Canonical Algebraic Generators in Automata Learning

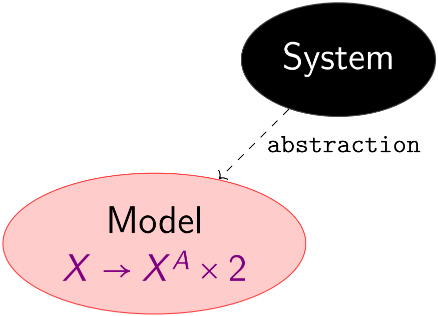
Stefan Jens Zetzsche [1]

[1]University College London

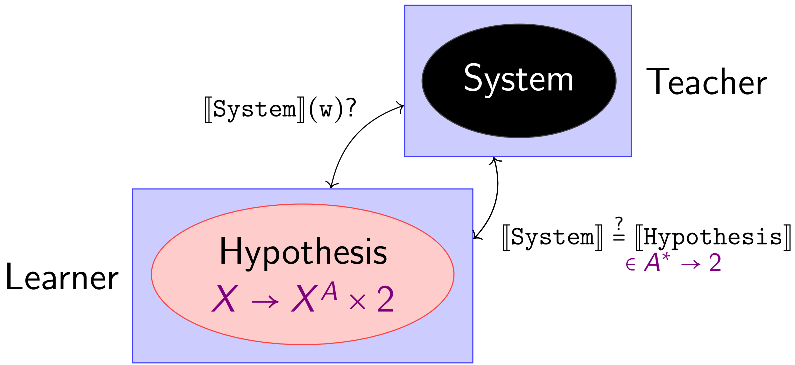
April 27, 2023

Introduction

Introduction: Automata Learning



$$[\![\text{System}]\!] \overset{\supseteq}{=} [\![\text{Model}]\!] \in A^* \to 2$$



L*-algorithm

Angluin. *Learning Regular Sets from Queries and Counterexamples* (1987).

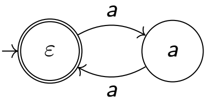
L* for $[\![1 + a \cdot a \cdot a^*]\!] \subseteq \{a\}^*$

	ε
ε	1
a	0

(a)

	ε
ε	1
a	0
aa	1

(b)



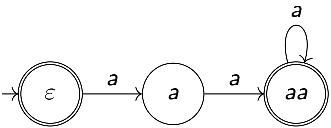
(c)

	ε	a	aa	aaa
ε	1	0	1	1
a	0	1	1	1
aa	1	1	1	1

(d)

	ε	a	aa	aaa
ε	1	0	1	1
a	0	1	1	1
aa	1	1	1	1
aaa	1	1	1	1

(e)



(f)

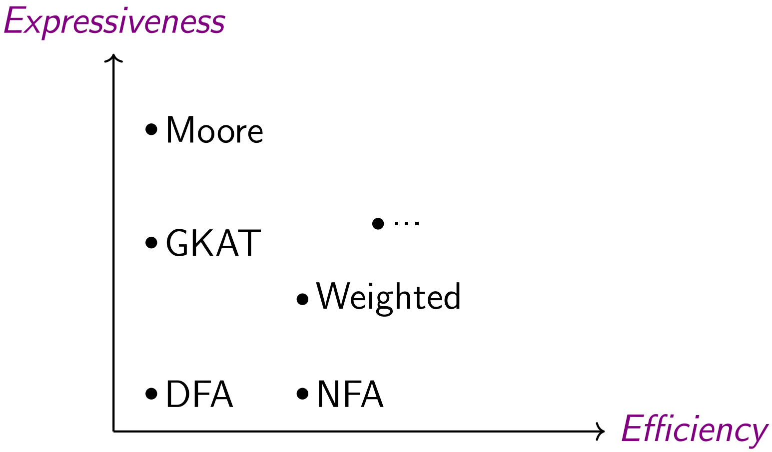
Introduction: The Canonical DFA

Theorem
If L^* is instantiated with $[\![\mathscr{X}]\!]$, then it terminates with the unique size-minimal DFA $m(\mathscr{X})$.

Angluin. *Learning Regular Sets from Queries and Counterexamples* (1987).

Introduction: Further Directions

Introduction: Further Directions



Contributions

Contributions

Learning Guarded Programs[1] (Chapter 3)

Canonical Automata[2] (Chapter 4)

Generating Monadic Closures[3] (Chapter 5)

[1] Accepted at MFPS 2022

[2] Accepted at MFPS 2021

[3] Submitted to CALCO 2023

Contributions: Learning Guarded Programs

Learning Guarded Programs (Chapter 3)

Canonical Automata (Chapter 4)

Generating Monadic Closures (Chapter 5)

Contributions: Learning Guarded Programs

Guarded Kleene Algebra with Tests (GKAT)

$$b, c \in \mathsf{BExp}_T ::= 0 \mid 1 \mid t \in T \mid b + c \mid b \cdot c \mid \overline{b}$$

$$\begin{aligned} e, f \in \mathsf{GExp}_{\Sigma,T} ::= {} & 0 \mid 1 \mid p \in \Sigma \mid e \cdot f \mid b \in \mathsf{BExp}_T \mid \\ & \texttt{if } b \texttt{ then } e \texttt{ else } f \mid \\ & \texttt{while } b \texttt{ do } e \mid \end{aligned}$$

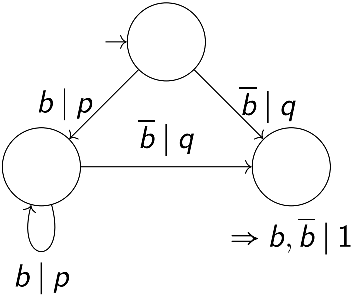
Smolka, Foster, Hsu, Kappe, Kozen, and Silva. *Guarded Kleene Algebra with Tests: Verification of Uninterpreted Programs in Nearly Linear Time* (2019).

Contributions: Learning Guarded Programs

$$[\![(\texttt{while } b \texttt{ do } p) \cdot q]\!] = \{\overline{b}qb, \overline{b}q\overline{b}, bp\overline{b}qb, bp\overline{b}q\overline{b}, \ldots\}$$

$$\in (\mathsf{At} \cdot \Sigma)^* \cdot \mathsf{At} \to 2$$

$$\cong (\mathsf{At} \cdot \Sigma)^* \to 2^{\mathsf{At}}$$



$$\in X \to (2 + \Sigma \times X)^{\mathsf{At}}$$

Kozen and Tseng. *The Böhm-Jacopini Theorem is False, Propositionally* (2008).

Contributions: Learning Guarded Programs

- Minimization of GKAT automata
- Native GL^* learning algorithm
- Correctness proof
- Complexity result
- GL^* is more efficient than L^*
- Implementation[4] of L^* and GL^* in OCaml

[4] https://github.com/zetzschest/gkat-automata-learning

Contributions: Canonical Automata

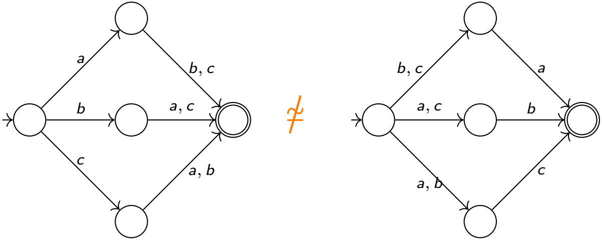
Learning Guarded Programs (Chapter 3)

Canonical Automata (Chapter 4)

Generating Monadic Closures (Chapter 5)

Contributions: Canonical Automata

$\{ab, ac, ba, bc, ca, cb\} \subseteq \{a, b, c\}^*$



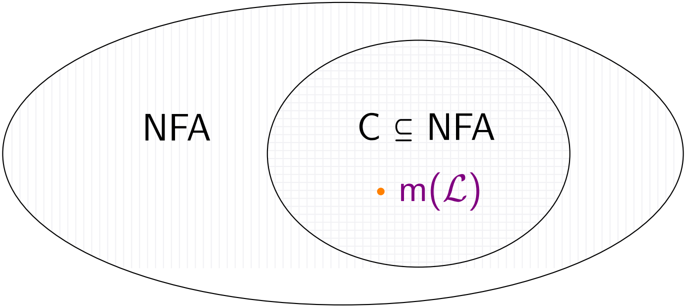
Arnold, Dicky, and Nivat. *A Note About Minimal Non-Deterministic Automata* (1992).

Contributions: Canonical Automata

Is there a *canonical* minimal NFA?

Contributions: Canonical Automata

Canonical RFSA[5], Distromaton[6], Átomaton[7], ...



[5] Denis, Lemay, and Terlutte. *Residual Finite State Automata* (2001)

[6] Myers, Adamek, Milius, and Urbat. *Coalgebraic Constructions of Canonical Nondeterministic Automata* (2015).

[7] Brzozowski and Tamm. *Theory of Átomata* (2014)

Contributions: Canonical Automata

$$\begin{array}{c} X \to X^A \times B \\ \Big\downarrow \\ Y \to T(Y)^A \times B \end{array}$$

$$B, X \in \mathrm{Alg}(S)$$

Rutten, Bonsangue, Bonchi, and Silva. *Generalizing Determinization from Automata to Coalgebras* (2013).
Arbib and Manes. *Fuzzy Machines in a Category* (1975).

Contributions: Canonical Automata

Name	S	T
Canonical RFSA[8]	CSL	CSL
Canonical Nominal RFSA[9]	Nom-CSL	Nom-CSL
Minimal Xor Automaton[10]	$\mathbb{Z}_2$-VSP	$\mathbb{Z}_2$-VSP
Átomaton[11]	CABA	CSL
Distromaton[12]	CDL	CSL
Xor-CABA Automaton[13]	CABA	$\mathbb{Z}_2$-VSP

[8] Example 4.4.2

[9] Example 4.4.3

[10] Example 4.4.4

[11] Section 4.5.3

[12] Section 4.5.4

[13] Section 4.5.5

Contributions: Canonical Automata

- Unifying category-theoretical framework
- Bialgebras, monads, distributive laws, generators
- Improve expressivity of previous work
- Uncover new canonical acceptor
- Abstract minimality result and size comparison

Contributions: Generating Monadic Closures

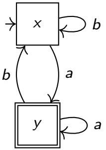
Learning Guarded Programs (Chapter 3)

Canonical Automata (Chapter 4)

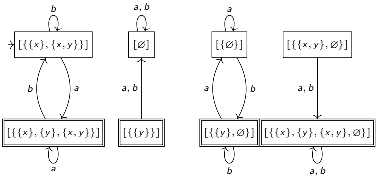
Generating Monadic Closures (Chapter 5)

Contributions: Generating Monadic Closures

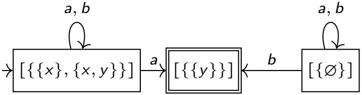
$(a+b)^*a \subseteq \{a,b\}^*$



(a) Minimal Coalgebra



(b) Minimal Bialgebra



(c) Canonical Automaton

Step 1: (a) → (b)   Step 2: (b) → (c)

Contributions: Generating Monadic Closures

Step 1: (a) → (b)

$$\overline{(\cdot)}^{\mathbb{X}} : \mathsf{Sub}_{\mathscr{C}}(X) \to \mathsf{Sub}_{\mathscr{C}^T}(\mathbb{X})$$

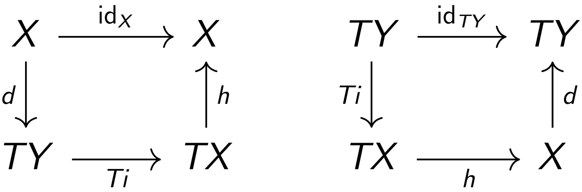
(a) Minimal Coalgebra

(b) Minimal Bialgebra

Theorem 5.2.1

Contributions: Generating Monadic Closures

Step 2: (b) → (c)



(b) Minimal Bialgebra

(c) Canonical Automaton

Definition 5.3.1

Contributions: Generating Monadic Closures

Step 1:

- Generalisation of algebraic closure
- Monad on subobjects in factorization system
- Morphisms between monads

Step 2:

- Abstract theory of generators and bases
- Generalisation of representation theory
- Bases for bialgebras and monoidal product

Summary

Learning Guarded Programs[14] (Chapter 3)

Canonical Automata[15] (Chapter 4)

Generating Monadic Closures[16] (Chapter 5)

[14] Accepted at MFPS 2022

[15] Accepted at MFPS 2021

[16] Submitted to CALCO 2023

The End

Thanks for listening!