Inferring Symbolic Automata

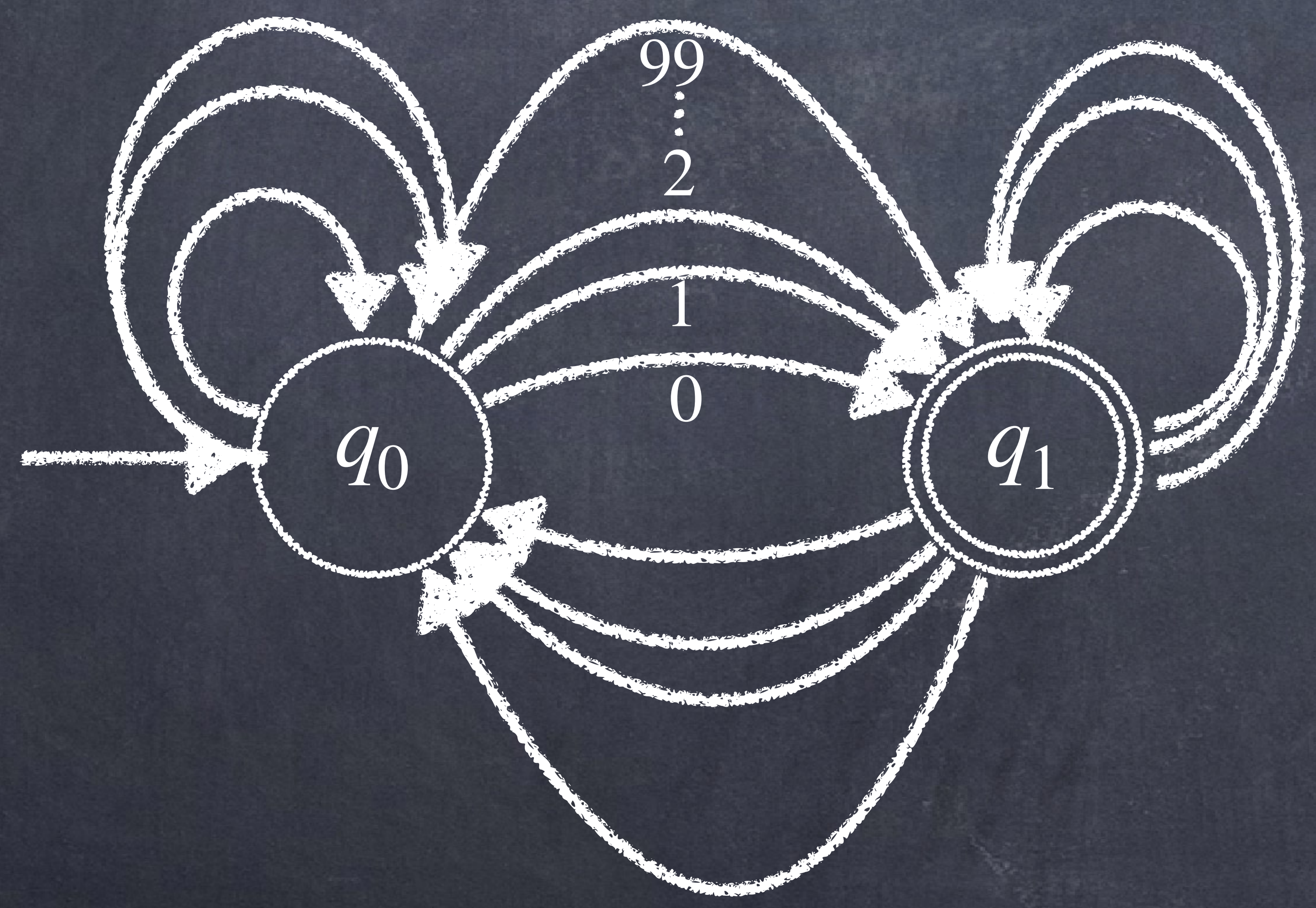
Hadar Frenkel

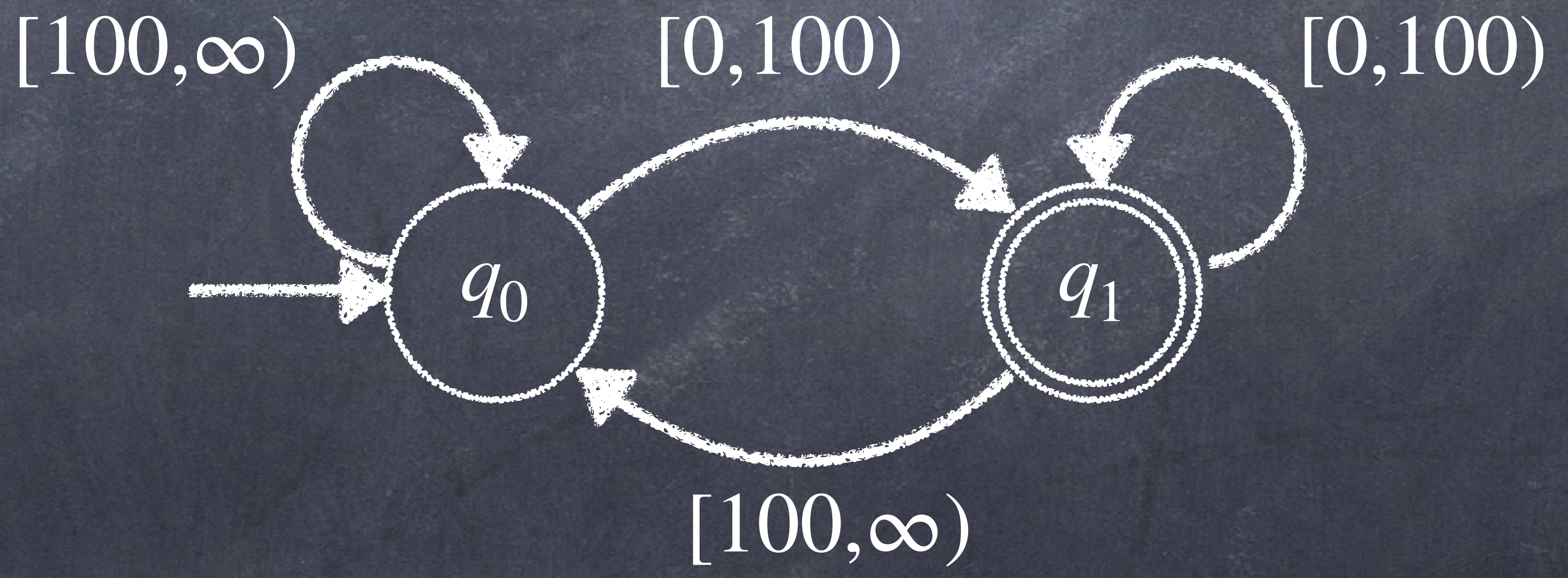
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Joint work with Dana Fisman and Sandra Zilles

Symbolic Automata - SFAs

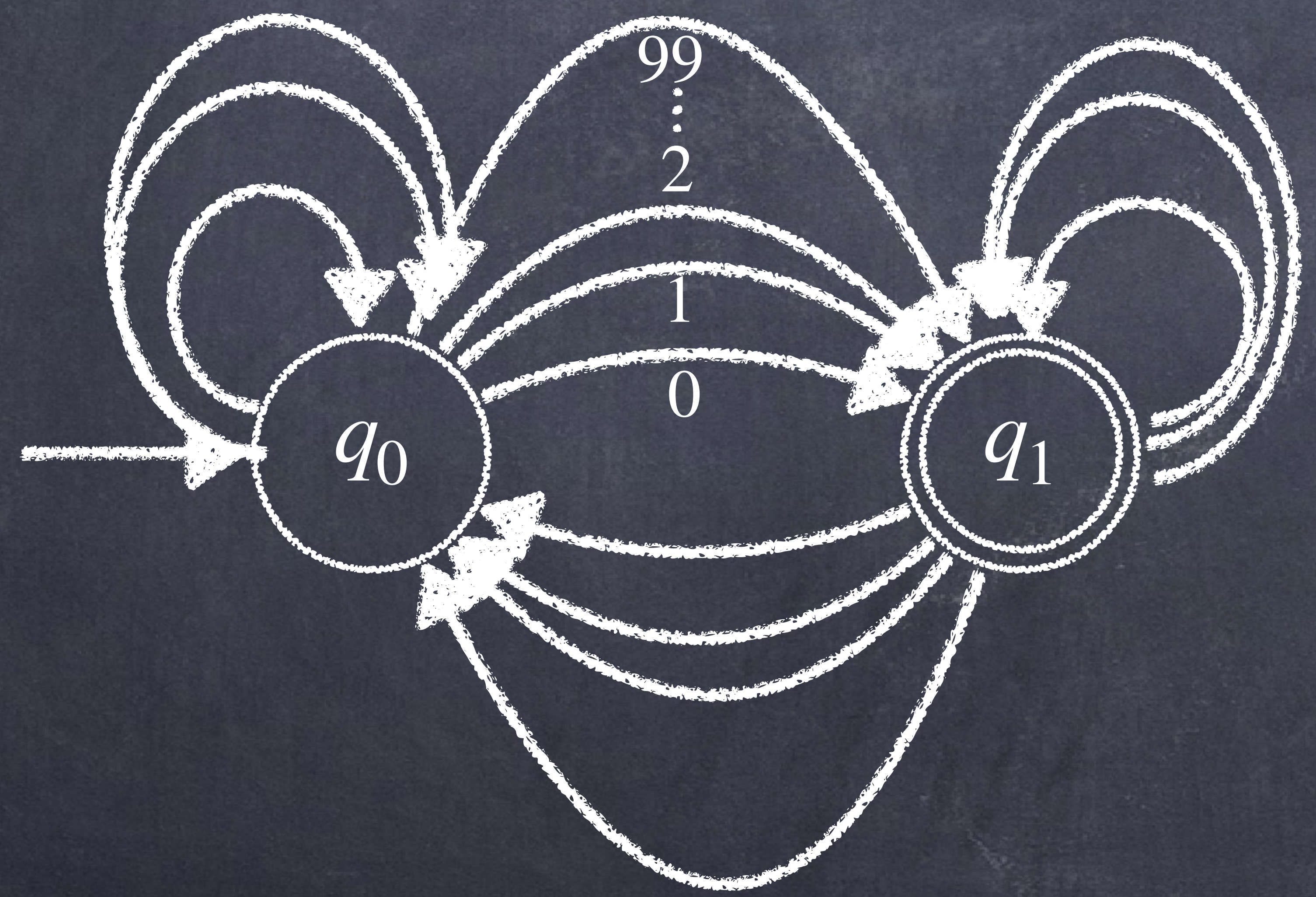
- Finite state automata
- Defined w.r.t. a Boolean algebra
- Transitions are over predicates

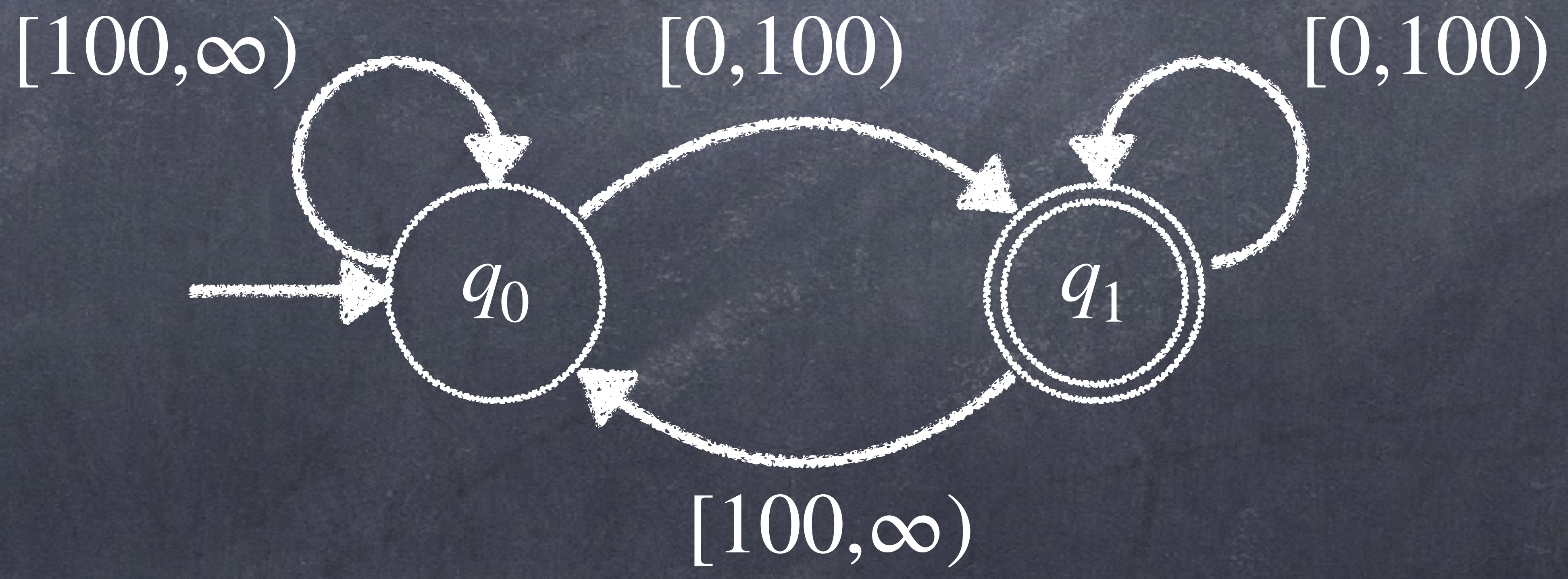




Symbolic Automata – SFAs

- Finite state automata
- Defined w.r.t. a Boolean algebra
- Transitions are over predicates
- Concise
- Reason about infinite domains





Monotonic Algebras

- Predicates correspond to a total order over the domain elements
- $[\psi] = \{d \mid a \leq d \leq b\}$
- Monotonic: Interval algebras (over $\mathbb{N}, \mathbb{Z}, \mathbb{R}$)

Monotonic Algebras

- Predicates correspond to a total order over the domain elements
- $[\psi] = \{d \,|\, a \leq d \leq b\}$
- Monotonic: Interval algebras (over $\mathbb{N}, \mathbb{Z}, \mathbb{R}$)

Propositional algebra

- Predicates are Boolean combinations of atomic propositions
- $(p_1 \wedge p_2) \vee (\neg p_1 \wedge p_3)$
- Not monotonic!

Automata Learning

- Active learning - L* style learning [Angluin 1987]
- Passive learning

Automata Learning

- Active learning – L* style learning [Angluin 1987]

- Passive learning

Automata Learning

- Active learning - L* style learning [Angluin 1987]

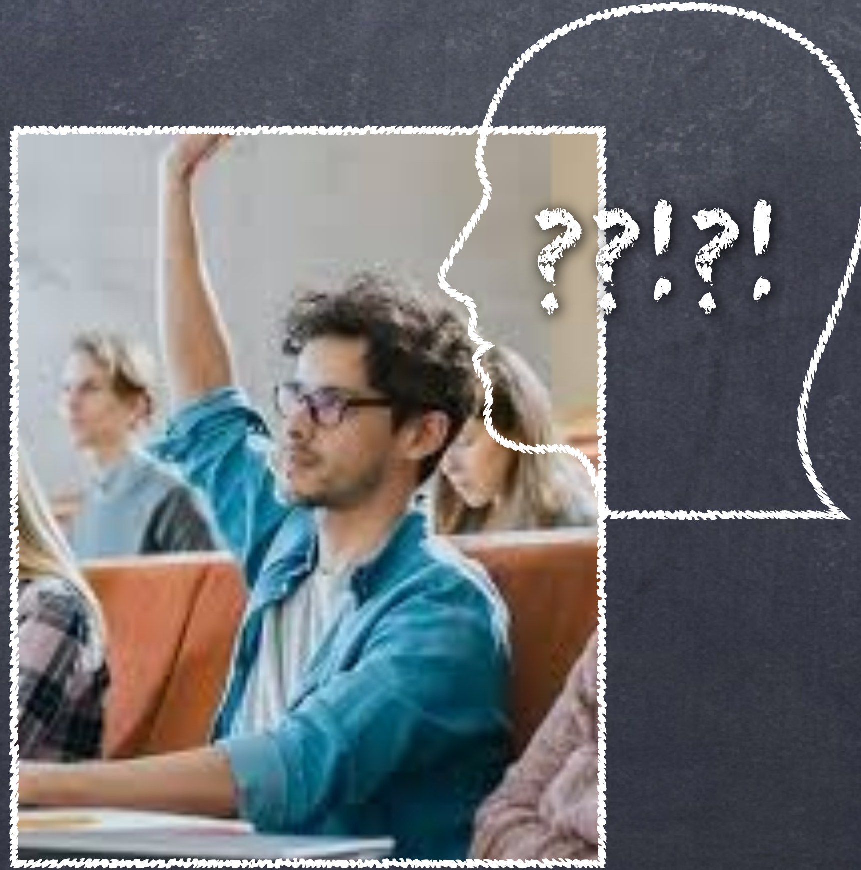
- Passive learning

$\langle w_1, \bot \rangle$

$\langle w_2, \top \rangle$

$\vdots$

$\langle w_n, \bot \rangle$



Learnability of a Class of languages via Representation $\mathscr{R}$

- Different representations of languages

- E.g. regular languages – DFAs, NFAs

Learnability of a Class of languages via Representation $\mathscr{R}$

- Different representations of languages
- A class of languages $\mathscr{L}$ is learnable via representation in $\mathscr{R}$ if there is an algorithm ALG such that we can apply ALG to learn a representation in $\mathscr{R}$ for every language in $\mathscr{L}$

- E.g. regular languages – DFAs, NFAs

Learnability of a Class of languages via Representation $\mathscr{R}$

- Different representations of languages
- A class of languages $\mathscr{L}$ is learnable via representation in $\mathscr{R}$ if there is an algorithm ALG such that we can apply ALG to learn a representation in $\mathscr{R}$ for every language in $\mathscr{L}$

- E.g. regular languages – DFAs, NFAs
- E.g. L* algorithm for regular languages via representation in DFAs

Active Learning of SFAs

- Positive results
  - Learning of SFAs over monotonic algebras using membership and equivalence queries
    - [Maler & Mens 2014], [Maler & Mens 2017]
    - [Chubachi, Diptarama, Yoshinaka, Shinohara 2017]
  - MAT* algorithm for learning SFAs
    - [Argyros & D'Antoni 2018]

Active Learning of SFAs

- First negative result

Active Learning of SFAs

- First negative result
- Necessary condition:
- We can **polynomially learn SFAs** over a Boolean algebra $\mathscr{A}$ using membership and equivalence queries only if we can **polynomially learn the predicates** of $\mathscr{A}$ using membership and equivalence queries

Active Learning of SFAs

- First negative result
- Necessary condition:
- We can polynomially learn SFAs over a Boolean algebra $\mathscr{A}$ using membership and equivalence queries only if we can polynomially learn the predicates of $\mathscr{A}$ using membership and equivalence queries

-> SFAs over the propositional algebra are not polynomially learnable

Passive Learning

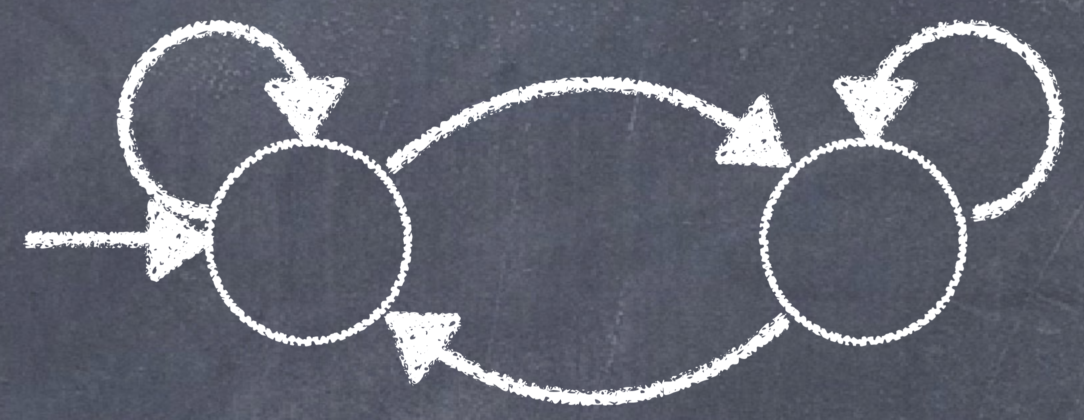
set
$\mathcal{S}'$

Infer

Identification in the Limit Using Polynomial Time and Data

set $\mathcal{S}$

← Characteristic



set $\mathcal{S}'$

Infer →

[Gold 1978, de la Higuera 1997]

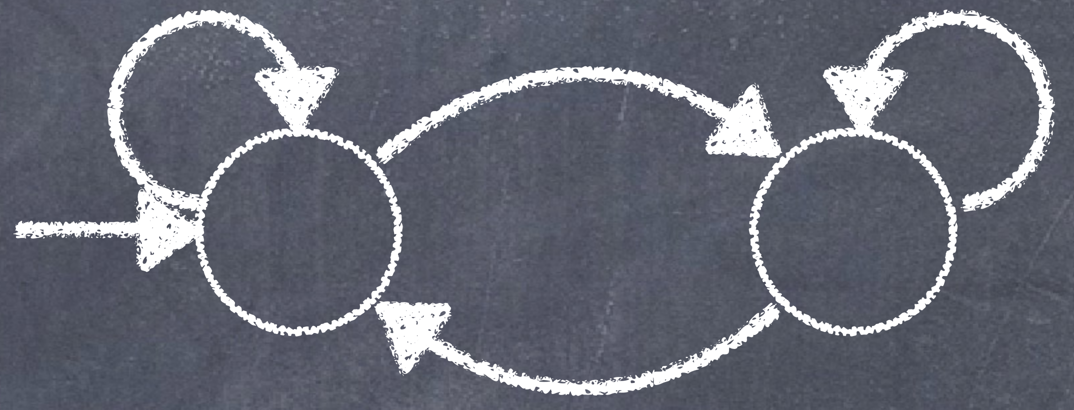
Identification in the Limit Using Polynomial Time and Data

[Gold 1978, de la Higuera 1997]

Identification in the Limit Using Polynomial Time and Data

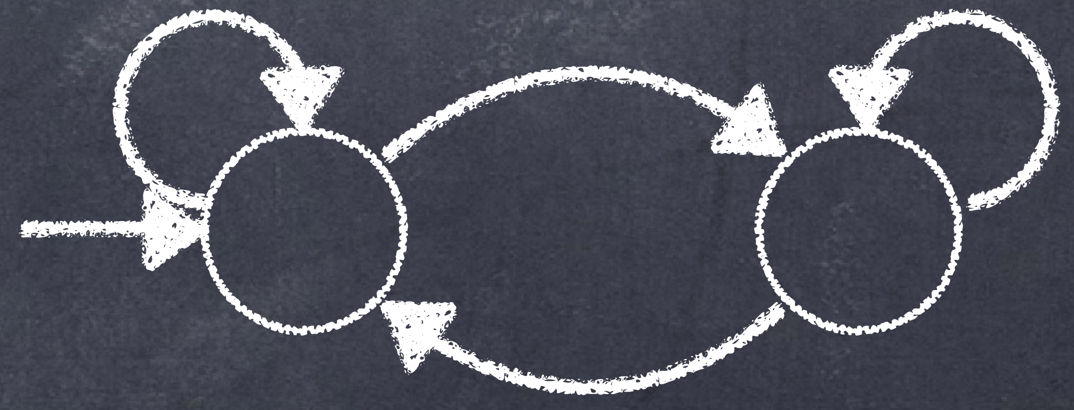
set $\mathcal{S}$

Characteristic ←



set $\mathcal{S}'$ | set $\mathcal{S}$

Infer →

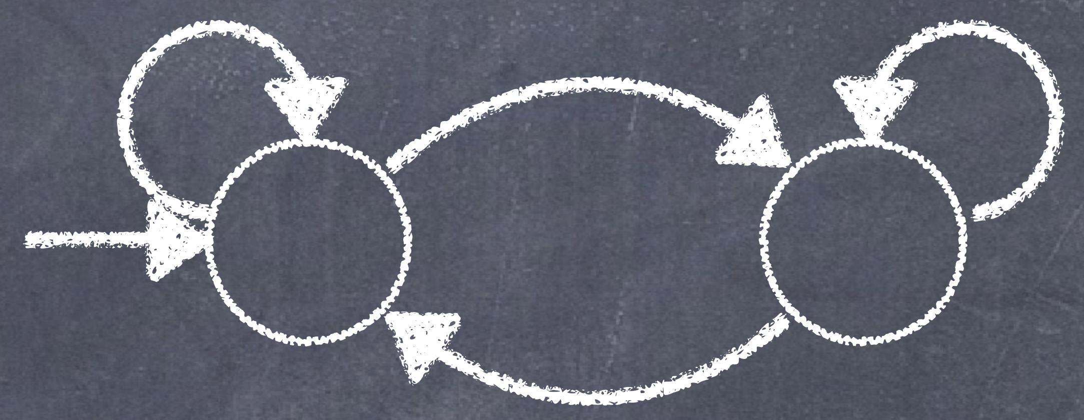


[Gold 1978, de la Higuera 1997]

Identification in the Limit Using Polynomial Time and Data

set $\mathcal{S}$

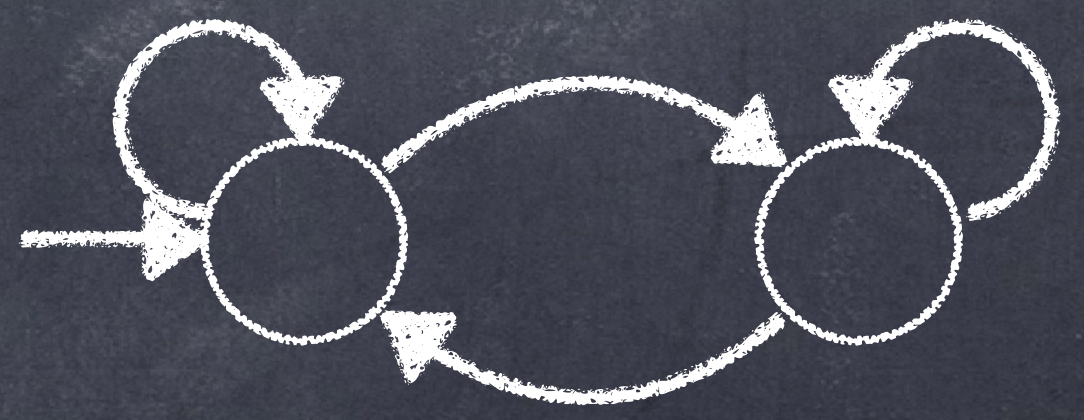
Characteristic



set $\mathcal{S}'$

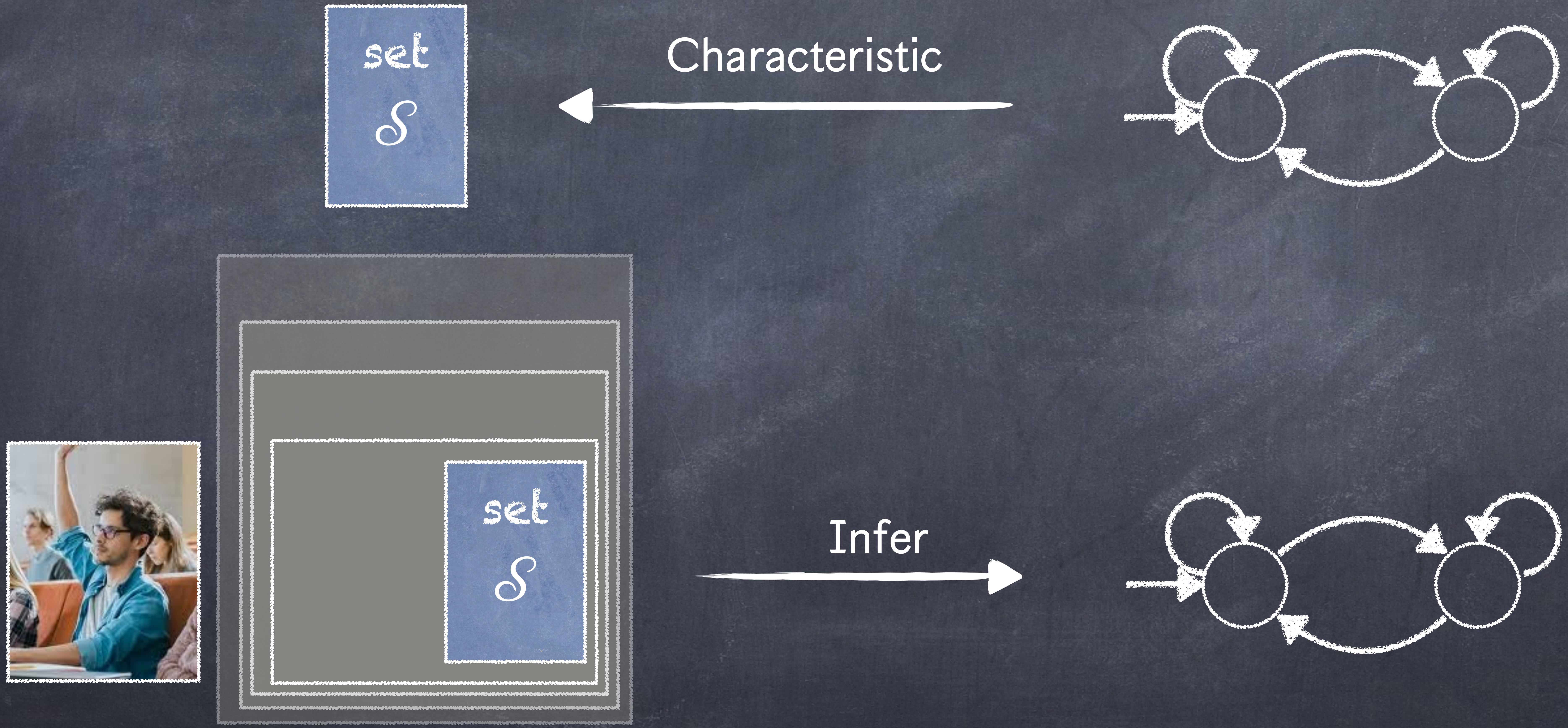
set $\mathcal{S}$

Infer



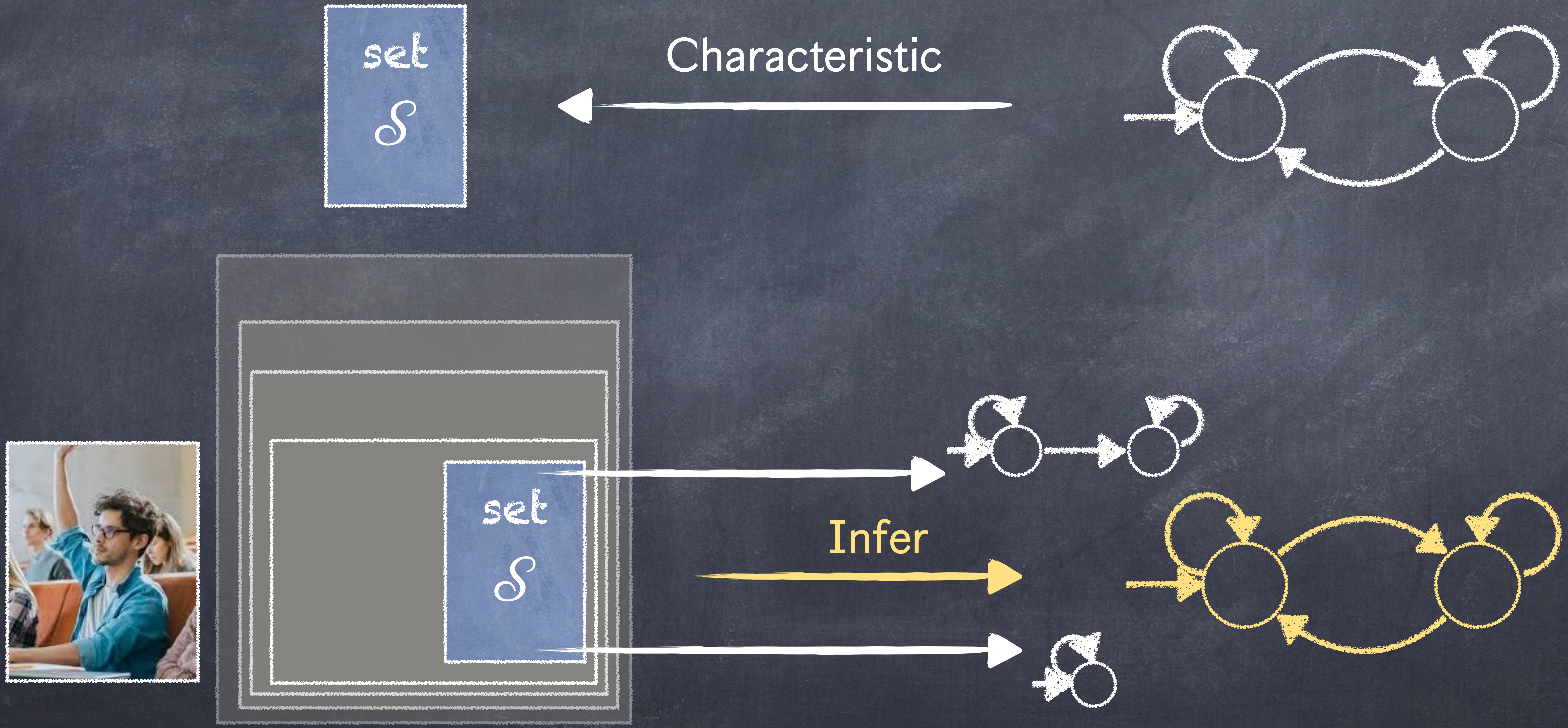
[Gold 1978, de la Higuera 1997]

Identification in the Limit Using Polynomial Time and Data



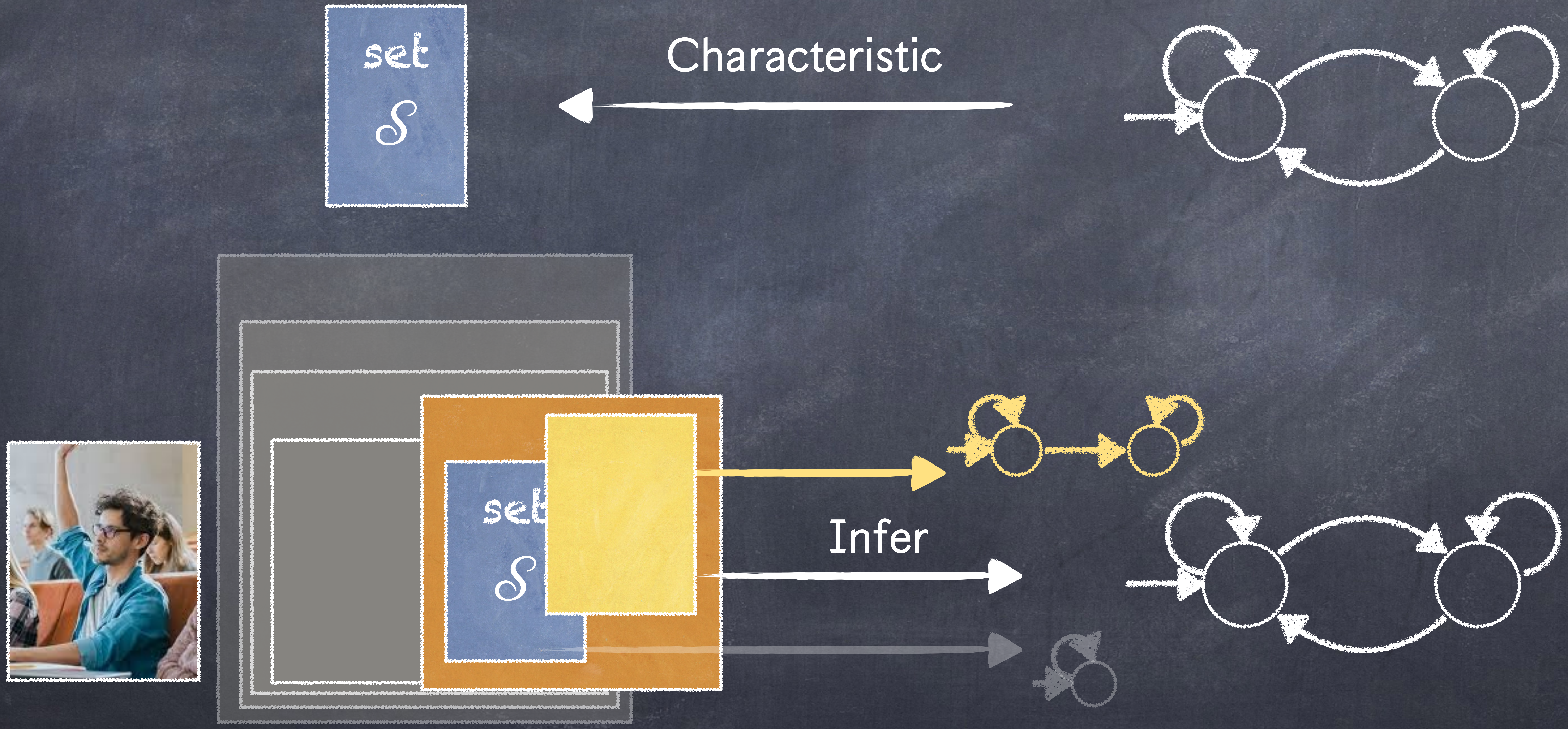
[Gold 1978, de la Higuera 1997]

Identification in the Limit Using Polynomial Time and Data



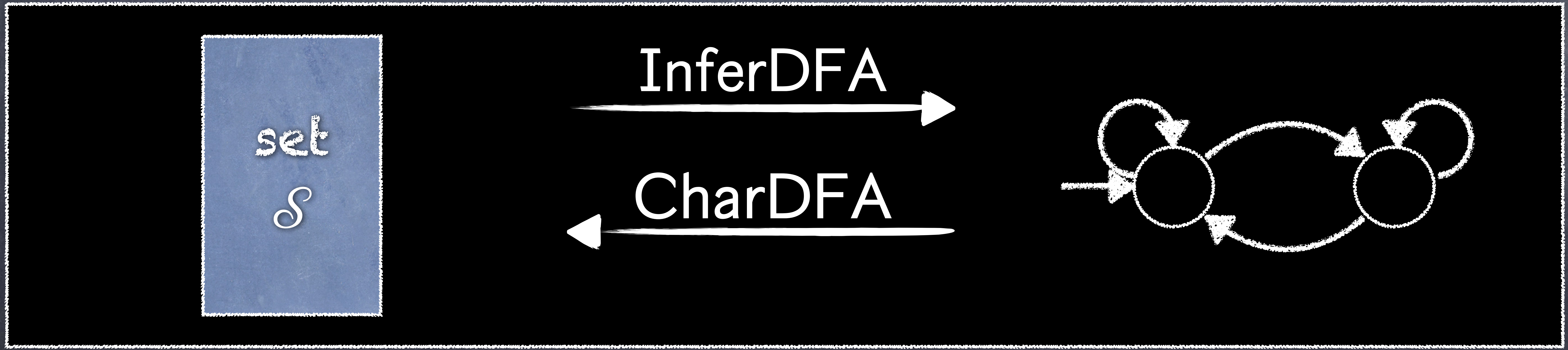
[Gold 1978, de la Higuera 1997]

Identification in the Limit Using Polynomial Time and Data



[Gold 1978, de la Higuera 1997]

Identification in the Limit for DFAs



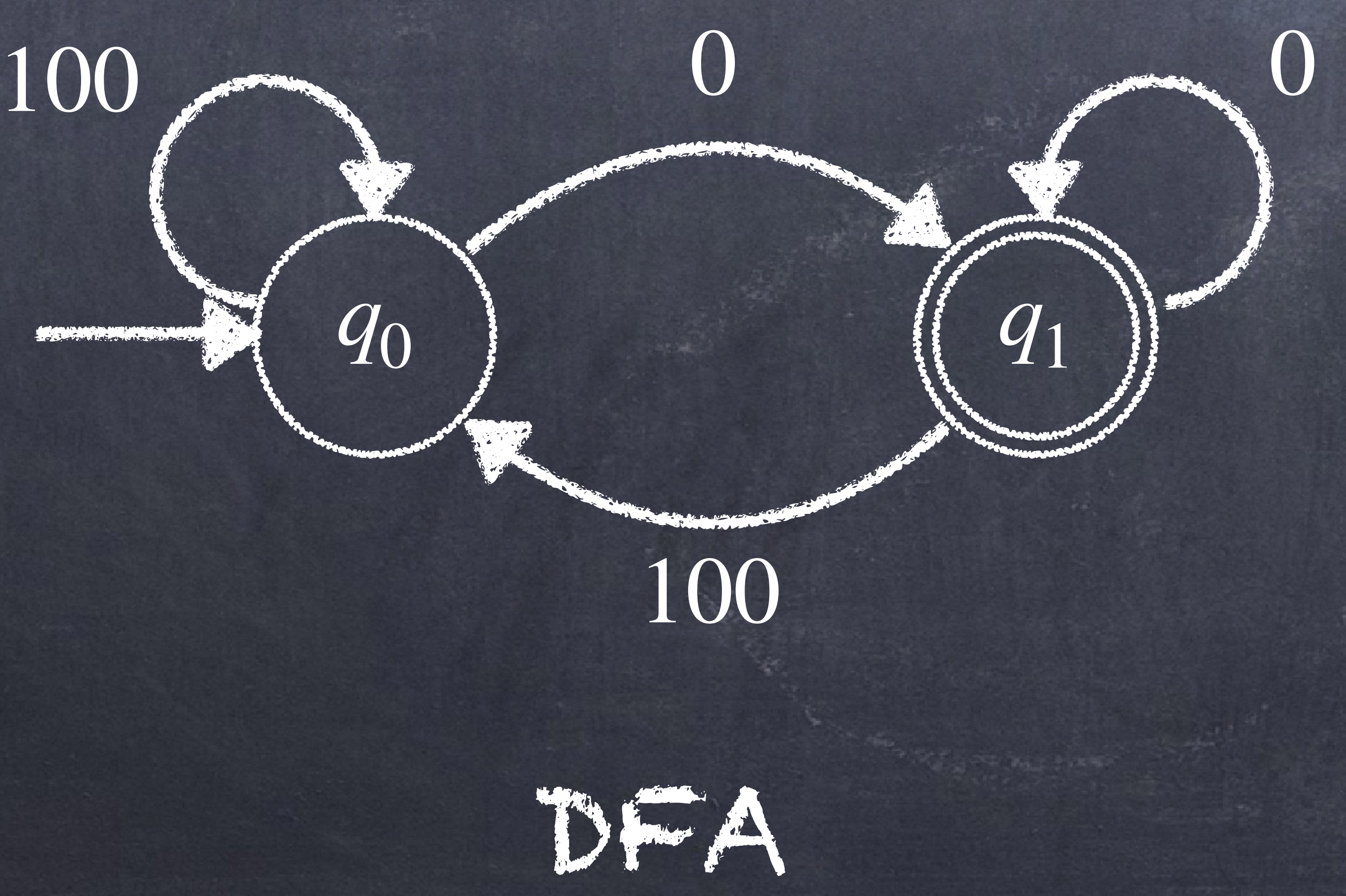
Concrete Sample set

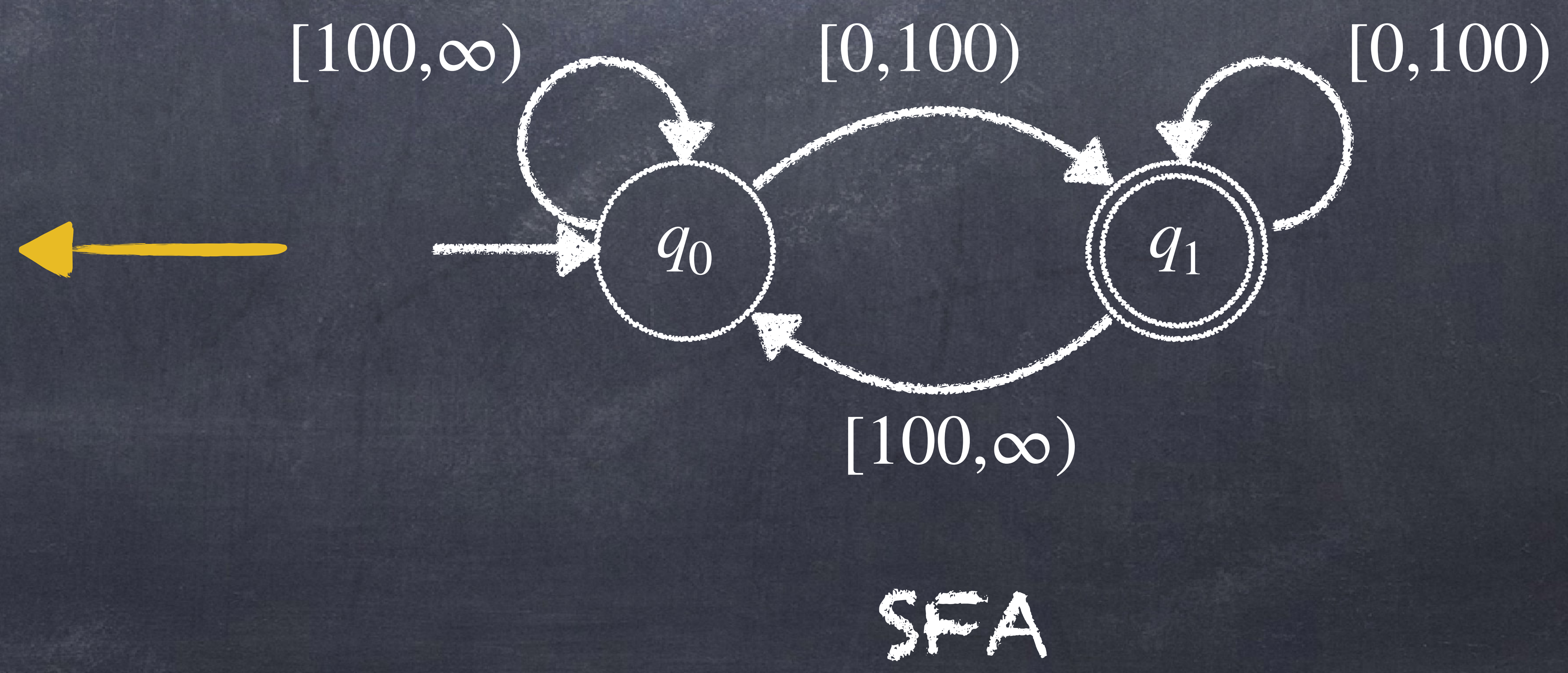
DFA

[Oncina&García 1992]

Identification in the Limit for SFAs - CharSFA

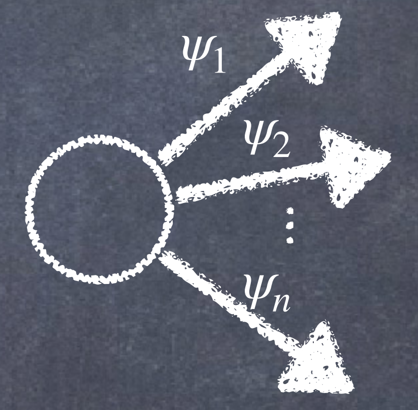
- Learning with respect to the concrete alphabet
- Creating a set of concrete words





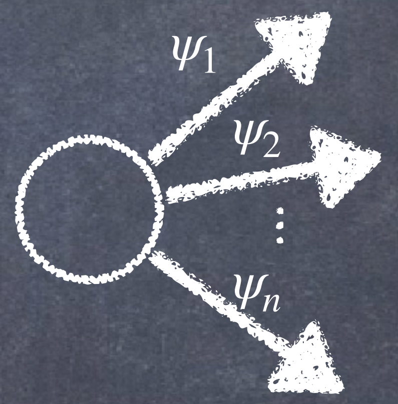
Identification in the Limit for SFAs - CharSFA

- Learning with respect to the concrete alphabet
- Creating a set of concrete words
- concretize$(\langle \psi_1, \ldots, \psi_n \rangle) = \langle \Gamma_1, \ldots, \Gamma_n \rangle$

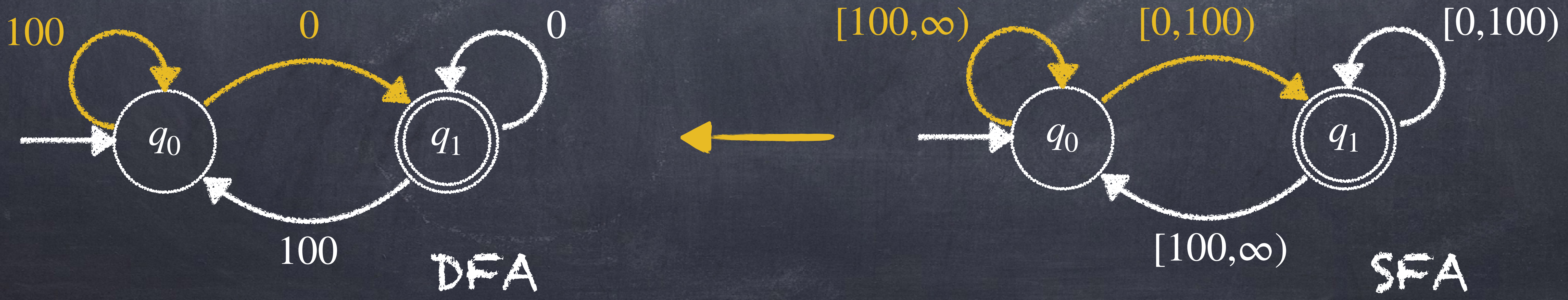


Identification in the Limit for SFAs - CharSFA

- Learning with respect to the concrete alphabet
- Creating a set of concrete words
- concretize($\langle \psi_1, \ldots, \psi_n \rangle$) $= \langle \Gamma_1, \ldots, \Gamma_n \rangle$



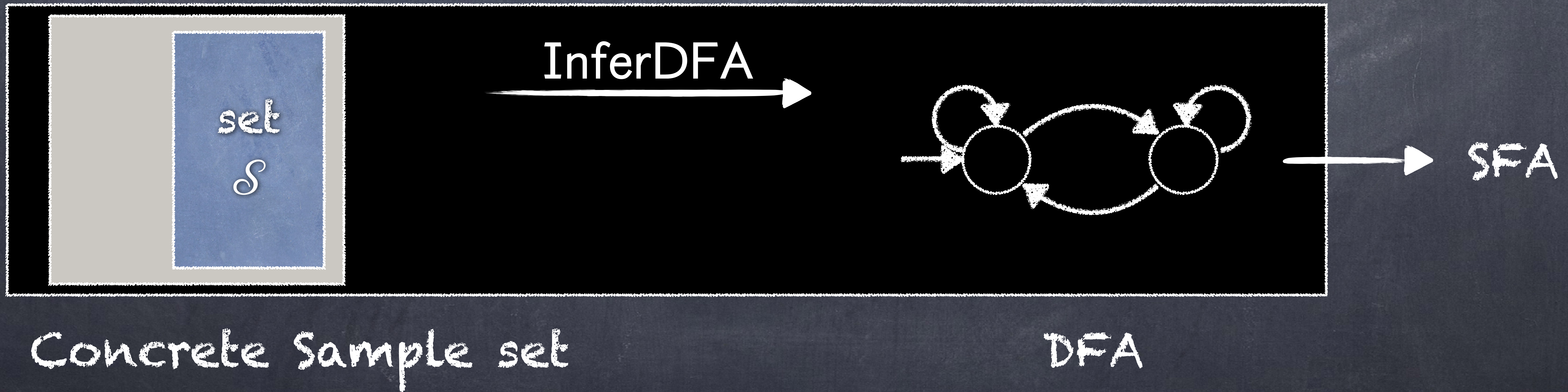
- concretize($\langle [0,100), [100,\infty) \rangle$) $= \langle \{0\}, \{100\} \rangle$



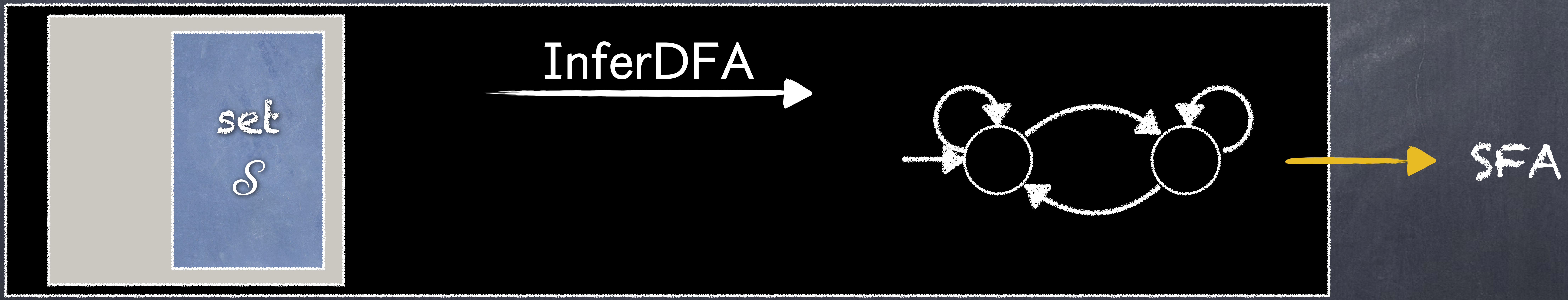
Identification in the Limit for SFAs - CharSFA

- Creating a set of concrete words

Identification in the Limit for SFAs - InferSFA



Identification in the Limit for SFAs – InferSFA

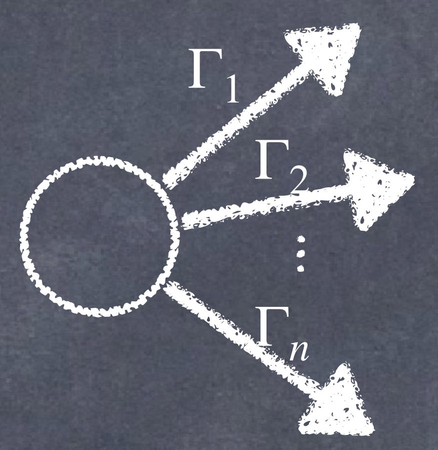


Concrete Sample set

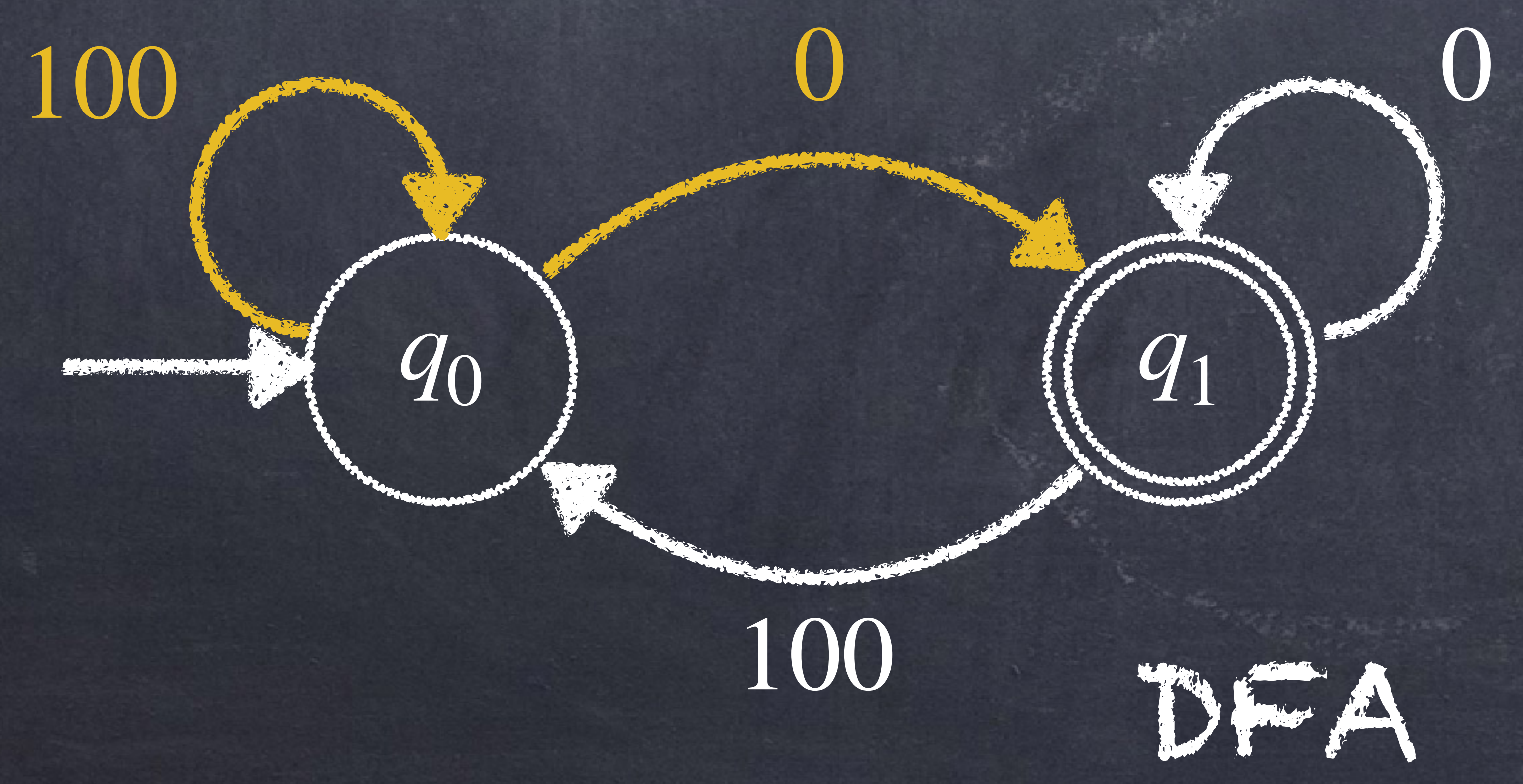
DFA Generalize

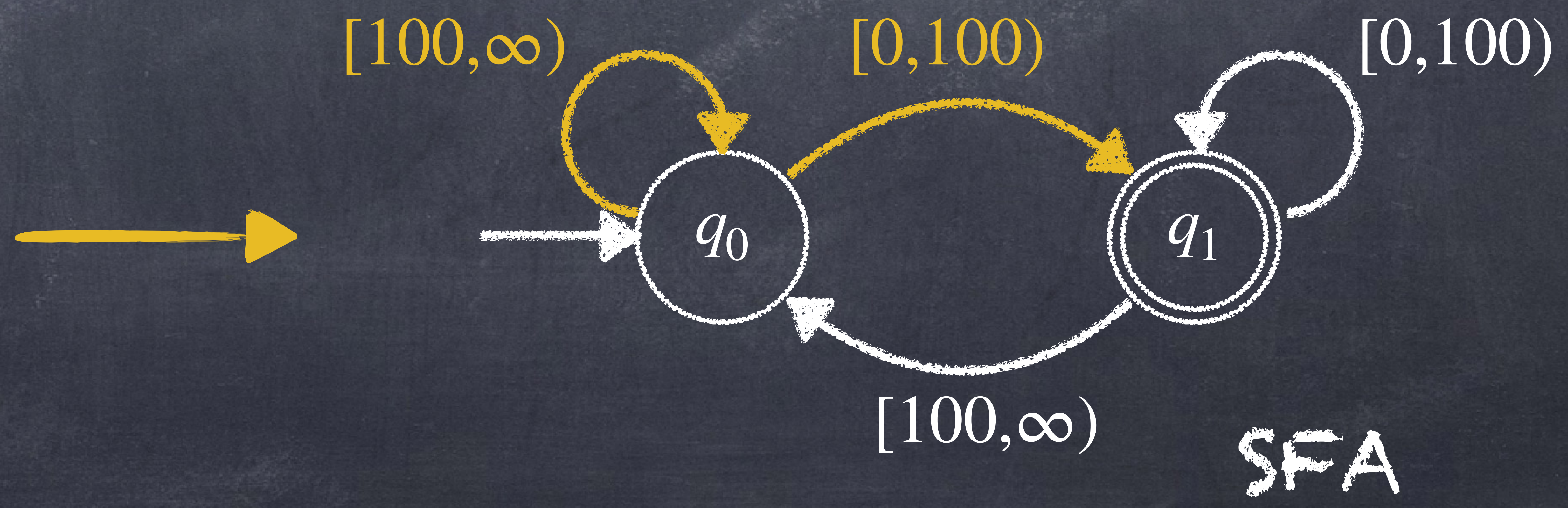
Identification in the Limit for SFAs - InferSFA

- generalize$(\langle \Gamma_1, \ldots, \Gamma_n \rangle) = \langle \psi_1, \ldots \psi_n \rangle$

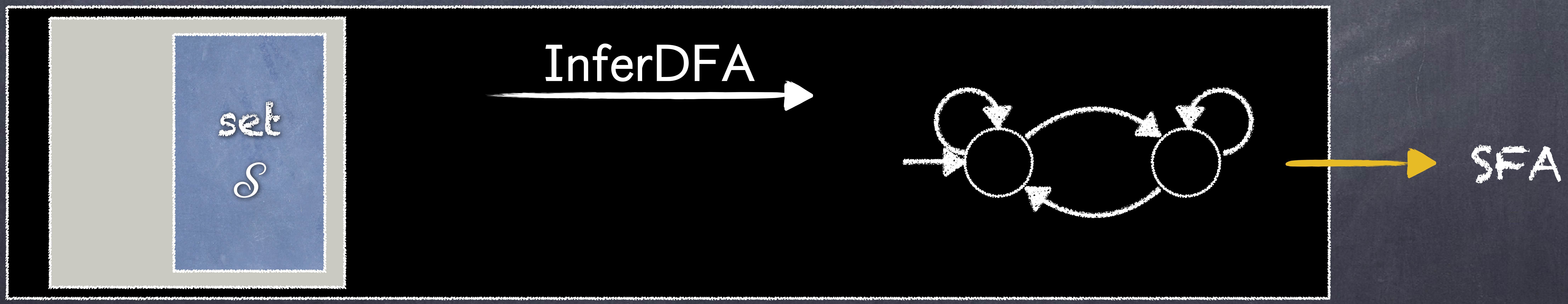


- generalize$(\langle \{0\}, \{100\} \rangle) = \langle [0,100), [100,\infty) \rangle$





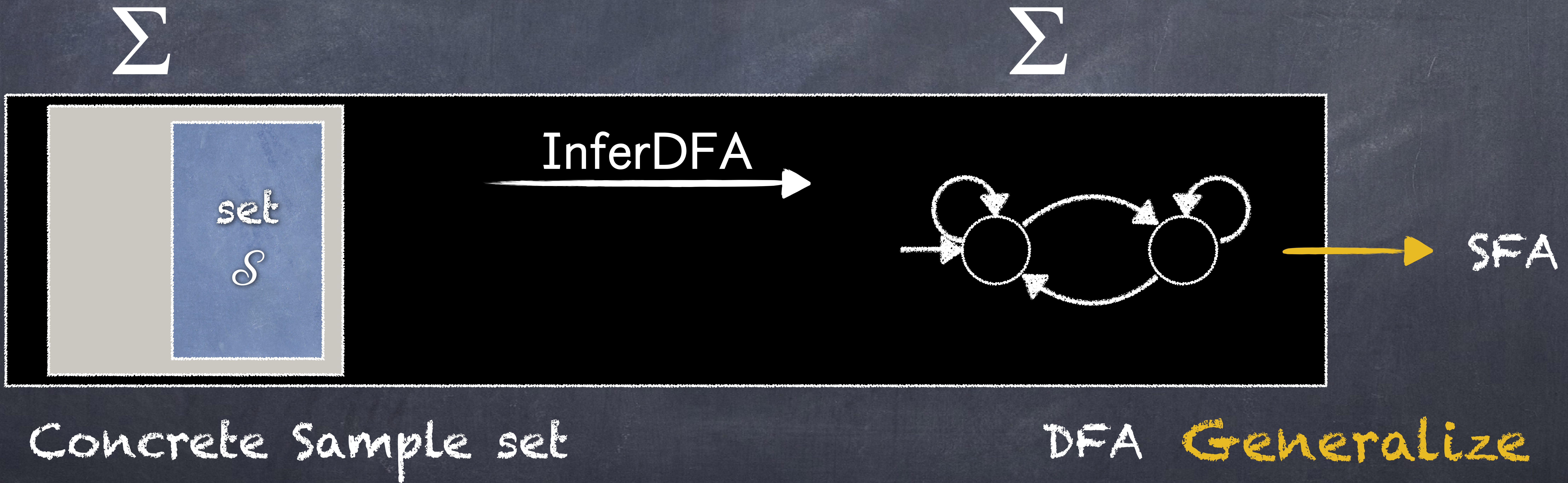
Identification in the Limit for SFAs - InferSFA



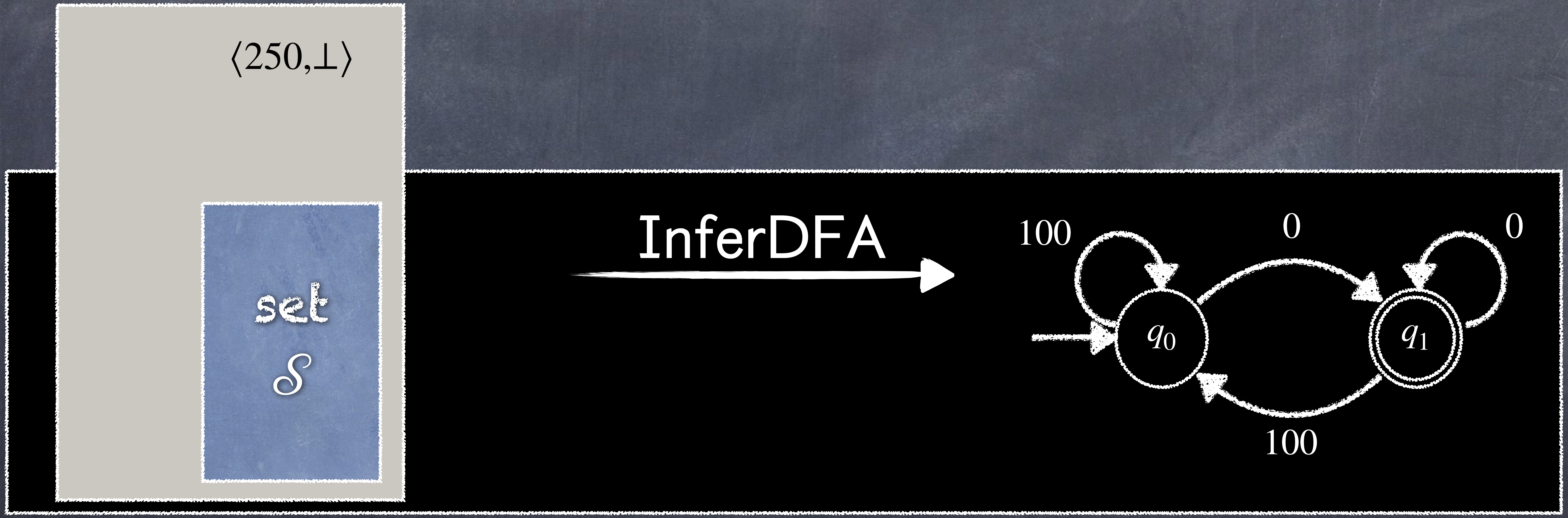
Concrete Sample set

DFA Generalize

Identification in the Limit for SFAs - InferSFA



Identification in the Limit for SFAs – InferSFA

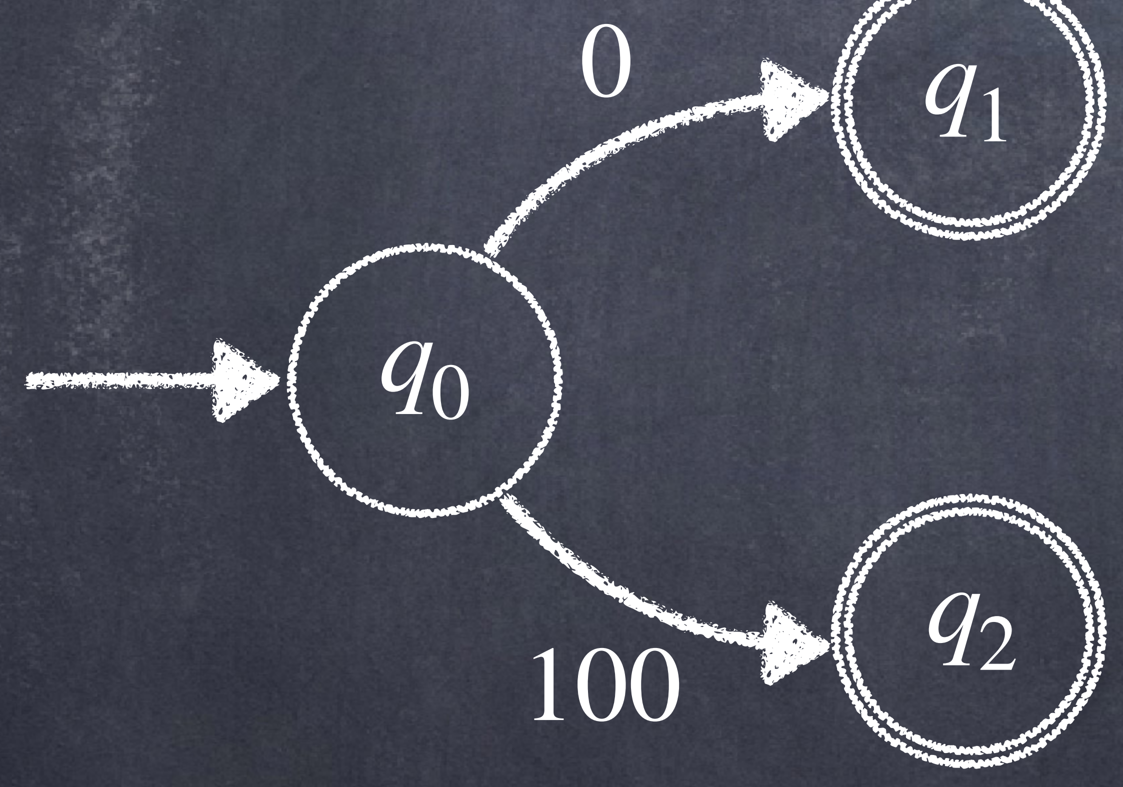


Concrete Sample set

DFA

Identification in the Limit for SFAs - InferSFA

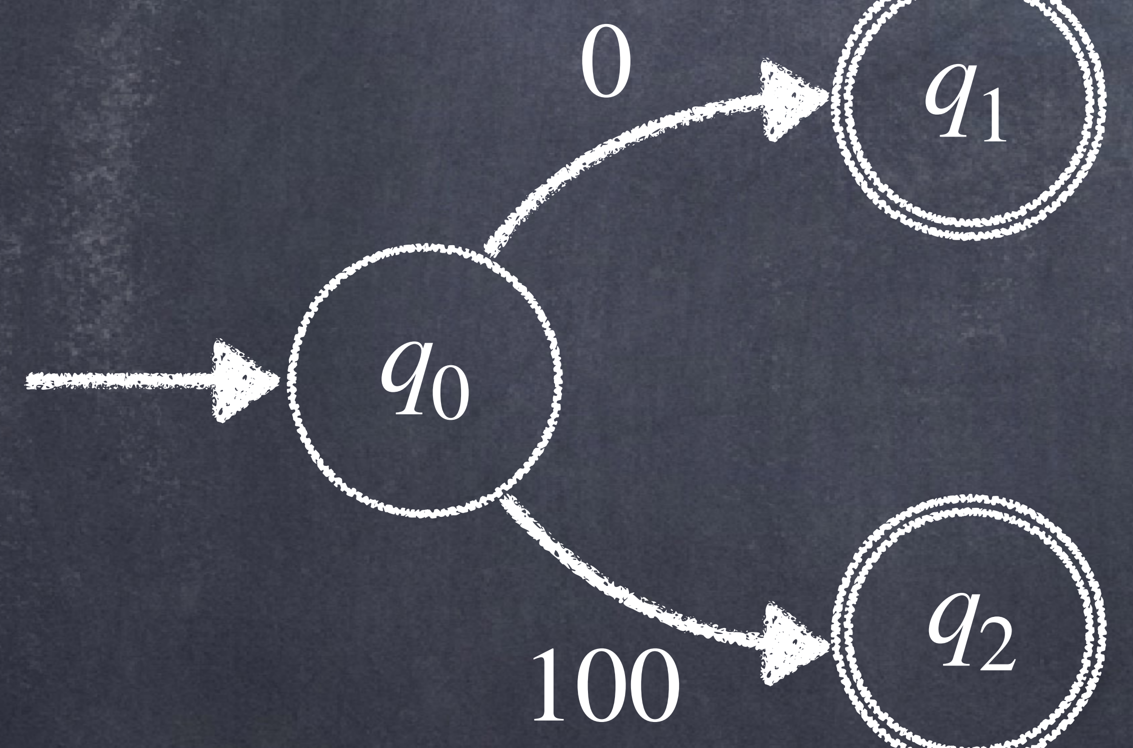
- Additional alphabet adds confusion



$\langle \epsilon, \perp \rangle, \langle 0, \top \rangle, \langle 100, \top \rangle$

Identification in the Limit for SFAs - InferSFA

- Additional alphabet adds confusion

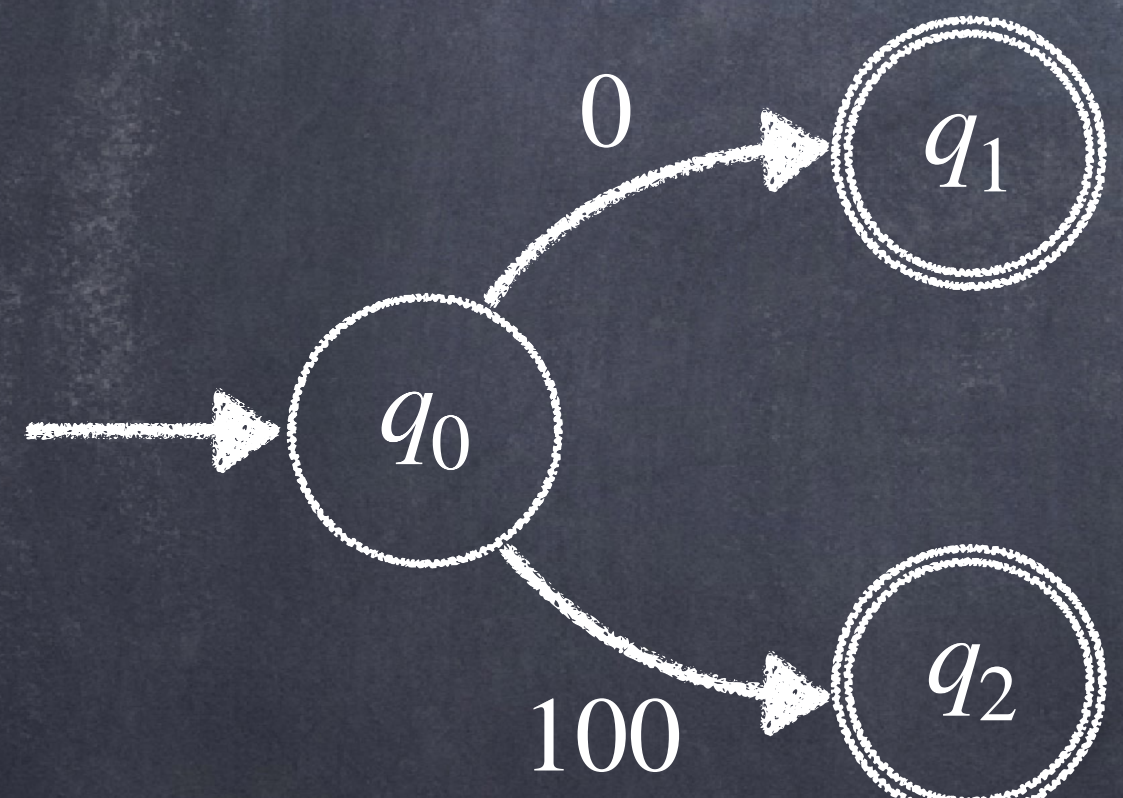


$\langle \epsilon, \bot \rangle, \langle 0, \top \rangle, \langle 100, \top \rangle$

$\langle 0 \cdot 0, \top \rangle, \langle 100 \cdot 0, \bot \rangle$

Identification in the Limit for SFAs - InferSFA

- Additional alphabet adds confusion



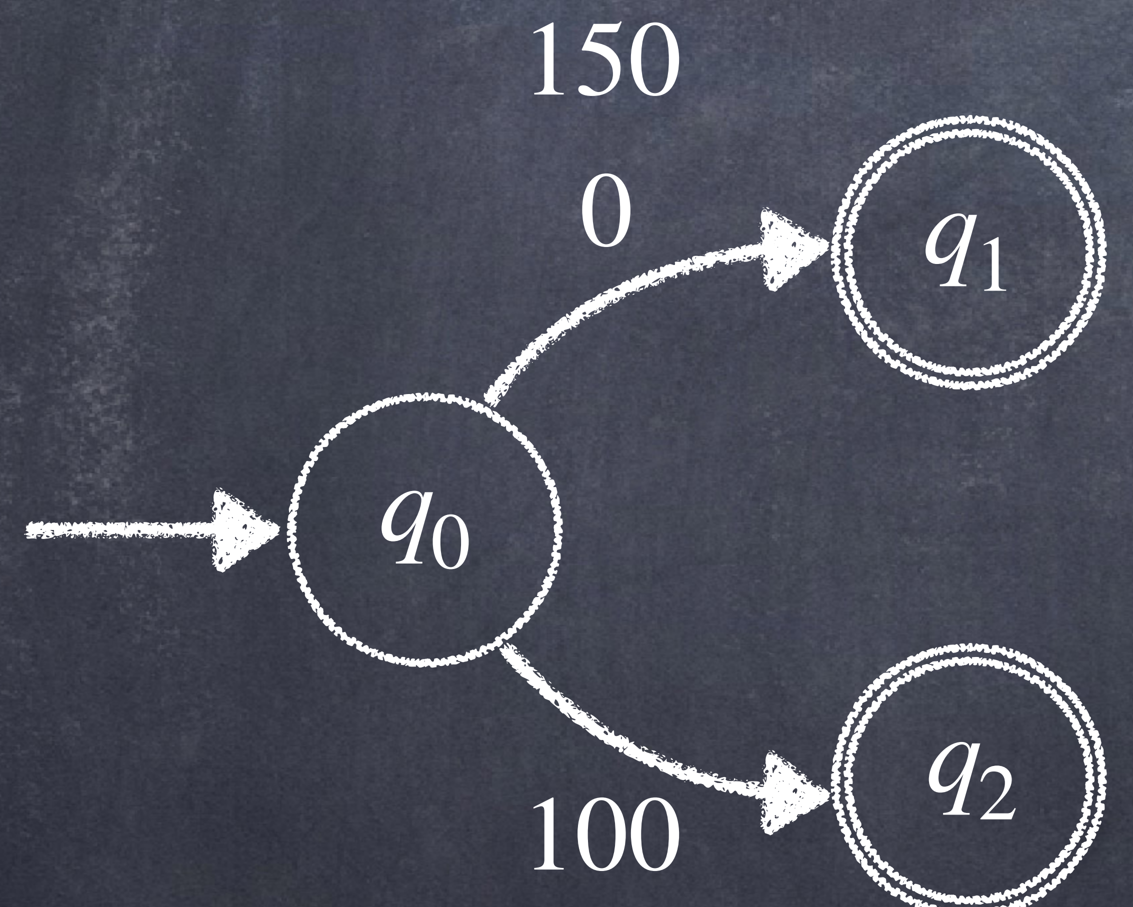
$\langle \epsilon, \bot \rangle, \langle 0, \top \rangle, \langle 100, \top \rangle$

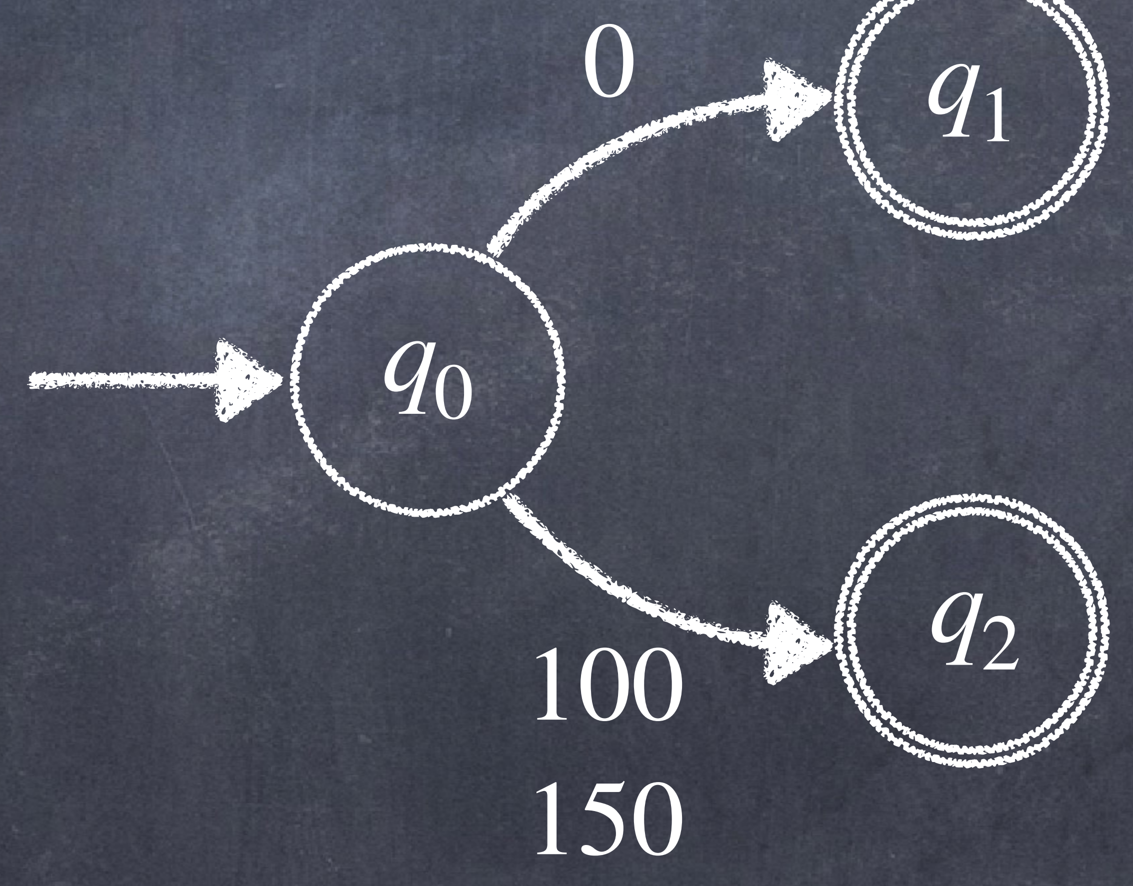
$\langle 0 \cdot 0, \top \rangle, \langle 100 \cdot 0, \bot \rangle$

$\langle 150, \top \rangle$

Identification in the Limit for SFAs - InferSFA

- Additional alphabet adds confusion





$\langle \epsilon, \bot \rangle, \langle 0, \top \rangle, \langle 100, \top \rangle$

$\langle 0 \cdot 0, \top \rangle, \langle 100 \cdot 0, \bot \rangle$

$\langle 150, \top \rangle$

Identification in the Limit for SFAs – InferSFA

$\langle 250, \bot \rangle$

$\langle 75, \top \rangle$

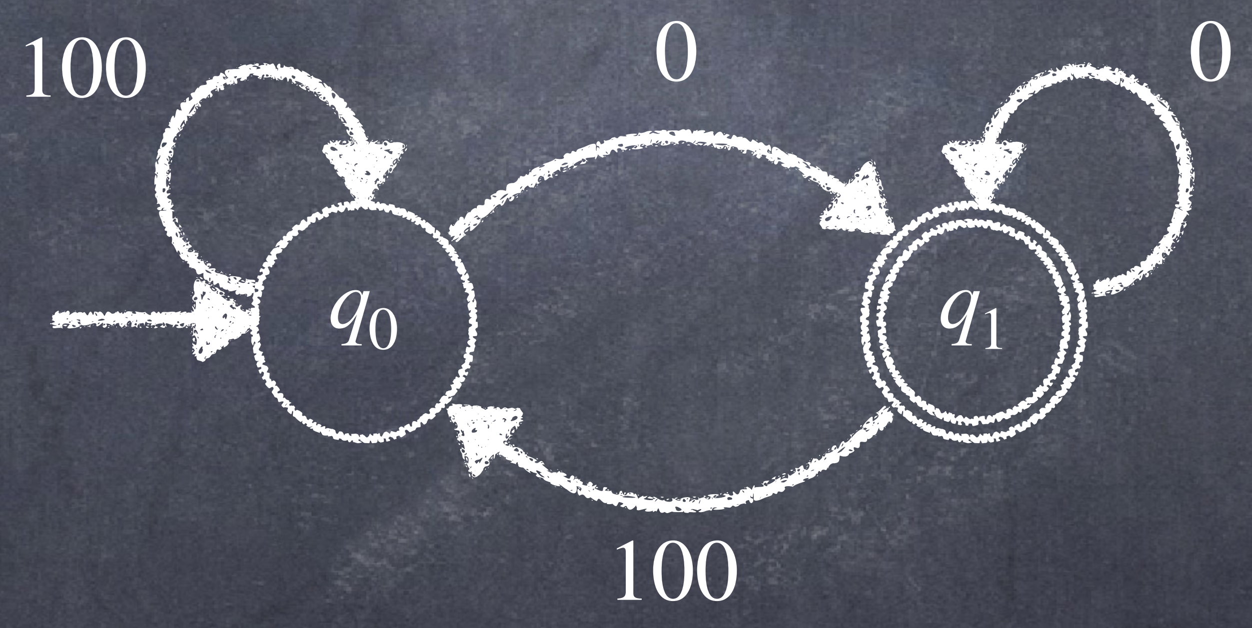
$\langle 100 \cdot 0, \top \rangle$

$\langle 100, \bot \rangle$

$\langle 0, \top \rangle$

Decontaminate

Concrete Sample set



DFA

Identification in the Limit for SFAs – InferSFA

$\langle 250,\bot\rangle$

$\langle 75,\top\rangle$

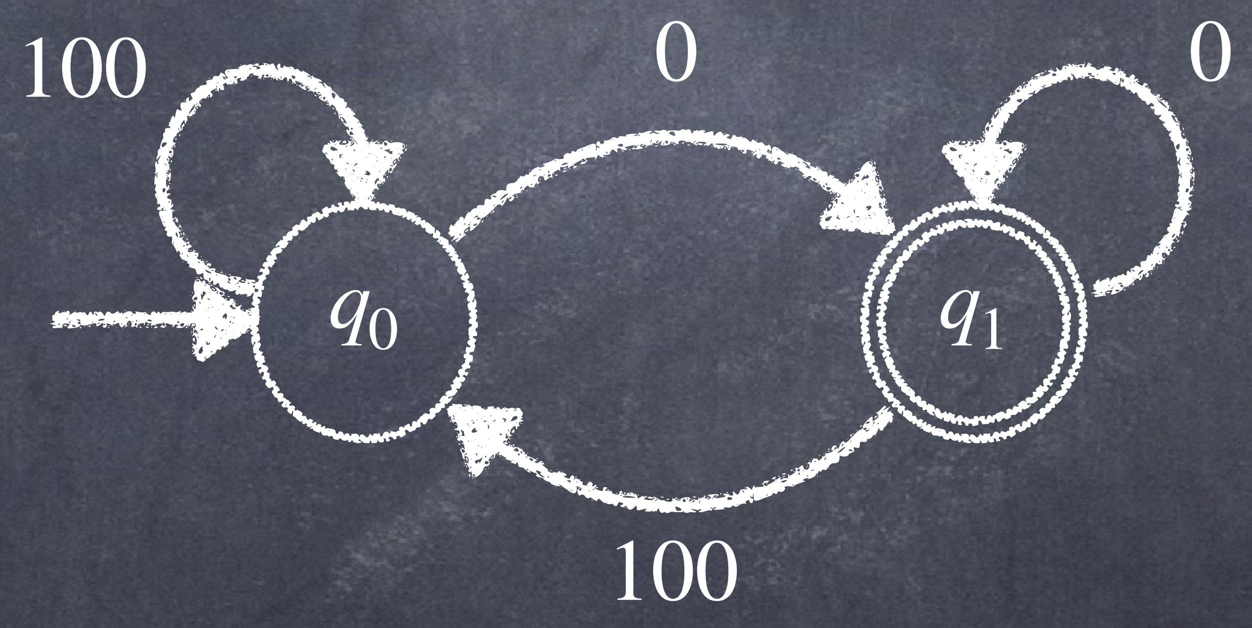
$\langle 100 \cdot 0,\top\rangle$

$\langle 100,\bot\rangle$

$\langle 0,\top\rangle$

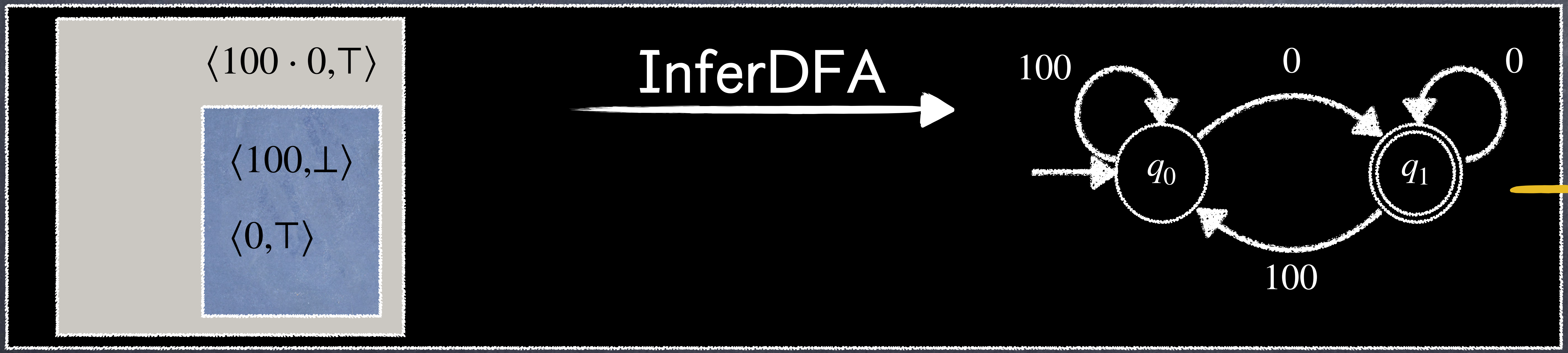
Decontaminate

Concrete Sample set



DFA

Identification in the Limit for SFAs - InferSFA



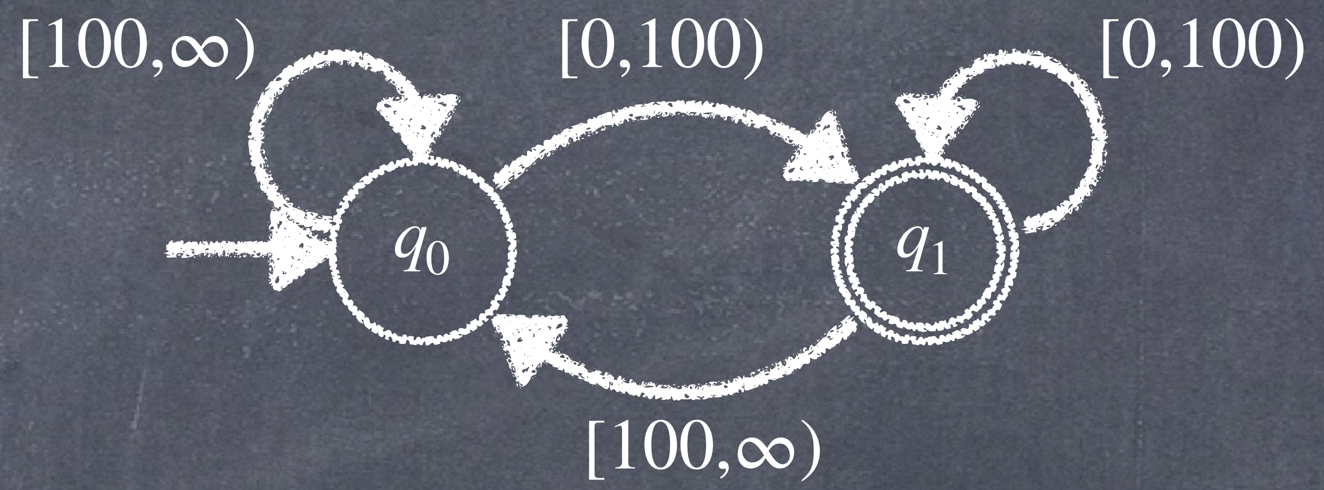
SFA

Concrete Sample set

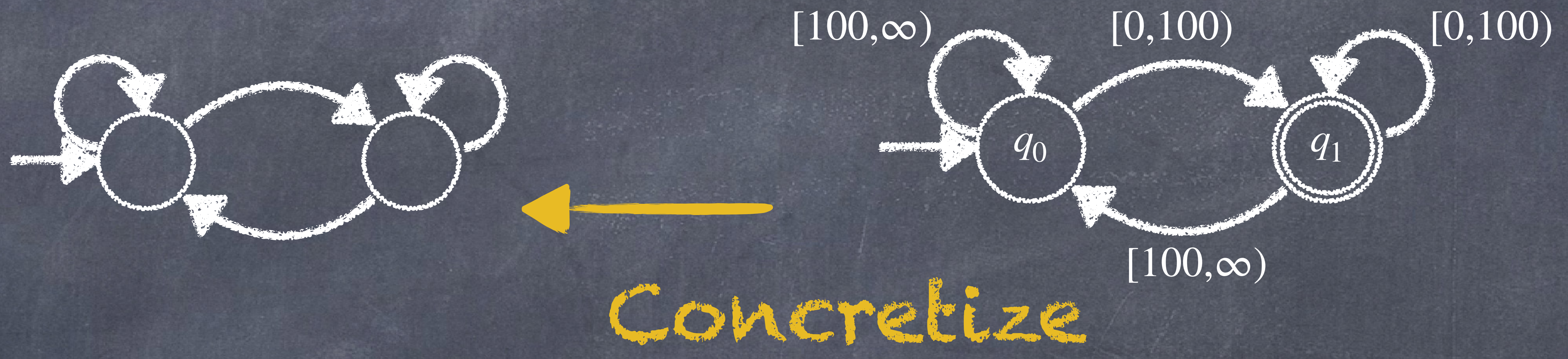
DFA Generalize

The Whole Process

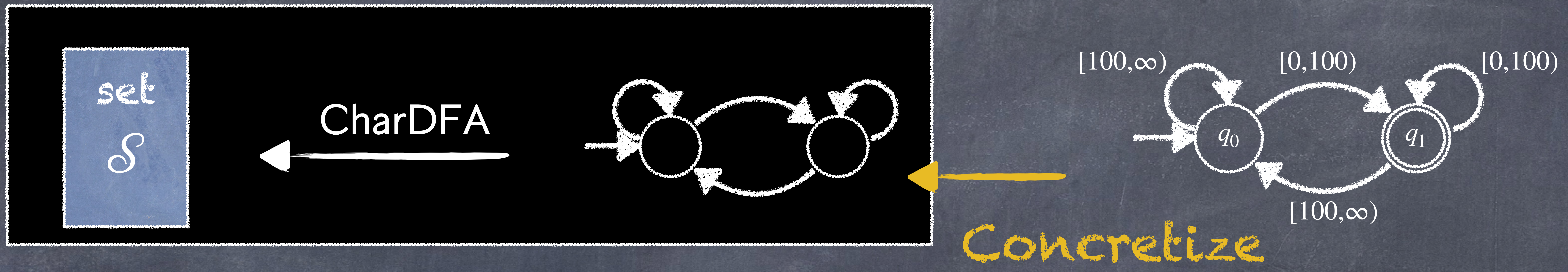
The Whole Process



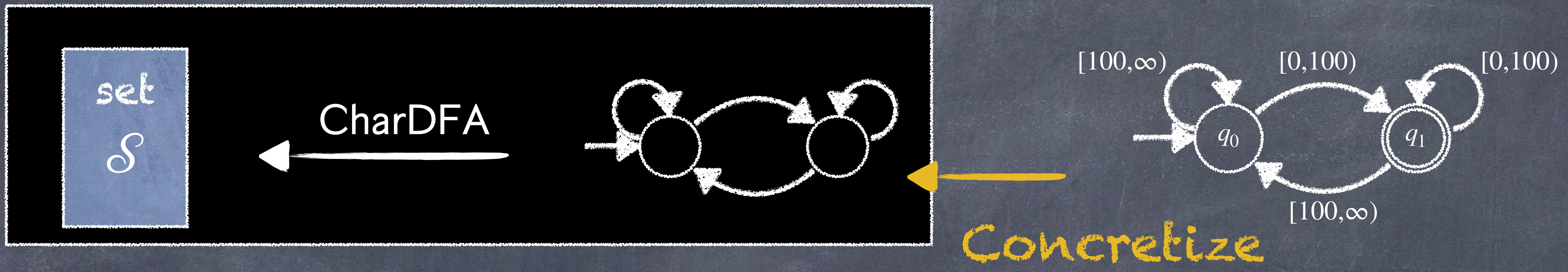
The Whole Process

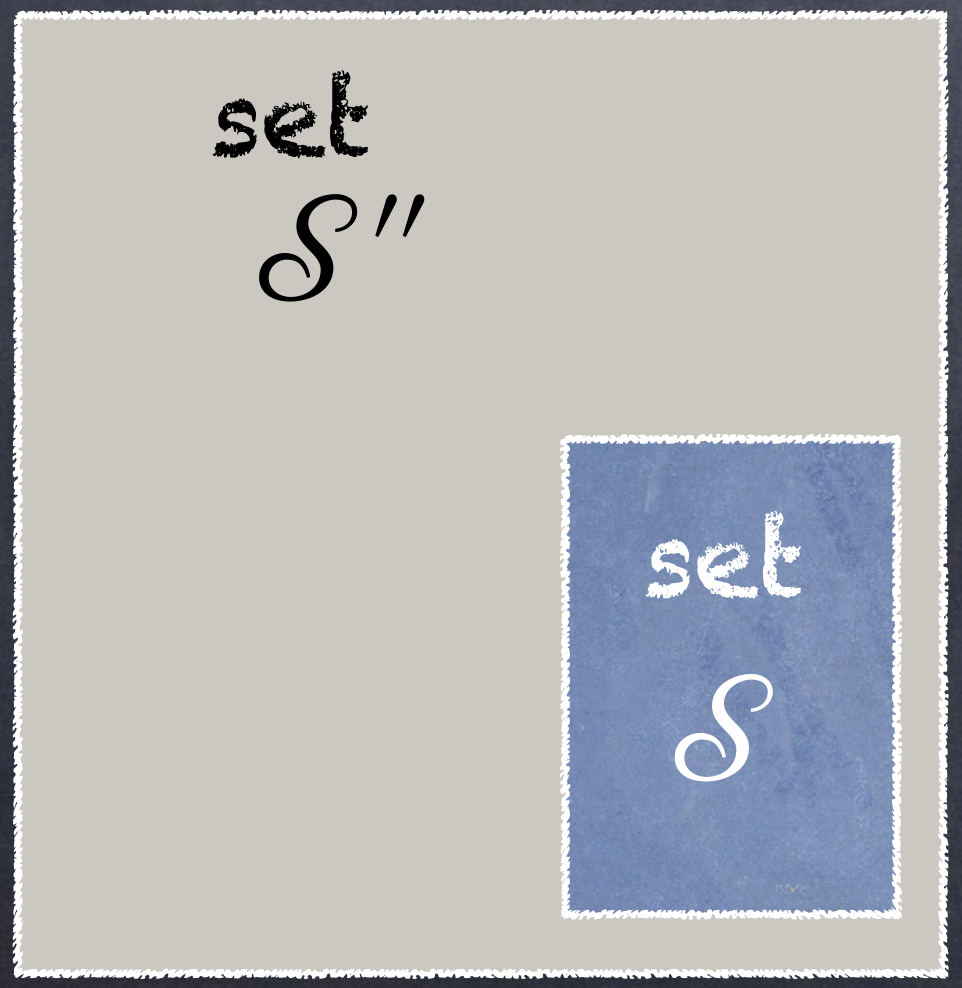


The Whole Process

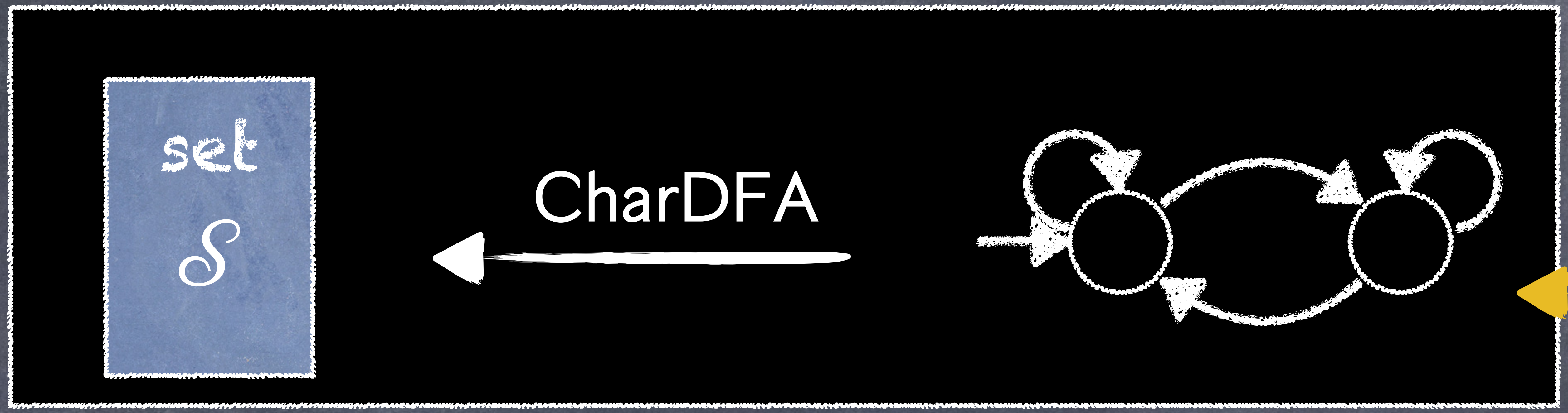


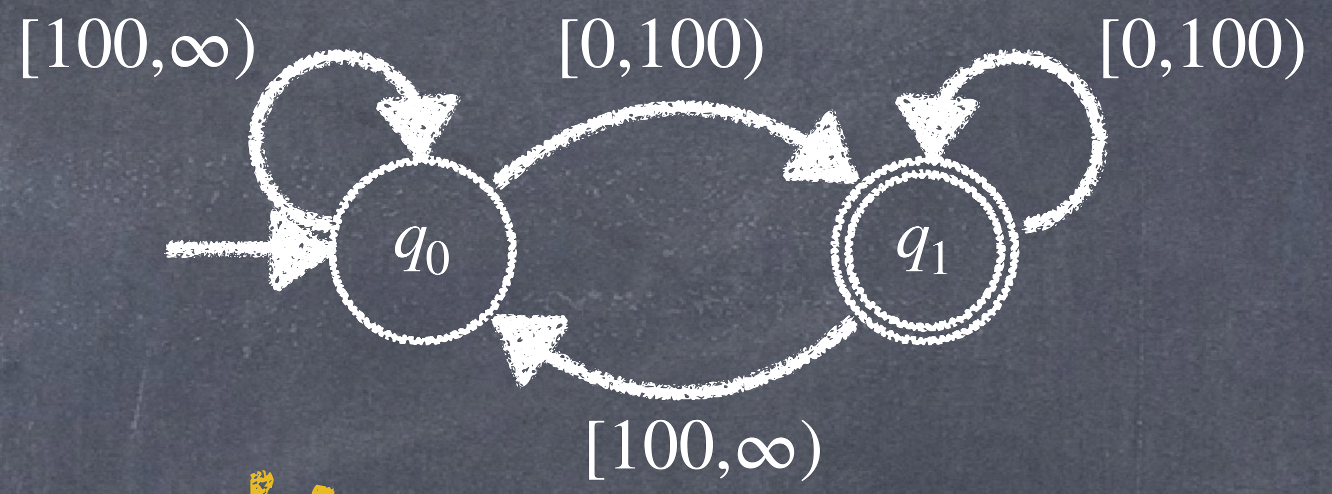
The Whole Process





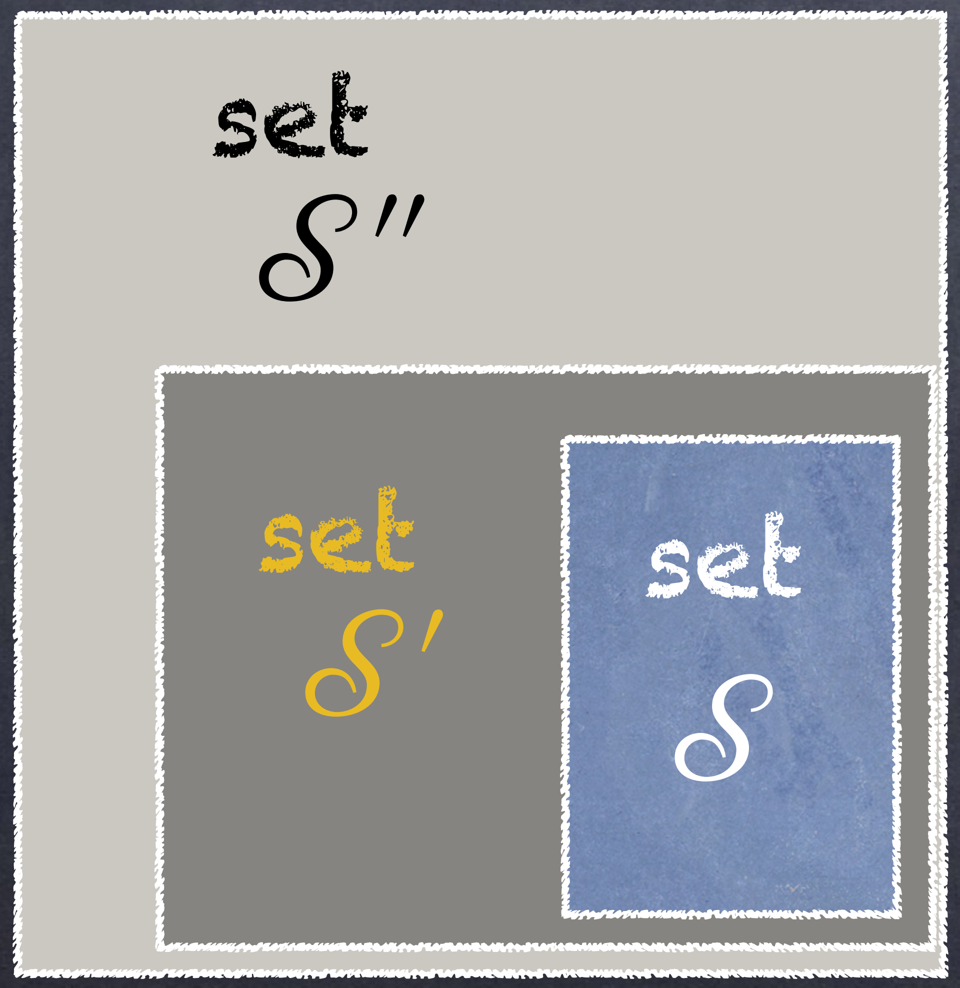
The Whole Process



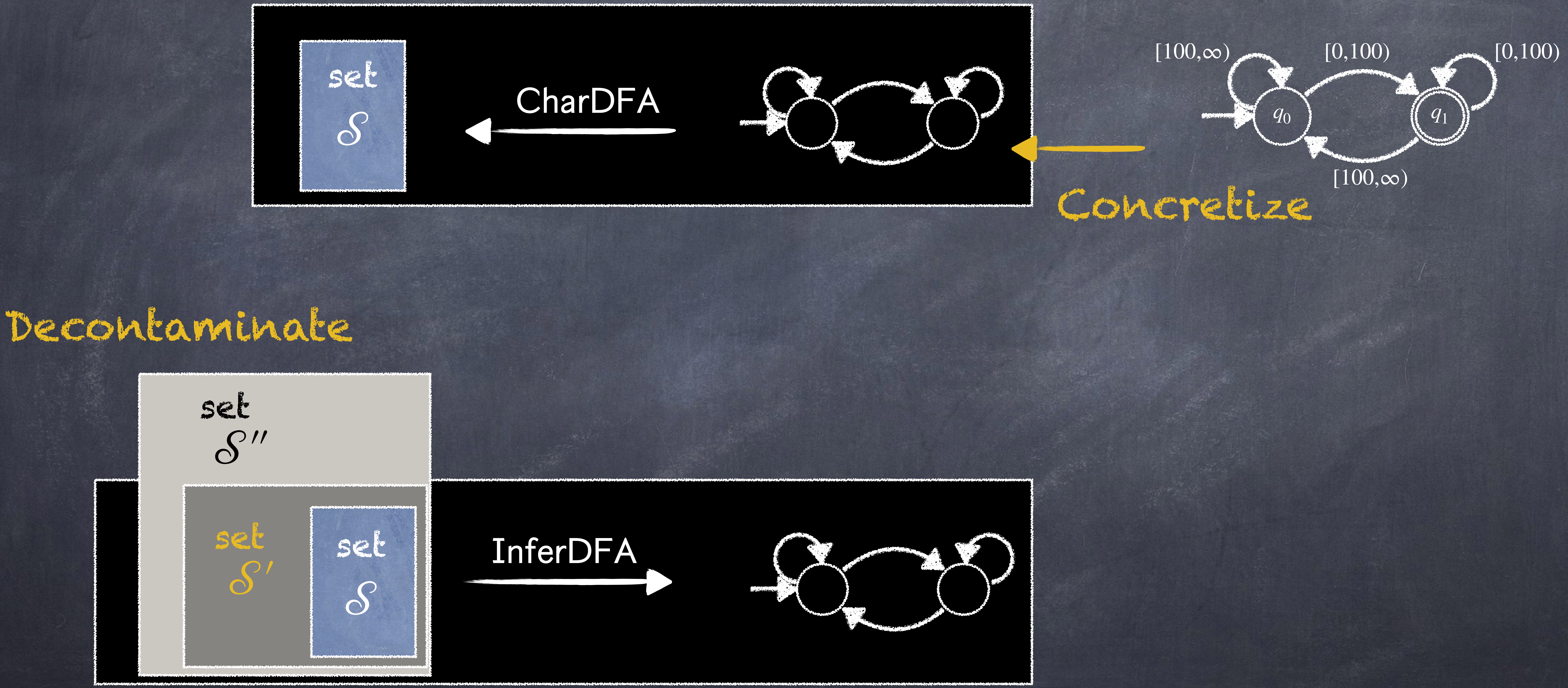


Concretize

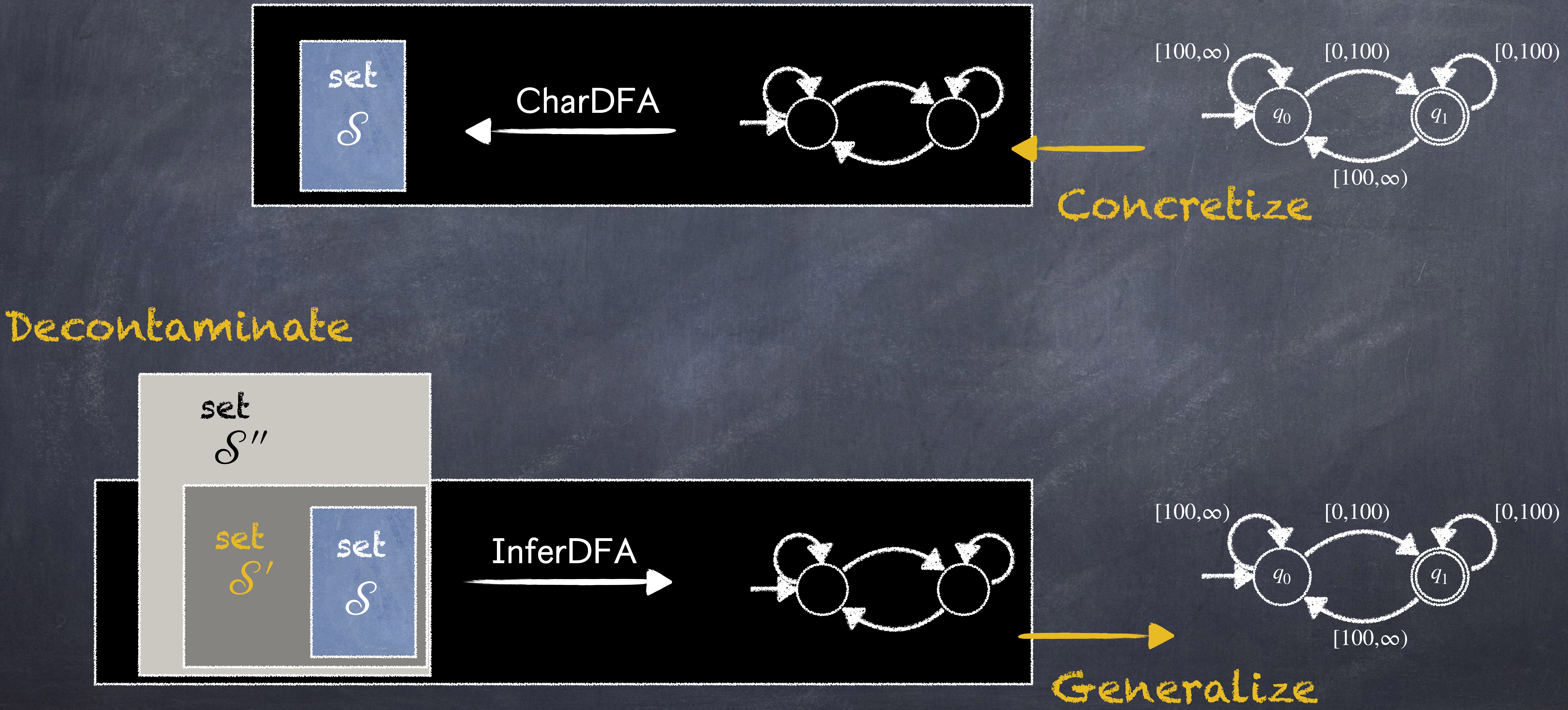
Decontaminate



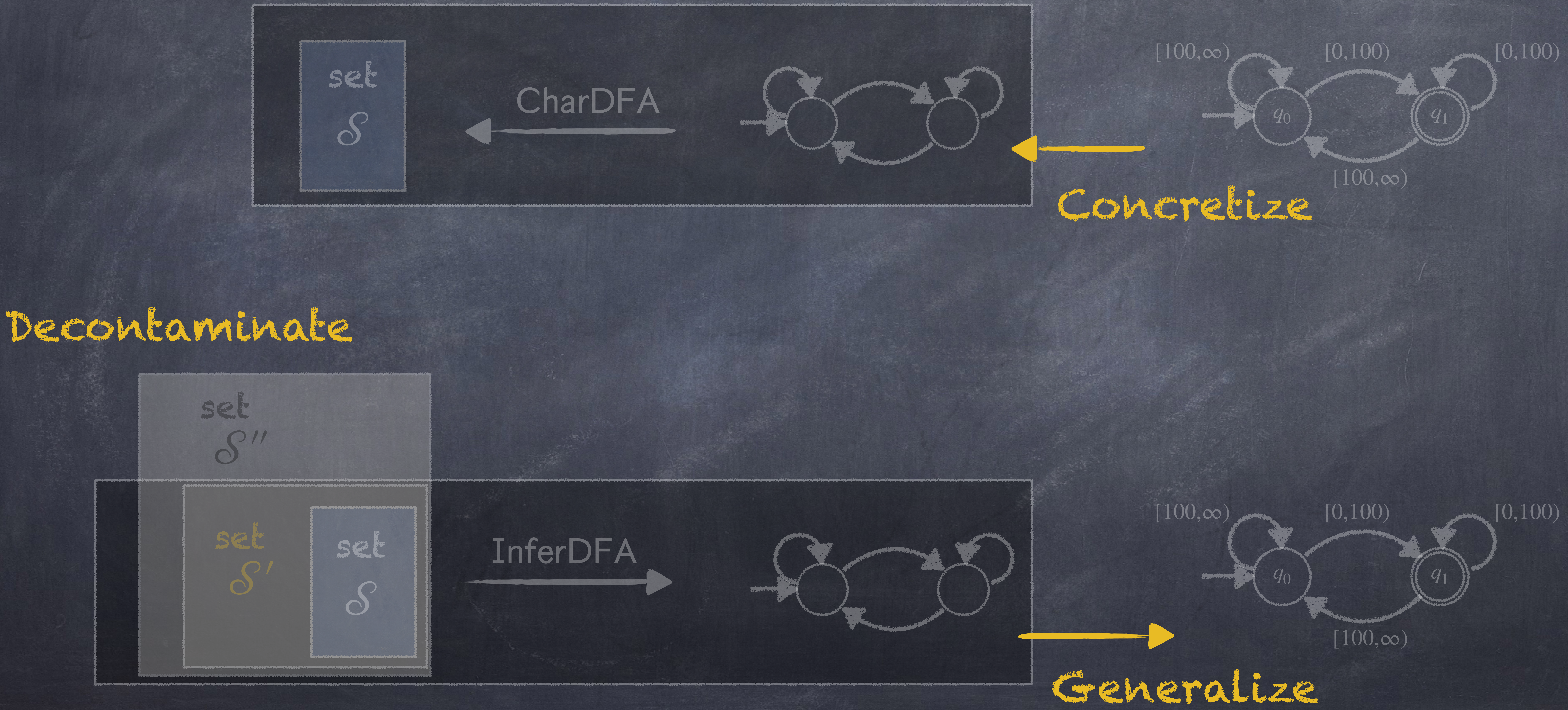
The Whole Process



The Whole Process

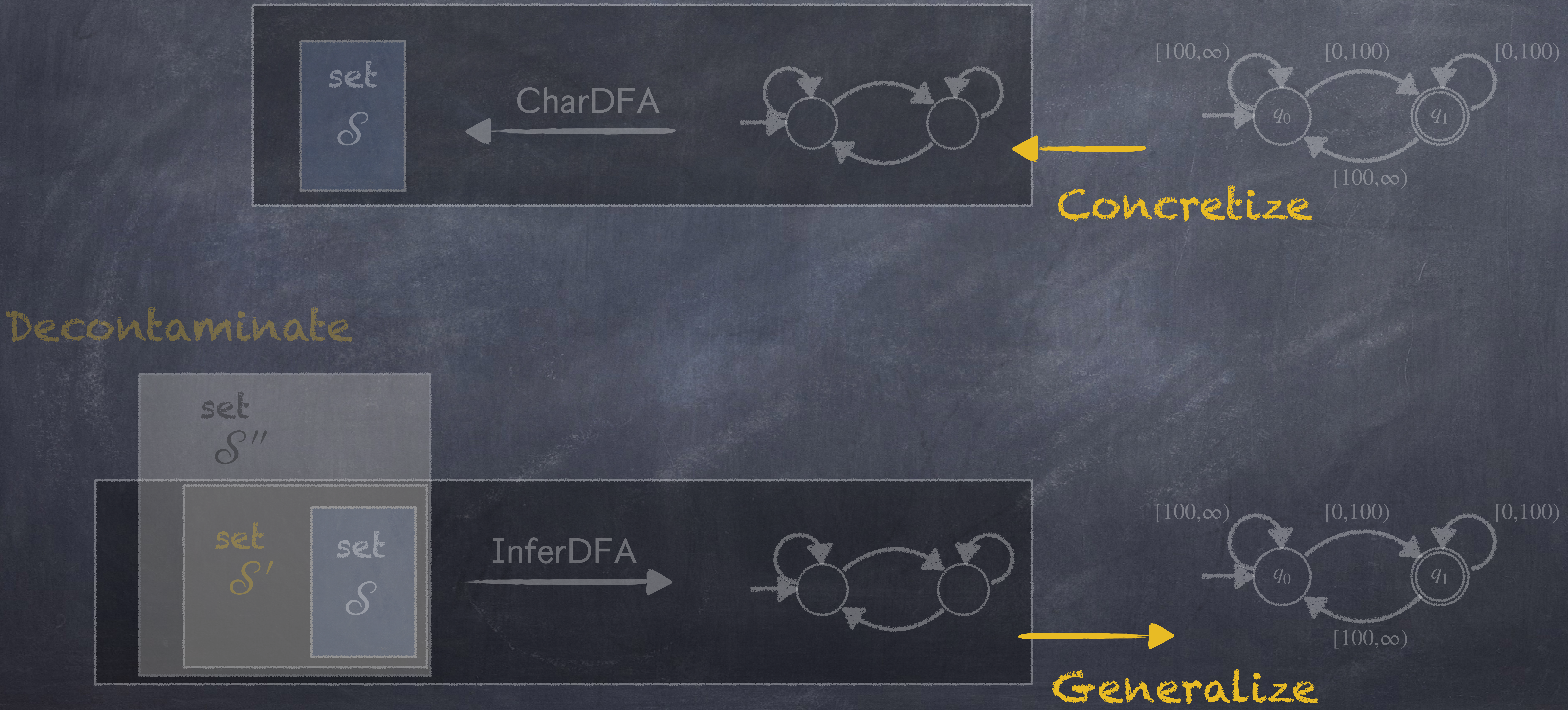


Sufficient Condition



Sufficient Condition

Necessary Condition



Necessary Condition

Summary

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

Summary

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

- Passive Learning

  - Necessary condition & sufficient condition for learning SFAs

Summary

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

- Passive Learning

  - Necessary condition & sufficient condition for learning SFAs
  - SFAs over the propositional algebra are not polynomially learnable

Summary

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

- Passive Learning

  - Necessary condition & sufficient condition for learning SFAs
  - SFAs over the propositional algebra are not polynomially learnable
  - Learning algorithm for SFAs over monotonic algebras

Summary

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

- Passive Learning

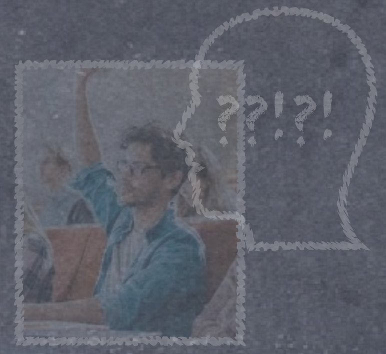
  - Necessary condition & sufficient condition for learning SFAs
  - SFAs over the propositional algebra are not polynomially learnable
  - Learning algorithm for SFAs over monotonic algebras
  - Learning scheme for the paradigm of identification in the limit of SFAs

THANK YOU!

- Active Learning
  - Necessary condition
  - SFAs over the propositional algebra are not polynomially learnable

Questions?

- Passive Learning



  - Necessary condition & sufficient condition for learning SFAs
  - SFAs over the propositional algebra are not polynomially learnable
  - Learning algorithm for SFAs over monotonic algebras
  - Learning scheme for the paradigm of identification in the limit of SFAs