Brzozowski's algorithm (co)algebraically

Jan Rutten

CWI & Radboud University

Motivation

- duality between reachability and observability:
beautiful, not very well-known.
- 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, Silva (CWI, 2011) and Hansen.
- based on Arbib and Manes, 1975.
- Cf. Panangaden, Kupke, Koenig, Adamek, Milius, Gehrke, Pin, Roumen, , . . .

Motivation

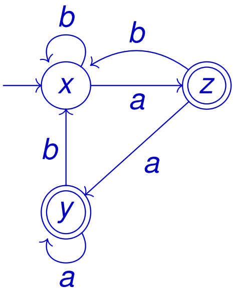
- duality between reachability and observability: beautiful, not very well-known.
- combined use of algebra and coalgebra.
- our understanding of automata is still very limited; cf. recent research: universal automata, àtomata, weighted automata (Sakarovitch, Brzozowski, . . .)
- joint work with Bonchi, Bonsangue, Silva (CWI, 2011) and Hansen.
- based on Arbib and Manes, 1975.
- Cf. Panangaden, Kupke, Koenig, Adamek, Milius, Gehrke, Pin, Roumen, , . . .

Overview

1. Brzozowski's algorithm: example
2. (Co)algebra
3. Automata (co)algebraically
4. The duality between reachability and observability
5. Reversing the automaton
6. Conclusions

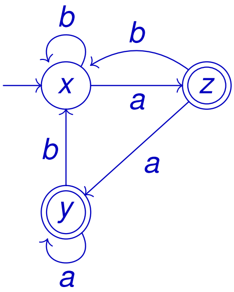
1. Brzozowski's algorithm: example

A deterministic automaton $X = \{x, y, z\}$



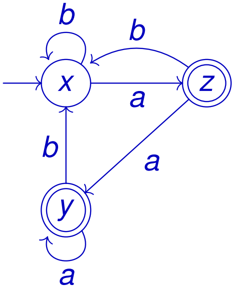
- initial state: x
- final states: y and z
- $L(x) = \{a, b\}^* a$

A deterministic automaton $X = \{x, y, z\}$



- initial state: x
- final states: y and z
- $L(x) = \{a, b\}^* a$

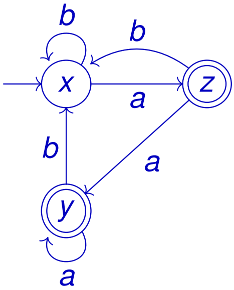
A deterministic automaton $X = \{x, y, z\}$



- initial state: x
- final states: y and z
- $L(x) = \{a, b\}^* a$

Reversing the automaton: $rev(X)$

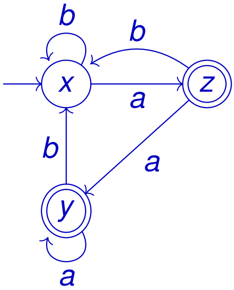
$X =$

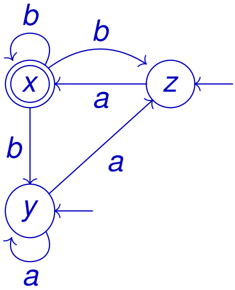


$rev(X) =$

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

Reversing the automaton: $rev(X)$

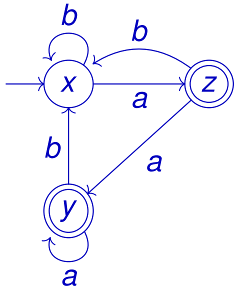
$X =$ 

$rev(X) =$ 

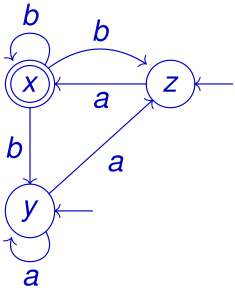
- transitions are reversed
- initial states $\Leftrightarrow$ final states
- $rev(X)$ is non-deterministic

Reversing the automaton: $rev(X)$

$X =$



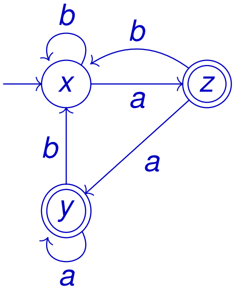
$rev(X) =$



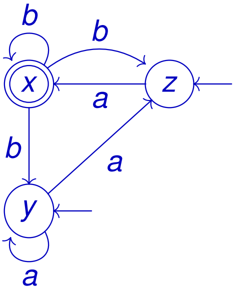
- transitions are reversed
- initial states $\Leftrightarrow$ final states
- $rev(X)$ is non-deterministic

Reversing the automaton: $rev(X)$

$X =$



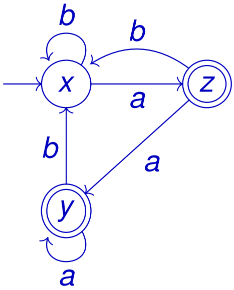
$rev(X) =$



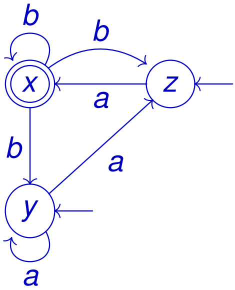
- transitions are reversed
- initial states $\Leftrightarrow$ final states
- $rev(X)$ is non-deterministic

Reversing the automaton: *rev*(X)

$X =$

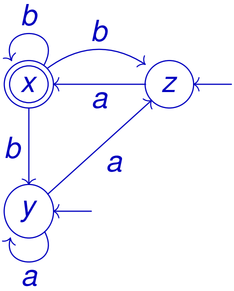


rev$(X) =$



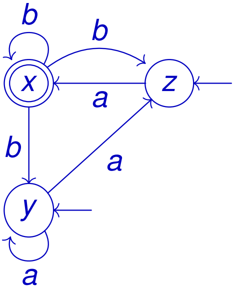
- transitions are reversed
- initial states $\Leftrightarrow$ final states
- *rev*(X) is non-deterministic

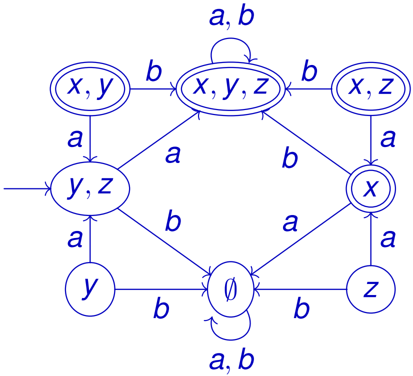
Making it deterministic again: $det(rev(X))$



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

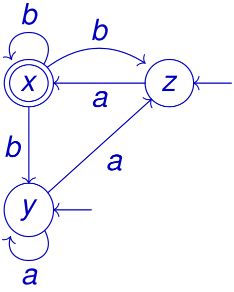
Making it deterministic again: $det(rev(X))$

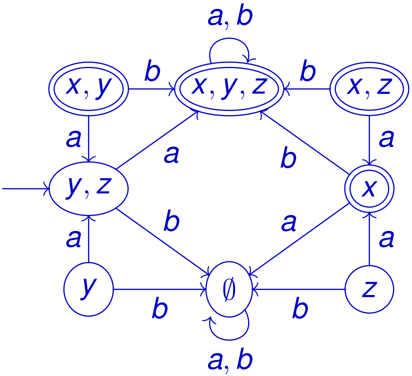




- new state space: $2^X = \{ V \mid V \subseteq \{x, y, z\} \}$
- $V \xrightarrow{a} W \quad \Leftrightarrow \quad \forall w \in W \, \exists v \in V \; : \; v \xrightarrow{a} w$
- initial state: $\{y, z\}$
- final states: all V with $x \in V$

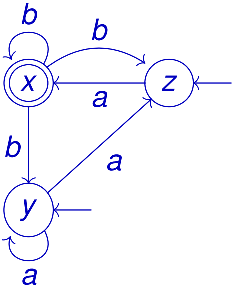
Making it deterministic again: $det(rev(X))$

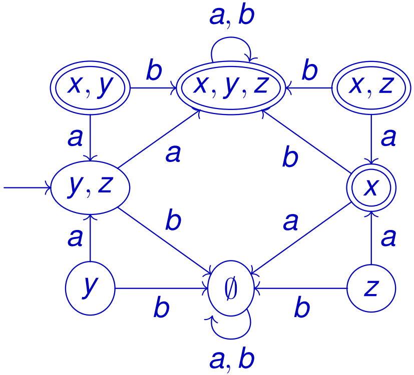




- new state space: $2^X = \{ V \mid V \subseteq \{x, y, z\} \}$
- $V \xrightarrow{a} W \quad \Leftrightarrow \quad \forall w \in W \, \exists v \in V \; : \; v \xrightarrow{a} w$
- initial state: $\{y, z\}$ • final states: all V with $x \in V$

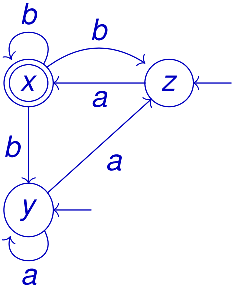
Making it deterministic again: $det(rev(X))$

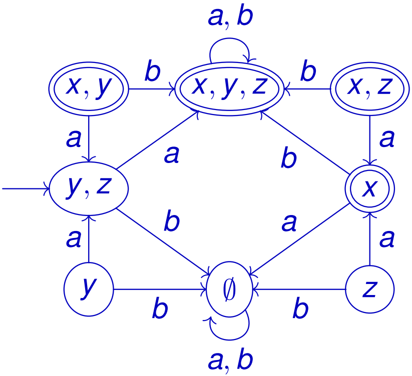




- new state space: $2^X = \{\, V \mid V \subseteq \{x, y, z\}\,\}$
- $V \xrightarrow{a} W \quad \Leftrightarrow \quad \forall w \in W\ \exists v \in V \ :\ \ v \xrightarrow{a} w$
- initial state: $\{y, z\}$ • final states: all V with $x \in V$

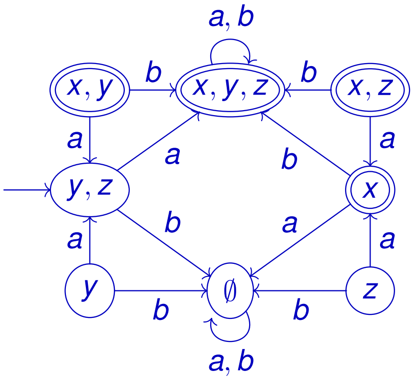
Making it deterministic again: $det(rev(X))$





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

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

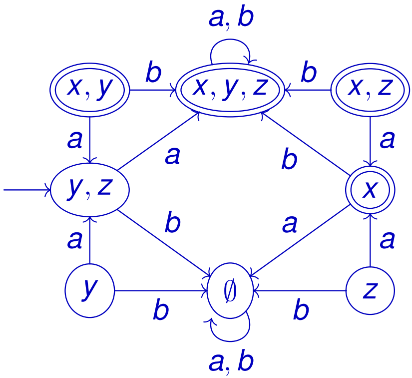


- . . . accepts the reverse of the language accepted by X:

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

- . . . and is minimal!

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

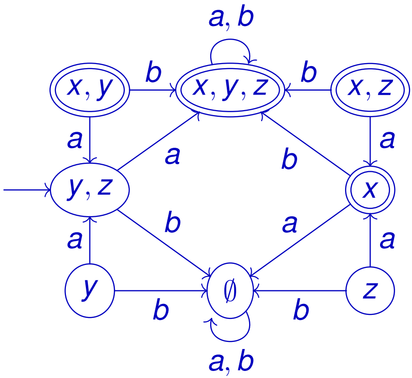


- . . . accepts the reverse of the language accepted by X:

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

- . . . and is minimal!

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 minimal!

Today's Theorem

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

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

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

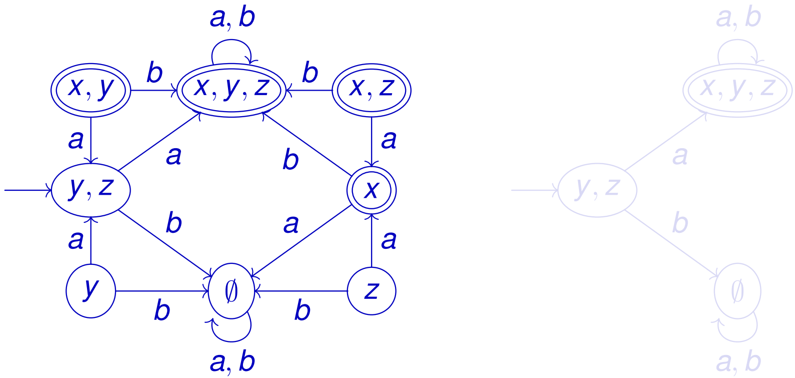
Today's Theorem

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

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

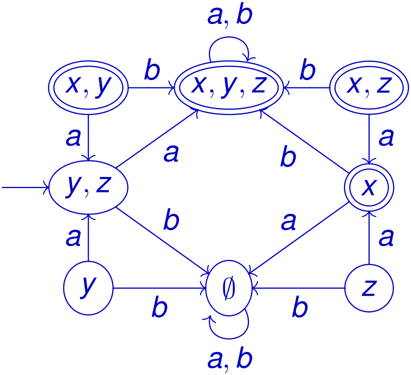
$$L(\,det(rev(X))\,) \;=\; reverse(\,L(X)\,)$$

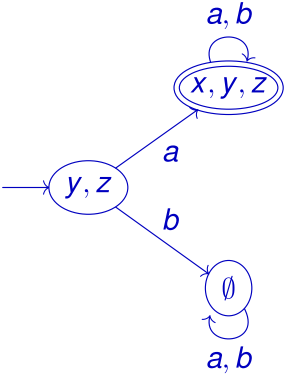
Taking the reachable part of *det*(*rev*(*X*))



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

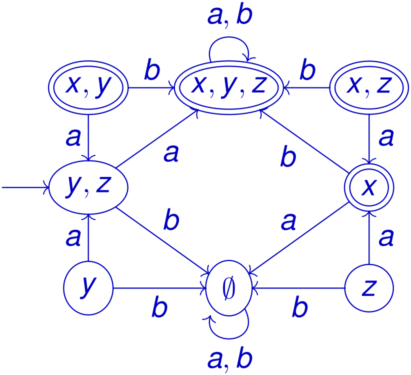
Taking the reachable part of $det(rev(X))$

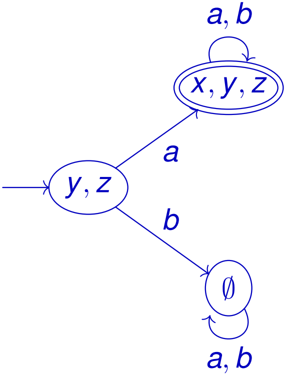




- $reach(det(rev(X)))$ is reachable (by construction)

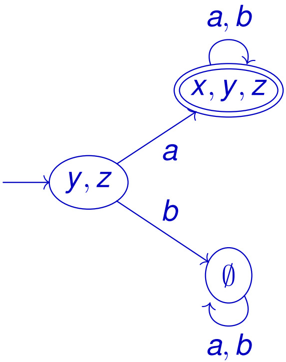
Taking the reachable part of *det*(*rev*(*X*))





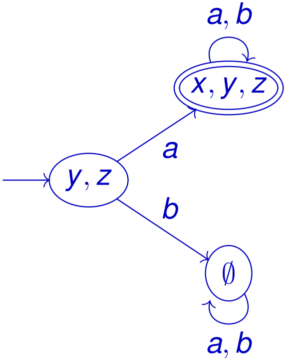
- *reach*(*det*(*rev*(*X*))) is reachable (by construction)

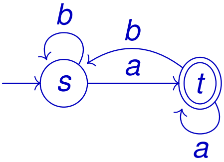
Repeating everything, now for *reach*(*det*(*rev*(*X*)))



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

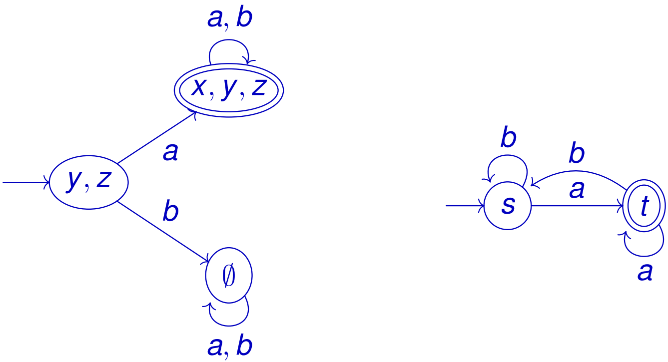
Repeating everything, now for *reach*(*det*(*rev*(X)))





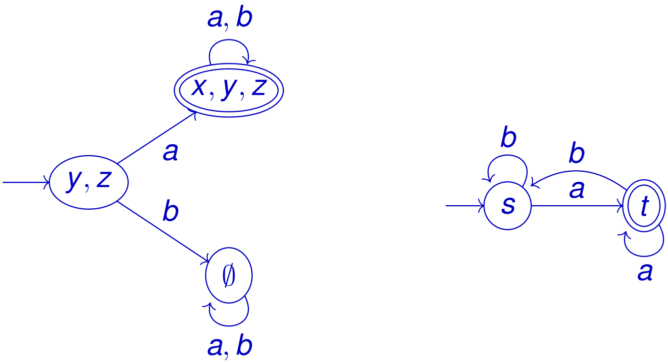
- . . . gives us *reach*(*det*(*rev*(*reach*(*det*(*rev*(X))))))
- which is (reachable and) minimal and accepts $\{a, b\}^* a$.

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



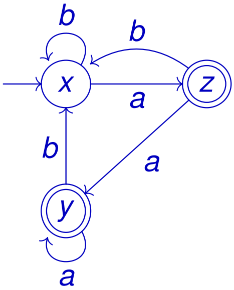
- . . . gives us $reach(det(rev(reach(det(rev(X))))))$
- which is (reachable and) minimal and accepts $\{a, b\}^* a$.

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



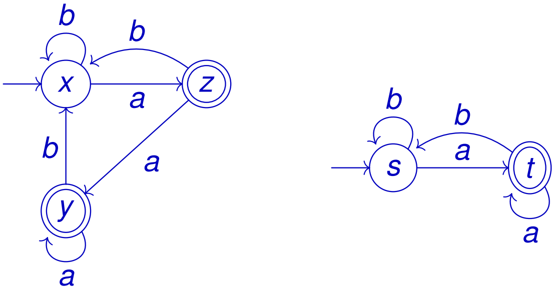
- . . . gives us *reach*(*det*(*rev*(*reach*(*det*(*rev*(*X*))))))
- which is (reachable and) minimal and accepts $\{a, b\}^* a$.

All in all: Brzozowski's algorithm



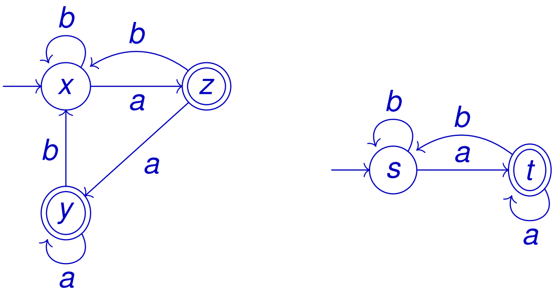
- X is reachable and accepts $\{a, b\}^* a$
- $reach(det(rev(reach(det(rev(X))))))$ also accepts $\{a, b\}^* a$
- . . . and is minimal!!

All in all: Brzozowski's algorithm



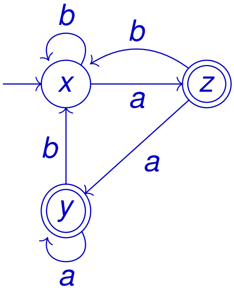
- X is reachable and accepts $\{a, b\}^* a$
- $reach(det(rev(reach(det(rev(X))))))$ also accepts $\{a, b\}^* a$
- . . . and is minimal!!

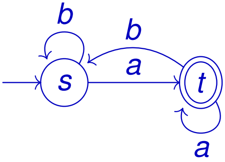
All in all: Brzozowski's algorithm



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

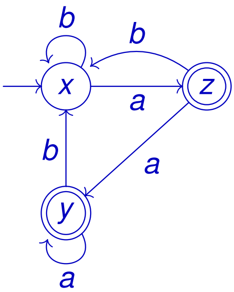
All in all: Brzozowski's algorithm

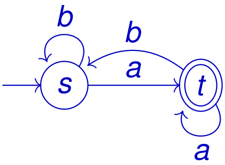




- X is reachable and accepts $\{a, b\}^* \, a$
- $reach(det(rev(reach(det(rev(X))))))$ also accepts $\{a, b\}^* \, a$
- . . . and is minimal!!

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!!

2. (Co)algebra

algebras:
$$\begin{array}{c} F(X) \\ \big\downarrow{\scriptstyle f} \\ X \end{array}$$

coalgebras:
$$\begin{array}{c} X \\ \big\downarrow{\scriptstyle f} \\ F(X) \end{array}$$

Examples of algebras

$$
\begin{tikzcd}
\mathbb{N} \times \mathbb{N} \arrow[d, "+"'] \\
\mathbb{N}
\end{tikzcd}
$$

$$
\begin{tikzcd}
1 + \mathbb{N} \arrow[d, "{[0, S]}"'] \\
\mathbb{N}
\end{tikzcd}
\equiv
\begin{tikzcd}
1 \arrow[dr, "0"'] & & \mathbb{N} \arrow[dl, "S"] \\
& \mathbb{N} &
\end{tikzcd}
\equiv
\begin{tikzcd}
1 \arrow[dr, "0"'] & \\
& \mathbb{N} \arrow[d, "S"] \\
& \mathbb{N}
\end{tikzcd}
$$

Examples of algebras

$$
\begin{tikzcd}
\mathbb{N} \times \mathbb{N} \arrow[d, "+"'] \\
\mathbb{N}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
1 + \mathbb{N} \arrow[d, "{[0, S]}"'] \\
\mathbb{N}
\end{tikzcd}
\equiv
\begin{tikzcd}
1 \arrow[dr, "0"'] & & \mathbb{N} \arrow[dl, "S"] \\
& \mathbb{N} &
\end{tikzcd}
\equiv
\begin{tikzcd}
1 \arrow[dr, "0"'] & \\
& \mathbb{N} \arrow[d, "S"] \\
& \mathbb{N}
\end{tikzcd}
$$

Examples of coalgebras

$$\begin{tikzcd} X \arrow[d, "t"'] \\ \mathcal{P}(A \times X) \end{tikzcd} \qquad x \xrightarrow{a} y \iff \langle a, y \rangle \in t(x)$$

$$\begin{tikzcd} X \arrow[d, "\langle Left, label, Right \rangle"'] \\ X \times A \times X \end{tikzcd}$$

Examples of coalgebras

$$\begin{tikzcd} X \arrow[d, "t"'] \\ \mathcal{P}(A \times X) \end{tikzcd} \qquad x \xrightarrow{a} y \quad \Longleftrightarrow \quad \langle a, y \rangle \in t(x)$$

$$\begin{tikzcd} X \arrow[d, "\langle Left, label, Right \rangle"'] \\ X \times A \times X \end{tikzcd}$$

Examples of coalgebras

$$
\begin{tikzcd}
2^\omega \arrow[d, "\langle\, \mathrm{head},\ \mathrm{tail}\,\rangle"'] \\
2 \times 2^\omega
\end{tikzcd}
\quad \equiv \quad
\begin{tikzcd}
 & 2^\omega \arrow[dl, "\mathrm{head}"'] \arrow[dr, "\mathrm{tail}"] & \\
2 & & 2^\omega
\end{tikzcd}
\quad \equiv \quad
\begin{tikzcd}
 & 2 \\
2^\omega \arrow[ur, "\mathrm{head}"'] \arrow[d, "\mathrm{tail}"] & \\
2^\omega &
\end{tikzcd}
$$

$$
\begin{aligned}
\mathrm{head}((b_0, b_1, b_2, \ldots)) &= b_0 \\
\mathrm{tail}((b_0, b_1, b_2, \ldots)) &= (b_1, b_2, b_3 \ldots)
\end{aligned}
$$

Homomorphisms

$$
\begin{tikzcd}
F(X) \arrow[r, "F(h)"] \arrow[d, "f"'] & F(Y) \arrow[d, "g"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "f"'] & Y \arrow[d, "g"] \\
F(X) \arrow[r, "F(h)"'] & F(Y)
\end{tikzcd}
$$

Homomorphisms

$$
\begin{tikzcd}
F(X) \arrow[r, "F(h)"] \arrow[d, "f"'] & F(Y) \arrow[d, "g"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
X \arrow[r, "h"] \arrow[d, "f"'] & Y \arrow[d, "g"] \\
F(X) \arrow[r, "F(h)"'] & F(Y)
\end{tikzcd}
$$

Initiality, finality

$$\begin{tikzcd} F(A) \arrow[r, dashed, "F(h)"] \arrow[d, "\alpha"'] & F(X) \arrow[d, "f"] \\ A \arrow[r, dashed, "\exists!\, h"'] & X \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, dashed, "\exists!\, h"] \arrow[d, "f"'] & Z \arrow[d, "\beta"] \\ F(X) \arrow[r, dashed, "F(h)"'] & F(Z) \end{tikzcd}$$

- initial algebras $\Longleftrightarrow$ *induction*
- final coalgebras $\Longleftrightarrow$ *coinduction*

Initiality, finality

$$\begin{tikzcd} F(A) \arrow[r, dashed, "F(h)"] \arrow[d, "\alpha"'] & F(X) \arrow[d, "f"] \\ A \arrow[r, dashed, "\exists ! \, h"'] & X \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, dashed, "\exists ! \, h"] \arrow[d, "f"'] & Z \arrow[d, "\beta"] \\ F(X) \arrow[r, dashed, "F(h)"'] & F(Z) \end{tikzcd}$$

- initial algebras $\Longleftrightarrow$ *induction*
- final coalgebras $\Longleftrightarrow$ *coinduction*

Initiality, finality

$$
\begin{tikzcd}
F(A) \arrow[r, dashed, "F(h)"] \arrow[d, "\alpha"'] & F(X) \arrow[d, "f"] \\
A \arrow[r, dashed, "\exists!\, h"'] & X
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
X \arrow[r, dashed, "\exists!\, h"] \arrow[d, "f"'] & Z \arrow[d, "\beta"] \\
F(X) \arrow[r, dashed, "F(h)"'] & F(Z)
\end{tikzcd}
$$

- initial algebras $\Longleftrightarrow$ *induction*
- final coalgebras $\Longleftrightarrow$ *coinduction*

3. Automata, (co)algebraically

- Automata are complicated structures:

  part of them is algebra - part of them is coalgebra

- (. . . in two different ways . . .)

3. Automata, (co)algebraically

- Automata are complicated structures:

  part of them is algebra - part of them is coalgebra
- (. . . in two different ways . . .)

A deterministic automaton

$$\begin{tikzcd} 1 \arrow[dr, "i"] & & 2 \\ & X \arrow[ur, "f"] \arrow[d, "t"] & \\ & X^A & \end{tikzcd}$$

where

$$1 = \{0\} \qquad 2 = \{0, 1\} \qquad X^A = \{\, g \mid g : A \to X \,\}$$

$$x \xrightarrow{a} y \quad \Longleftrightarrow \quad t(x)(a) = y$$

$i(0) \in X$ is the initial state

x is final (or accepting) $\Longleftrightarrow$ $f(x) = 1$

A deterministic automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"'] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
$$

where

$$1 = \{0\} \qquad 2 = \{0,1\} \qquad X^A = \{\, g \mid g : A \to X \,\}$$

$$x \xrightarrow{a} y \quad \Longleftrightarrow \quad t(x)(a) = y$$

$i(0) \in X$ is the initial state

x is final (or accepting) $\Longleftrightarrow$ $f(x) = 1$

Automata: algebra or coalgebra?

- initial state: algebraic – final states: coalgebraic

$$\begin{tikzcd} 1 \arrow[dr, "i"] & & 2 \\ & X \arrow[ur, "f"'] & \end{tikzcd}$$

- transition function: both algebraic and coalgebraic

$$\frac{\dfrac{X \xrightarrow{t} X^A}{X \longrightarrow (A \longrightarrow X)}}{X \times A \xrightarrow{t} X}$$

Automata: algebra or coalgebra?

- initial state: algebraic – final states: coalgebraic

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"] &
\end{tikzcd}
$$

- transition function: both algebraic and coalgebraic

$$
\frac{\dfrac{X \xrightarrow{t} X^A}{X \longrightarrow (A \longrightarrow X)}}{X \times A \xrightarrow{t} X}
$$

Automata: algebra *and* coalgebra!

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

To take home: this picture!! . . . which we'll explain next . . .

Automata: algebra *and* coalgebra!

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[ur, "f"] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

To take home: this picture!! . . . which we'll explain next . . .

Automata: algebra *and* coalgebra!

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

To take home: this picture!! . . . which we'll explain next . . .

The "automaton" of languages

$$\begin{tikzcd} 2 \\ 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\ (2^{A^*})^A \end{tikzcd}$$

$$\epsilon?(L) = 1 \iff \epsilon \in L$$

$$2^{A^*} = \{g \mid g : A^* \to 2\} \cong \{L \mid L \subseteq A^*\}$$

$$\beta(L)(a) = L_a = \{w \in A^* \mid a \cdot w \in L\}$$

- We say "automaton": it does not have an initial state.

The "automaton" of languages

$$\begin{tikzcd} 2 \\ 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\ (2^{A^*})^A \end{tikzcd}$$

$$\epsilon?(L) = 1 \iff \epsilon \in L$$

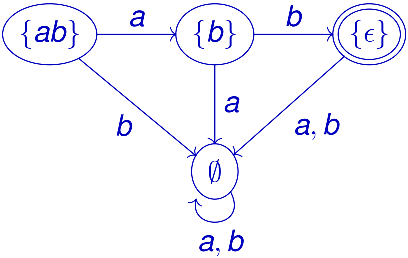
$$2^{A^*} = \{g \mid g : A^* \to 2\} \cong \{L \mid L \subseteq A^*\}$$

$$\beta(L)(a) = L_a = \{w \in A^* \mid a \cdot w \in L\}$$

- We say "automaton": it does not have an initial state.

The automaton of languages

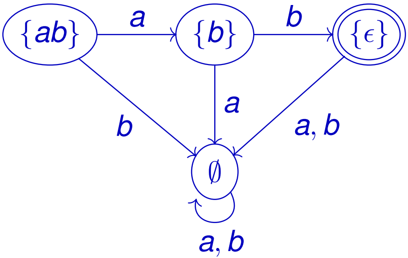
- transitions: $L \xrightarrow{a} L_a$ where $L_a = \{w \in A^* \mid a \cdot w \in L\}$
- for instance:

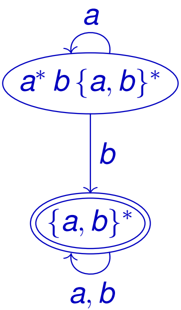


- note: every *state* L accepts . . . . the *language* L !!

The automaton of languages

- transitions: $L \xrightarrow{a} L_a$ where $L_a = \{w \in A^* \mid a \cdot w \in L\}$
- for instance:

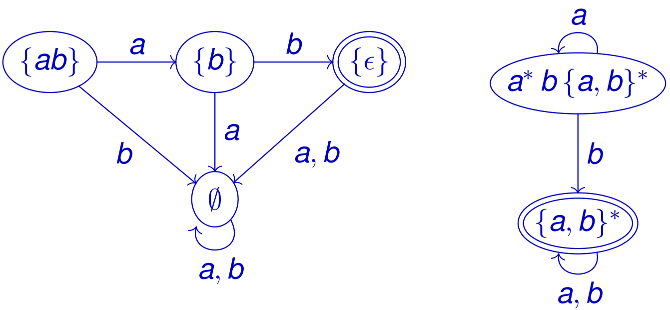




- note: every *state* L accepts . . . . the *language* L !!

The automaton of languages

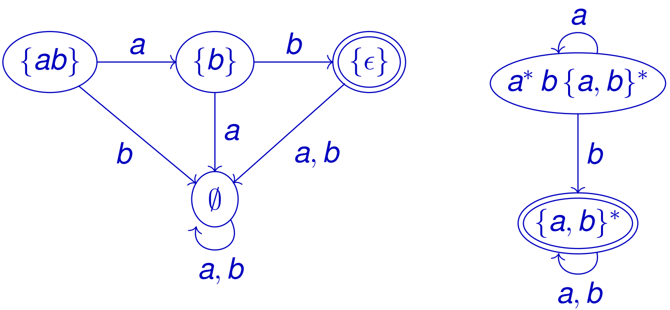
- transitions: $L \xrightarrow{a} L_a$ where $L_a = \{w \in A^* \mid a \cdot w \in L\}$
- for instance:



- note: every *state* L accepts . . . . . . the *language* L !!

The automaton of languages

- transitions: $L \xrightarrow{a} L_a$ where $L_a = \{w \in A^* \mid a \cdot w \in L\}$
- for instance:



- note: every *state* L accepts . . . . . . the *language* L !!

The automaton of languages is . . . *final*

$$
\begin{tikzcd}
 & 2 \\
X \arrow[ur, "f"] \arrow[r, dashed, "\exists!\, o"'] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

$$
\begin{aligned}
o(x) &= \{w \in A^* \mid f(x_w) = 1\} \\
&= \text{the language accepted by } x
\end{aligned}
$$

where: x_w is the state reached after inputting the word w,

and: $o^A(g) = o \circ g$, all $g \in X^A$.

The automaton of languages is . . . *final*

$$\begin{tikzcd} & & 2 \\ X \arrow[urr, "f"] \arrow[rr, dashed, "\exists!\, o"'] \arrow[d, "t"] & & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\ X^A \arrow[rr, dashed, "o^A"'] & & (2^{A^*})^A \end{tikzcd}$$

$$\begin{aligned} o(x) &= \{w \in A^* \mid f(x_w) = 1\} \\ &= \text{the language accepted by } x \end{aligned}$$

where: x_w is the state reached after inputting the word w,

and: $o^A(g) = o \circ g$, all $g \in X^A$.

An aside: a proof by coinduction

Question: $a^* \, b \, \{a, b\}^* = \{a, b\}^* \, b \, a^*$??

Answer: consider

and conclude: *yes*, since the relation

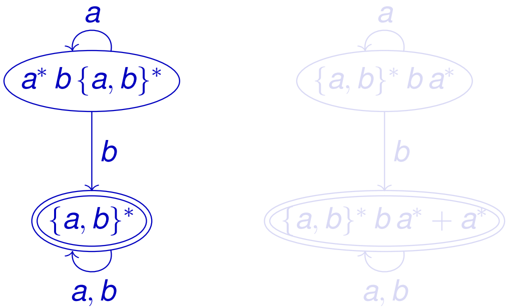
$$R = \{ \langle a^* \, b \, \{a, b\}^*, \{a, b\}^* \, b \, a^* \rangle , \; \langle \{a, b\}^*, \{a, b\}^* \, b \, a^* + a^* \rangle \}$$

is a (language!) bisimulation.

An aside: a proof by coinduction

Question: $a^* \, b \, \{a,b\}^* \; = \; \{a,b\}^* \, b \, a^*$??

Answer: consider



and conclude: *yes*, since the relation

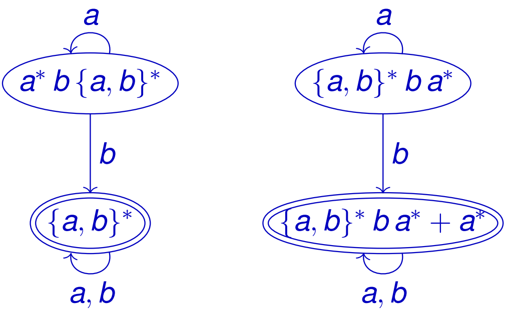
$$R = \{ \langle a^* \, b \, \{a,b\}^*, \{a,b\}^* \, b \, a^* \rangle \, , \; \langle \{a,b\}^*, \{a,b\}^* \, b \, a^* + a^* \rangle \}$$

is a (language!) bisimulation.

An aside: a proof by coinduction

Question: $a^* \, b \, \{a, b\}^* = \{a, b\}^* \, b \, a^*$??

Answer: consider



and conclude: *yes*, since the relation

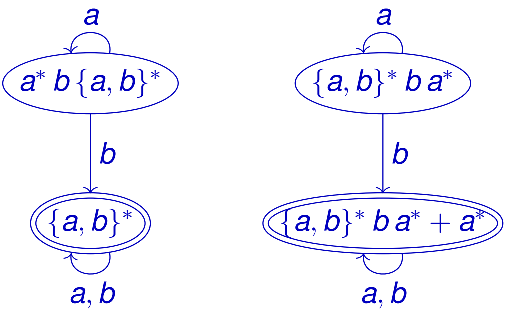
$$R = \{ \langle a^* \, b \, \{a, b\}^*, \{a, b\}^* \, b \, a^* \rangle , \langle \{a, b\}^*, \{a, b\}^* \, b \, a^* + a^* \rangle \}$$

is a (language!) bisimulation.

An aside: a proof by coinduction

Question: $a^* \, b \, \{a, b\}^* \; = \; \{a, b\}^* \, b \, a^*$??

Answer: consider



and conclude: *yes*, since the relation

$$R = \; \{ \, \langle \; a^* \, b \, \{a, b\}^*, \; \{a, b\}^* \, b \, a^* \, \rangle \; , \;\; \langle \, \{a, b\}^*, \; \{a, b\}^* \, b \, a^* + a^* \, \rangle \; \}$$

is a (language!) bisimulation.

Back to today's picture

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

On the right: final coalgebra

On the left: initial algebra . . .

Back to today's picture

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

On the right: final coalgebra

On the left: initial algebra . . .

Back to today's picture

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}$$

On the right: final coalgebra

On the left: initial algebra . . .

The "automaton" of words

$$\begin{tikzcd} 1 \arrow[d, "\epsilon"'] \\ A^* \arrow[d, "\alpha"'] \\ (A^*)^A \end{tikzcd}$$

ϵ is initial state

$\alpha(w)(a) = w \cdot a$

that is, transitions: $w \xrightarrow{a} w \cdot a$

The automaton of words is . . . *initial*

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, dashed, "\exists!\, r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A
\end{tikzcd}$$

$$\begin{aligned}
i \in X &= \text{initial state} \\
&\quad\ \text{(to be precise: } i(0)) \\
r(w) &= i_w \\
&= \text{the state } \textit{reached} \text{ from } i \\
&\quad\ \text{after inputting } w
\end{aligned}$$

- Proof: easy exercise.
- Proof: formally, because A^* is an initial $1 + A \times (-)$-algebra!

4. Duality

- Reachability and observability are dual:

  Arbib and Manes, 1975.

- (here observable = minimal)

Reachability and observability

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}$$

$$r(w) = \text{state reached on input } w$$

$$o(x) = \text{language accepted by } x$$

- We call X *reachable* if r is *surjective*.
- We call X *observable* (= minimal) if o is *injective*.

Reachability and observability

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

$r(w)$ = state reached on input w

$o(x)$ = language accepted by x

- We call X *reachable* if r is *surjective*.
- We call X *observable* (= minimal) if o is *injective*.

Reachability and observability

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

$r(w)$ = state reached on input w

$o(x)$ = language accepted by x

- We call X *reachable* if r is *surjective*.
- We call X *observable* (= minimal) if o is *injective*.

Reachability and observability

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[r, dashed, "o"'] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}$$

$r(w) =$ state reached on input w

$o(x) =$ language accepted by x

- We call X *reachable* if r is *surjective*.
- We call X *observable* (= minimal) if o is *injective*.

5. Reversing the automaton

- Reachability $\Longleftrightarrow$ observability
- Being precise about homomorphisms is crucial.
- Forms the basis for proof Brzozowski's algorithm.

Powerset construction

$$
2^{(-)} \;:\quad
\begin{tikzcd}
V \arrow[d, "g"'] \\
W
\end{tikzcd}
\quad\mapsto\quad
\begin{tikzcd}
2^V \\
2^W \arrow[u, "2^g"']
\end{tikzcd}
$$

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

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

- This construction is *contravariant* !!
- Note: if g is *surjective*, then 2^g is *injective*.

Powerset construction

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

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

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

- This construction is *contravariant* !!
- Note: if g is *surjective*, then 2^g is *injective*.

Powerset construction

$$
2^{(-)} : \quad \begin{tikzcd} V \arrow[d, "g"'] \\ W \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} 2^V \\ 2^W \arrow[u, "2^g"'] \end{tikzcd}
$$

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

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

- This construction is *contravariant* !!
- Note: if g is *surjective*, then 2^g is *injective*.

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 $S \subseteq W$,

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

- This construction is *contravariant* !!
- Note: if g is *surjective*, then 2^g is *injective*.

Reversing transitions

$$
\begin{tikzcd}
X \arrow[d, "t"'] \\
X^A
\end{tikzcd}
\;\Bigg\|\;
\begin{tikzcd}
X \times A \arrow[d] \\
X
\end{tikzcd}
\quad \overset{2^{(-)}}{\longmapsto} \quad
\begin{tikzcd}
2^{X \times A} \\
2^X \arrow[u]
\end{tikzcd}
\;\Bigg\|\;
\begin{tikzcd}
(2^X)^A \\
2^X \arrow[u]
\end{tikzcd}
\;\Bigg\|\;
\begin{tikzcd}
2^X \arrow[d, "2^t"] \\
(2^X)^A
\end{tikzcd}
$$

Reversing transitions

$$
\begin{array}{c|c}
X & X \times A \\
\downarrow t & \downarrow \\
X^A & X
\end{array}
\quad \overset{2^{(-)}}{\longmapsto} \quad
\begin{array}{c|c|c}
2^{X \times A} & (2^X)^A & 2^X \\
\uparrow & \uparrow & \downarrow 2^t \\
2^X & 2^X & (2^X)^A
\end{array}
$$

Reversing transitions

$$
\begin{tikzcd}
X \arrow[d, "t"'] \\
X^A
\end{tikzcd}
\quad\Big\|\quad
\begin{tikzcd}
X \times A \arrow[d] \\
X
\end{tikzcd}
\quad \xmapsto{\;2^{(-)}\;} \quad
\begin{tikzcd}
2^{X \times A} \\
2^X \arrow[u]
\end{tikzcd}
\quad\Big\|\quad
\begin{tikzcd}
(2^X)^A \\
2^X \arrow[u]
\end{tikzcd}
\quad\Big\|\quad
\begin{tikzcd}
2^X \arrow[d, "2^t"] \\
(2^X)^A
\end{tikzcd}
$$

Reversing transitions

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

Reversing transitions

$$
\begin{tikzcd}
X \arrow[d, "t"'] \\
X^A
\end{tikzcd}
\Bigg\|
\begin{tikzcd}
X \times A \arrow[d] \\
X
\end{tikzcd}
\quad \xmapsto{\;2^{(-)}\;} \quad
\begin{tikzcd}
2^{X \times A} \\
2^X \arrow[u]
\end{tikzcd}
\Bigg\|
\begin{tikzcd}
(2^X)^A \\
2^X \arrow[u]
\end{tikzcd}
\Bigg\|
\begin{tikzcd}
2^X \arrow[d, "2^t"] \\
(2^X)^A
\end{tikzcd}
$$

Initial $\Longleftrightarrow$ final

$$\begin{tikzcd} 1 \arrow[dr, "i"] & \\ & X \end{tikzcd} \qquad \overset{2^{(-)}}{\longmapsto} \qquad \begin{tikzcd} & 2 \\ 2^X \arrow[ur, "2^i"] & \end{tikzcd}$$

$$\begin{tikzcd} & 2 \\ X \arrow[ur, "f"] & \end{tikzcd} \qquad \longleftrightarrow \qquad \begin{tikzcd} 1 \arrow[dr, "f"] & \\ & 2^X \end{tikzcd}$$

Initial $\Longleftrightarrow$ final

$$1 \xrightarrow{i} X \qquad \overset{2^{(-)}}{\longmapsto} \qquad 2^X \xrightarrow{2^i} 2$$

$$X \xrightarrow{f} 2 \qquad \longleftrightarrow \qquad 1 \xrightarrow{f} 2^X$$

Initial $\Longleftrightarrow$ final

$$1 \xrightarrow{i} X \quad \xmapsto{2^{(-)}} \quad 2^X \xrightarrow{2^i} 2$$

$$X \xrightarrow{f} 2 \quad \longleftrightarrow \quad 1 \xrightarrow{f} 2^X$$

Initial $\Longleftrightarrow$ final

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & & & 2 \\
& X & \overset{2^{(-)}}{\longmapsto} & 2^X \arrow[ur, "2^i"] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
& 2 & & 1 \arrow[dr, "f"] & \\
X \arrow[ur, "f"] & & \longleftrightarrow & & 2^X
\end{tikzcd}
$$

Reversing the entire automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
\qquad \xmapsto{2^{(-)}} \qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[d, "2^t"] & \\
& (2^X)^A &
\end{tikzcd}
$$

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

Reversing the entire automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
\qquad \overset{2^{(-)}}{\longmapsto} \qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[d, "2^t"] & \\
& (2^X)^A &
\end{tikzcd}
$$

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

Reversing the entire automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
\quad \xmapsto{2^{(-)}} \quad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[d, "2^t"] & \\
& (2^X)^A &
\end{tikzcd}
$$

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

Reversing the entire automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"] & & 2 \\
& X \arrow[ur, "f"] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
\qquad \xmapsto{2^{(-)}} \qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[d, "2^t"] & \\
& (2^X)^A &
\end{tikzcd}
$$

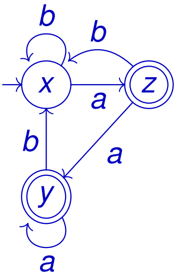
- Initial and final are exchanged . . .
- transitions are reversed . . .
- and the result is again deterministic!

Reversing the entire automaton

$$
\begin{tikzcd}
1 \arrow[dr, "i"'] & & 2 \\
& X \arrow[ur, "f"'] \arrow[d, "t"] & \\
& X^A &
\end{tikzcd}
\qquad \xmapsto{2^{(-)}} \qquad
\begin{tikzcd}
1 \arrow[dr, "f"'] & & 2 \\
& 2^X \arrow[ur, "2^i"'] \arrow[d, "2^t"] & \\
& (2^X)^A &
\end{tikzcd}
$$

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

Our previous example

$X =$ 

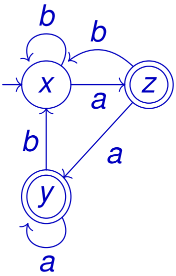
$2^X =$

- Note that X has been reversed and determinized:

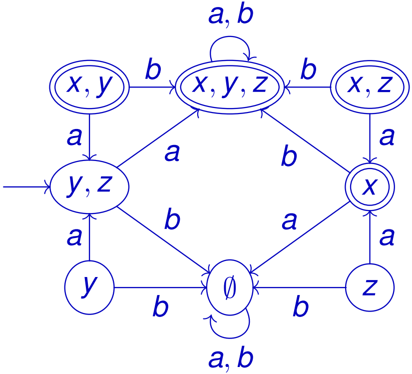
$$2^X = det(rev(X))$$

Our previous example

$X =$



$2^X =$

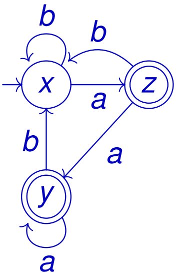


- Note that X has been reversed and determinized:

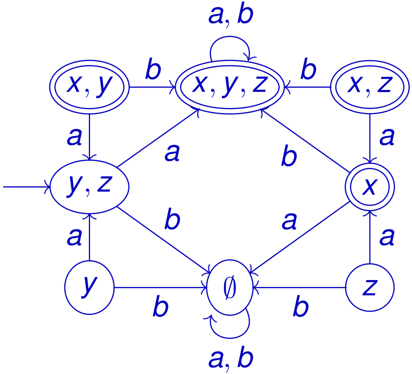
$$2^X = det(rev(X))$$

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 *minimal/observable* and

$$L(2^X) = reverse(L(X))$$

Proving today's Theorem

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

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

$$L(2^X) = reverse(L(X))$$

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

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\quad \xmapsto{2^{(-)}} \quad
\begin{tikzcd}
 & 2 \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[d, "2^\alpha"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\qquad \xmapsto{\;2^{(-)}\;} \qquad
\begin{tikzcd}
 & 2 \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[d, "2^\alpha"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\qquad \xmapsto{2^{(-)}} \qquad
\begin{tikzcd}
 & 2 \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[d, "2^\alpha"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\qquad \xmapsto{\;2^{(-)}\;} \qquad
\begin{tikzcd}
 & 2 \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[d, "2^\alpha"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

$$\begin{tikzcd} 1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\ A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\ (A^*)^A \arrow[r] & X^A \end{tikzcd} \quad \xmapsto{2^{(-)}} \quad \begin{tikzcd} & 2 \\ 2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[d, "2^\alpha"] \\ (2^X)^A \arrow[r] & (2^{A^*})^A \end{tikzcd}$$

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

Reachable becomes observable

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\quad \overset{2^{(-)}}{\longmapsto} \quad
\begin{tikzcd}
 & 2 & \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[r, dashed, "rev"'] \arrow[d, "2^\alpha"] & 2^{A^*} \arrow[ul, "\epsilon?"'] \arrow[d, "\beta"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

Reachable becomes observable

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\quad \overset{2^{(-)}}{\longmapsto} \quad
\begin{tikzcd}
 & 2 & \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[r, dashed, "rev"'] \arrow[d, "2^\alpha"] & 2^{A^*} \arrow[ul, "\epsilon?"'] \arrow[d, "\beta"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

Reachable becomes observable

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\quad \xmapsto{2^{(-)}} \quad
\begin{tikzcd}
& 2 & \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[r, dashed, "rev"'] \arrow[d, "2^\alpha"] & 2^{A^*} \arrow[ul, "\epsilon?"'] \arrow[d, "\beta"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

Reachable becomes observable

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & \\
A^* \arrow[r, "r"'] \arrow[d, "\alpha"'] & X \arrow[d, "t"] \\
(A^*)^A \arrow[r] & X^A
\end{tikzcd}
\quad \xmapsto{2^{(-)}} \quad
\begin{tikzcd}
& 2 & \\
2^X \arrow[ur, "2^i"] \arrow[r, "2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "2^\epsilon"'] \arrow[r, dashed, "rev"'] \arrow[d, "2^\alpha"] & 2^{A^*} \arrow[ul, "\epsilon?"'] \arrow[d, "\beta"] \\
(2^X)^A \arrow[r] & (2^{A^*})^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

Summarizing

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[d, "t"] & \\
(A^*)^A \arrow[r, dashed] & X^A &
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "rev \circ 2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
& (2^X)^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}$$

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

  then: $rev \circ 2^r$ is injective, i.e., 2^X is observable = minimal.
- And: $rev(2^r(f)) = rev(o(i))$, i.e., $L(\,2^X\,) \; = \; reverse(\,L(X)\,)$

Summarizing

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[d, "t"'] & \\
(A^*)^A \arrow[r, dashed] & X^A &
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "rev \circ 2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
& (2^X)^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

  then: $rev \circ 2^r$ is injective, i.e., 2^X is observable = minimal.
- And: $rev(2^r(f)) = rev(o(i))$, i.e., $L(\,2^X\,) \;=\; reverse(\,L(X)\,)$

Summarizing

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[d, "t"'] & \\
(A^*)^A \arrow[r, dashed] & X^A &
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "rev \circ 2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
& (2^X)^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

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

Summarizing

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"'] \arrow[d, "t"'] & \\
(A^*)^A \arrow[r, dashed] & X^A &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "rev \circ 2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
& (2^X)^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

  then: $rev \circ 2^r$ is injective, i.e., 2^X is observable = minimal.
- And: $rev(2^r(f)) = rev(o(i))$, i.e., $L(2^X) = reverse(L(X))$

Summarizing

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[d, "t"'] & \\
(A^*)^A \arrow[r, dashed] & X^A &
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[dr, "f"] & & 2 \\
& 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "rev \circ 2^r"'] \arrow[d, "2^t"'] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
& (2^X)^A \arrow[r, dashed] & (2^{A^*})^A
\end{tikzcd}
$$

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

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

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

$$reverse(reverse(L(X))) = L(X)$$

Corollary: Brzozowski's algorithm

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

$$reverse(reverse(L(X))) = L(X)$$

Corollary: Brzozowski's algorithm

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

$$reverse(reverse(L(X))) = L(X)$$

6. Conclusions

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[ur, "f"] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

6. Conclusions

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[r, dashed, "o"'] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
$$

Generalizations

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}$$

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & B \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & B^{A^*} \arrow[u] \arrow[d] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (B^{A^*})^A
\end{tikzcd}$$

- A Brzozowski minimization algorithm for *Moore* automata.
- Non-deterministic and weighted automata: under way (with Bonsangue et al).
- Probabilistic systems: under investigation.

Generalizations

$$\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & B \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & B^{A^*} \arrow[u] \arrow[d] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (B^{A^*})^A
\end{tikzcd}$$

- A Brzozowski minimization algorithm for *Moore* automata.
- Non-deterministic and weighted automata: under way (with Bonsangue et al).
- Probabilistic systems: under investigation.

Generalizations

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"'] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & B \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[d, "t"] \arrow[ur, "f"] & B^{A^*} \arrow[u] \arrow[d] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (B^{A^*})^A
\end{tikzcd}
$$

- A Brzozowski minimization algorithm for *Moore* automata.
- Non-deterministic and weighted automata: under way (with Bonsangue et al).
- Probabilistic systems: under investigation.

Generalizations

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[ur, "f"] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & B \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[r, dashed, "o"'] \arrow[ur, "f"] \arrow[d, "t"] & B^{A^*} \arrow[u] \arrow[d] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (B^{A^*})^A
\end{tikzcd}
$$

- A Brzozowski minimization algorithm for *Moore* automata.
- Non-deterministic and weighted automata: under way (with Bonsangue et al).
- Probabilistic systems: under investigation.

Generalizations

$$
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[r, dashed, "o"'] \arrow[d, "t"] & 2^{A^*} \arrow[u, "\epsilon?"'] \arrow[d, "\beta"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "\epsilon"'] \arrow[dr, "i"] & & B \\
A^* \arrow[r, dashed, "r"'] \arrow[d, "\alpha"'] & X \arrow[ur, "f"] \arrow[r, dashed, "o"'] \arrow[d, "t"] & B^{A^*} \arrow[u] \arrow[d] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (B^{A^*})^A
\end{tikzcd}
$$

- A Brzozowski minimization algorithm for *Moore* automata.
- Non-deterministic and weighted automata: under way (with Bonsangue et al).
- Probabilistic systems: under investigation.