Learning Automata with Side-Effects

Gerco van Heerdt Matteo Sammartino Alexandra Silva

University College London

September 21, 2020

The $\mathtt{L}^\star$ algorithm (Angluin, 1987)

Finite alphabet A

System behaviour captured by a **regular language** $\mathcal{L} \subseteq A^*$

$\mathtt{L}^\star$ learns *minimal* DFA for $\mathcal{L}$ assuming an *oracle* that answers

- **Membership queries**

$$\boxed{w \in \mathcal{L}?}$$

- **Equivalence queries**

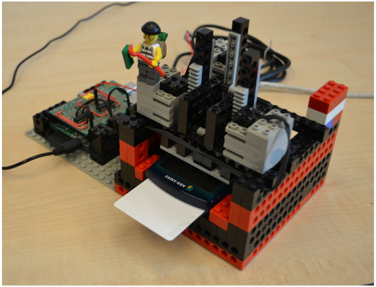
$$\boxed{\mathcal{L}(H) = \mathcal{L}?}$$

Negative result $\implies$ *counterexample*

Applications of L*

Through learning, verification methods for automata become available for black box systems

- Network protocols
- Devices such as smartcard readers
- Legacy software
- ...



Source: *Automated Reverse Engineering using Lego®*
Chalupar et al., WOOT 2014

L⋆ observation table

L⋆ maintains $S, E \subseteq A^*$ inducing a table

$$
\begin{array}{rl|cc}
 & & \multicolumn{2}{c}{\overbrace{\hspace{3em}}^{E}} \\
 & & \varepsilon & \mathrm{a} \\ \hline
 & \varepsilon & 1 & 0 \\
S & \mathrm{a} & 0 & 1 \\
 & \mathrm{aa} & 1 & 1 \\ \hline
S \cdot A & \mathrm{aaa} & 1 & 1
\end{array}
$$

L* observation table

L* maintains $S, E \subseteq A^*$ inducing a table

$$
\begin{array}{r l|cc}
 & & \multicolumn{2}{c}{\overbrace{\hphantom{\varepsilon \quad a}}^{E}} \\
 & & \varepsilon & a \\
\hline
\multirow{3}{*}{S} & \varepsilon & 1 & 0 \\
 & a & 0 & 1 \\
 & aa & 1 & 1 \\
\hline
S \cdot A & aaa & 1 & 1
\end{array}
$$

$\mathcal{L} = \{a^n \mid n \neq 1\}$

$aa \cdot a \in \mathcal{L}$

Prepend *row label* to *column label* and pose **membership query**

$$
(s, e) \mapsto \begin{cases} 1 & \text{if } se \in \mathcal{L} \\ 0 & \text{if } se \notin \mathcal{L} \end{cases}
$$

L* hypothesis DFA

Hypothesis states are upper rows of the table; transitions append symbols to row labels

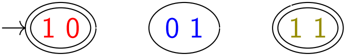
	ε	a
ε	1	0
a	0	1
aa	1	1
aaa	1	1

Requires properties **closedness** and **consistency** to be well-defined

L* hypothesis DFA

Hypothesis states are upper rows of the table; transitions append symbols to row labels

	ε	a
ε	1	0
a	0	1
aa	1	1
aaa	1	1

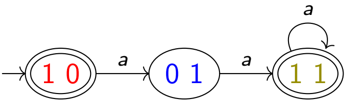


Requires properties **closedness** and **consistency** to be well-defined

$L^\star$ hypothesis DFA

Hypothesis states are upper rows of the table; transitions append symbols to row labels

	ε	a
ε	1	0
a	0	1
aa	1	1
aaa	1	1



Requires properties **closedness** and **consistency** to be well-defined

L* algorithm overview

table updated using
membership queries

1. Initialise $S = E = \{\varepsilon\}$
2. Satisfy closedness and consistency (by augmenting S and E)
3. Construct hypothesis
4. Pose equivalence query
5. On a counterexample, add its prefixes to S and repeat from 2

Motivation

Deterministic automata are often too large for tools to handle

A **side-effect** like non-determinism can provide succinctness

Idea: use side-effects to **optimise** the algorithm

Monads

We consider side-effects given by a **monad**

A monad T assigns to each set X a set

$$TX = \textbf{combinations of } X$$

It lifts functions to combinations and has

Unit:
$X \to TX$

Multiplication:
$TTX \to TX$

Side-effects as monads

Side-effect	**Monad**
Partiality	$(-) + 1$
Non-determinism	$\mathcal{P}(-)$
Alternation	$\mathcal{P}_{\uparrow}\mathcal{P}(-)$
Monoid value	$\mathbb{M} \times (-)$
Weighted sum	$V(-)$

T-algebras

A T*-algebra* on a set X **interprets combinations within** X

$$TX \to X$$

T*-algebra homomorphisms* preserve this structure:
$$\begin{array}{ccc} TX & \xrightarrow{Tf} & TY \\ \downarrow & & \downarrow \\ X & \xrightarrow{f} & Y \end{array}$$

T-algebras

A T*-algebra* on a set X **interprets combinations within** X

$$TX \to X$$

T*-algebra homomorphisms* preserve this structure:
$$\begin{array}{ccc} TX & \xrightarrow{Tf} & TY \\ \downarrow & & \downarrow \\ X & \xrightarrow{f} & Y \end{array}$$

TX itself is the **free** T**-algebra on** X, with

$$TTX \to TX$$

being the monad's multiplication

Non-deterministic automaton

A non-deterministic automaton is a set Q with $q_0 \in \mathcal{P}Q$ and functions

$$
\begin{tikzcd}
Q \arrow[r, "\mathsf{out}"] \arrow[d, "\delta"] & \{0,1\} \\
(\mathcal{P}Q)^A &
\end{tikzcd}
$$

Non-deterministic automaton

A non-deterministic automaton is a set Q with $q_0 \in \mathcal{P}Q$ and functions

$$\begin{tikzcd} & \{0,1\} \\ Q \arrow[ur, "\mathrm{out}"] \arrow[d, "\delta"] & \\ (\mathcal{P}Q)^A & \end{tikzcd}$$

$\underline{\mathcal{P}\text{-algebra:}}$

sets containing $1 \mapsto 1$

other sets $\mapsto 0$

Automaton with side-effects

An **automaton with side-effects in** T is a set Q with $q_0 \in TQ$ and functions

$$\begin{tikzcd} & O \\ Q \arrow[ur, "\mathsf{out}"] \arrow[d, "\delta"] & \\ (TQ)^A & \end{tikzcd}$$

output set with T-algebra structure
$TO \to O$

These **do not minimise** in general; no target for the learner

Target automaton

No unique (up to iso) minimal NFA

Target automaton

No unique (up to iso) minimal NFA

NFA semantics defined in terms of its **determinisation**

automaton with free $\mathcal{P}$-algebra
state space $\mathcal{P}Q$

Automata with T-algebra structure **minimise uniquely**

T-automaton

A T*-automaton* is a T-algebra Q with T-algebra homomorphisms

$$\begin{tikzcd} q_0 \in Q \arrow[r, "\mathsf{out}"] \arrow[d, "\delta"] & O \\ Q^A & \end{tikzcd}$$

point-wise T-algebra

Contributions

①

General adaptation of $\mathtt{L}^\star$ for T-automata

$\Downarrow$ method to find generators

General adaptation of $\mathtt{L}^\star$ for automata with T-side-effects

②

Adaptation of optimised counterexample handling

Contributions

① General adaptation of $\mathtt{L}^\star$ for T-automata

$\Downarrow$ method to find generators

General adaptation of $\mathtt{L}^\star$ for automata with T-side-effects

② Adaptation of optimised counterexample handling

Structure on the table

T-algebra on output set $\implies$ pointwise T-algebra on table rows

$\boxed{\text{Implicit row for each combination of row labels}}$

	ε	a
ε	1	0
a	0	1
aa	1	1

	ε	a
$\emptyset$	0	0
$\{\varepsilon, a\}$	1	1
$\{a, aa\}$	1	1

Structure on the table

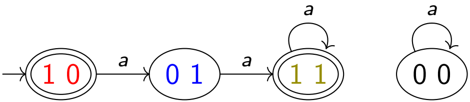
T-algebra on output set $\implies$ pointwise T-algebra on table rows

$\boxed{\text{Implicit row for each combination of row labels}}$

	ε	a
ε	1	0
a	0	1
aa	1	1

	ε	a
$\emptyset$	0	0
$\{\varepsilon, a\}$	1	1
$\{a, aa\}$	1	1

Hypothesis given by all (upper) rows



Generators

Hypothesis is generated by the explicit rows

Every generating set of rows induces a T-succinct hypothesis

Minimising generators:
While there is a row that is a combination of other rows, remove it

Generators example

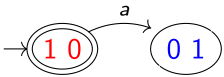
	ε	a
ε	1	0
a	0	1
aa	1	1

	ε	a
$\emptyset$	0	0
$\{\varepsilon, a\}$	1	1

Generators example

	ε	a
ε	1	0
a	0	1
aa	1	1

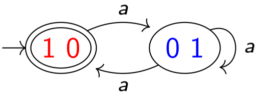
	ε	a
$\emptyset$	0	0
$\{\varepsilon, a\}$	1	1



Generators example

	ε	a
ε	1	0
a	0	1
aa	1	1

	ε	a
$\emptyset$	0	0
$\{\varepsilon, a\}$	1	1



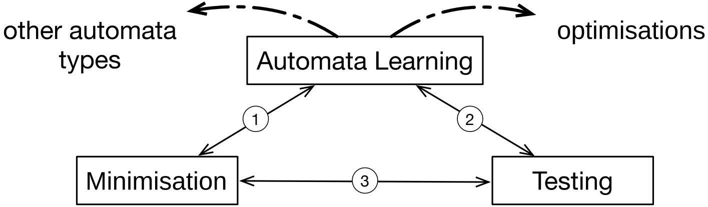
Future work

Monads not preserving finite sets

- Weighted automata over infinite PIDs
- Subsequential transducers
- Nominal automata

Conformance testing

CALF: Categorical Automata Learning Framework



Project: `calf-project.org`