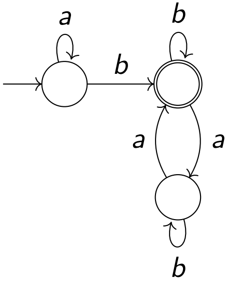
Tree Automata as Algebras: Minimisation and Determinisation

Gerco van Heerdt Tobias Kappé Jurriaan Rot
Matteo Sammartino Alexandra Silva

University College London

June 3, 2019

DFAs: nice diagrams



Tree automata: no nice diagrams

Transitions originate from *multiple states*
Automaton reads *trees* rather than words

Alphabet Γ with arities $\mathsf{ar}\colon \Gamma \to \mathbb{N}$
Sets I, O

Tree automaton $(Q, \{\delta_\gamma\}_{\gamma \in \Gamma}, i, o)$

- transitions $\delta_\gamma\colon Q^{\mathsf{ar}(\gamma)} \to Q$
- initial states $i\colon I \to Q$
- output $o\colon Q \to O$

DFAs: every γ has arity 1

Tree automata: no nice diagrams :(

Transitions originate from *multiple states*
Automaton reads *trees* rather than words

Alphabet Γ with arities $\mathsf{ar}\colon \Gamma \to \mathbb{N}$
Sets I, O

Tree automaton $(Q, \{\delta_\gamma\}_{\gamma \in \Gamma}, i, o)$

- transitions $\delta_\gamma\colon Q^{\mathsf{ar}(\gamma)} \to Q$
- initial states $i\colon I \to Q$
- output $o\colon Q \to O$

DFAs: every γ has arity 1

Example

$$I = \{a, b\} \qquad \Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2 \qquad O = \{0, 1\}$$

Example

$$I = \{a, b\} \qquad \Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2 \qquad O = \{0, 1\}$$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$

Example

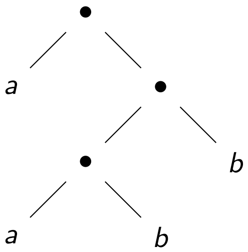
$I = \{a, b\}$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

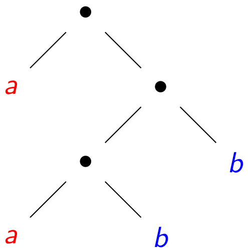
$$I = \{a, b\} \qquad \Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2 \qquad O = \{0, 1\}$$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

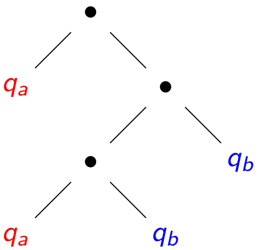
$I = \{a, b\}$ $\quad\quad$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $\quad\quad$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

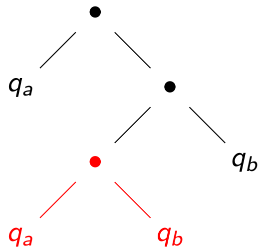
$I = \{a, b\}$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

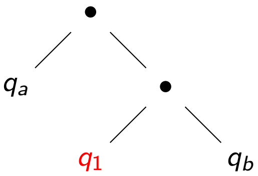
$I = \{a, b\}$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

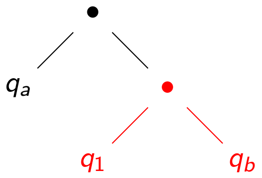
$I = \{a, b\}$ $\quad$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $\quad$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

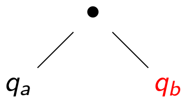
$$I = \{a, b\} \qquad \Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2 \qquad O = \{0, 1\}$$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

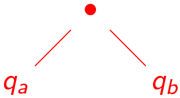
$I = \{a, b\}$ $\qquad$ $\Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $\qquad$ $O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$



Example

$I = \{a, b\}$ $\qquad \Gamma = \{\bullet\}, \mathrm{ar}(\bullet) = 2$ $\qquad O = \{0, 1\}$

$Q = \{q_a, q_b, q_1, q_\perp\}$

q_1

$i(a) = q_a$
$i(b) = q_b$

$\delta_\bullet(q_a, q_b) = q_1$
$\delta_\bullet(q_a, q_1) = q_a$
$\delta_\bullet(q_1, q_b) = q_b$
$\delta_\bullet(-) = q_\perp$

$o(q_a) = o(q_b) = o(q_\perp) = 0$
$o(q_1) = 1$

Tree automata as functor algebras

Tree automata over Γ can be seen as algebras for *signature endofunctor*

$$\Sigma X = \coprod_{\gamma \in \Gamma} X^{\mathsf{ar}(\gamma)}$$

(together with initial states and output)

Free Σ-algebra with generators X written $\Sigma^\diamond X$

- trees over the signature with additional X constants

$\Sigma^\diamond$ forms a *monad*

Tree language

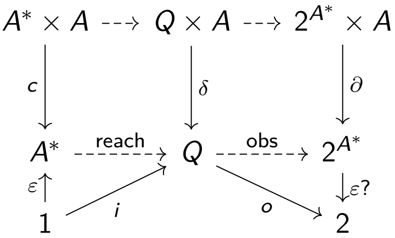
Given: tree automaton (Q, δ, i, o), $\delta \colon \Sigma Q \to Q$

$I \xrightarrow{i} Q$ extends to **reachability map** $\Sigma^{\diamond} I \xrightarrow{\text{reach}} Q$

Accepted language is $\mathcal{L}_Q = \Sigma^{\diamond} I \xrightarrow{\text{reach}} Q \xrightarrow{o} O$

In general, a *language* is a morphism $\Sigma^{\diamond} I \to O$

Minimality of DFAs: nice diagram



Reachable: reach surjective

Observable: obs injective

Minimal: reachable + observable

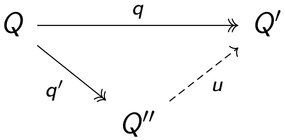
Minimality of tree automata: no nice diagram

Reachable: $\Sigma^\diamond I \xrightarrow{\text{reach}} Q$ surjective

Minimal: final among automata that

- are reachable and
- accept the same language

Minimisation of an automaton: final quotient automaton



Minimisations of reachable automata are minimal

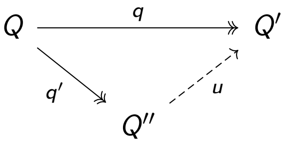
Minimality of tree automata: no nice diagram :(

Reachable: $\Sigma^{\diamond} I \xrightarrow{\text{reach}} Q$ surjective

Minimal: final among automata that

- are reachable and
- accept the same language

Minimisation of an automaton: final quotient automaton



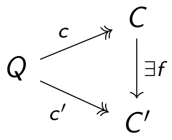
Minimisations of reachable automata are minimal

Minimisation via cobase

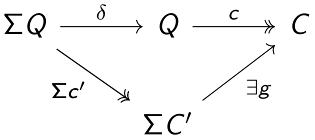
Automaton (Q, δ, i, o)

Complete lattice on quotients

$c \leq c' \iff$



Minimisation: gfp of monotone operator involving **cobase**

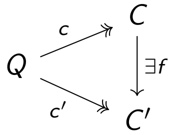


Minimisation via cobase

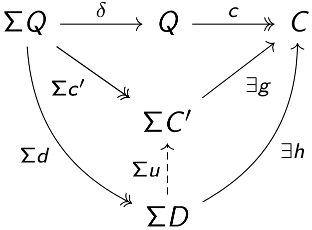
Automaton (Q, δ, i, o)

Complete lattice on quotients

$c \leq c' \iff$



Minimisation: gfp of monotone operator involving **cobase**



DFAs as monad algebras

$Q \times A \xrightarrow{\delta} Q$ extends to $Q \times A^* \xrightarrow{\delta^*} Q$

δ^* monoid action $\implies$ bijective correspondence

algebra for monad $(-) \times A^*$

Unit

$X \to X \times A^*$ by **empty word**

Multiplication

$X \times A^* \times A^* \to X \times A^*$ by **concatenation**

Automata based on monad algebras

Given objects I and O and a monad T, we can define automata

$$\begin{array}{ccccc} & & TQ & & \\ & & \downarrow{\scriptstyle \delta} & & \\ & & Q & & \\ & \overset{i}{\nearrow} & & \overset{o}{\searrow} & \\ I & & & & O \end{array}$$

where δ is a T-algebra

Reachability map: $TI \xrightarrow{i^\sharp} Q$

Language: $TI \xrightarrow{i^\sharp} Q \xrightarrow{o} O$

Automata based on monad algebras

DFA case:

$$
\begin{array}{ccccc}
 & & Q \times A^* & & \\
 & & \big\downarrow \delta & & \\
 & & Q & & \\
 & \overset{i}{\nearrow} & & \overset{o}{\searrow} & \\
1 & & & & 2
\end{array}
$$

where δ is a monoid action

Reachability map: $1 \times A^* \xrightarrow{i^\sharp} Q$

Language: $1 \times A^* \xrightarrow{i^\sharp} Q \xrightarrow{o} 2$

Nerode equivalence: classic

Used to find minimal automaton

Given language $\mathcal{L}\colon A^* \to 2$,

$$R = \{(u, v) \in A^* \times A^* \mid \forall w \in A^*. \, \mathcal{L}(uw) = \mathcal{L}(vw)\}$$

Largest relation making this diagram commute:

$$\begin{array}{ccccc}
R \times A^* & & \xrightarrow{p_2 \times \mathrm{id}} & & A^* \times A^* \\
 & & & & \big\downarrow \text{concat} \\
p_1 \times \mathrm{id} \big\downarrow & & & & A^* \\
 & & & & \big\downarrow \mathcal{L} \\
A^* \times A^* & \xrightarrow{\text{concat}} & A^* & \xrightarrow{\mathcal{L}} & 2
\end{array}$$

Nerode equivalence: classic

Used to find minimal automaton

Given language $\mathcal{L} \colon A^* \to 2$,

$$R = \{(u, v) \in A^* \times A^* \mid \forall w \in A^*. \mathcal{L}(uw) = \mathcal{L}(vw)\}$$

Largest relation making this diagram commute:

$$\begin{array}{ccc}
R \times A^* & \xrightarrow{p_2 \times \mathrm{id}} & 1 \times A^* \times A^* \\
 & & \downarrow \mu \\
\Big\downarrow p_1 \times \mathrm{id} & & 1 \times A^* \\
 & & \downarrow \mathcal{L} \\
1 \times A^* \times A^* \overset{\mu}{\to} 1 \times A^* & \xrightarrow{\mathcal{L}} & 2
\end{array}$$

Nerode equivalence: abstract

$\exists R, p_1, p_2$ s.t.

$$
\begin{array}{ccccc}
TR & & \xrightarrow{Tp_2} & & TTI \\
 & & & & \downarrow \mu \\
\downarrow Tp_1 & & & & TI \\
 & & & & \downarrow \mathcal{L} \\
TTI & \xrightarrow{\mu} & TI & \xrightarrow{\mathcal{L}} & O
\end{array}
$$

and if $\exists S, q_1, q_2$ s.t.

$$
\begin{array}{ccccc}
TS & & \xrightarrow{Tq_2} & & TTI \\
 & & & & \downarrow \mu \\
\downarrow Tq_1 & & & & TI \\
 & & & & \downarrow \mathcal{L} \\
TTI & \xrightarrow{\mu} & TI & \xrightarrow{\mathcal{L}} & O
\end{array}
\qquad \text{then} \qquad
\begin{array}{ccccc}
 & & S & & \\
 & \swarrow q_1 & \downarrow u & q_2 \searrow & \\
TI & \xleftarrow{p_1} & R & \xrightarrow{p_2} & TI
\end{array}
$$

Theorem: Nerode equivalence $\iff$ minimal automaton

Nerode equivalence: syntactic congruence

$I = A$, $T = (-)^*$, $\mu_X \colon (X^*)^* \to X^*$ flattens

Given $\mathcal{L} \colon A^* \to 2$, R instantiates to the largest relation s.t.

$$\frac{(u_1, v_1), \ldots, (u_n, v_n) \in R}{\mathcal{L}(u_1 \cdots u_n) = \mathcal{L}(v_1 \cdots v_n)}$$

or equivalently (syntactic congruence)

$$\frac{(u, v) \in R \qquad w, x \in A^*}{\mathcal{L}(wux) = \mathcal{L}(wvx)}$$

Minimal automaton: syntactic monoid

Nerode equivalence in **Set**

Exists for any finitary monad

- Equivalence under *contexts* $T(I+1)$
- Element of 1 is a hole where trees can be plugged in

Plans: learning

Tree automata learning algorithms exist

- But not modulo equations
- Pomset automata

Axiomatisation of semantic object

Further paper overview

Nominal tree automata

- Abstraction
- Parse trees for λ-terms

Relations between notions of minimality

- Minimal, minimisation, simple

Determinisation (lifting Kleisli adjunction)

- Nondeterministic (nominal) tree automata
- Multiplicity/weighted tree automata