CALF: Categorical Automata Learning Framework

Gerco van Heerdt Matteo Sammartino Alexandra Silva

May 23, 2017

Active automata learning

- Active automata learning algorithms learn an automaton describing the behaviour of a system by providing inputs and observing outputs
- Enables verification methods that work on an automaton
- Allows comparison of different implementations of e.g. a network protocol

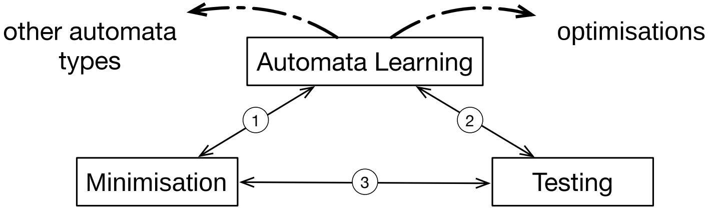
Capturing systems more precisely requires more complex types of automata and more complicated learning algorithms

Idea: understanding the main concepts on an abstract level helps developing and reasoning about new algorithms

CALF

Our Categorical Automata Learning Framework

- Gives an abstract view on the ingredients and constructions of learning algorithms, leading to new adaptations
- Covers also minimisation and equivalence testing
- Allows transferring optimisations among these areas



Active learning of DFAs: the basic setting

- Finite alphabet set A
- Target regular language $\mathcal{L}\colon A^* \to 2 = \{0, 1\}$
- Oracle that can tell whether a given word is in $\mathcal{L}$ (*membership queries*)

Aim is to learn a DFA accepting $\mathcal{L}$, in particular the minimal one

A simple data structure used to conjecture a DFA is the *observation table*

Observation table

Given $S, E \subseteq A^*$, define

$$\mathrm{row_t} \colon S \to 2^E \qquad \mathrm{row_t}(s)(e) = \mathcal{L}(se)$$

$$\mathrm{row_b} \colon S \cdot A \to 2^E \qquad \mathrm{row_b}(sa)(e) = \mathcal{L}(sae)$$

		E: ε	a	aa
S	ε	0	0	1
$S \cdot A$	a	0	1	0
	b	0	0	0

S and E evolve throughout runs of learning algorithms

Hypothesis

Given an observation table defined by $S, E \subseteq A^*$, the *hypothesis* DFA is given by

$$H = \{\mathsf{row_t}(s) \mid s \in S\} \subseteq 2^E$$

$$\begin{aligned}
\mathsf{init} &\in H & \mathsf{init} &= \mathsf{row_t}(\varepsilon) \\
\delta\colon H \times A &\to H & \delta(\mathsf{row_t}(s), a) &= \mathsf{row_b}(sa) \\
\mathsf{out}\colon H &\to 2 & \mathsf{out}(\mathsf{row_t}(s)) &= \mathsf{row_t}(s)(\varepsilon)
\end{aligned}$$

provided that $\varepsilon \in S \cap E$ and two properties hold

Closedness and consistency

- **Closedness** states that each transition leads to a state of the hypothesis. The table is closed if for all $t \in S \cdot A$ there is $s \in S$ such that $\mathrm{row_t}(s) = \mathrm{row_b}(t)$

- **Consistency** states that there is no ambiguity in determining transitions. The table is consistent if for all $s_1, s_2 \in S$ with

$$\mathrm{row_t}(s_1) = \mathrm{row_t}(s_2)$$

we have, for any $a \in A$,

$$\mathrm{row_b}(s_1 a) = \mathrm{row_b}(s_2 a)$$

Closedness

The table is closed if for all $t \in S \cdot A$ there is $s \in S$ such that $\mathrm{row_t}(s) = \mathrm{row_b}(t)$

	ε
ε	1
a	0

If no such s exists, add the word t to S

	ε
ε	1
a	0
aa	1

Consistency

The table is consistent if for all $s_1, s_2 \in S$ with $\mathrm{row_t}(s_1) = \mathrm{row_t}(s_2)$ we have, for any $a \in A$, $\mathrm{row_b}(s_1 a) = \mathrm{row_b}(s_2 a)$

	ε
ε	1
a	1
aa	0

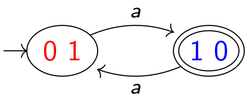
If $\mathrm{row_b}(s_1 a)(e) \neq \mathrm{row_b}(s_2 a)(e)$, add ae to E to distinguish $\mathrm{row_t}(s_1)$ and $\mathrm{row_t}(s_2)$

	ε	$a\varepsilon$
ε	1	1
a	1	0
aa	0	0

Hypothesis construction

- State space: distinct top rows (image of row_t)
- Initial state: ε row
- Output: taken from ε column
- Transitions: appending symbols to row labels

	ε	a
ε	0	1
a	1	0
aa	0	1



ID algorithm

Assume a given set $S \subseteq A^*$ such that for every state of the minimal DFA accepting $\mathcal{L}$ there is a word in S reaching that state

Closedness will automatically hold

1. Initialise $E = \{\varepsilon\}$
2. Enforce consistency
3. Construct the hypothesis

The hypothesis will be isomorphic to the minimal DFA

L* algorithm

Assume an oracle that can tell whether a hypothesis accepts the right language, and if not provides a counterexample word (*equivalence queries*)

1. Enforce closedness and consistency
2. Construct the hypothesis
3. Ask the oracle if the hypothesis is correct
4. If not, add all prefixes of the counterexample to S and restart

The hypothesis will be correct after finitely many iterations, and it will be isomorphic to the minimal DFA

DA of words

Given the language $\mathcal{L}\colon A^* \to 2$, we have a DA accepting $\mathcal{L}$:

- State space: A^*
- Initial state: $\varepsilon \in A^*$
- Output: $\mathcal{L}\colon A^* \to 2$
- Transitions:

$$c\colon A^* \times A \to A^* \qquad\qquad c(u, a) = ua$$

Reachability map

If Q is a DA accepting $\mathcal{L}$, there is a unique DA homomorphism $r\colon A^* \to Q$ given by

$$r(\varepsilon) = \mathrm{init}_Q \qquad\qquad r(ua) = \delta_Q(r(u), a)$$

called the *reachability map*, which assigns to each word the state it reaches in Q

Q is *reachable* if r is surjective: every state is reached by a word

DA of languages

Given the language $\mathcal{L}\colon A^* \to 2$, we have a DA accepting $\mathcal{L}$:

- State space: 2^{A^*}
- Initial state: $\mathcal{L} \in 2^{A^*}$
- Output:

$$\varepsilon?\colon 2^{A^*} \to 2 \qquad\qquad \varepsilon?(l) = l(\varepsilon)$$

- Transitions:

$$\partial\colon 2^{A^*} \times A \to 2^{A^*} \qquad\qquad \partial(l, a)(v) = l(av)$$

e.g. $\partial(\{a, ba, abb\}, a) = \{\varepsilon, bb\}$

Observability map

If Q is a DA accepting $\mathcal{L}$, there is a unique DA homomorphism $o\colon Q \to 2^{A^*}$ given by

$$o(q)(\varepsilon) = \mathrm{out}_Q(q) \qquad\qquad o(q)(av) = o(\delta_Q(q,a))(v)$$

called the *observability map*, which assigns to each state the language it accepts

The DA Q is *observable* if o is injective: different states accept different languages

A DA is *minimal* if it is both reachable and observable

Total response

The language $\mathcal{L}\colon A^* \to 2$ induces DAs A^* and 2^{A^*} accepting $\mathcal{L}$

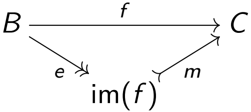
The reachability map of 2^{A^*} coincides with the observability map of A^* in the DA homomorphism called the *total response of* $\mathcal{L}$:

$$t_{\mathcal{L}}\colon A^* \to 2^{A^*} \qquad\qquad t_{\mathcal{L}}(u)(v) = \mathcal{L}(uv)$$

If Q is any DA accepting $\mathcal{L}$, then $t_{\mathcal{L}} = A^* \xrightarrow{r} Q \xrightarrow{o} 2^{A^*}$

Function factorisation

Every function can be written as a surjection followed by an injection:



$$e(b) = f(b)$$
$$m(c) = c$$

Factorisation uniqueness

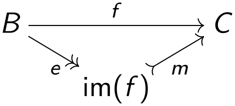
In a commutative square of functions as on the left,

$$\begin{array}{ccc} U & \overset{i}{\twoheadrightarrow} & V \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle h} \\ W & \overset{j}{\rightarrowtail} & X \end{array} \qquad\qquad \begin{array}{ccc} U & \overset{i}{\twoheadrightarrow} & V \\ {\scriptstyle g}\downarrow & {\scriptstyle d}\swarrow & \downarrow{\scriptstyle h} \\ W & \overset{j}{\rightarrowtail} & X \end{array}$$

where i is surjective and j injective, there is a unique diagonal d making the triangles commute: $d(i(u)) = g(u)$

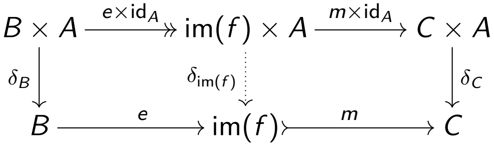
DA homomorphism factorisation

In an image factorisation



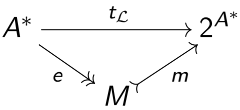
if f is a DA homomorphism, then so are e and m, given this DA structure on $\mathrm{im}(f)$:

- Initial state: initial state of C
- Output: output of C
- Transitions: the unique diagonal

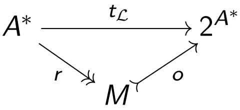


Minimal DA

The minimal DA accepting $\mathcal{L} \colon A^* \to 2$ can be obtained in theory by factorising the total response $t_{\mathcal{L}}$:

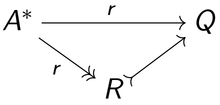


Since e and m are DA homomorphisms, we must have $e = r_M$ and $m = o_M$ by the uniqueness properties

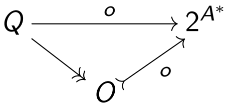


Minimisation

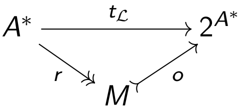
Similarly, the reachable part of a DA Q is obtained by factorising its reachability map:



Equivalent states are merged by factorising the observability map:



The hypothesis approximates the minimal DA



Concretely, the minimal DA is given by

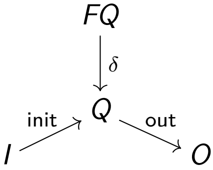
$$M = \{t_{\mathcal{L}}(u) \mid u \in A^*\}$$

$$\begin{aligned} \text{init} &\in M & \text{init} &= t_{\mathcal{L}}(\varepsilon) \\ \delta\colon M \times A &\to M & \delta(t_{\mathcal{L}}(u), a) &= t_{\mathcal{L}}(ua) \\ \text{out}\colon M &\to 2 & \text{out}(t_{\mathcal{L}}(u)) &= t_{\mathcal{L}}(u)(\varepsilon) \end{aligned}$$

This is equivalent to the hypothesis for $S = E = A^*$

Abstract automaton

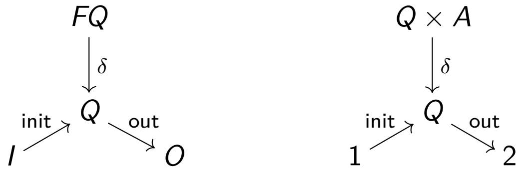
Given a category $\mathbf{C}$, objects I and O in $\mathbf{C}$, and a functor $F\colon \mathbf{C} \to \mathbf{C}$, an *automaton* is an object Q in $\mathbf{C}$ with three morphisms:



DAs as automata

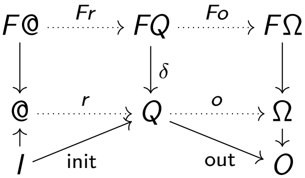
For DAs:

- A singleton 1 serves as the initial state selector
- The set $2 = \{0, 1\}$ captures rejection (0) and acception (1)
- The functor $(-) \times A$ provides the transition domain



Reachability and observability maps

Assume an initial object @ among automata without output and a final object Ω among automata without initial state:



Languages can be defined as morphisms $I \to \Omega$ or $@ \to O$, which correspond bijectively to each other through the total response

The total response may be defined as the reachability map of Ω or as the observability map of @

Factorisation system

We assume two classes of **C**-morphisms:

- "surjective" morphisms $\mathcal{E}$ and
- "injective" morphisms $\mathcal{M}$

such that

- every **C**-morphism $f\colon A \to B$ can be factored as $f = m \circ e$, with $e \in \mathcal{E}$ and $m \in \mathcal{M}$;
- $\mathcal{E}$ and $\mathcal{M}$ are closed under composition and contain all isos;
- everything in $\mathcal{E}$ is an epi, and everything in $\mathcal{M}$ is a mono; and
- we have the unique diagonal property that does not fit on this slide but is the same as before

Lifts to the category of automata if F preserves $\mathcal{E}$

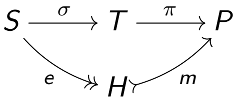
Approximating an object

A *wrapper for an object* T is a pair of morphisms

$$w = (S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$$

- T is called the *target* of w
- σ selects from T
- π classifies T

The (unstructured) *hypothesis* H is the image of $\xi = \pi \circ \sigma$:



Observation table wrapper

For $S, E \subseteq A^*$, we have a wrapper for the minimal DA M for $\mathcal{L}$:

$$(S \xrightarrow{\alpha} A^* \xrightarrow{r} M, M \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E),$$

where

- α is the inclusion and
- ω restricts to E

Recall $o \circ r = t_{\mathcal{L}}$ and note that

$$\xi = \omega \circ t_{\mathcal{L}} \circ \alpha = \mathsf{row}_{\mathsf{t}}$$

The image of $\mathsf{row}_{\mathsf{t}}$ is precisely the state space of the hypothesis in learning

Approximating algebraic structure

Consider a wrapper

$$w = (S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$$

Given a functor F and an F-algebra $f\colon FT \to T$, we have the approximation

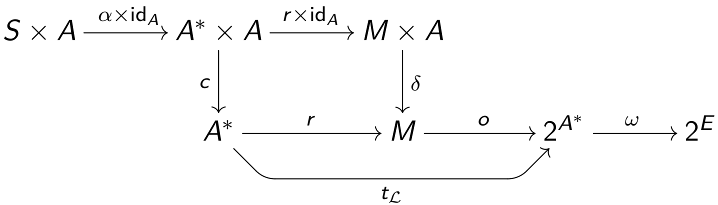
$$\begin{array}{rcl} \xi_f = FS & \xrightarrow{F\sigma} & FT \\ & & \downarrow f \\ & & T \xrightarrow{\pi} P \end{array}$$

Approximating the minimal DFA transition function

For the observation table wrapper

$$(S \xrightarrow{\alpha} A^* \xrightarrow{r} M, M \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E)$$

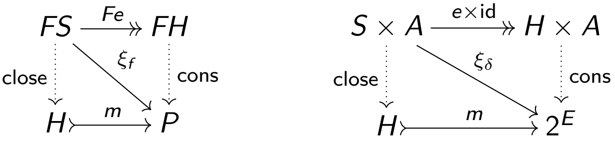
and transition function $\delta \colon M \times A \to M$, we have $\xi_\delta = \mathsf{row_b}$ (up to $S \times A \cong S \cdot A$):



$$\mathsf{row}_b(sa)(e) = \mathcal{L}(sae)$$

Closedness and consistency

A wrapper ($S \xrightarrow{\sigma} T$, $T \xrightarrow{\pi} P$) is *f-closed*, for $f\colon FT \to T$, if a morphism close exists making the left triangle commute

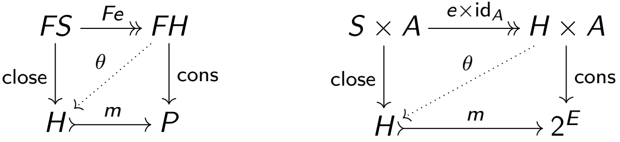


It is *f-consistent* if a morphism cons exists making the right triangle commute

For the observation table wrapper, δ-closedness and δ-consistency are the classical notions of closedness and consistency

Structured hypothesis

If $(S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$ is f-closed and f-consistent, for $f \colon FT \to T$, we have an algebra



For an observation table wrapper and $f = \delta_M$, $\theta = \delta_H$

(We only consider F that preserve "surjective" morphisms)

Initial state

The initial state of M can be seen as an algebra

$$\mathsf{init}\colon 1 \to M$$

for $1 = \{*\}$ an arbitrary singleton

This gives a closedness property init-closedness (init-consistency is trivial) stating that there must be $s \in S$ s.t. $\xi(s) = \xi_{\mathsf{init}}(*)$, where

$$\xi_{\mathsf{init}}\colon 1 \to 2^E \qquad\qquad \xi_{\mathsf{init}}(*)(e) = \mathcal{L}(e)$$

is the row of the empty word

Thus, this property is weaker than requiring $\varepsilon \in S$

Output

The set of accepting states can be seen as a coalgebra

$$\mathsf{out}\colon M \to 2$$

for $2 = \{0, 1\}$

This gives a consistency property (technically coclosedness) stating that for all $s_1, s_2 \in S$ s.t. $\xi(s_1) = \xi(s_2)$ we must have $\xi_{\mathsf{out}}(s_1) = \xi_{\mathsf{out}}(s_2)$, where

$$\xi_{\mathsf{out}}\colon S \to 2 \qquad\qquad \xi_{\mathsf{init}}(s) = \mathcal{L}(s)$$

is the column of the empty word

Again, this property is weaker than requiring $\varepsilon \in E$

Simple correctness conditions

Consider a wrapper $(S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$

If σ is surjective, we have a diagonal

$$\begin{array}{ccc} S & \overset{\sigma}{\twoheadrightarrow} & T \\ {\scriptstyle e}\big\downarrow & \overset{\phi}{\swarrow} & \big\downarrow{\scriptstyle \pi} \\ H & \overset{m}{\rightarrowtail} & P \end{array}$$

If π is injective, we have a diagonal

$$\begin{array}{ccc} S & \overset{e}{\twoheadrightarrow} & H \\ {\scriptstyle \sigma}\big\downarrow & \overset{\psi}{\swarrow} & \big\downarrow{\scriptstyle m} \\ T & \overset{\pi}{\rightarrowtail} & P \end{array}$$

If both of these hold, then ϕ and ψ are inverse to each other

Results

Consider a wrapper ($S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P$)

If σ is surjective, we have a diagonal

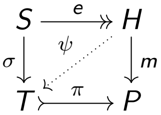
$$\begin{array}{ccc} S & \overset{\sigma}{\twoheadrightarrow} & T \\ {\scriptstyle e}\downarrow & \overset{\phi}{\swarrow} & \downarrow{\scriptstyle \pi} \\ H & \overset{m}{\rightarrowtail} & P \end{array}$$

- For any $f\colon FT \to T$, the wrapper is f-closed
- If the wrapper is f-consistent, ϕ is an F-algebra homomorphism

Results

Consider a wrapper $(S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$

If π is injective, we have a diagonal



- For any $f\colon FT \to T$, the wrapper is f-consistent
- If the wrapper is f-closed, ψ is an F-algebra homomorphism

Results

Consider a wrapper $(S \xrightarrow{\sigma} T, T \xrightarrow{\pi} P)$

If σ is surjective and π injective, we have diagonals

$$
\begin{array}{ccc}
S & \overset{\sigma}{\twoheadrightarrow} & T \\
{\scriptstyle e}\downarrow & \overset{\phi}{\swarrow} & \downarrow{\scriptstyle \pi} \\
H & \overset{m}{\rightarrowtail} & P
\end{array}
\qquad\qquad
\begin{array}{ccc}
S & \overset{e}{\twoheadrightarrow} & H \\
{\scriptstyle \sigma}\downarrow & \overset{\psi}{\swarrow} & \downarrow{\scriptstyle m} \\
T & \overset{\pi}{\rightarrowtail} & P
\end{array}
$$

- ϕ and ψ are inverse to each other
- For any $f : FT \to T$, the wrapper is f-closed and f-consistent
- ϕ and ψ are F-algebra homomorphisms (and thus isos)

Simple correctness conditions for observation tables

For an observation table wrapper

$$(S \xrightarrow{\alpha} A^* \xrightarrow{r} M, M \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E),$$

- $r \circ \alpha$ is surjective if and only if for each state of M there is a word in S reaching that state
- $\omega \circ o$ is injective if and only if for each pair of distinct states of M there is a word in E on which they behave differently

Less simple correctness conditions (1)

Let $(S \xrightarrow{\sigma} Q, Q \xrightarrow{\pi} P)$ be a wrapper for an automaton Q

If

- σ is surjective;
- Q is observable;
- the wrapper is out-consistent; and
- the wrapper is δ-consistent

then H is an automaton isomorphic to Q

Less simple correctness conditions (2)

Let $(S \xrightarrow{\sigma} Q, Q \xrightarrow{\pi} P)$ be a wrapper for an automaton Q

If

- π is injective;
- Q is reachable;
- the wrapper is init-closed; and
- the wrapper is δ-closed

then H is an automaton isomorphic to Q

`ID` correctness

If

- σ is surjective;
- Q is observable;
- the wrapper is out-consistent; and
- the wrapper is δ-consistent

then H is an automaton isomorphic to Q

- `ID` assumes a set S such that $\sigma = S \xrightarrow{\alpha} A^* \xrightarrow{r} M$ is surjective
- $Q = M$ is observable by definition
- out-consistency holds because $\varepsilon \in E$
- δ-consistency is what the algorithm enforces

L* correctness

If

- σ is surjective;
- Q is observable;
- the wrapper is out-consistent; and
- the wrapper is δ-consistent

then H is an automaton isomorphic to Q

- Adding all prefixes of a counterexample to S increases the image of $\sigma = S \xrightarrow{\alpha} A^* \xrightarrow{r} M$
- $Q = M$ is observable by definition
- out-consistency holds because $\varepsilon \in E$
- δ-consistency is enforced before constructing the hypothesis

Reachability analysis

To find the reachable part R of a known DFA Q, we can use a wrapper of inclusions

$$(S \to R, R \to Q),$$

where $S \subseteq R$

- init-closedness: $\text{init}_R \in S$
- δ-closedness: for each $s \in S$ and $a \in A$, $\delta_R(s, a) \in S$

Since $\text{init}_R = \text{init}_Q$ and $\delta_R(s, a) = \delta_Q(s, a)$, this leads to the usual algorithm:

- initialise $S = \{\text{init}_Q\}$
- while $\delta_Q(s, a) \notin S$, add it

Reachability analysis correctness

$$(S \xrightarrow{\sigma} R, R \xrightarrow{\pi} Q)$$

If

- π is injective;
- R is reachable;
- the wrapper is init-closed; and
- the wrapper is δ-closed

then H is an automaton isomorphic to R

State merging

To merge equivalent states of a DFA Q, we could use a wrapper

$$(Q \xrightarrow{\sigma} O, O \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E)$$

where O is the automaton of languages accepted by Q and σ classifies states according to their language

out-consistency says that states of Q equivalent under ξ must have the same output (accept/reject)

δ-consistency says that equivalent states of Q must have equivalent successors for each $a \in A$

These can be satisfied as in learning

State merging correctness

$$(Q \xrightarrow{\sigma} O, O \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E)$$

If

- σ is surjective;
- O is observable;
- the wrapper is out-consistent; and
- the wrapper is δ-consistent

then H is an automaton isomorphic to O

General equivalence testing theorem

For U and V DFAs and $S, E \subseteq A^*$ we have wrappers

$$w_U = (\sigma_U, \pi_U) = (S \xrightarrow{\alpha} A^* \xrightarrow{r} U, U \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E)$$
$$w_V = (\sigma_V, \pi_V) = (S \xrightarrow{\alpha} A^* \xrightarrow{r} V, V \xrightarrow{o} 2^{A^*} \xrightarrow{\omega} 2^E)$$

Suppose

- σ_U is surjective;
- π_U is injective; and
- either σ_V is surjective and V observable or π_V is injective and V reachable

Then $U \cong V$ if and only if all of the below hold

$$\xi^{w_U} = \xi^{w_V} \qquad \xi_\delta^{w_U} = \xi_\delta^{w_V} \qquad \xi_{\text{init}}^{w_U} = \xi_{\text{init}}^{w_V} \qquad \xi_{\text{out}}^{w_U} = \xi_{\text{out}}^{w_V}$$

W-method

Let U be a known minimal DFA and V an unknown one

Using minimization-like algorithms inspired by learning, we can find

- $S \subseteq A^*$ such that $S \xrightarrow{\sigma_U} U$ is surjective and
- $E \subseteq A^*$ such that $U \xrightarrow{\pi_U} 2^E$ is injective

These are the first two conditions for the theorem

They also ensure that the hypothesis of w_U is isomorphic to U

W-method

Assume that at this point the equalities hold:

$$\xi^{w_U} = \xi^{w_V} \qquad \xi_\delta^{w_U} = \xi_\delta^{w_V} \qquad \xi_{\mathsf{init}}^{w_U} = \xi_{\mathsf{init}}^{w_V} \qquad \xi_{\mathsf{out}}^{w_U} = \xi_{\mathsf{out}}^{w_V}$$

Then the two hypotheses coincide and are isomorphic to U

Assume a given upper bound n on $|V|$

Updating S to $S \cdot A^{\leq(n-|U|)}$ ensures that (assuming $\varepsilon \in S$)

- σ_V is surjective (and we know that V is observable)

which triggers the theorem: $U \cong V$ if and only if

$$\xi^{w_U} = \xi^{w_V} \qquad \xi_\delta^{w_U} = \xi_\delta^{w_V} \qquad \xi_{\mathsf{init}}^{w_U} = \xi_{\mathsf{init}}^{w_V} \qquad \xi_{\mathsf{out}}^{w_U} = \xi_{\mathsf{out}}^{w_V}$$

W-method

Determining the equalities

$$\xi^{w_U} = \xi^{w_V} \qquad \xi_\delta^{w_U} = \xi_\delta^{w_V} \qquad \xi_{\mathsf{init}}^{w_U} = \xi_{\mathsf{init}}^{w_V} \qquad \xi_{\mathsf{out}}^{w_U} = \xi_{\mathsf{out}}^{w_V}$$

consists in testing whether U and V agree on a set of words:

$$\xi^{w_U}(s)(e) = \mathcal{L}_U(se) \qquad \xi_\delta^{w_U}(s)(a)(e) = \mathcal{L}_U(sae)$$

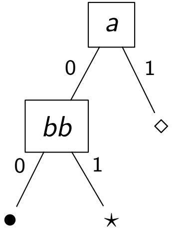
etc., and analogously for V

Optimising learning

We distinguish rows by adding a word to E, but this requires a query for every row in the table

More efficient is to handle the classification using a *classification tree*

Classification tree



- Internal nodes represent experiments, the result of which determines the next subtree
- Classification is into the set L of labels making up the leaves
- A tree τ classifies languages, $2^{A^*} \xrightarrow{\omega_\tau} L$, and states of an automaton using the composition $Q \xrightarrow{o} 2^{A^*} \xrightarrow{\omega_\tau} L$

Sifting

Given the target language $\mathcal{L}$, a tree also classifies words: given a word u we move on a node v to the subtree of $\mathcal{L}(uv)$

This is called *sifting*

The closedness and consistency that follow from our general definitions are conveniently described using this classification:

- Closedness states that all words in $S \cdot A$ must sift into a leaf into which a word from S sifts
- Consistency states that for $s_1, s_2 \in S$ sifting into the same leaf, $s_1 a$ and $s_2 a$ for each $a \in A$ must also sift into the same leaf

Optimised algorithms

- $L^\star$: Kearns and Vazirani's algorithm
- State merging: splitting tree algorithm
- Conformance testing: HSI-method

Other instances

- Nondeterministic automata (more generally JSL automata)
- Weighted automata over a field
- Nominal automata
- Automata with a state space that is an algebra for a monad preserving finite sets (naive general algorithm: CONCUR submission)

Future work: more instances

- Register automata
- Tree automata
- Büchi-style automata
- Alternating automata
- (Subclasses of) probabilistic automata

More future work

- Optimised algorithms for automata with structure
- Implementation of the CONCUR algorithm
- Describing iterative algorithms abstractly
- Finding other (possibly even non-automaton?) applications of the general wrapper theory

Reading material

- Master thesis: *An Abstract Automata Learning Framework*
  Gerco van Heerdt
- CSL submission: *CALF: Categorical Automata Learning Framework*
  Gerco van Heerdt, Matteo Sammartino, Alexandra Silva
- CONCUR submission: *Learning Automata with Side-Effects*
  Gerco van Heerdt, Matteo Sammartino, Alexandra Silva

These and others can be found on our website:

`http://calf-project.org`