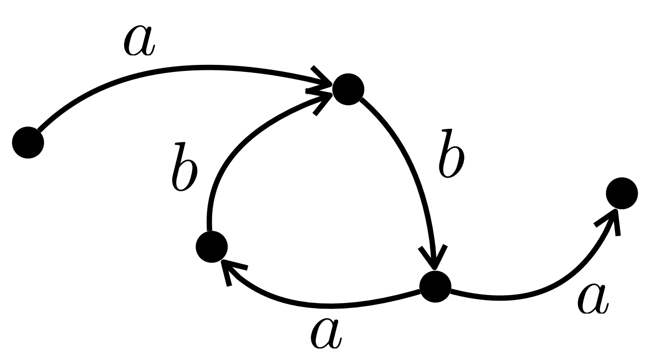
Mikołaj Bojańczyk, Bartek Klin, Julian Salamanca

University of Warsaw

13 May 2020

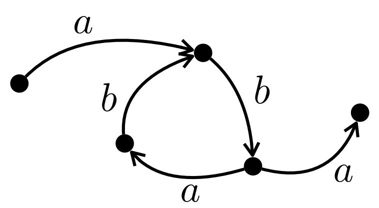
Languages of finite words

accepted by finite automata



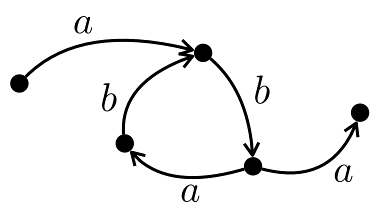
Languages of finite words

accepted by finite automata



Languages of finite words

accepted by finite automata

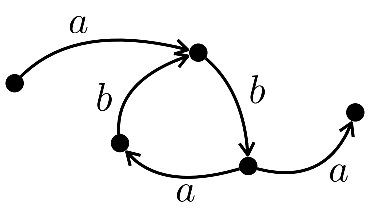


defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

Languages of finite words

accepted by finite automata

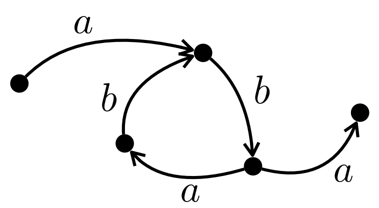


defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

Languages of finite words

accepted by finite automata



defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

MSO-definable

$$x < y \qquad Q_a(x) \qquad x \in X$$

$$\phi \vee \psi \qquad \neg\phi \qquad \exists X.\phi$$

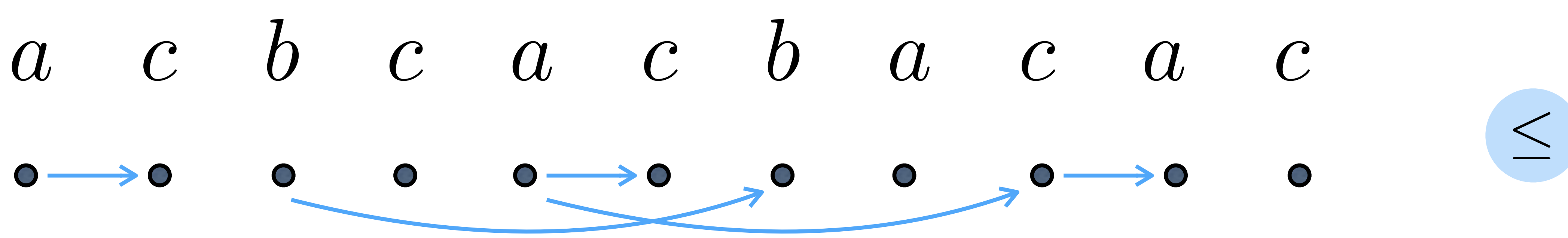
Monadic Second-Order Logic

- words as relational structures:

$$a \quad c \quad b \quad c \quad a \quad c \quad b \quad a \quad c \quad a \quad c$$

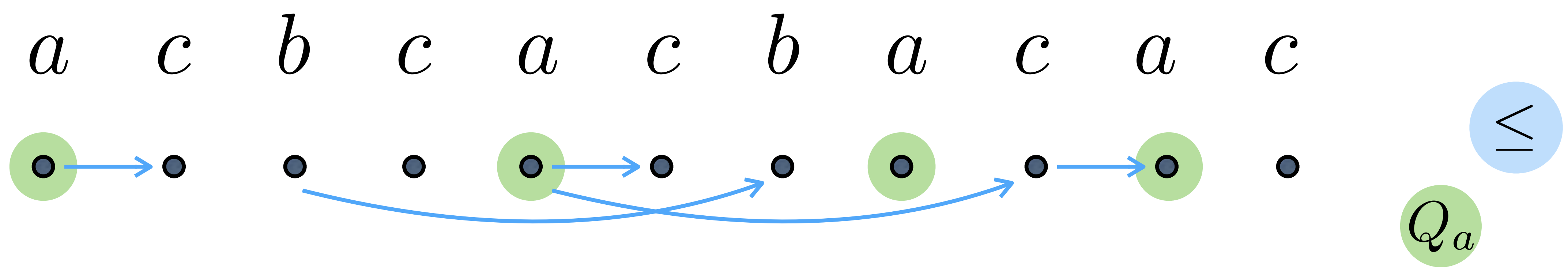
Monadic Second-Order Logic

- words as relational structures:



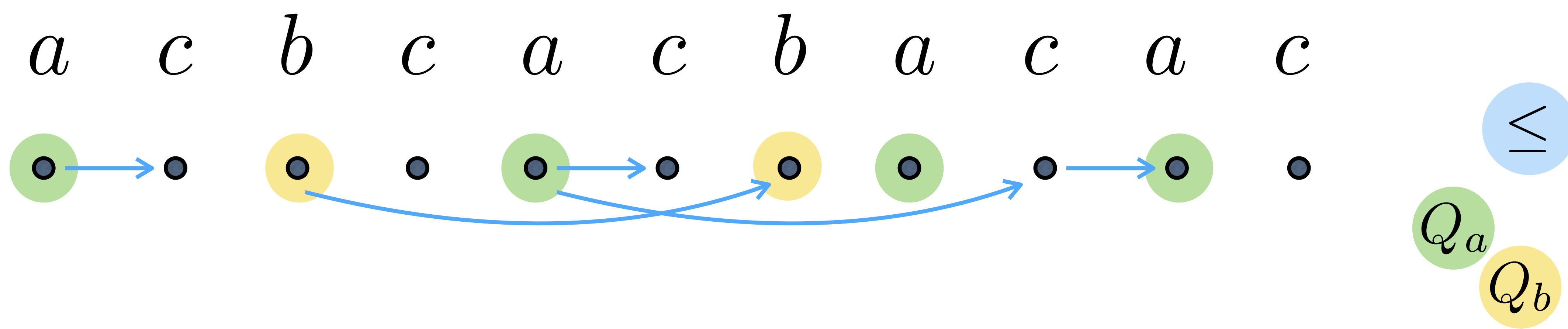
Monadic Second-Order Logic

- words as relational structures:



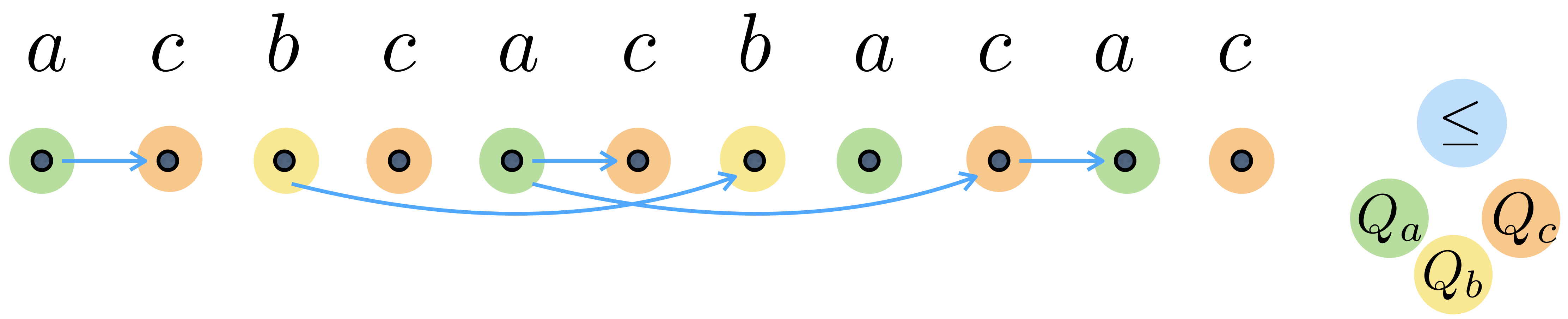
Monadic Second-Order Logic

- words as relational structures:



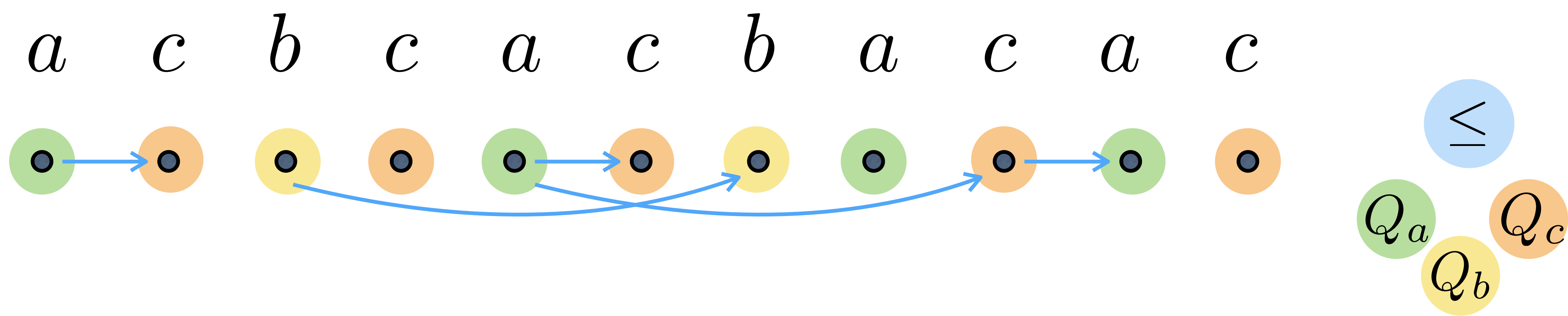
Monadic Second-Order Logic

- words as relational structures:



Monadic Second-Order Logic

- words as relational structures:

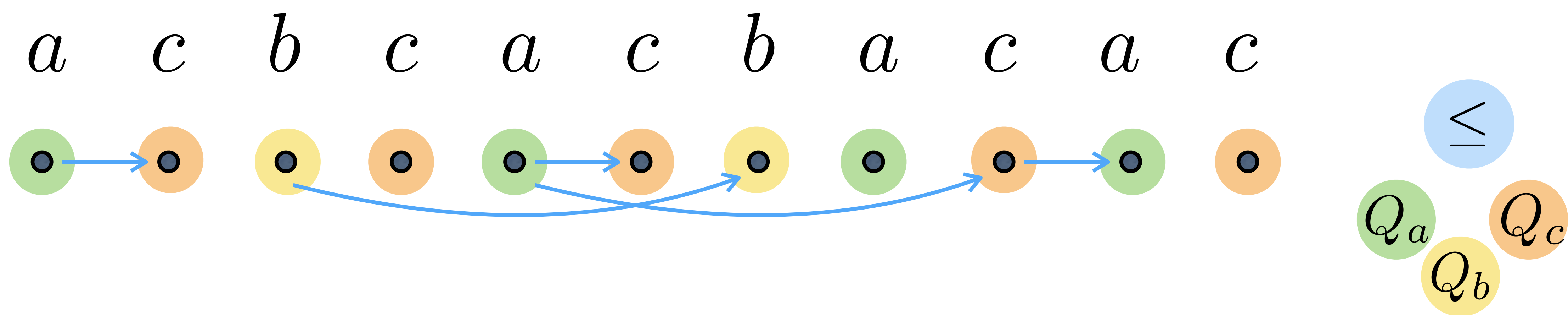


- examples:

$$\forall x. Q_a(x) \Rightarrow \exists y. x < y \wedge Q_c(y)$$

Monadic Second-Order Logic

- words as relational structures:



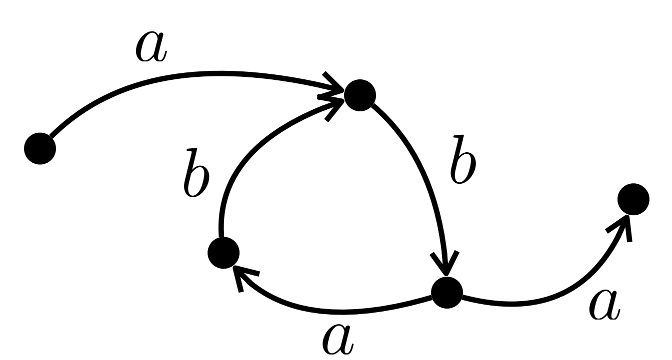
- examples:

$$\forall x. Q_a(x) \Rightarrow \exists y. x < y \wedge Q_c(y)$$

$$\begin{aligned}\exists X.(&\forall x\ \exists y\ y \leq x \wedge y \in X) \wedge \\ (&\forall x\ \exists y\ y \geq x \wedge y \in X) \wedge \\ (&\forall x\ \forall y\ (x < y \wedge \neg(\exists z\ x < z < y)) \Rightarrow (x \in X \Leftrightarrow y \notin X)).\end{aligned}$$

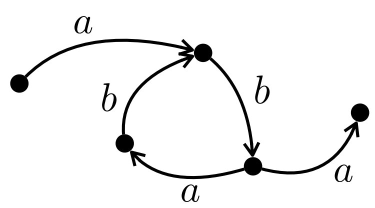
Languages of finite words

accepted by finite automata



Languages of finite words

accepted by finite automata



defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

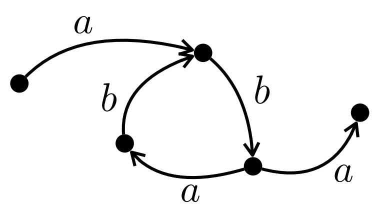
MSO-definable

$x < y$ $Q_a(x)$ $x \in X$

$\phi \vee \psi$ $\neg\phi$ $\exists X.\phi$

Languages of finite words

accepted by finite automata



defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

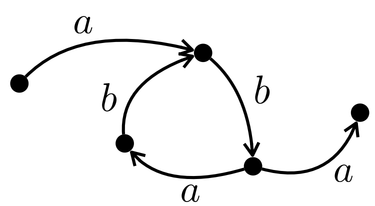
MSO-definable

$x < y \quad Q_a(x) \quad x \in X$

$\phi \vee \psi \quad \neg\phi \quad \exists X.\phi$

Languages of finite words

accepted by finite automata



defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

MSO-definable

$x < y \quad Q_a(x) \quad x \in X$

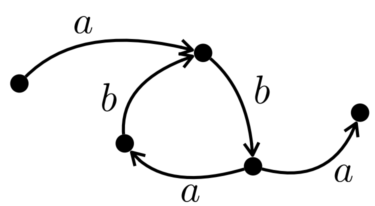
$\phi \vee \psi \quad \neg\phi \quad \exists X.\phi$

recognized by finite monoids

$$\begin{array}{ccccc} \overleftarrow{h}(A) = & L & & & A \\ & \cap\!\!\!\mid & & & \cap\!\!\!\mid \\ & \Sigma^* & \xrightarrow[h]{\quad\quad} & & M \end{array}$$

Languages of finite words

accepted by finite automata



defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

MSO-definable

$x < y$ $\quad Q_a(x)$ $\quad x \in X$

$\phi \vee \psi$ $\quad \neg\phi$ $\quad \exists X.\phi$

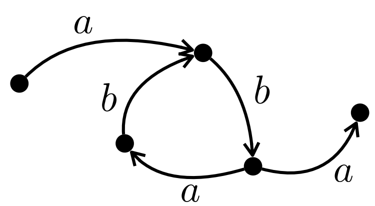
recognized by finite monoids

$$\overleftarrow{h}(A) = L \qquad A$$

$$\begin{array}{ccc} L & & A \\ \cap\!\!| & & \cap\!\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

Languages of finite words

accepted by finite automata



⟺

defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

⟺

MSO-definable

$$x < y \qquad Q_a(x) \qquad x \in X$$

$$\phi \vee \psi \qquad \neg\phi \qquad \exists X.\phi$$

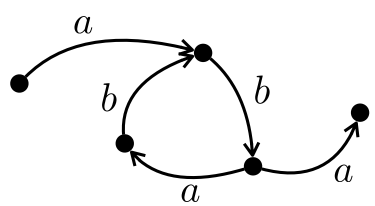
⟺

recognized by finite monoids

$$\overleftarrow{h}(A) = L \subseteq \Sigma^* \xrightarrow{h} M \supseteq A$$

Languages of finite words

accepted by finite automata



⇔

defined by regular expressions

$$E ::= \epsilon \mid a \mid E + E \mid EE \mid E^*$$

⇕

MSO-definable

$x < y \qquad Q_a(x) \qquad x \in X$

$\phi \vee \psi \qquad \neg\phi \qquad \exists X.\phi$

⇔

recognized by finite monoids

$$\overleftarrow{h}(A) = L \subseteq \Sigma^* \xrightarrow{h} M \supseteq A$$

Things in this talk

- finite words
- ω-words
- countable total orders
- scattered total orders
- total orders of size $\leq \mathfrak{c}$
- finite trees
- infinite trees
- graphs of bounded treewidth
- graphs of bounded cliquewidth
- ...
- ...

Our focus

MSO-definable

$x < y \qquad Q_a(x) \qquad x \in X$

$\phi \vee \psi \qquad \neg\phi \qquad \exists X.\phi$

$\Leftrightarrow$

recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \subseteq & & \subseteq \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

Our focus

MSO-definable

$x < y \quad Q_a(x) \quad x \in X$

$\phi \vee \psi \quad \neg\phi \quad \exists X.\phi$

$\Longleftrightarrow$

recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

$\Longleftarrow$

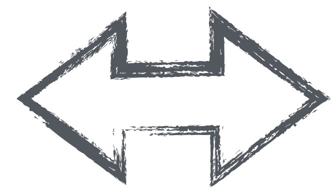
- quite easy for finite words or trees
- difficult (or open) for other structures
- structure-specific arguments

Our focus

MSO-definable

$x < y \qquad Q_a(x) \qquad x \in X$

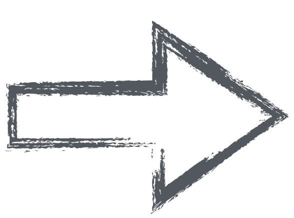
$\phi \vee \psi \qquad \neg\phi \qquad \exists X.\phi$



recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!\!| & & \cap\!\!| \\ \Sigma^* & \xrightarrow[h]{\qquad} & M \end{array}$$

- quite easy for finite words or trees
- difficult (or open) for other structures
- structure-specific arguments



- relatively easy for all cases
- the arguments look generic

Our focus

MSO-definable

$x < y \quad Q_a(x) \quad x \in X$

$\phi \vee \psi \quad \neg\phi \quad \exists X.\phi$

$\Leftrightarrow$

recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!\!| & & \cap\!\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

Our focus

MSO-definable

$x < y \quad Q_a(x) \quad x \in X$

$\phi \vee \psi \quad \neg\phi \quad \exists X.\phi$

⟺

recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

⇕

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

Our focus

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

$\Leftrightarrow$

recognized by finite monoids

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \rotatebox{90}{$\subseteq$} & & \rotatebox{90}{$\subseteq$} \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

Definable implies recognizable, for finite words

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

⇒

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- $0^*1^* \subseteq \{0,1\}^*$ recognized

Definable implies recognizable, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- $0^*1^* \subseteq \{0,1\}^*$ recognized
- L_i rec. by $h_i : \Sigma^* \to M_i$ (for $i = 1, 2$)
  implies $L_1 \cap L_2$ rec. by $\langle h_1, h_2 \rangle : \Sigma^* \to M_1 \times M_2$

Definable implies recognizable, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

recognized by a finite monoid

$$\overleftarrow{h}(A) = L \subseteq \Sigma^* \xrightarrow{h} M \supseteq A$$

- $0^*1^* \subseteq \{0,1\}^*$ recognized
- L_i rec. by $h_i : \Sigma^* \to M_i$ (for $i = 1, 2$)

  implies $L_1 \cap L_2$ rec. by $\langle h_1, h_2 \rangle : \Sigma^* \to M_1 \times M_2$

  $\Sigma^* \setminus L_i$ rec. by h_i

Definable implies recognizable, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

⇒

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- $0^*1^* \subseteq \{0,1\}^*$ recognized
- L_i rec. by $h_i : \Sigma^* \to M_i$ (for $i = 1, 2$)
  implies $L_1 \cap L_2$ rec. by $\langle h_1, h_2 \rangle : \Sigma^* \to M_1 \times M_2$
  $\Sigma^* \setminus L_i$ rec. by h_i
- L rec. by $h : \Gamma^* \to M$, $\quad g : \Sigma \to \Gamma^*$
  implies $\overleftarrow{g}(L)$ rec. by $h \circ \hat{g}$ $\quad \hat{g} : \Sigma^* \to \Gamma^*$

Closure under direct images, for finite words

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

$\Rightarrow$

recognized by a finite monoid

$$\overleftarrow{h}(A) = L \subseteq \Sigma^* \xrightarrow[h]{} M \supseteq A$$

Closure under direct images, for finite words

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

⇒

recognized by a finite monoid

$$\overleftarrow{h}(A) = L \quad L \subseteq \Sigma^* \xrightarrow{h} M \supseteq A$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$

Closure under direct images, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

recognized by a finite monoid

$$\overleftarrow{h}(A) = L \subseteq \Sigma^* \xrightarrow{h} M \supseteq A$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$
- take $g : \Sigma \to \Gamma$

Closure under direct images, for finite words

least class closed under:

- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!\!| & & \cap\!\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$
- take $g : \Sigma \to \Gamma$
- define a monoid on $\mathcal{P}M$:

$$S \cdot T = \{s \cdot t \mid s \in S, t \in T\}$$

Closure under direct images, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

⇒

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!| & & \cap\!| \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$
- take $g : \Sigma \to \Gamma$
- define a monoid on $\mathcal{P}M$:

$$S \cdot T = \{s \cdot t \mid s \in S, t \in T\}$$

- put $k : \Gamma^* \to \mathcal{P}M$ s.t. $k(c) = \{h(a) \mid g(a) = c\}$

Closure under direct images, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \cap\!\shortmid & & \cap\!\shortmid \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$
- take $g : \Sigma \to \Gamma$
- define a monoid on $\mathcal{P}M$:

$$S \cdot T = \{s \cdot t \mid s \in S, t \in T\}$$

- put $k : \Gamma^* \to \mathcal{P}M$ s.t. $k(c) = \{h(a) \mid g(a) = c\}$

  $B \subseteq \mathcal{P}M$ s.t. $B = \{S \mid S \cap A \neq \emptyset\}$

Closure under direct images, for finite words

least class closed under:
- $0^*1^* \subseteq \{0,1\}^*$
- boolean combinations
- inv. images along $h : \Sigma \to \Gamma^*$
- dir. images along $h : \Sigma \twoheadrightarrow \Gamma$

⇨

recognized by a finite monoid

$$\overleftarrow{h}(A) = \begin{array}{ccc} L & & A \\ \rotatebox{90}{$\subseteq$} & & \rotatebox{90}{$\subseteq$} \\ \Sigma^* & \xrightarrow[h]{} & M \end{array}$$

- let $L \subseteq \Sigma^*$ be recognized by $h : \Sigma^* \to M$
- take $g : \Sigma \to \Gamma$
- define a monoid on $\mathcal{P}M$:

$$S \cdot T = \{s \cdot t \mid s \in S, t \in T\}$$

- put $k : \Gamma^* \to \mathcal{P}M$ s.t. $k(c) = \{h(a) \mid g(a) = c\}$

  $B \subseteq \mathcal{P}M$ s.t. $B = \{S \mid S \cap A \neq \emptyset\}$
- then k and B recognize $g^*(L)$

Definable implies recognizable, for finite words

We have just shown:

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

Definable implies recognizable, for finite words

We have just shown:

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

We want to generalize this to other things.

Monads

Monads are *ways to collect stuff*

Monads are *ways to collect stuff*

A monad T:

- given a set X, returns a set TX

Monads

Monads are *ways to collect stuff*

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T:

- given a set X, returns a set TX

Monads

Monads are *ways to collect stuff*

A monad T:

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

- given a set X, returns a set TX
- given a function $f : X \to Y$, returns a function $Tf : TX \to TY$

Monads

Monads are *ways to collect stuff*

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T:

- given a set X, returns a set TX
- given a function $f : X \to Y$, returns a function $Tf : TX \to TY$

so that:

- $T(\mathrm{id}_X) = \mathrm{id}_{TX}$, and
- $T(g \circ f) = Tg \circ Tf$

Monads

Monads are *ways to collect stuff*

A monad T:

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

- given a set X, returns a set TX
- given a function $f : X \to Y$,
  returns a function $Tf : TX \to TY$

so that:

- $T(\mathrm{id}_X) = \mathrm{id}_{TX}$, and
- $T(g \circ f) = Tg \circ Tf$

functor

Monads

Monads are *ways to collect stuff*

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T:

- given a set X, returns a set TX
- given a function $f : X \to Y$,
  returns a function $Tf : TX \to TY$

so that:

- $T(\mathrm{id}_X) = \mathrm{id}_{TX}$, and
- $T(g \circ f) = Tg \circ Tf$

functor

to be ctd...

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$

unit

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$ unit
- $\mu_X : TTX \to TX$

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$ — unit
- $\mu_X : TTX \to TX$ — multiplication

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$ — unit
- $\mu_X : TTX \to TX$ — multiplication

such that (for every $f : X \to Y$):

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & TX \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Tf} \\ Y & \xrightarrow[\eta_Y]{} & TY \end{array} \quad \text{and} \quad \begin{array}{ccc} TTX & \xrightarrow{\mu_X} & TX \\ {\scriptstyle TTf}\downarrow & & \downarrow{\scriptstyle Tf} \\ TTY & \xrightarrow[\mu_Y]{} & TY \end{array}$$

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$ unit
- $\mu_X : TTX \to TX$ multiplication

such that (for every $f : X \to Y$):

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & TX \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Tf} \\ Y & \xrightarrow[\eta_Y]{} & TY \end{array} \quad \text{and} \quad \begin{array}{ccc} TTX & \xrightarrow{\mu_X} & TX \\ {\scriptstyle TTf}\downarrow & & \downarrow{\scriptstyle Tf} \\ TTY & \xrightarrow[\mu_Y]{} & TY \end{array}$$

naturality

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

A monad T comes with (for every set X):

- $\eta_X : X \to TX$ — unit
- $\mu_X : TTX \to TX$ — multiplication

such that (for every $f : X \to Y$):

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & TX \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Tf} \\ Y & \xrightarrow[\eta_Y]{} & TY \end{array} \quad \text{and} \quad \begin{array}{ccc} TTX & \xrightarrow{\mu_X} & TX \\ {\scriptstyle TTf}\downarrow & & \downarrow{\scriptstyle Tf} \\ TTY & \xrightarrow[\mu_Y]{} & TY \end{array}$$

naturality

to be ctd...

Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

Further axioms on $\eta_X : X \to TX$

$\mu_X : TTX \to TX$:

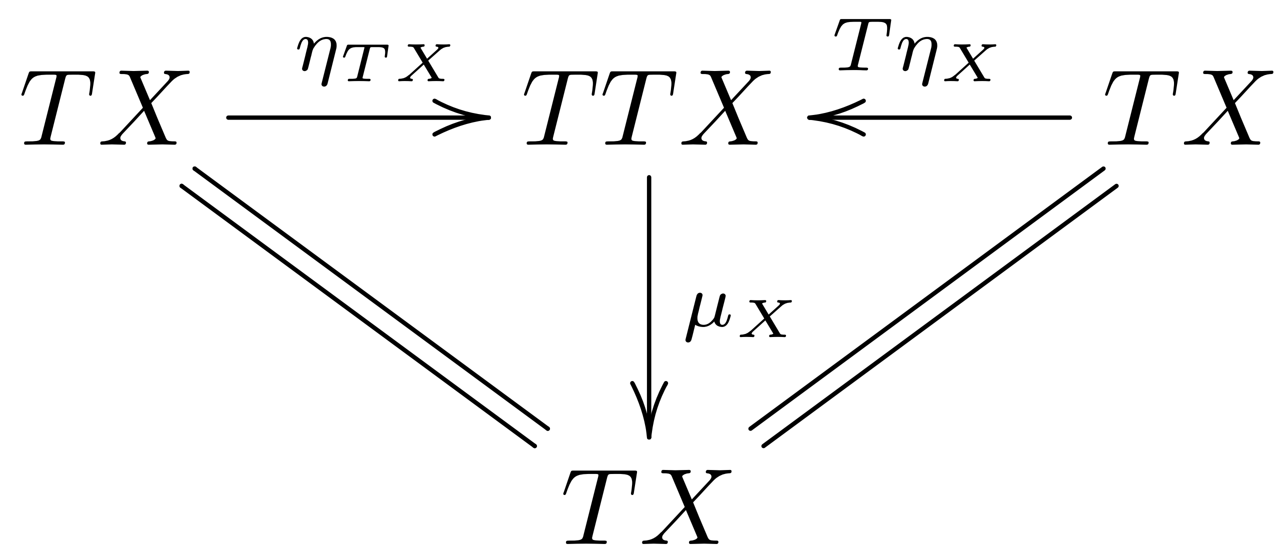
$$\begin{array}{ccccc} TX & \xrightarrow{\eta_{TX}} & TTX & \xleftarrow{T\eta_X} & TX \\ & \searrow^{=} & \downarrow{\scriptstyle \mu_X} & \swarrow_{=} & \\ & & TX & & \end{array}$$

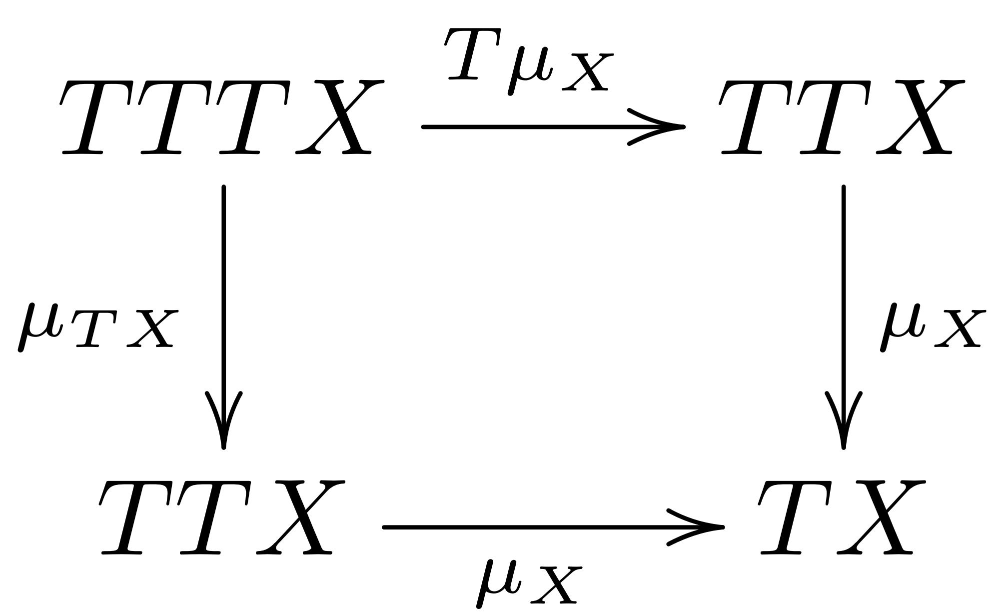
Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

Further axioms on $\eta_X : X \to TX$

$\mu_X : TTX \to TX$:



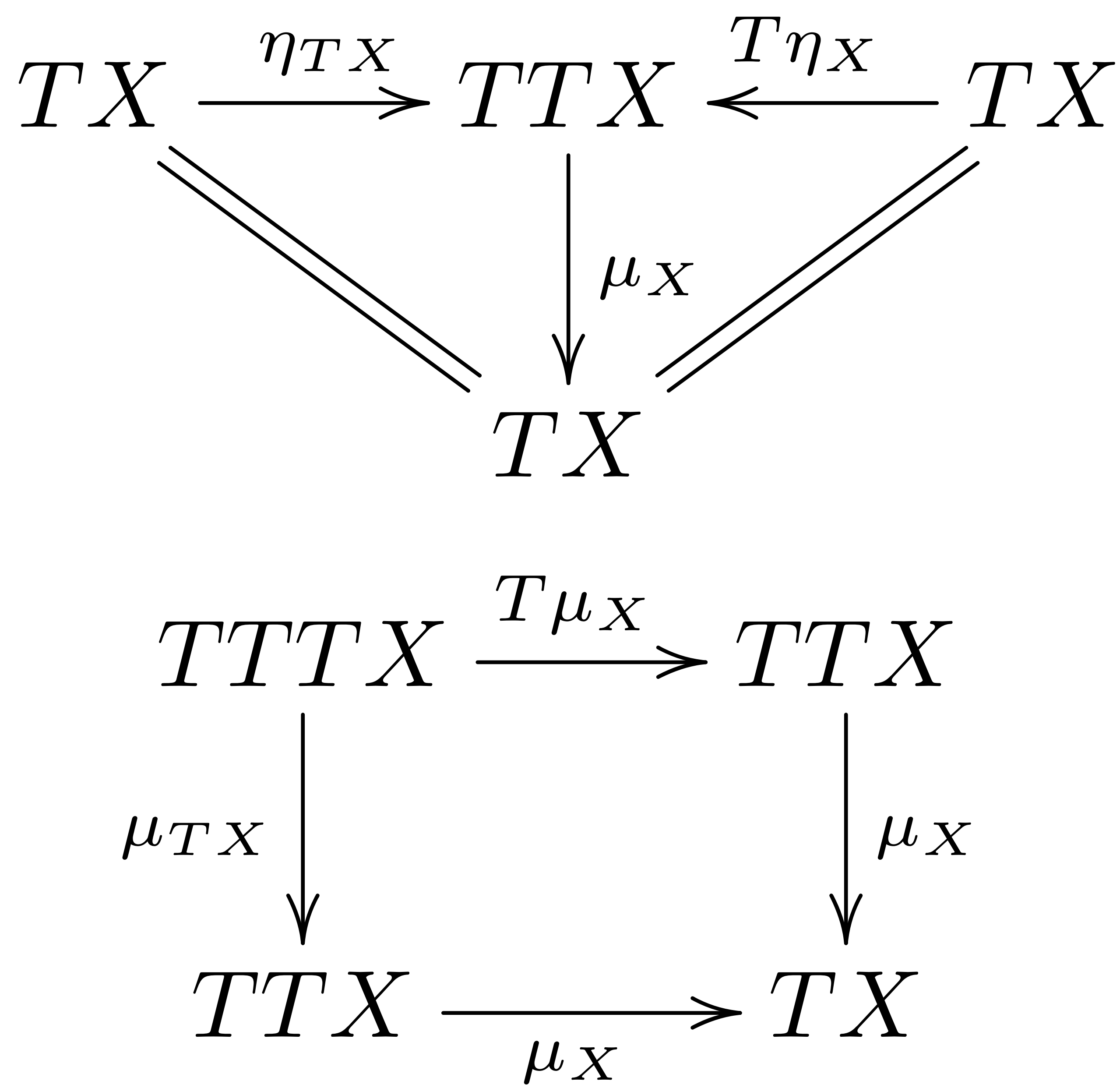


Monads ctd.

Examples: X^*, X^ω, X^∞, $\mathcal{P}X$, $\mathbb{N}^X$

Further axioms on $\eta_X : X \to TX$

$\mu_X : TTX \to TX$:



That's it!

Examples

1. The list monad

$$TX = X^*$$

$$Tf(x_1 \cdots x_n) = f(x_1) \cdots f(x_n)$$

$$\eta_X(x) = x$$

$$\mu_X(w_1 w_2 \cdots w_n) = w_1{}^\frown w_2{}^\frown \cdots {}^\frown w_n$$

Examples

1. The list monad

$$TX = X^*$$

$$Tf(x_1 \cdots x_n) = f(x_1) \cdots f(x_n)$$

$$\eta_X(x) = x$$

$$\mu_X(w_1 w_2 \cdots w_n) = w_1 ^\frown w_2 ^\frown \cdots ^\frown w_n$$

2. The powerset monad

$$TX = \mathcal{P}X$$

$$Tf = \overrightarrow{f}$$

$$\eta_X(x) = \{x\}$$

$$\mu_X(\Phi) = \bigcup \Phi$$

Examples ctd.

3. The reader monad

$$TX = X^\omega \qquad Tf(x_1x_2\cdots) = f(x_1)f(x_2)\cdots$$

$$\eta_X(x) = xxx\cdots \qquad \mu_X(w_1w_2\cdots) = w_{11}w_{22}w_{33}\cdots$$

Examples ctd.

3. The reader monad

$$TX = X^\omega \qquad Tf(x_1x_2\cdots) = f(x_1)f(x_2)\cdots$$

$$\eta_X(x) = xxx\cdots \qquad \mu_X(w_1w_2\cdots) = w_{11}w_{22}w_{33}\cdots$$

4,5,...: term monads

For an equational presentation (Σ, E) , put:

$TX =$ Σ -terms over X modulo the equations

Tf - variable substitution

η - variables as terms

μ - term flattening

What we want to talk about

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

$L \subseteq T\Sigma$

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective)
  letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite monoids
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite monoids ?
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

Algebras

A T-algebra is:

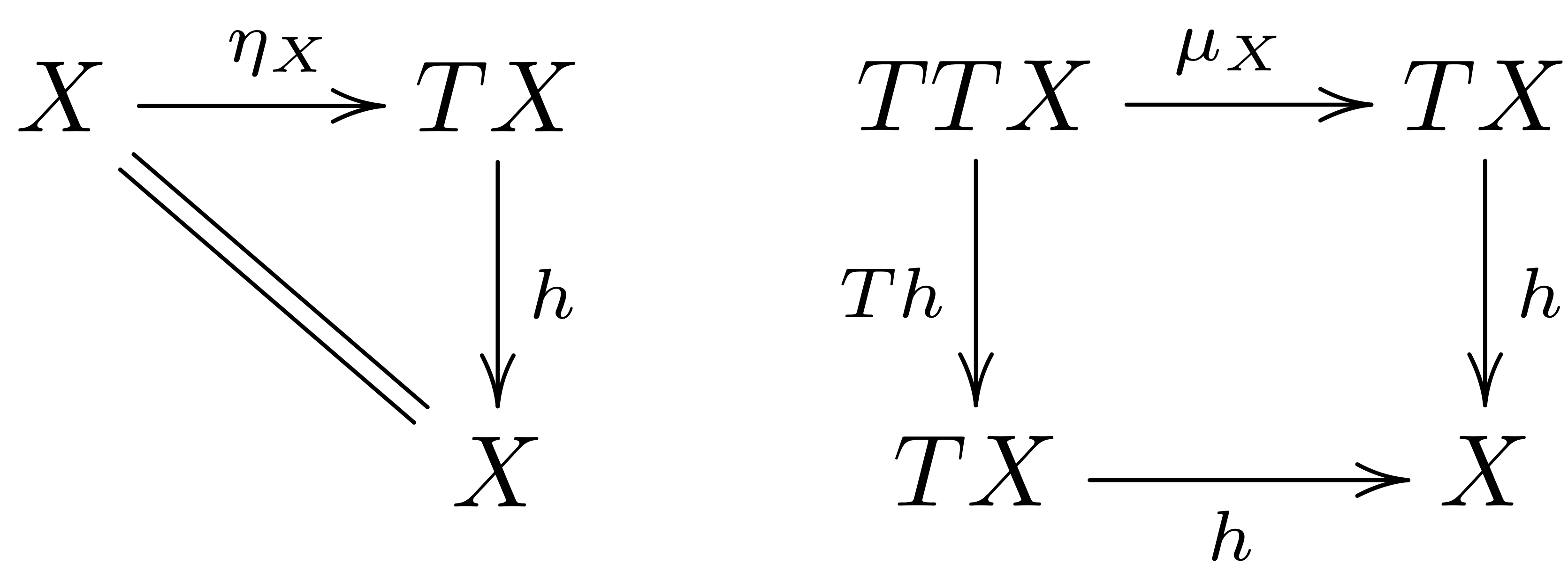
- a set X and a function $f : TX \to X$

Algebras

A T-algebra is:

- a set X and a function $f : TX \to X$

such that:

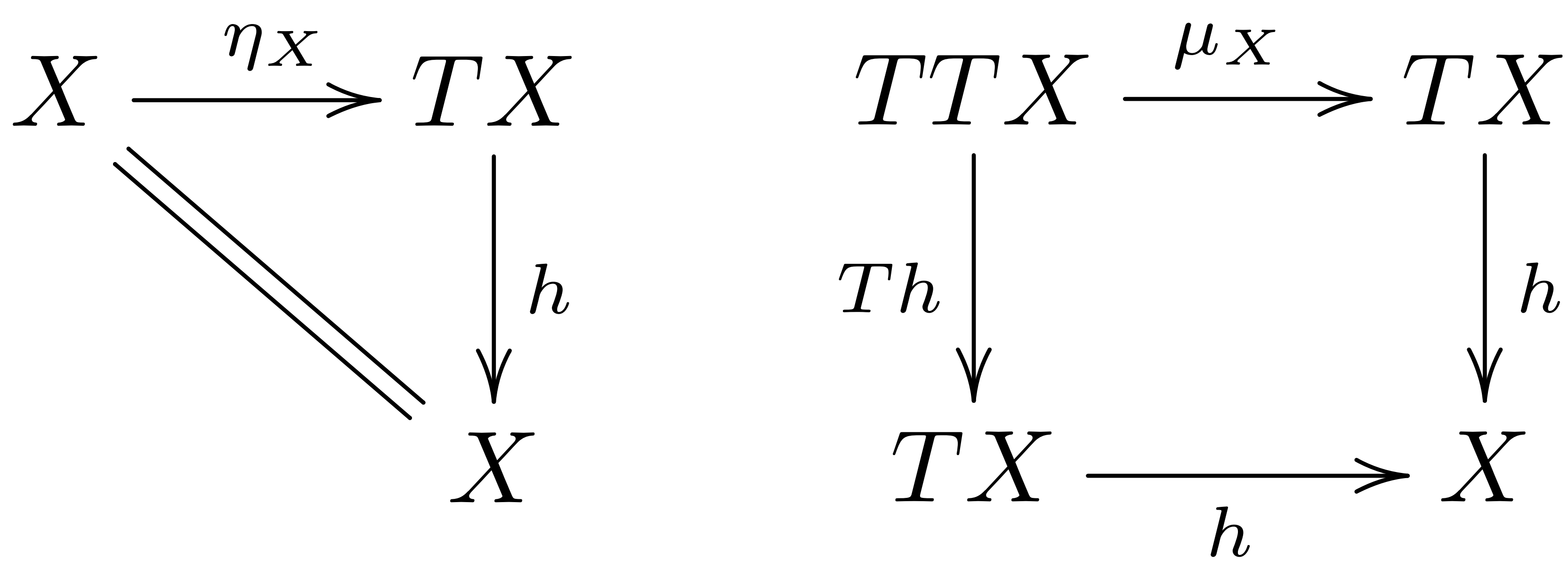


Algebras

A T-algebra is:

- a set X and a function $f : TX \to X$

such that:



Examples:

$(-)^*$-algebras are monoids

Algebras

A T-algebra is:

- a set X and a function $f : TX \to X$

such that:

$$
\begin{array}{ccc}
X & \xrightarrow{\eta_X} & TX \\
 & \searrow\!\!\!= & \downarrow h \\
 & & X
\end{array}
\qquad\qquad
\begin{array}{ccc}
TTX & \xrightarrow{\mu_X} & TX \\
{\scriptstyle Th}\downarrow & & \downarrow{\scriptstyle h} \\
TX & \xrightarrow[h]{} & X
\end{array}
$$

Examples:

$(-)^*$-algebras are monoids

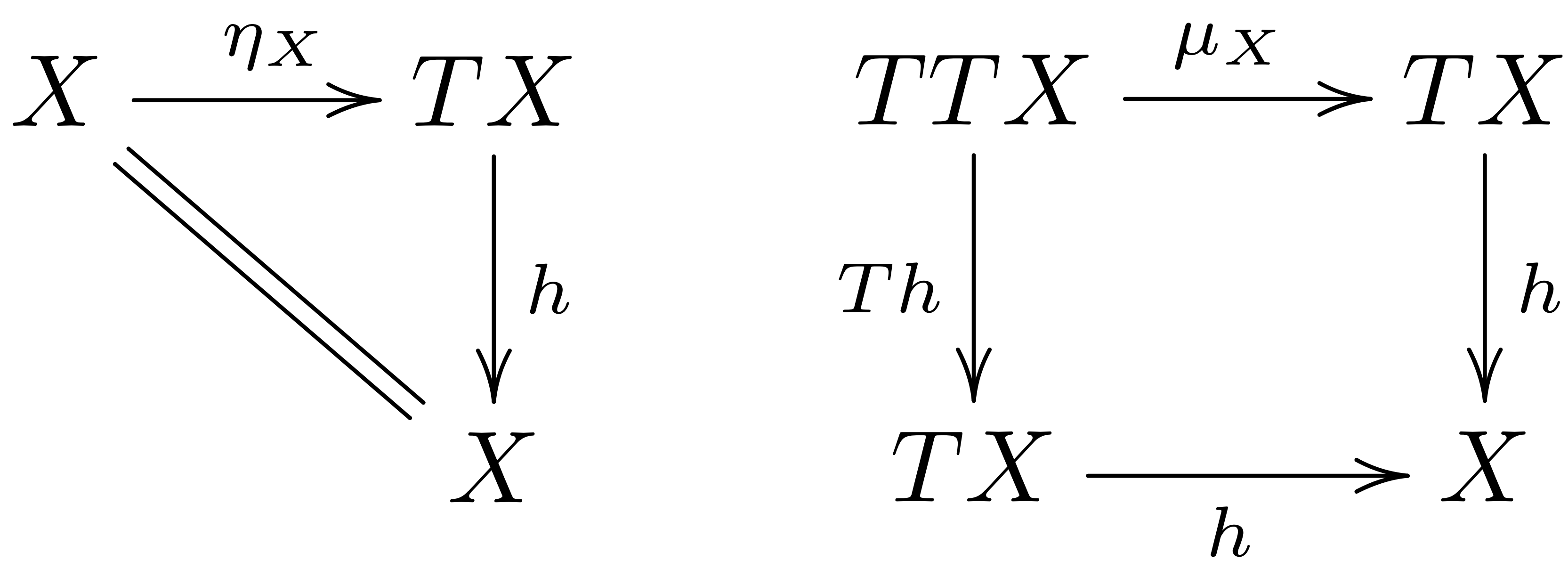
$\mathcal{P}_{\text{fin}}$-algebras are semilattices

Algebras

A T-algebra is:

- a set X and a function $f : TX \to X$

such that:



Examples:

$(-)^*$-algebras are monoids

$\mathcal{P}_{\text{fin}}$-algebras are semilattices

Term-monad algebras are what you expect

Homomorphisms

A homomorphism from $f : TX \to X$
to $g : TY \to Y$:

a function $h : X \to Y$ such that:

$$\begin{array}{ccc} TX & \xrightarrow{Th} & TY \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ X & \xrightarrow[h]{} & Y \end{array}$$

Recognizing languages with algebras

Fact: $\mu_X : TTX \to TX$ is always a T-algebra.

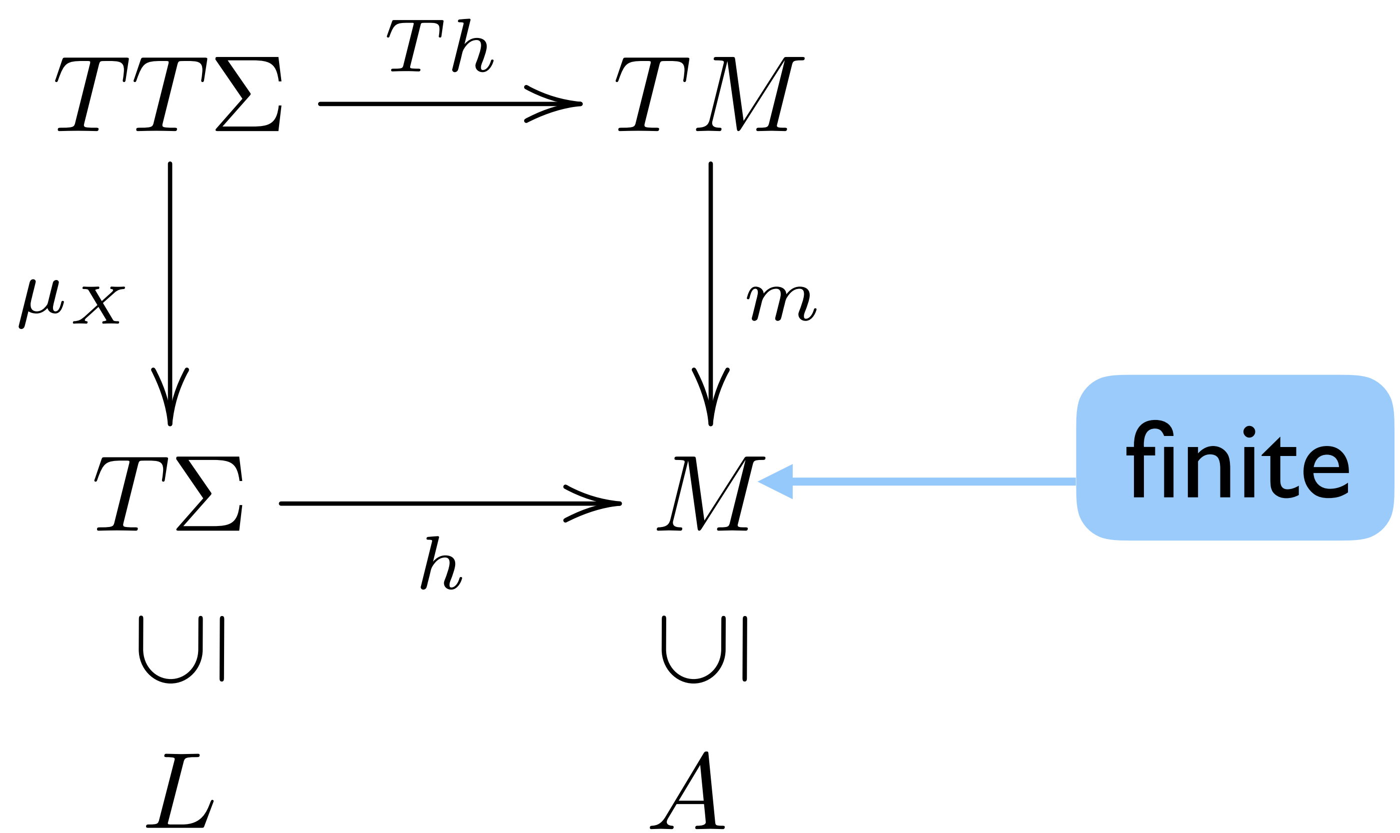
Recognizing languages with algebras

Fact: $\mu_X : TTX \to TX$ is always a T-algebra.

$$
\begin{array}{ccc}
TT\Sigma & \xrightarrow{Th} & TM \\
\downarrow{\scriptstyle \mu_X} & & \downarrow{\scriptstyle m} \\
T\Sigma & \xrightarrow[h]{} & M \\
\cup\!| & & \cup\!| \\
L & & A
\end{array}
$$

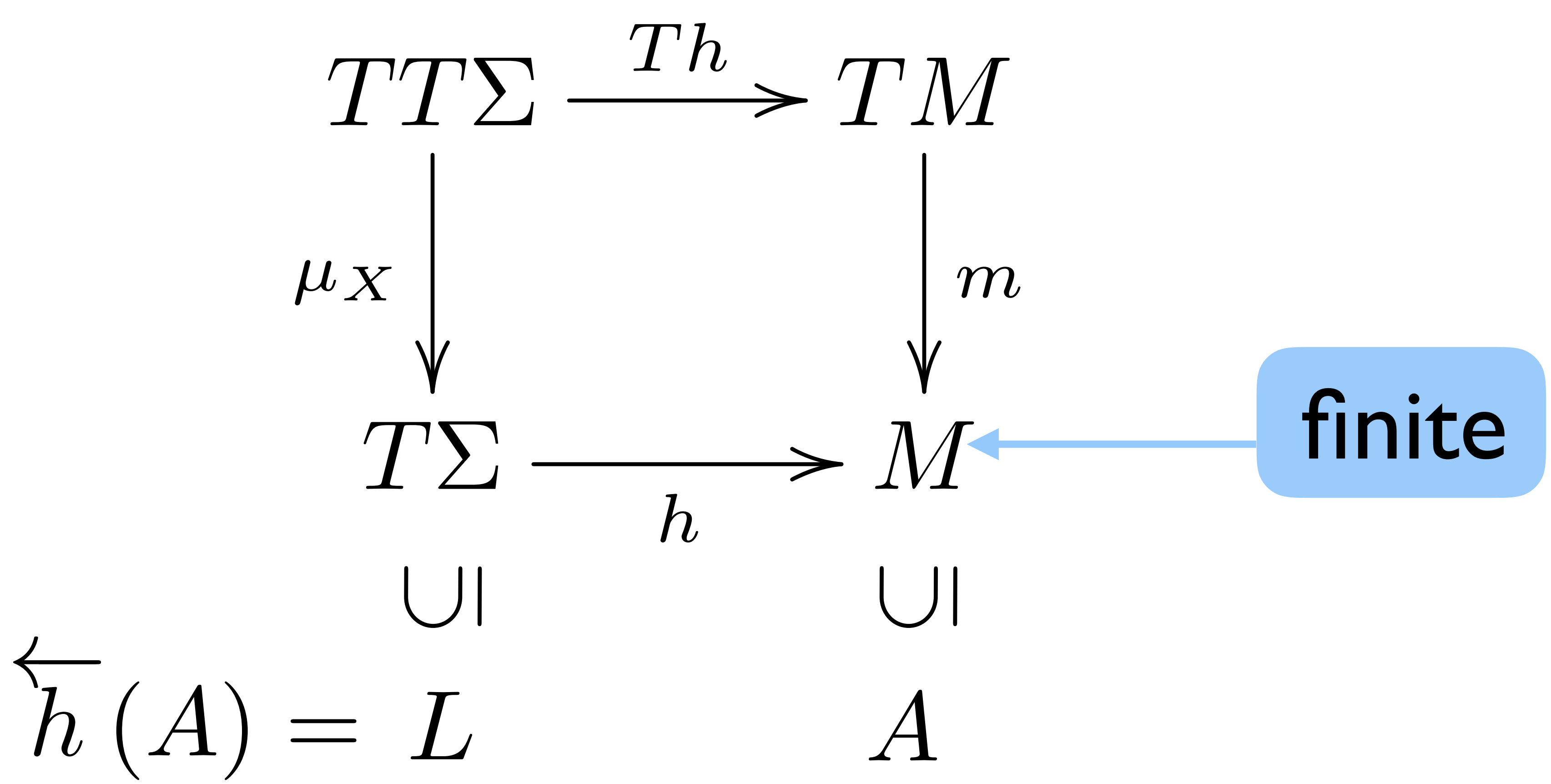
Recognizing languages with algebras

Fact: $\mu_X : TTX \to TX$ is always a T-algebra.



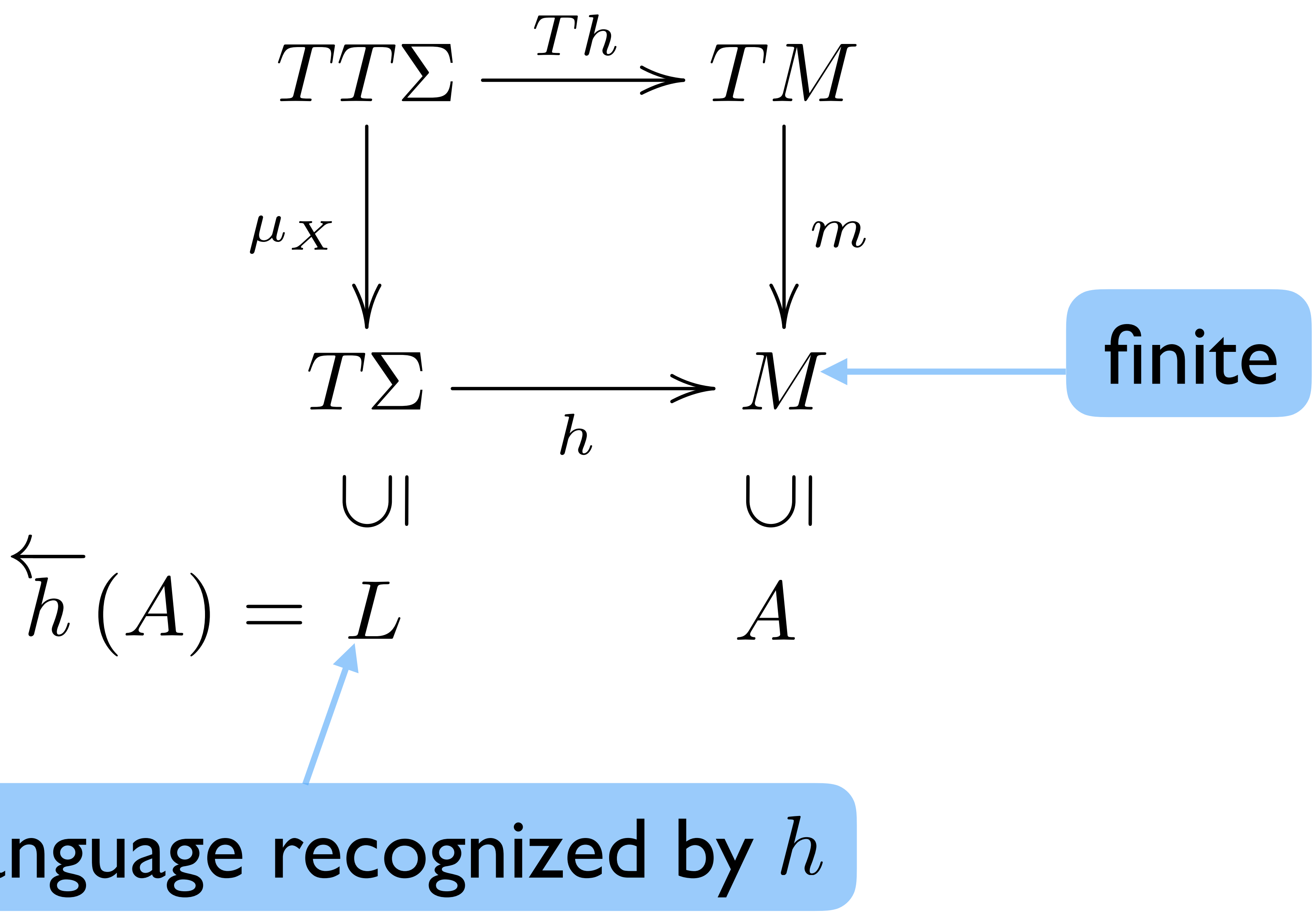
Recognizing languages with algebras

Fact: $\mu_X : TTX \to TX$ is always a T-algebra.



Recognizing languages with algebras

Fact: $\mu_X : TTX \to TX$ is always a T-algebra.



What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms. ?

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms. ?

those of the form $Tf : T\Sigma \to T\Gamma$

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

those of the form $Tf : T\Sigma \to T\Gamma$

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

- boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

✓ - boolean combinations
- inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

✓ - boolean combinations
✓ - inverse images along homomorphisms,
- direct images along (surjective) letter-to-letter homomorphisms.

What we want to talk about

The class of languages
recognized by finite algebras
is closed under:

✓ - boolean combinations
✓ - inverse images along homomorphisms,
✗ - direct images along (surjective)
letter-to-letter homomorphisms.

Counterexample

Let T be the list monad quotiented by:

$$x \cdot x \cdot x = x \cdot x$$

Counterexample

Let T be the list monad quotiented by:

$$x \cdot x \cdot x = x \cdot x \quad \text{B}$$

Counterexample

Let T be the list monad quotiented by:

$$x \cdot x \cdot x = x \cdot x \quad \mathcal{B}$$

A language $L \subseteq T\Sigma$ corresponds to
a language $L \subseteq \Sigma^*$ closed under $\mathcal{B}$
(in the sense of (sub)word rewriting)

Counterexample

Let T be the list monad quotiented by:

$$x \cdot x \cdot x = x \cdot x \quad \mathcal{B}$$

A language $L \subseteq T\Sigma$ corresponds to
a language $L \subseteq \Sigma^*$ closed under $\mathcal{B}$
(in the sense of (sub)word rewriting)

A T-algebra is a monoid that satisfies $\mathcal{B}$

Counterexample

Let T be the list monad quotiented by:

$$x \cdot x \cdot x = x \cdot x \qquad \mathcal{B}$$

A language $L \subseteq T\Sigma$ corresponds to
a language $L \subseteq \Sigma^*$ closed under $\mathcal{B}$
(in the sense of (sub)word rewriting)

A T-algebra is a monoid that satisfies $\mathcal{B}$

Fact: $L \subseteq T\Sigma$ is recognizable iff
(the corresponding) $L \subseteq \Sigma^*$ is
regular and closed under $\mathcal{B}$.

Counterexample ctd.

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Fact: L is closed under B .

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Fact: L is closed under $\mathcal{B}$.

So: L is T -recognizable.

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Fact: L is closed under B .

So: L is T-recognizable.

Put $\Gamma = \Delta \cup \{0\}$ and $h : \Sigma \to \Gamma$ s.t. $h(1) = 0$.

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Fact: L is closed under $\mathcal{B}$.

So: L is T-recognizable.

Put $\Gamma = \Delta \cup \{0\}$ and $h : \Sigma \to \Gamma$ s.t. $h(1) = 0$.

Then $\overrightarrow{Th}(L)$ is the $\mathcal{B}$-closure of $\Delta^* 0 \Delta^* 0 \subseteq \Gamma^*$

Counterexample ctd.

For $\Delta = \{a, b, c\}$ and $\Sigma = \Delta \cup \{0, 1\}$, let

$$L = \Delta^* 0 \Delta^* 1 \subseteq \Sigma^*$$

Fact: L is closed under $\mathcal{B}$.

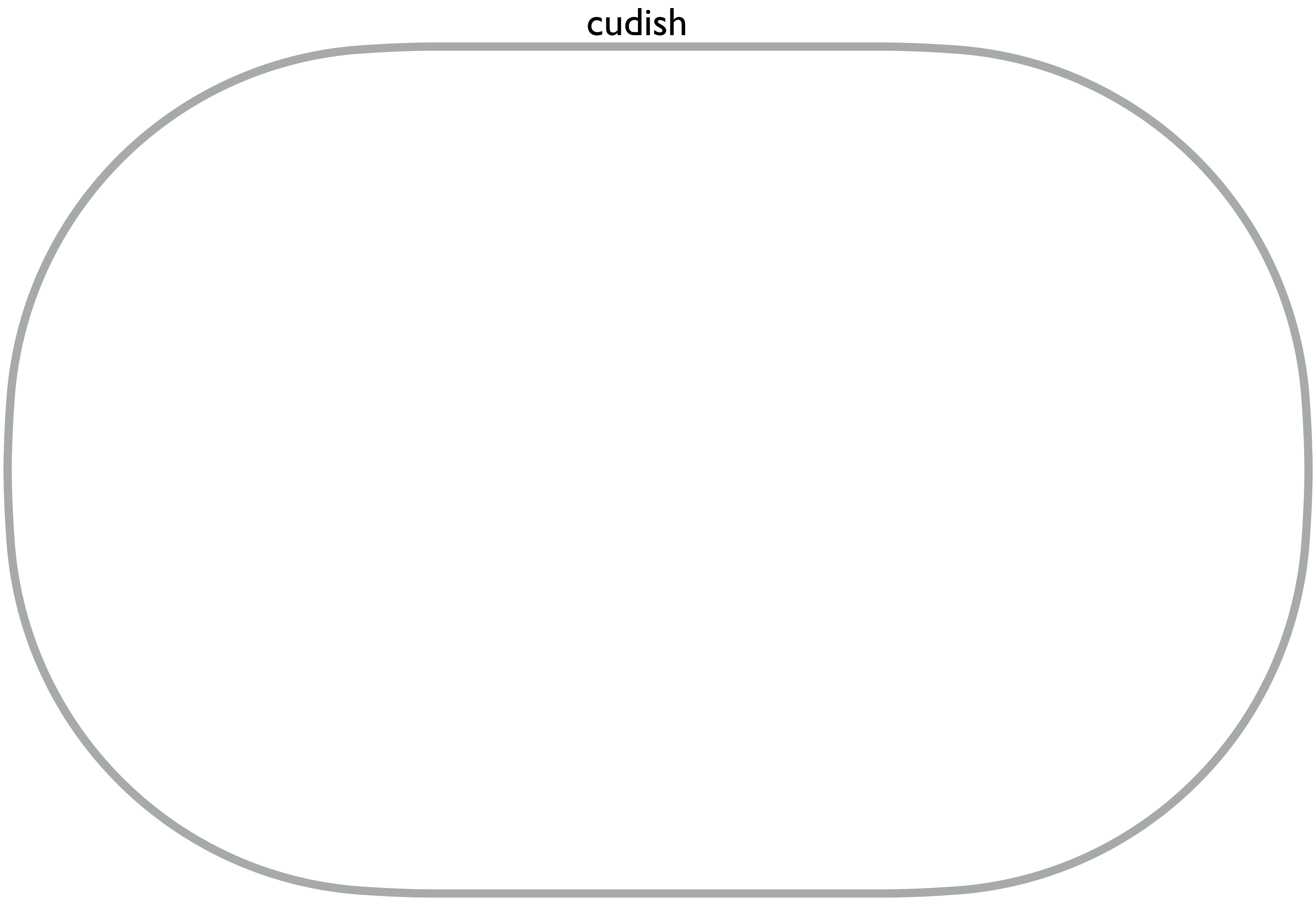
So: L is T-recognizable.

Put $\Gamma = \Delta \cup \{0\}$ and $h : \Sigma \to \Gamma$ s.t. $h(1) = 0$.

Then $\overrightarrow{Th}(L)$ is the $\mathcal{B}$-closure of $\Delta^* 0 \Delta^* 0 \subseteq \Gamma^*$

Fact: $\overrightarrow{Th}(L)$ is not regular, so not T-recognizable.

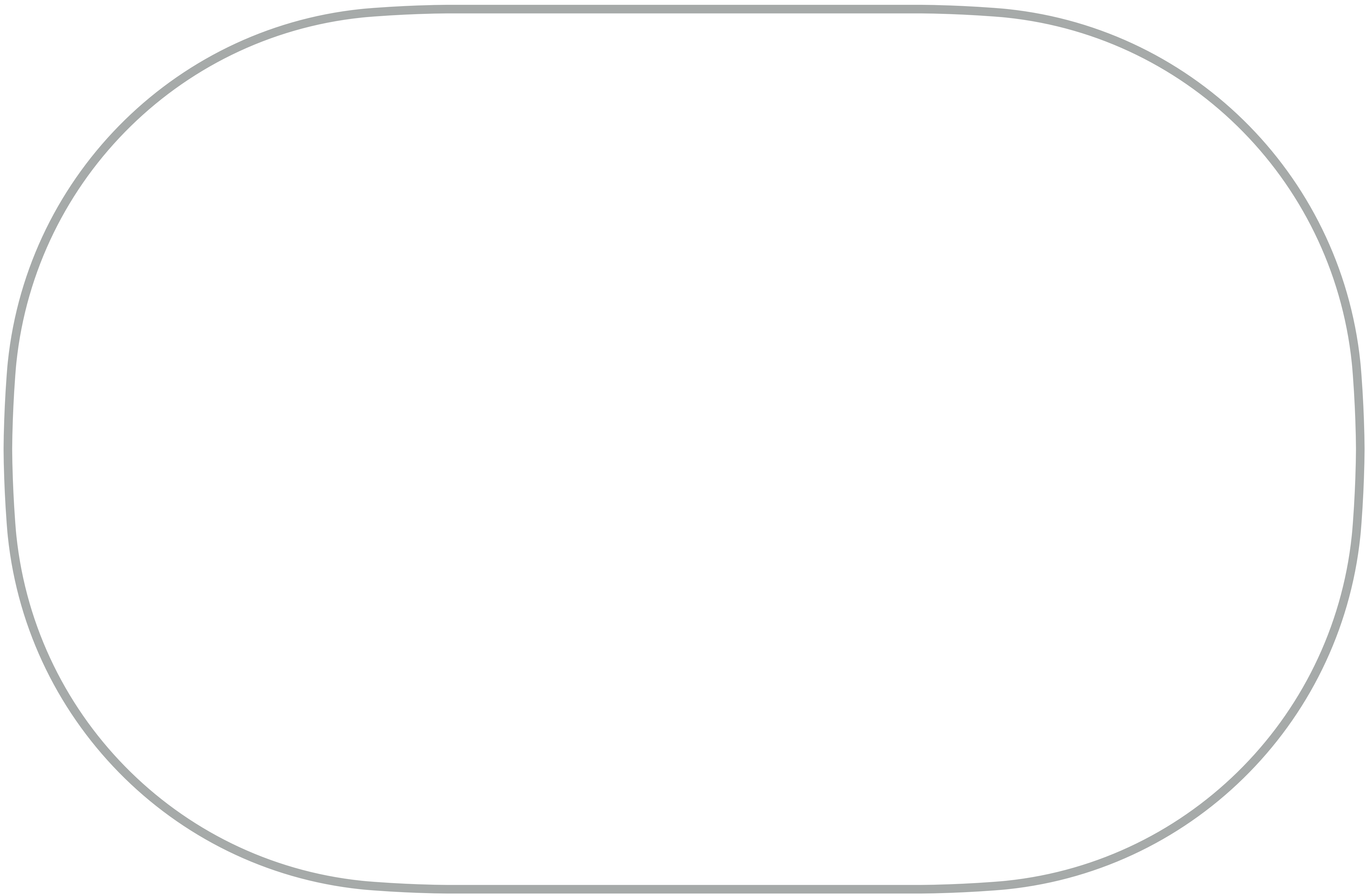
The landscape of monads



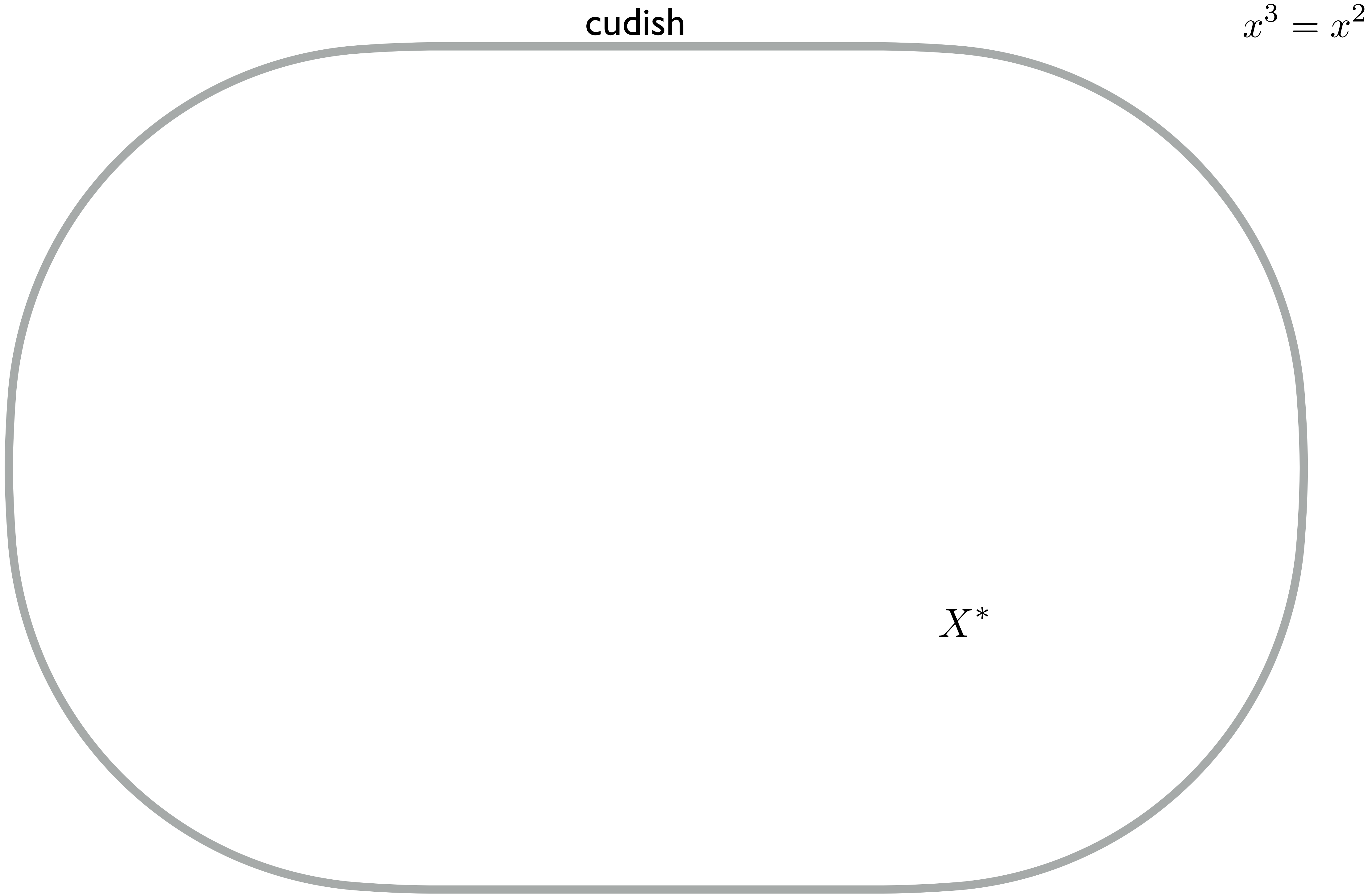
The landscape of monads

cudish

$x^3 = x^2$

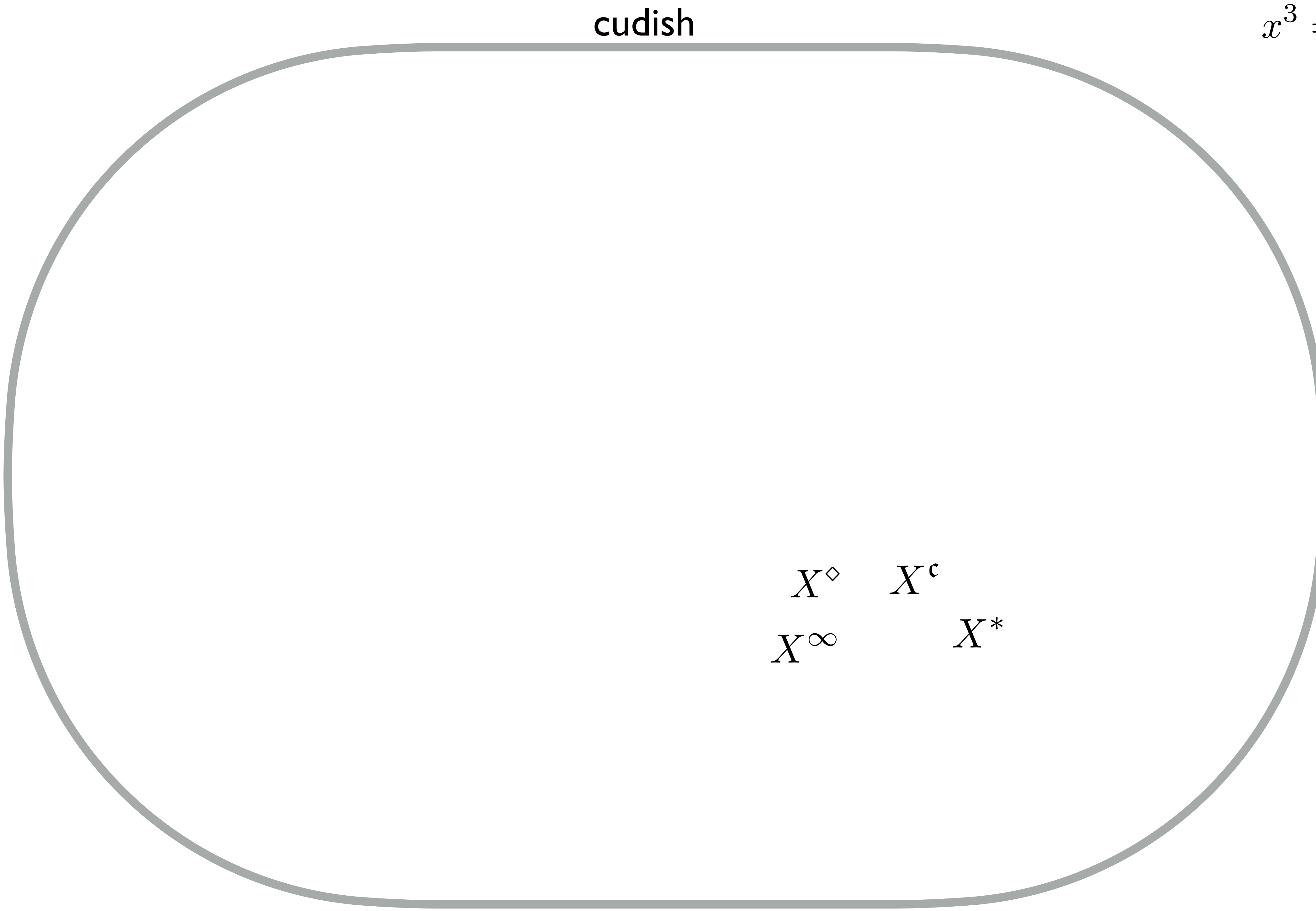


The landscape of monads



The landscape of monads

$x^3 = x^2$



Sufficient condition 1

Fact: if T preserves finiteness
then every language on a finite alphabet Σ
is recognizable (by $T\Sigma$).

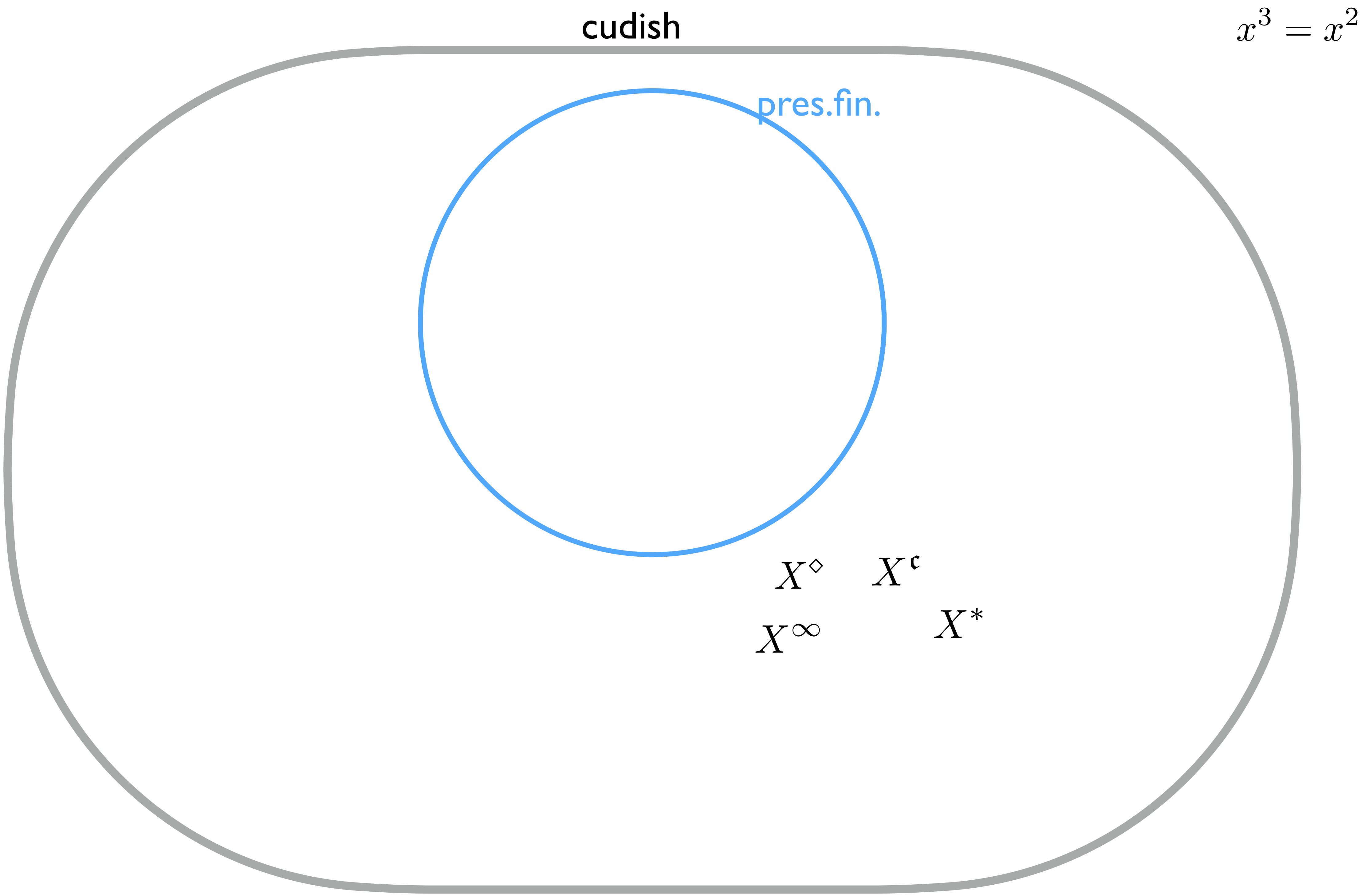
Sufficient condition 1

Fact: if T preserves finiteness
then every language on a finite alphabet Σ
is recognizable (by $T\Sigma$).

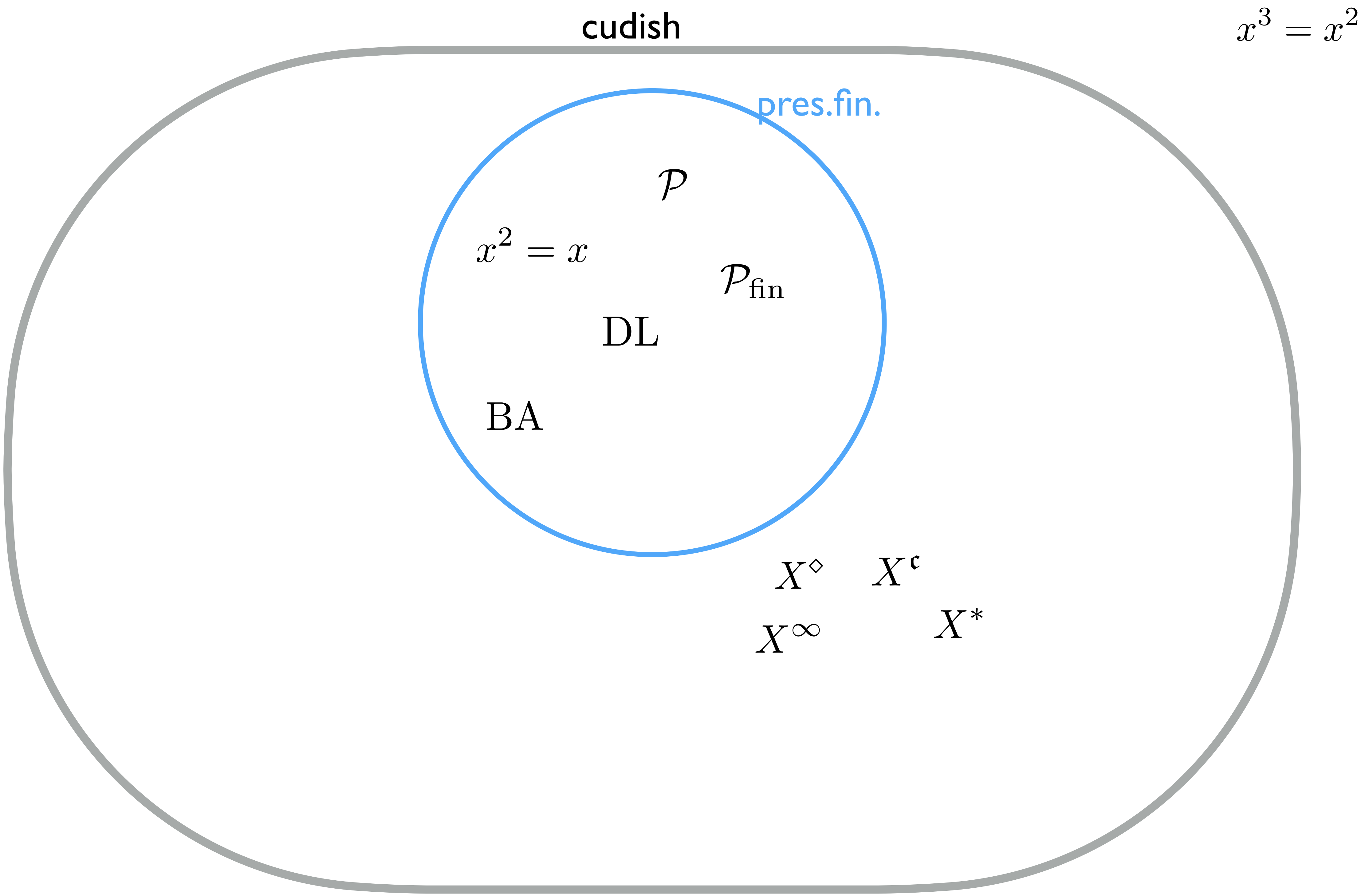
Examples:

- $\mathcal{P}$, $\mathcal{P}^+$, $\mathcal{P}_{\mathrm{fin}}$
- idempotent monoids/semigroups
- distributive lattices
- Boolean algebras

The landscape of monads



The landscape of monads



Sufficient condition II

Def.: a monad is Malcevian
if it admits (an eq. presentation with)
a ternary term $t(x, y, z)$ such that

$$t(x, x, y) = y = t(y, x, x)$$

Sufficient condition II

Def.: a monad is Malcevian
if it admits (an eq. presentation with)
a ternary term $t(x, y, z)$ such that

$$t(x, x, y) = y = t(y, x, x)$$

Fact: Malcevian monads are cudish.

Sufficient condition II

Def.: a monad is Malcevian
if it admits (an eq. presentation with)
a ternary term $t(x, y, z)$ such that

$$t(x, x, y) = y = t(y, x, x)$$

Fact: Malcevian monads are cudish.

Examples:

- groups $\quad t(x, y, z) = xy^{-1}z$

Sufficient condition II

Def.: a monad is Malcevian
if it admits (an eq. presentation with)
a ternary term $t(x, y, z)$ such that

$$t(x, x, y) = y = t(y, x, x)$$

Fact: Malcevian monads are cudish.

Examples:

- groups $t(x, y, z) = xy^{-1}z$
- Boolean algebras

$$t(x, y, z) = (x \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z)$$

Sufficient condition II

Def.: a monad is Malcevian
if it admits (an eq. presentation with)
a ternary term $t(x, y, z)$ such that

$$t(x, x, y) = y = t(y, x, x)$$

Fact: Malcevian monads are cudish.

Examples:

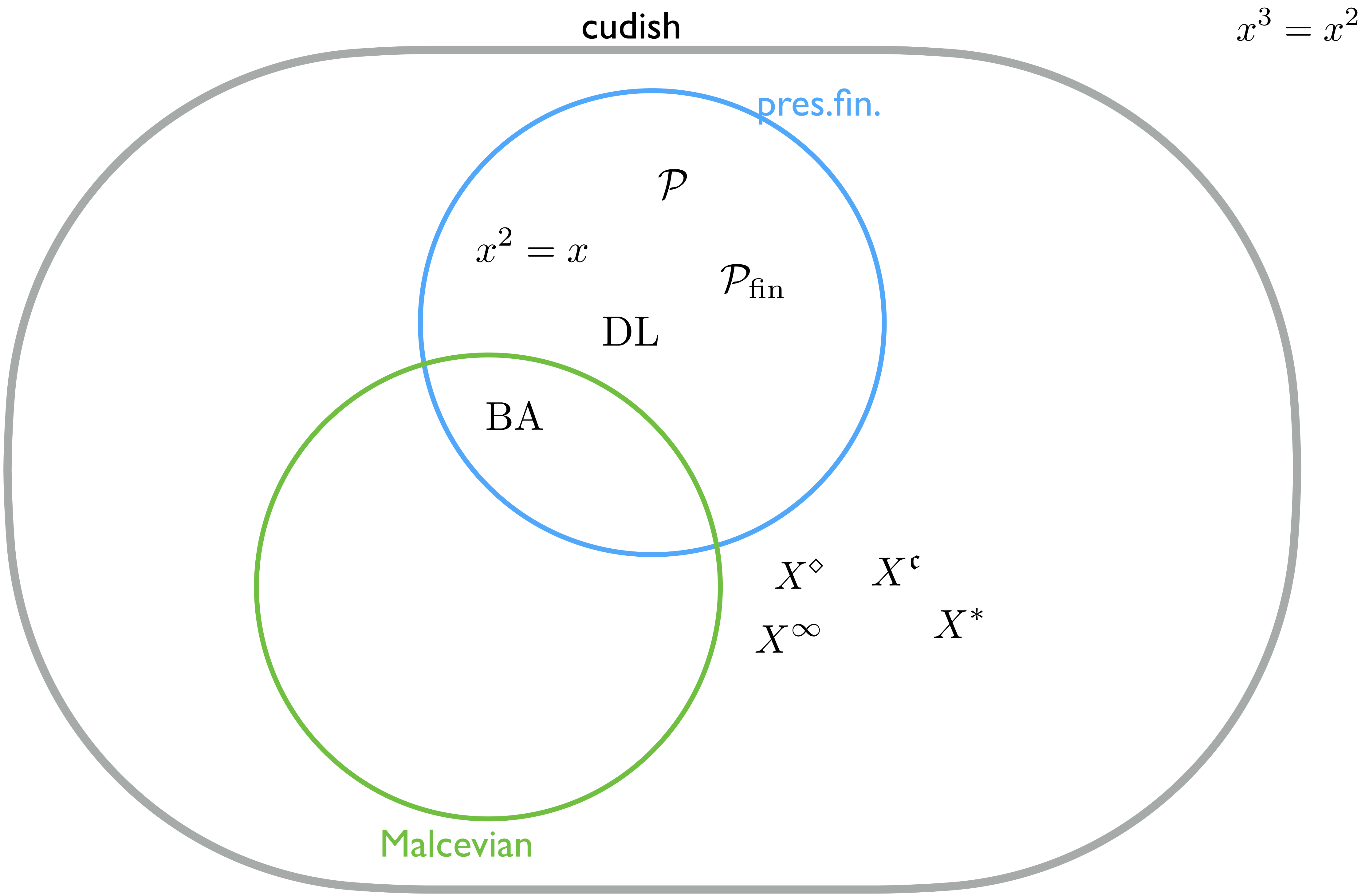
- groups $t(x, y, z) = xy^{-1}z$
- Boolean algebras

$$t(x, y, z) = (x \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z)$$

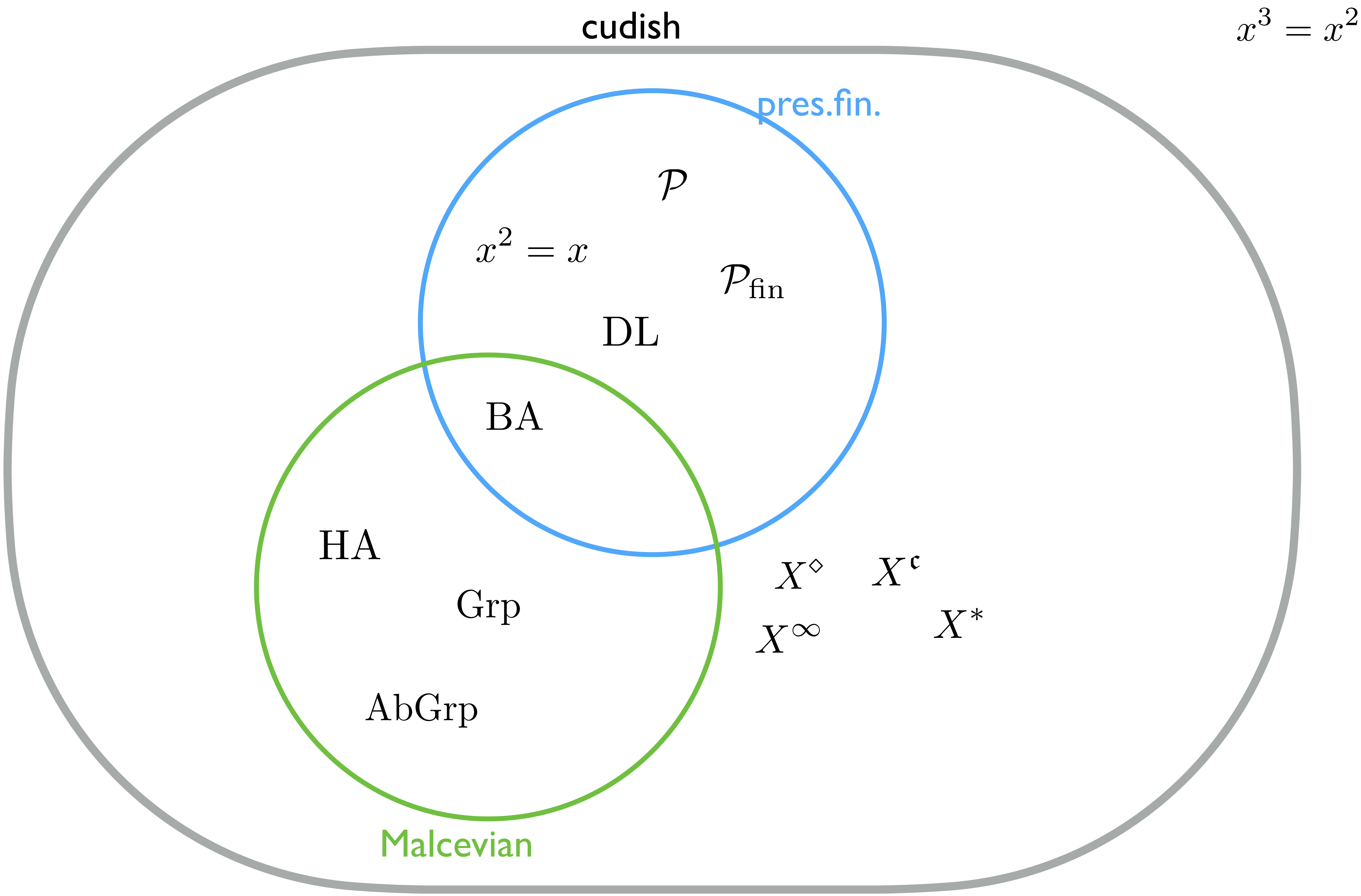
- Heyting algebras

$$t(x, y, z) = ((x \to y) \to z) \wedge ((z \to y) \to z) \wedge (x \vee z)$$

The landscape of monads



The landscape of monads



Sufficient condition III

Def.: a monad T is weakly Cartesian
if:
- T preserves weak pullbacks
- all naturality squares for η and μ are weak pullbacks.

Sufficient condition III

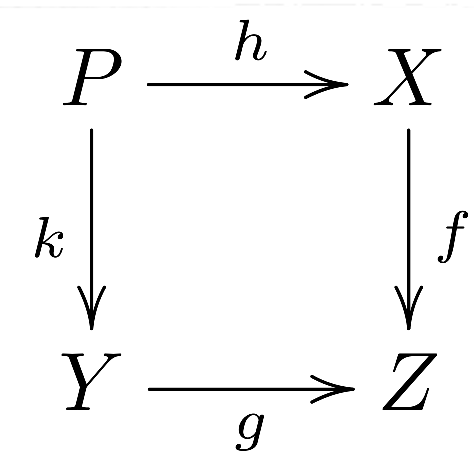
Def.: a monad T is weakly Cartesian

if:
- T preserves weak pullbacks
- all naturality squares for η and μ are weak pullbacks.

Sufficient condition III

Def.: a monad T is weakly Cartesian
if: - T preserves weak pullbacks
- all naturality squares for η and μ are weak pullbacks.

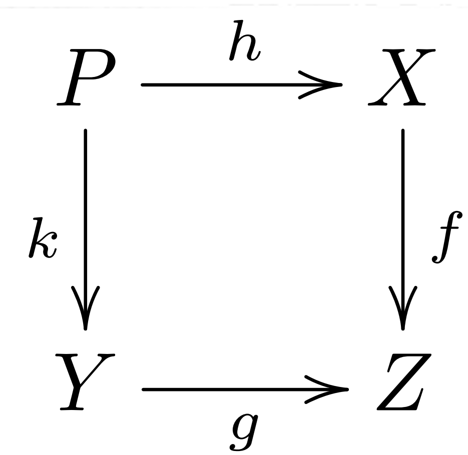
weak pullback:
for all $x \in X$, $y \in Y$ s.t. $f(x) = g(y)$
there is $p \in P$ s.t. $h(p) = x$, $k(p) = y$



Sufficient condition III

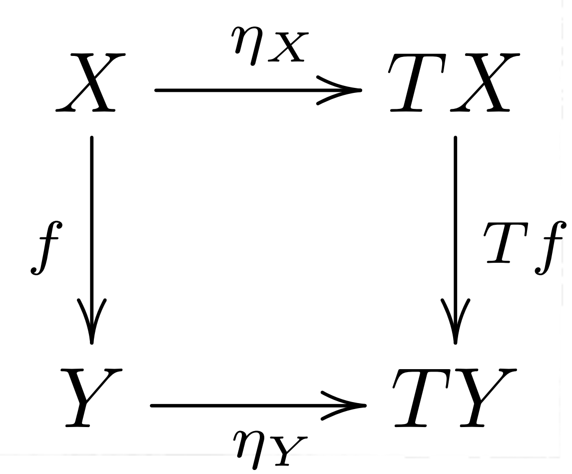
Def.: a monad T is weakly Cartesian
if: - T preserves weak pullbacks
- all naturality squares for η and μ are weak pullbacks.

weak pullback:
for all $x \in X$, $y \in Y$ s.t. $f(x) = g(y)$
there is $p \in P$ s.t. $h(p) = x$, $k(p) = y$



E.g. for η:

"a non-unit element never becomes a unit element after a substitution"



Sufficient condition III

Fact: weakly Cartesian monads are cudish.

(the powerset construction works)

Sufficient condition III

Fact: weakly Cartesian monads are cudish.

(the powerset construction works)

Examples:

- any monad presented by *linear regular equations:*

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad ✓$$

$$x \cdot y = y \cdot x \quad ✓$$

$$x \cdot x = x \quad ✗$$

$$x \cdot x^{-1} = e \quad ✗$$

Sufficient condition III

Fact: weakly Cartesian monads are cudish.

(the powerset construction works)

Examples:

- any monad presented by *linear regular equations:*

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad ✓$$

$$x \cdot y = y \cdot x \quad ✓$$

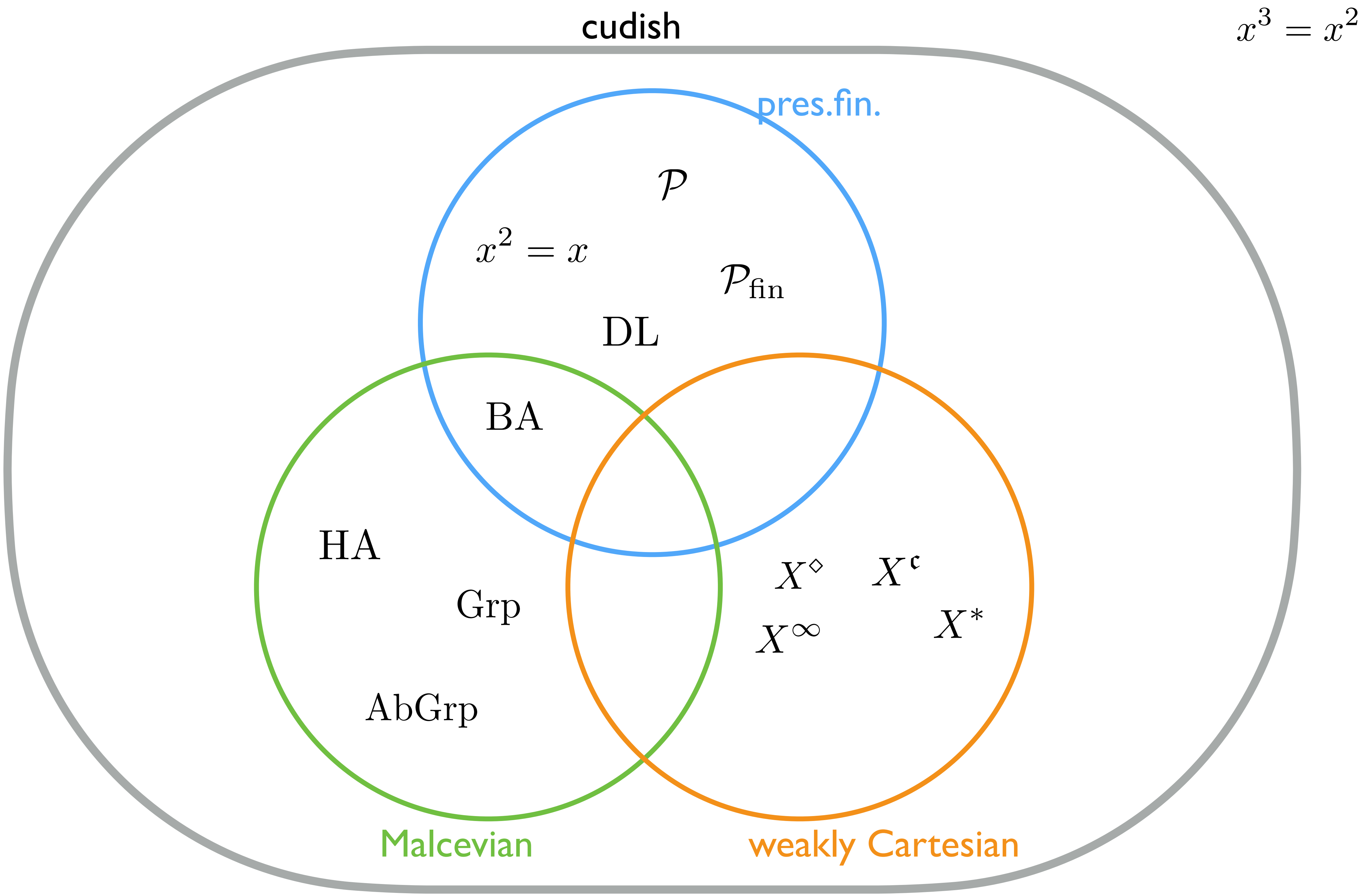
$$x \cdot x = x \quad ✗$$

$$x \cdot x^{-1} = e \quad ✗$$

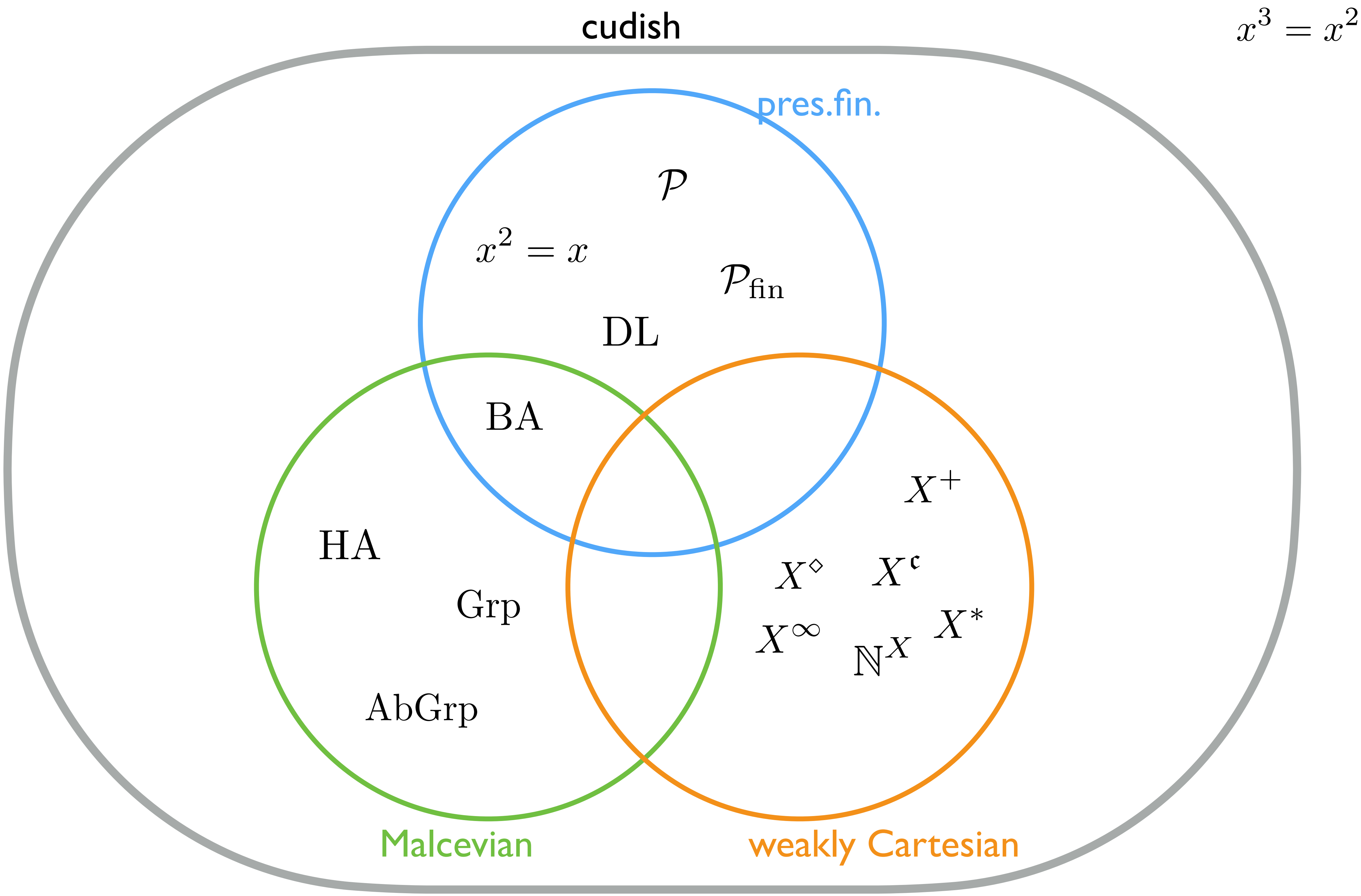
- T presented by a binary operation with:

$$x \cdot (x \cdot y) = x \cdot y$$

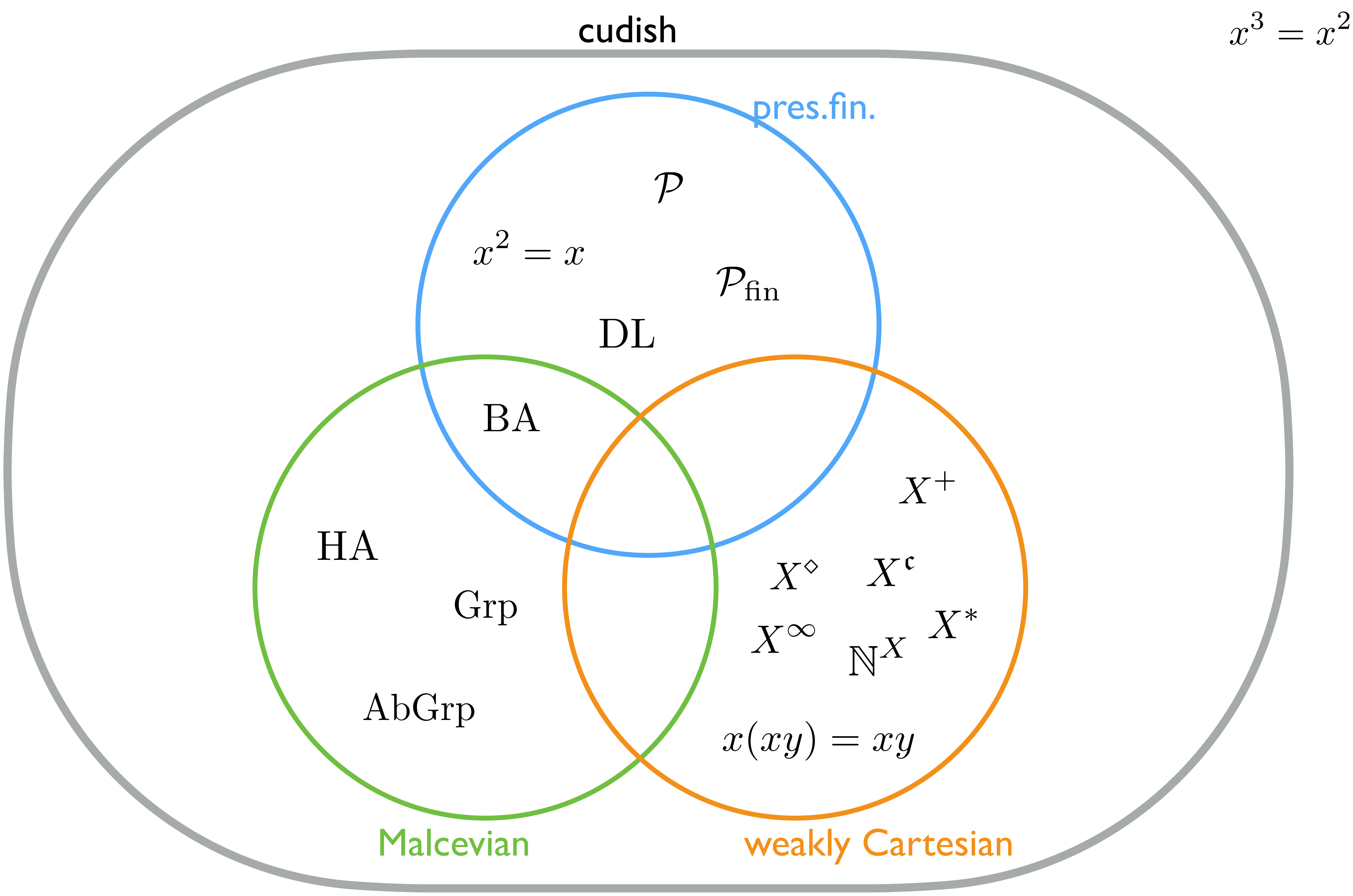
The landscape of monads



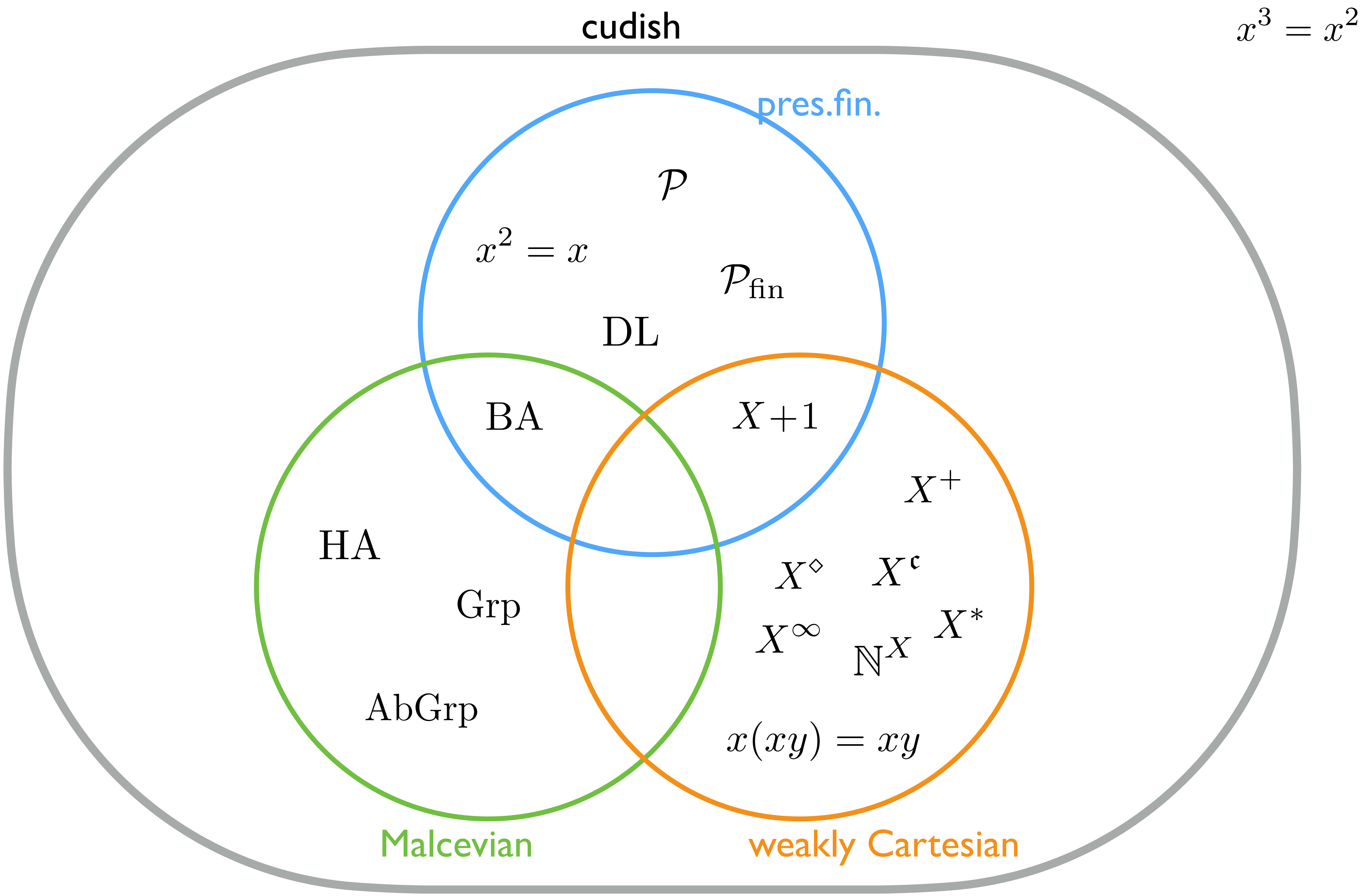
The landscape of monads



The landscape of monads



The landscape of monads



Other examples

1. The reader monad X^ω

(a compactness argument)

Other examples

1. The reader monad X^ω

(a compactness argument)

2. The free lattice monad Lat

(a convexity argument)

Other examples

1. The reader monad X^ω

   (a compactness argument)

2. The free lattice monad Lat

   (a convexity argument)

3. A binary operation with:

$$z \cdot (x \cdot (x \cdot y)) = z \cdot (x \cdot y)$$

(a "powerset squared" construction works)

Other examples

1. The reader monad X^ω

   (a compactness argument)

2. The free lattice monad Lat

   (a convexity argument)

3. A binary operation with:

$$z \cdot (x \cdot (x \cdot y)) = z \cdot (x \cdot y)$$

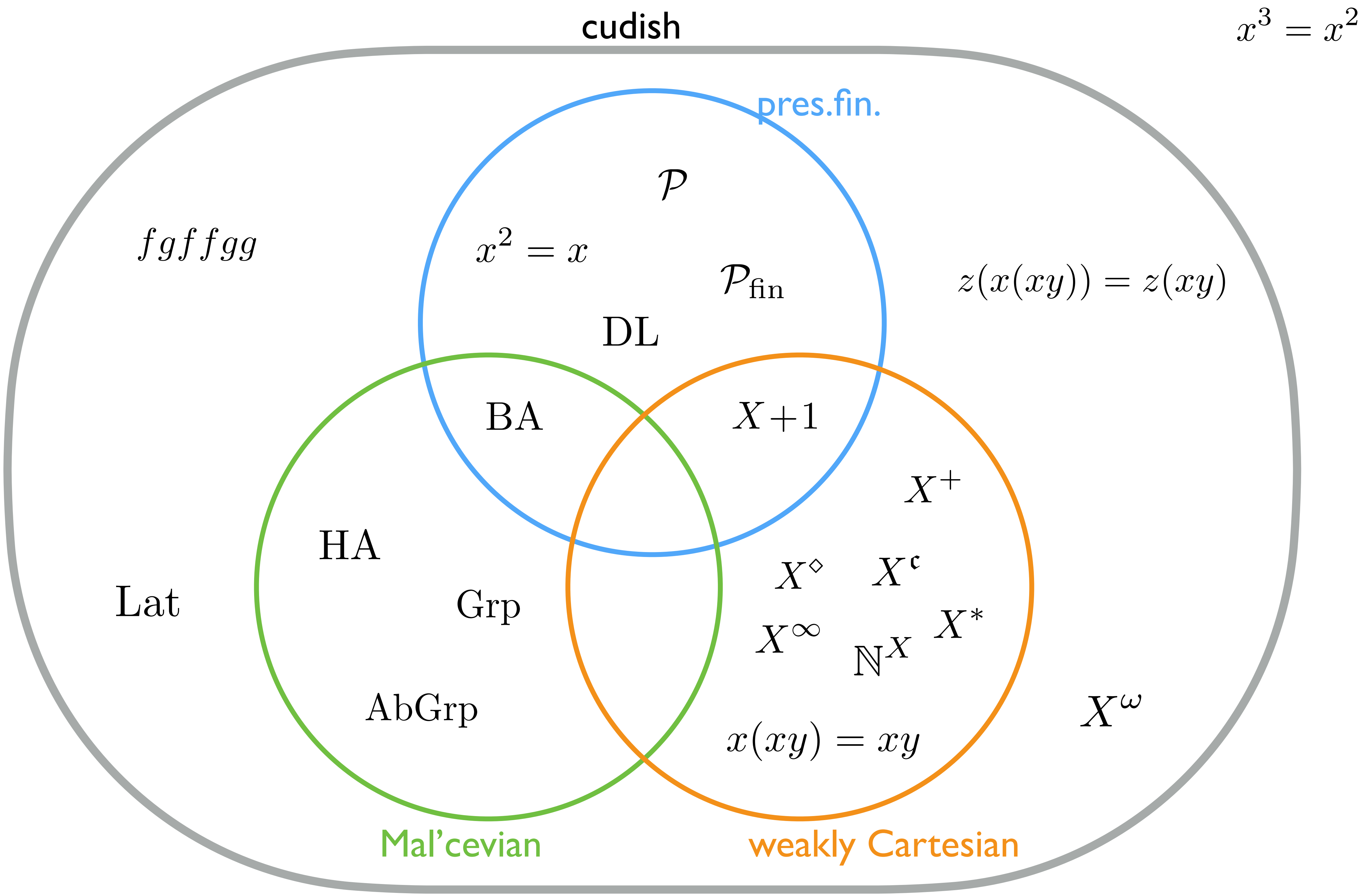
   (a "powerset squared" construction works)

4. Unary operations f, g with:

$$fgfgg(x) = x \qquad fgffgg(x) = fgffgg(y)$$

   (has no nontrivial finite algebras)

The landscape of monads



Counterexamples

1. Monoids with $x^3 = x^2$

Counterexamples

1. Monoids with $x^3 = x^2$

2. The "marked words" monad:

$$TX = \{(\beta, w) \mid \beta : X \to \mathbb{N},\ w \in X^*,\ \beta \leq w\}$$

Counterexamples

1. Monoids with $x^3 = x^2$

2. The "marked words" monad:

$$TX = \{(\beta, w) \mid \beta : X \to \mathbb{N},\ w \in X^*,\ \beta \leq w\}$$

3. The "balanced associativity" monad:
   a binary operation with

$$x \cdot (y \cdot x) = (x \cdot y) \cdot x$$

Counterexamples

1. Monoids with $x^3 = x^2$

2. The "marked words" monad:

$$TX = \{(\beta, w) \mid \beta : X \to \mathbb{N},\ w \in X^*,\ \beta \leq w\}$$

3. The "balanced associativity" monad:
   a binary operation with

$$x \cdot (y \cdot x) = (x \cdot y) \cdot x$$

4. The "almost Mal'cevian" monad:
   a ternary operation with

$$o(x, x, y) = o(y, x, x)$$

The landscape of monads