Brzozowski's algorithm (co)algebraically.

Helle Hvid Hansen and Alexandra Silva

Radboud Universiteit Nijmegen and CWI

Coalgebraic Logics, 9 Oct 2012

Motivation

- duality between reachability and observability (Arbib and Manes 1975): beautiful, not very well-known.

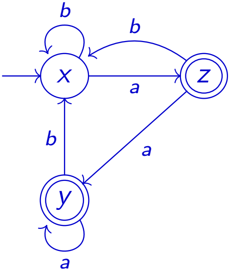
  Bidoit&Hennicker&Kurz. On the duality between observability and reachability (2001)

- combined use of algebra and coalgebra.

- our understanding of automata is still very limited;
  cf. recent research: universal automata, àtomata, weighted automata (Sakarovitch, Brzozowski, . . .)

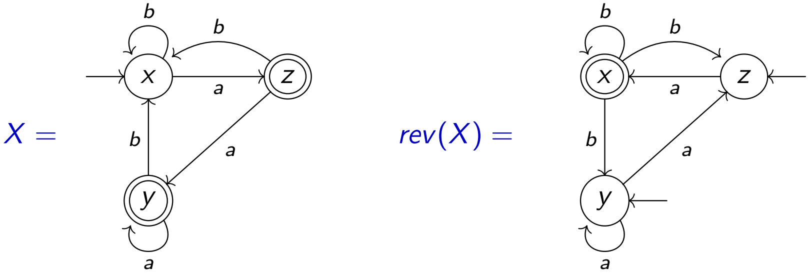
- joint work with Bonchi, Bonsangue, Rutten (Dexter's festschrift 2012) and Hansen, Panangaden and Bezhanishvilli, Kozen, Kupke.

Brzozowski algorithm (by example)



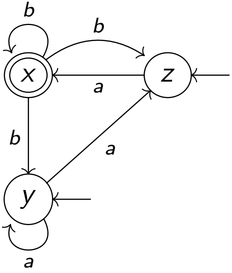
- initial state: x • final states: y and z
- $L(x) = \{a, b\}^* a$
- X is reachable but not minimal: $L(y) = \varepsilon + \{a, b\}^* a = L(z)$

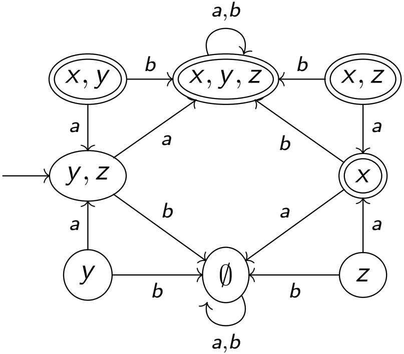
Reversing the automaton: $rev(X)$



- transitions are reversed
- initial states $\Leftrightarrow$ final states
- $rev(X)$ is non-deterministic

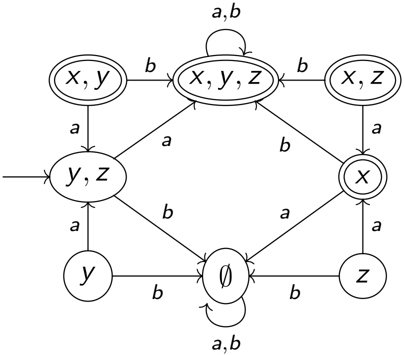
Making it deterministic again: $det(rev(X))$





- new state space: $2^X = \{V \mid V \subseteq \{x, y, z\}\}$
- $V \xrightarrow{a} W \qquad W = \{w \mid v \xrightarrow{a} w, v \in V\}$
- initial state: $\{y, z\}$
- final states: all V with $x \in V$

The automaton $det(rev(X))$ . . .



- . . . accepts the reverse of the language accepted by X:

$$L(\, det(rev(X))\,) \;=\; a\,\{a,b\}^* \;=\; reverse(\, L(X)\,)$$

- . . . and is observable!

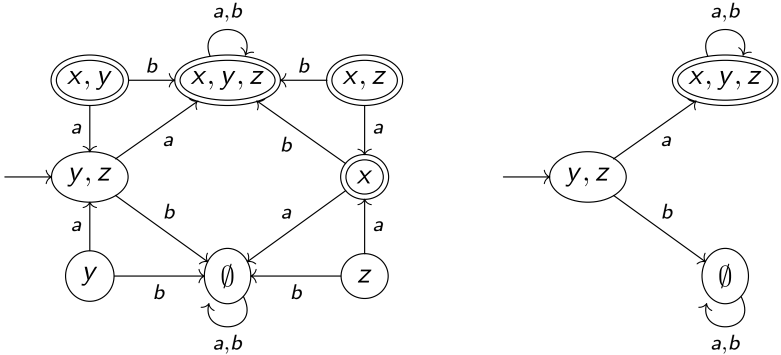
Today's Theorem

If: a deterministic automaton X is *reachable* and accepts $L(X)$

then: $det(rev(X))$ is *minimal* and

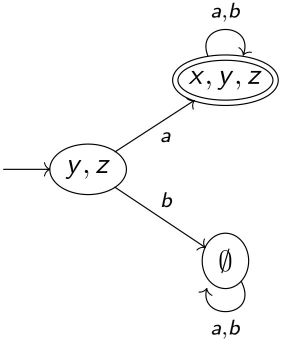
$$L(\,det(rev(X))\,) \;=\; reverse(\,L(X)\,)$$

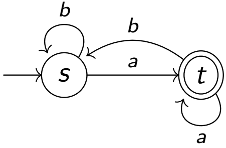
Taking the reachable part of *det*(*rev*(*X*))



- *reach*(*det*(*rev*(*X*))) is reachable (by construction)

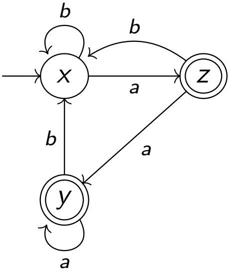
Repeating everything, now for *reach*(*det*(*rev*(X)))

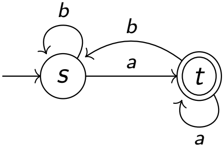




- . . . gives us *reach*(*det*(*rev*(*reach*(*det*(*rev*(X))))))
- which is (reachable and) minimal and accepts $\{a, b\}^* a$.

All in all: Brzozowski's algorithm





- X is reachable and accepts $\{a, b\}^* \, a$
- $reach(det(rev(reach(det(rev(X))))))$ also accepts $\{a, b\}^* \, a$
- . . . and is minimal!!

Goal of the day

- Correctness of Brzozowski's algorithm (co)algebraically
- Generalizations to other types of automata

Deterministic Automata are Algebras and Coalgebras

$(1 = \{0\})$ $\qquad$ $1 \xrightarrow{i} X \xrightarrow{f} 2$ $\qquad$ $(2 = \{0, 1\})$

$$X \xrightarrow{t} X^A$$

$$1 + A \times X \xrightarrow{[i,t]} X$$

$$\frac{\dfrac{X \to X^A}{A \times X \to X}}{A \to (X \to X)}$$

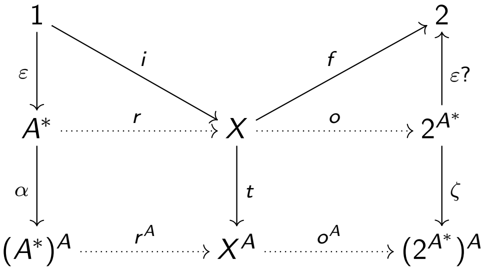
$$X \xrightarrow{\langle f,t \rangle} 2 \times X^A$$

initial state, transitions
$1 + A \times (-)$-algebra

transitions are both
algebra and coalgebra

output, transitions
$2 \times (-)^A$-coalgebra

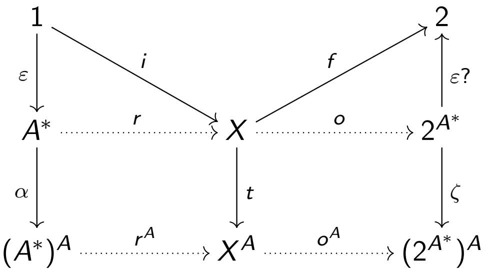
Initial Algebras and Final Coalgebras



For all $a \in A, w \in A^*$:

$$
\begin{aligned}
\alpha(w)(a) &= wa && \text{(append } a\text{)} \\
\zeta(S)(a) &= \{w \in A^* \mid aw \in S\} = a^{-1}S && \text{(left } a\text{-derivative)} \\
\\
r(w) &= t(i)(w) && \text{(state reached on input } w\text{)} \\
o(x) &= \{w \in A^* \mid f(t(x)(w)) = 1\} && \text{(language acepted by } x\text{)}
\end{aligned}
$$

Reachability, Observability, Minimality



Def. (Arbib & Manes)

Automaton $\langle X, t, i, f \rangle$ is ...

- reachable if r is surjective (no algebraic redundancy).
- observable if o is injective (no coalgebraic redundancy).
- minimal if it is *reachable and observable*.

(Contravariant) Powerset construction

$$2^{(-)} \; : \quad \begin{array}{c} V \\ \Big\downarrow g \\ W \end{array} \quad \mapsto \quad \begin{array}{c} 2^V \\ \Big\uparrow 2^g \\ 2^W \end{array}$$

where $2^V = \{S \mid S \subseteq V\}$ and, for all $S \subseteq W$,

$$2^g(S) = g^{-1}(S) \quad (= \{v \in V \mid g(v) \in S\})$$

- Note: if g is *surjective*, then 2^g is *injective*.

Reversing an Automaton

- <u>$2^{(-)}$ reverses transitions and determinises:</u>

$$
\begin{array}{c}X\\ \downarrow t\\ X^A\end{array}
\;\Bigg\|\;
\begin{array}{c}X\times A\\ \downarrow\\ X\end{array}
\quad\xmapsto{\;2^{(-)}\;}\quad
\begin{array}{c}2^{X\times A}\\ \uparrow\\ 2^X\end{array}
\;\Bigg\|\;
\begin{array}{c}(2^X)^A\\ \uparrow 2^t\\ 2^X\end{array}
$$

Reversed transitions: $S \xrightarrow{a} t_a^{-1}(S)$ (a-predecessors of S)

- <u>initial becomes final:</u>

$$i: 1 \to X \quad \longmapsto \quad 2^i: 2^X \to 2^1 = 2$$

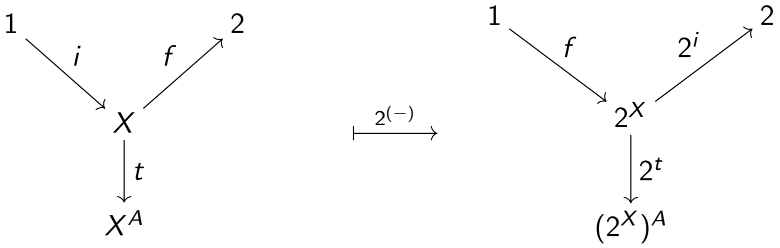
In reversed automaton: S is final iff $i \in S$.

- <u>final becomes initial:</u>

$$f: X \to 2 = 2^1 \quad \longmapsto \quad f: 1 \to 2^X$$

In reversed automaton: initial state is set of final states f.

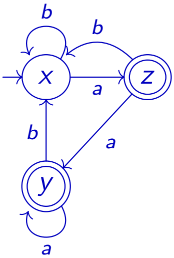
Reversing the entire automaton



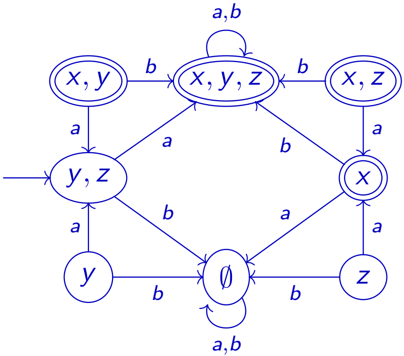
- Initial and final are exchanged . . .
- transitions are reversed . . .
- and the result is again deterministic!

Our previous example

$X =$



$2^X =$



- Note that X has been reversed and determinized:

$$2^X = det(rev(X))$$

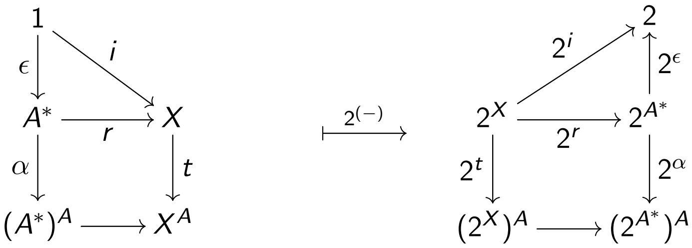
Proving today's Theorem

If: a deterministic automaton X is *reachable* and accepts $L(X)$

then: 2^X ($=$ $det(rev(X))$) is *observable* and

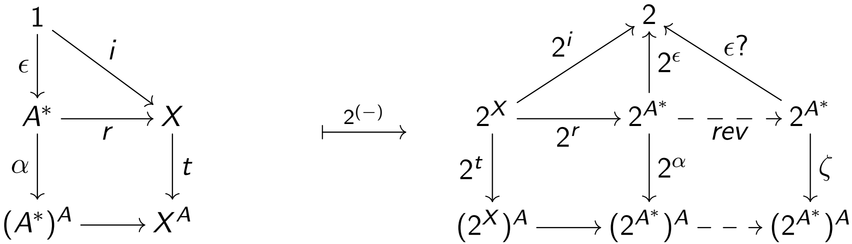
$$L(\, 2^X \,) \;=\; reverse(\, L(X)\,)$$

Proof: by reversing $A^* \xrightarrow{r} X$



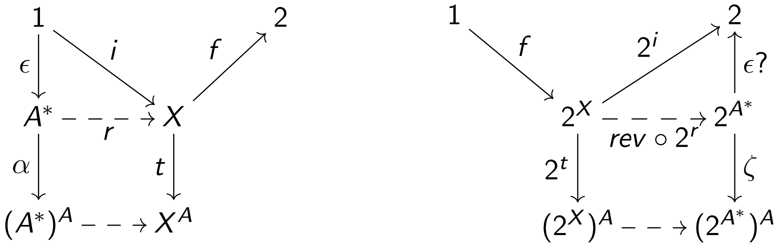
- X becomes 2^X
- initial automaton A^* becomes (almost) final automaton 2^{A^*}
- r is *surjective* $\Rightarrow$ 2^r is *injective*

Reachable becomes observable



- If r is *surjective* then (2^r and hence) $rev \circ 2^r$ is *injective*.
- That is, 2^X is observable.

Summarizing



- If: X is reachable, i.e., r is surjective

  then: $rev \circ 2^r$ is injective, i.e., 2^X is observable.

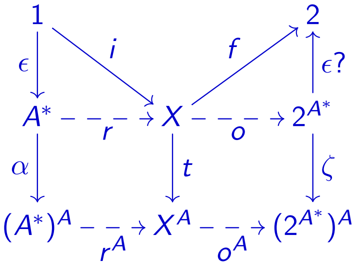
- And: $rev(2^r(f)) = rev(o(i))$, i.e., $L(\,2^X\,) \;=\; reverse(\,L(X)\,)$

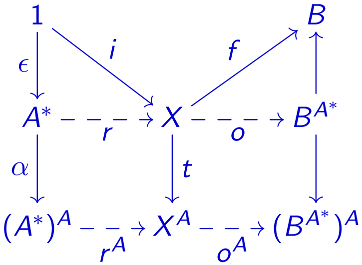
Corollary: Brzozowski's algorithm

- X becomes 2^X, accepting $\mathit{reverse}(L(X))$
- take reachable part: $Y = \mathit{reachable}(2^X)$
- Y becomes 2^Y, which is minimal and accepts

$$\mathit{reverse}(\mathit{reverse}(L(X))) = L(X)$$

Generalizations



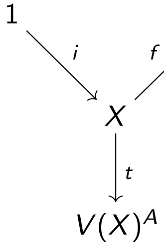


- A Brzozowski minimization algorithm for *Moore* automata.

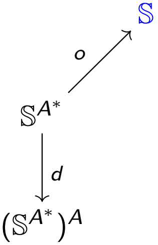
$$B^X = \{\phi \mid \phi\colon X \to B\} \qquad B^f(\phi) = \phi \circ f$$

Brzozowski for Weighted Automata

Weighted Automata

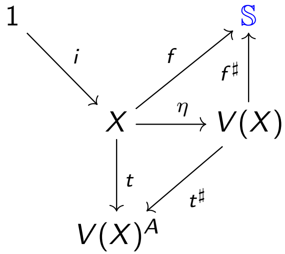


Weighted languages

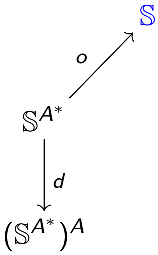


Brzozowski for Weighted Automata

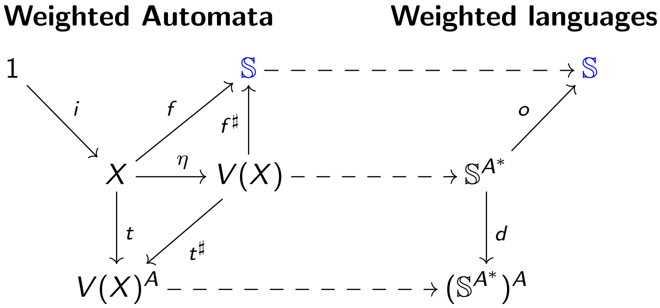
Weighted Automata



Weighted languages



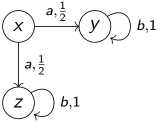
Brzozowski for Weighted Automata



Brzozowski for weighted languages: given a weighted automaton we want a **canonical** representative of the image in the final coalgebra – Moore automaton.

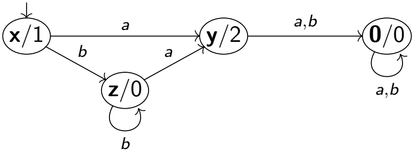
Brzozowski for weighted automata

Weighted automaton which recognizes $\sigma\colon A^* \to \mathbb{S}$:



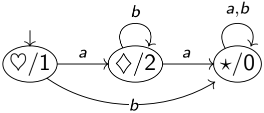
$ab^* \mapsto 2$

"Reverse and determinize" (Worthington):



$b^*a \mapsto 2$

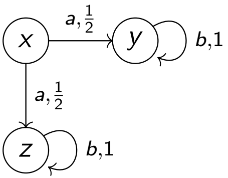
"Reverse and determinize" for Moore automata (using B^-):



$ab^* \mapsto 2$

Example

$$X = \{x, y, z\},\ i = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},\ f = \begin{pmatrix} 1 & 2 & 2 \end{pmatrix}$$



$$t_a = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \qquad t_b = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$ab \mapsto \begin{pmatrix} 1 & 2 & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 2$$

$$L(x) = \{\varepsilon \mapsto 1, ab^* \mapsto 2\}$$

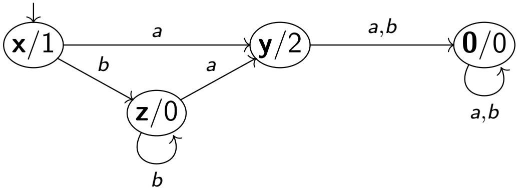
Reversing the automaton (Worthinghton)

Moore automaton that recognizes the reverse weighted language.
Initial vector: $f^T = \begin{pmatrix} 1 & 2 & 2 \end{pmatrix}^T$, final vector: $i^T = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ and transition function is **transposed**.

$$t_a^T = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad t_b^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$t^T \colon V(X) \to V(X)^A$$

Reachable automaton from $\mathbf{x} = f^T$:



$$L(\mathbf{x}) = \{\varepsilon \mapsto 1; b^* a \mapsto 2\} = rev(L(x))$$

Part II: Brzozowski's algorithm via adjunctions

Part II: Brzozowski's algorithm via adjunctions

Motivation: Gain deeper understanding of

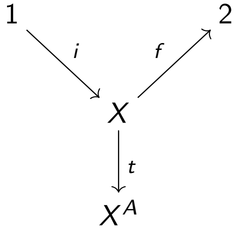
- the construction/algorithm,
- relation to similar constructions,
- uniform proofs.

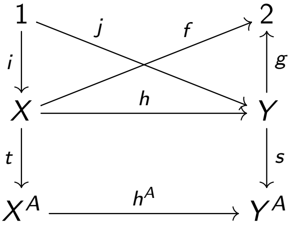
Overview:

- Categories of automata.
- Adjunction of automata via reversal.
- Brzozowski, functorially.
- Generalisations to Moore and weighted automata.
- Generalised dual adjunction.
- Related work.

Categories of Automata

Aut $=$ category of all deterministic automata, and automaton morphisms:





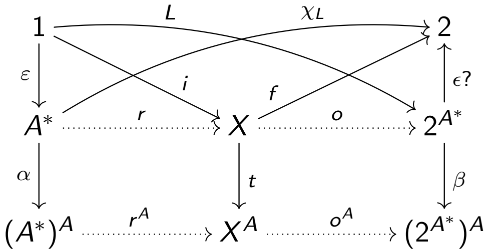
Note:

- Automaton morphisms preserve language.
- No initial object, no final object in Aut.

The Category Aut(L)

Aut(L) $=$ subcategory of Aut of automata accepting L.

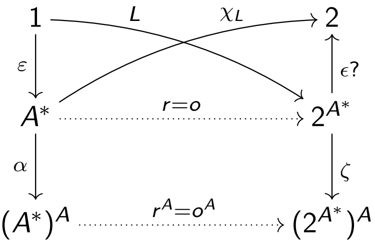
Initial and final objects regained:



Automaton $\langle X, t, i, f \rangle$ in Aut(L) is ...

- reachable if <u>initial morphism r is surjective</u>.
- observable if <u>final morphism o is injective</u>.

Myhill-Nerode via Aut(L)



Characterisation:

L regular

iff index($\equiv_L$) is finite

iff left-quotients(L) is finite

- $o(w) = \{u \in A^* \mid wu \in L\} = w^{-1}L$
- ker(o) is Myhill-Nerode-equivalence:
  $w \equiv_L v$ iff $\forall u \in A^* : wu \in L \iff vu \in L$
- img(o) is set of left-quotients of L.
- $|\mathrm{img}(o)| = \mathrm{index}(\equiv_L)$

Adjoint Automata: Main tools

- Adjunction of state spaces:

$$\mathsf{Set} \underset{2^{(-)^{\mathrm{op}}}}{\overset{2^{(-)}}{\rightleftarrows}} \mathsf{Set}^{\mathrm{op}} \quad (\perp) \qquad\qquad \frac{X \to 2^Y \quad \text{in Set}}{2^X \to Y \quad \text{in Set}^{\mathrm{op}}}$$

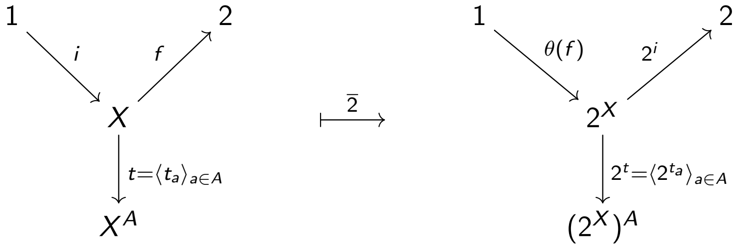
- Exponential transpose:

$$\frac{g\colon X \to 2^Y \quad \text{in Set}}{\theta(g)\colon Y \to 2^X \quad \text{in Set}}$$

- Transpose lemma:

$$\theta(X \xrightarrow{h} Y \xrightarrow{f} 2^Z) \quad = \quad Z \xrightarrow{\theta(f)} 2^Y \xrightarrow{2^h} 2^X$$

Reversing an Automaton



$$\mathcal{X}: \qquad 1 \xrightarrow{i} X \xrightarrow{t_{a_1}} X \quad \cdots \quad X \xrightarrow{t_{a_n}} X \xrightarrow{f} 2$$

$$\overline{2}(\mathcal{X}): \qquad 2 \xleftarrow{2^i} 2^X \xleftarrow{2^{t_{a_1}}} 2^X \quad \cdots \quad 2^X \xleftarrow{2^{t_{a_n}}} 2^X \xleftarrow{\theta(f)} 1$$

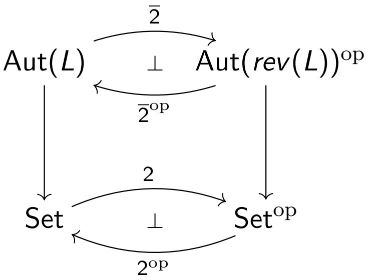
Reversal is Functorial

Theorem:

- $L(\overline{2}(\mathcal{X})) = rev(L(\mathcal{X}))$.
- Reversing is functor $\overline{2}\colon \mathsf{Aut} \to \mathsf{Aut}^{\mathrm{op}}$.
- Reversing is functor $\overline{2}\colon \mathsf{Aut}(L) \to \mathsf{Aut}(rev(L))^{\mathrm{op}}$.

Adjunction of Automata

Theorem: Reversal lifts dual adjunction on Set to dual adjunction of automata:



Corollary (duality): Let $\mathcal{A}$ be initial object in Aut(L), $\mathcal{Z}$ the final object in Aut($rev(L)$), and let $\mathcal{X}$ be an automaton in Aut(L).

$$r\colon \mathcal{A} \twoheadrightarrow \mathcal{X} \qquad \overset{\overline{2}}{\longmapsto} \qquad o\colon \overline{2}(\mathcal{X}) \rightarrowtail \overline{2}(\mathcal{A}) = \mathcal{Z}$$

$\mathcal{X}$ reachable $\Longrightarrow$ $\overline{2}(\mathcal{X})$ observable

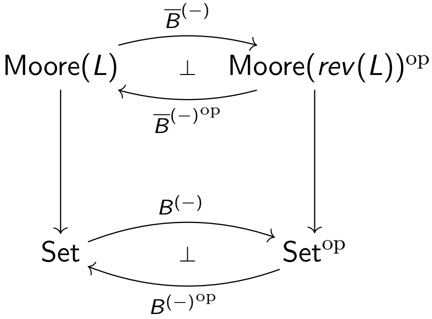
Brzozowski's algorithm, functorially

- Let: $\mathrm{rAut}(L) = $ reachable automata accepting L,
  $\mathrm{oAut}(L) = $ observable automata accepting L,
  $\mathrm{mAut}(L) = $ minimal automata accepting L.
- Reachability is functor $R\colon \mathrm{Aut}(L) \to \mathrm{rAut}(L)$ (coreflector).
  Restricts to $R\colon \mathrm{oAut}(L) \to \mathrm{mAut}(L)$.
- Brzozowski's algorithm is $R \circ \overline{2}^{\mathrm{op}} \circ R^{\mathrm{op}} \circ \overline{2}$:

$$\begin{array}{ccc} \mathrm{Aut}(L) & \xrightarrow{\overline{2}} & \mathrm{Aut}(rev(L))^{\mathrm{op}} \\ {\scriptstyle O}\,\vdots & & \downarrow {\scriptstyle R^{\mathrm{op}}} \\ \mathrm{oAut}(L) & \xleftarrow{\overline{2}^{\mathrm{op}}} & \mathrm{rAut}(rev(L))^{\mathrm{op}} \\ {\scriptstyle R}\,\downarrow & & \\ \mathrm{mAut}(L) & & \end{array}$$

Brzozowski for Moore Automata, revisited

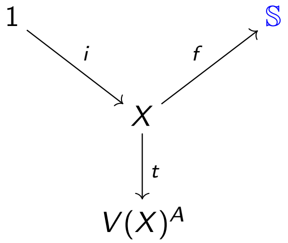
Adjunction of Moore Automata:



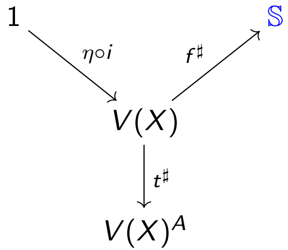
- $L\colon A^* \to B$ (B-weighted language), $rev(L)(w) = L(w^R)$.
- Reversal functor $B^{(-)} = \mathsf{Set}(-, B)$.
- Brzozowski minimization, functorially ✓

Brzozowski for Weighted Automata

Weighted Automaton in Set



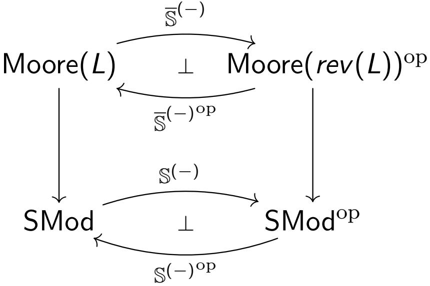
Moore Automaton in SMod



- $\mathbb{S}$ is a commutative semiring $(S, +, \cdot, 0, 1)$.
- SMod $=$ $\mathbb{S}$-semimodules and $\mathbb{S}$-linear maps
- $V(X) = \{s_1 x_1 + \ldots + s_n x_n \mid s_i \in \mathbb{S}, x_i \in X\}$ (free on X)

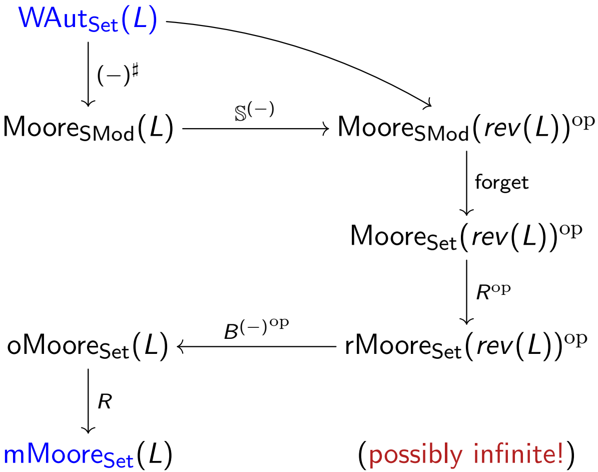
Brzozowski for Weighted Automata, revisited

Adjunction of Moore Automata over SMod:

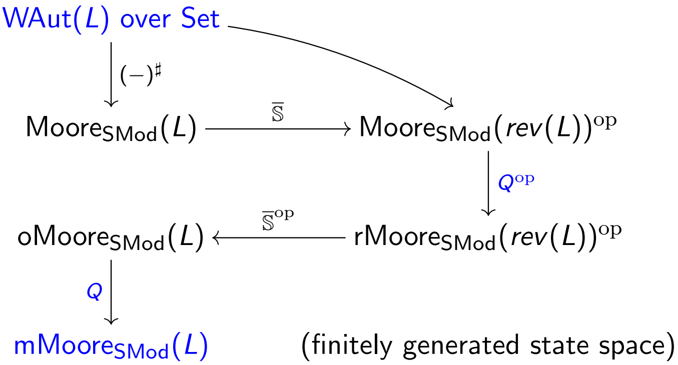


- $L \colon A^* \to \mathbb{S}$ (formal power series), $rev(L)(w) = L(w^R)$.
- Reversal functor: $(-)^* = \mathbb{S}^{(-)} = \mathsf{SMod}(-, \mathbb{S})$ (dual space)
- Note: $V(X)^* = V(X^*)$ for finite X.

Brzozowski for WAut via Brzozowski for Moore



Brzozowski for WAut in SMod

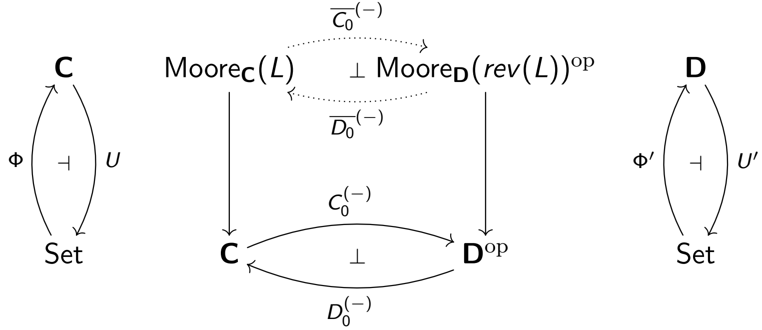


Note:

- reachability "illegal operation" in $\mathsf{Moore}_{\mathsf{SMod}}(L)$.
- $\mathcal{A} \twoheadrightarrow Q(\mathcal{X})$ (image/quotient of initial object?).
- $Q(\mathcal{X})$ finitely generated if $\mathbb{S}$ Noetherian.

Generalised Duality

Assume $\mathbf{C}, \mathbf{D}$ have products.

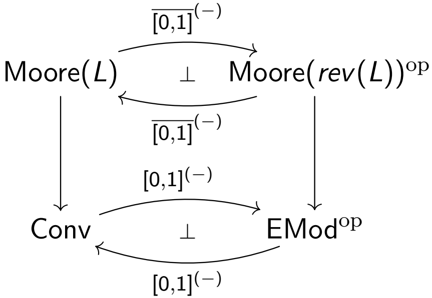


Moore automaton over $\mathbf{C}$:

state space:	$C \in \mathbf{C}$
initial state:	$i\colon 1 \to UC$
transitions:	$t_a\colon C \to C,\ a \in A$
output:	$f\colon C \to D_0^{\Phi' 1}$

Example: Quantum Automata

Cf. Bart Jacobs, Frank Roumen.

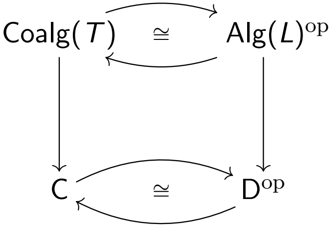


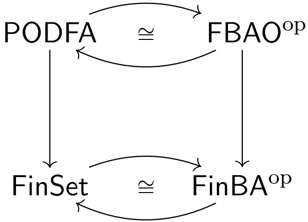
State	$\leftrightsquigarrow$	Observations
$\mathcal{DM}(\mathcal{H})$		$\mathcal{E}\!f\!f(\mathcal{H})$
Schrödinger		Heisenberg

Brzozowski minimization? Finitely generated resulting automaton?

Related Work

- Bezhanishvili, Kupke, Panangaden (WoLLIC 2012): minimisation via dual equivalence coalgebra-algebra (deterministic, linear weighted, belief automata).





(Both left and right adjoint must preserve epis.)

- Arbib, Manes, Gehrke, Pin, König, Hülsbusch, Milius, Adamek, Myers, Worthington,...

Conclusion

Summary:

- Brzozowski algorithm via dual adjunction of automata.
- Duality: state and observations.
- Generalisations: given Moore/nondeterm/weighted automaton accepting L, construct minimal Moore automaton accepting L (language equivalence!).
- Future work: other automaton types (probabilistic, multi-sorted, ...), combination with generalised powerset construction, algebraic-coalgebraic automata theory.

Message:

- duality $\rightsquigarrow$ algorithms.
- categories $\rightsquigarrow$ generalisations, clarification.