Monads, comonads, and Mealy machines

Structure and Power workshop 2022

Rafał Stefański, UCL

Regular languages, automata, and monads

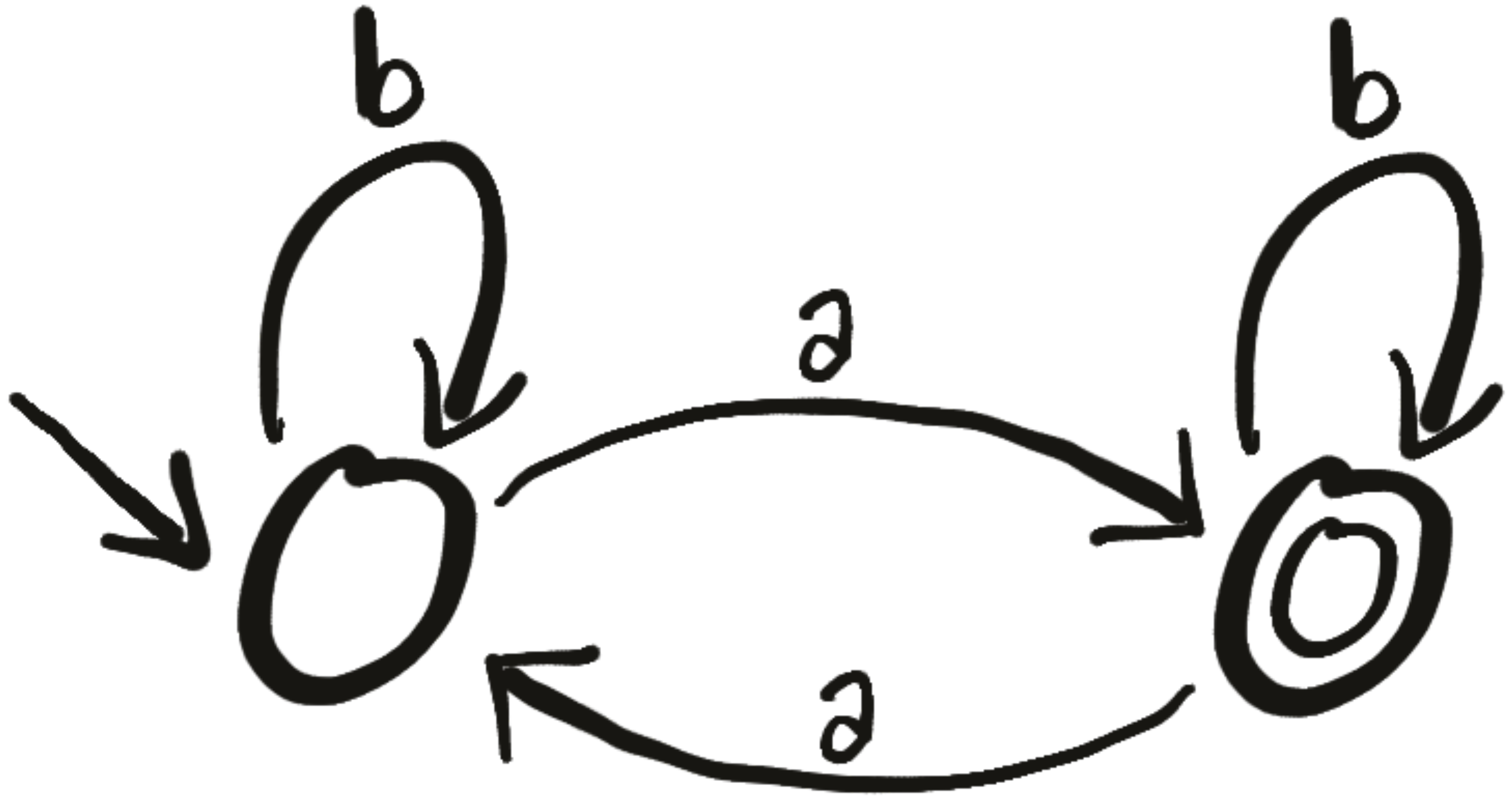
Regular languages, automata, and monads

M. Bojańczyk. *Recognisable languages over monads.*

M. Bojańczyk, B. Klin, J. Salamanca. *Monadic Monadic Second Order Logic.*

M. Bojańczyk. *Languages recognised by finite semigroups, and their generalisations to objects such as trees and graphs, with an emphasis on definability in monadic second-order logic.*

Deterministic finite automata



$$L \subseteq \{a, b\}^*$$

The input word contains an odd number of a's

Monoids

$$(A,\ 1,\ \cdot\ )$$

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$1 \cdot a = a = a \cdot 1$$

Monoids

$$(\mathbb{Z}_2,\ 0,\ +)$$

$$(\{a, b\}^*,\ \epsilon,\ \cdot\ )$$

$$(\mathbb{N},\ 0,\ \mathrm{max})$$

Finite monoids and regular languages

$A \qquad h : \Sigma \to A \qquad f : A \to \{\text{Yes, No}\}$

Finite monoids and regular languages

$A \qquad h : \Sigma \to A \qquad f : A \to \{\text{Yes, No}\}$

$$\Sigma^* \longrightarrow A^* \longrightarrow A \longrightarrow \{Yes, No\}$$

Finite monoids and regular languages

$M \qquad h : \Sigma \to M \qquad f : M \to \{\text{Yes, No}\}$

$$\underbrace{\Sigma^* \longrightarrow A^* \longrightarrow A}_{h^* : \Sigma^* \to A} \longrightarrow \{Yes, No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

a a b b a b

Finite monoids and regular languages

The input word contains an odd number of a's

a a b b a b

$$\Sigma = \{a, b\}$$
$$A = \mathbb{Z}_2$$

Finite monoids and regular languages

The input word contains an odd number of a's

a a b b a b

$$\Sigma = \{a, b\}$$
$$A = \mathbb{Z}_2$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes, No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

a a b b a b

$$h(a) = 1$$
$$h(b) = 0$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

1 a b b a b

$$h(a) = 1$$
$$h(b) = 0$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

1 1 b b a b

$$h(a) = 1$$
$$h(b) = 0$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

1 1 0 b a b

$$h(a) = 1$$
$$h(b) = 0$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

1 1 0 0 1 0

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

1

$$f(0) = No$$
$$f(1) = Yes$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes,\ No\}$$

Finite monoids and regular languages

The input word contains an odd number of a's

Yes

$$f(0) = No$$
$$f(1) = Yes$$

$$\{a, b\}^* \longrightarrow \mathbb{Z}_2^* \longrightarrow \mathbb{Z}_2 \longrightarrow \{Yes, No\}$$

Monoids: alternative defintion

$$(A, \mathtt{prod} : A^* \to A)$$

$$\mathtt{prod}([x]) = x$$

Monoids: alternative defintion

$$(A, \texttt{prod} : A^* \to A)$$

$$\begin{tikzcd}
(A^*)^* \arrow[r, "\texttt{flatten}"] \arrow[d, "\texttt{map prod}"'] & A^* \arrow[d, "\texttt{prod}"] \\
A^* \arrow[r, "\texttt{prod}"] & A
\end{tikzcd}$$

Monads

$$(M,\ \eta_X : X \to MX,\ \mu_X : MMX \to MX)$$

Monads

$$(M,\ \eta_X : X \to MX,\ \mu_X : MMX \to MX)$$

$$\begin{tikzcd}
MMMX \arrow[r, "\mu_{MX}"] \arrow[d, "M\mu_X"'] & MMX \arrow[d, "\mu_X"] \\
MMX \arrow[r, "\mu_X"] & MX
\end{tikzcd}$$

Monads

$$(M,\ \eta_X : X \to MX,\ \mu_X : MMX \to MX)$$

$$\begin{tikzcd}
MX \arrow[r, "\eta_{MX}"] \arrow[d, "M\mu_{\eta X}"'] \arrow[dr, "\mathrm{id}"] & MMX \arrow[d, "\mu_X"] \\
MMX \arrow[r, "\mu_X"'] & MX
\end{tikzcd}$$

Finite lists

Finite lists

$$\eta(x) = [a]$$

Finite lists

$$\eta(x) = [a]$$

$$\mu = \texttt{flatten}$$

Labelled orders

Labelled orders

$$\eta(a) = [a]$$

$$\mu = \texttt{flatten}$$

Labelled orders

- Contains all ω-words

Labelled orders

- Contains all ω-words
- Submonads:

Finite lists

Labelled orders

- Contains all ω-words
- Submonads:
  - Finite orders e.g. lists

Finite lists

Labelled orders

- Contains all ω-words
- Submonads:
  - Finite orders e.g. lists
  - Countable orders

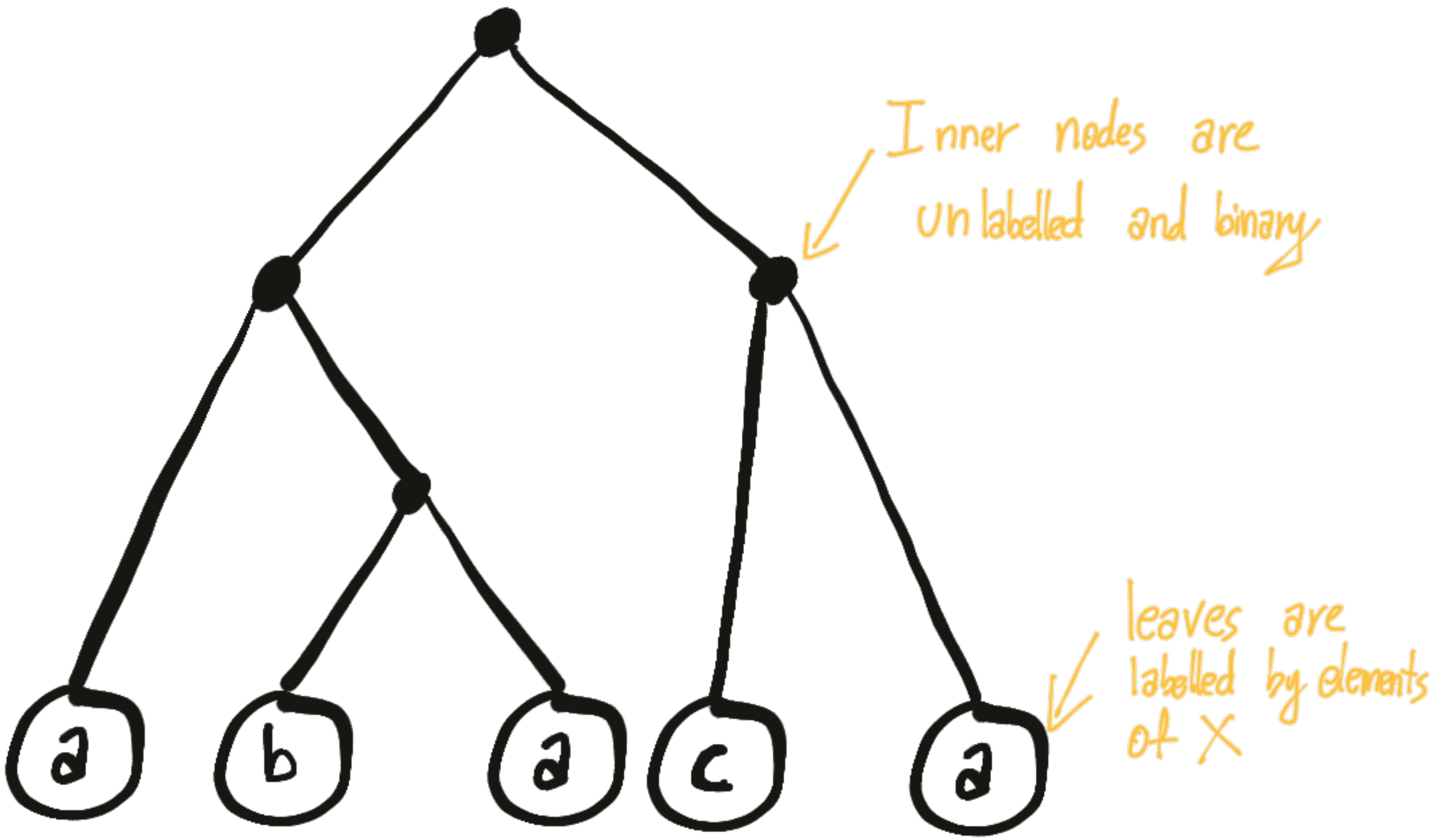
Labelled orders

- Contains all ω-words
- Submonads:
  - Finite orders e.g. lists
  - Countable orders
  - Well-founded orders

Labelled orders

- Contains all ω-words
- Submonads:
  - Finite orders e.g. lists
  - Countable orders
  - Well-founded orders
  - ...

Terms



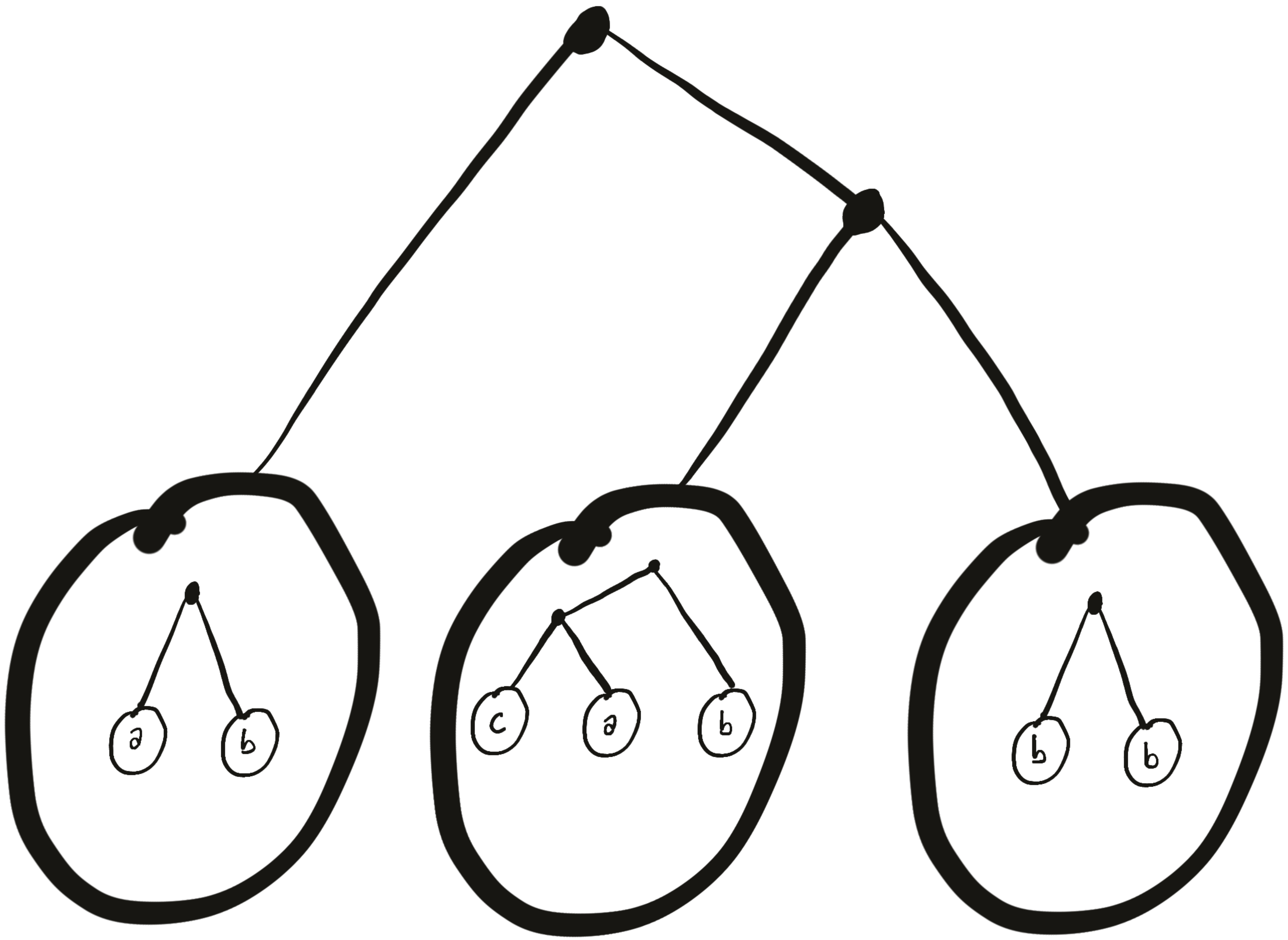
Finite lists **Labelled orders**

Terms

$$\partial \xmapsto{\eta} \partial$$

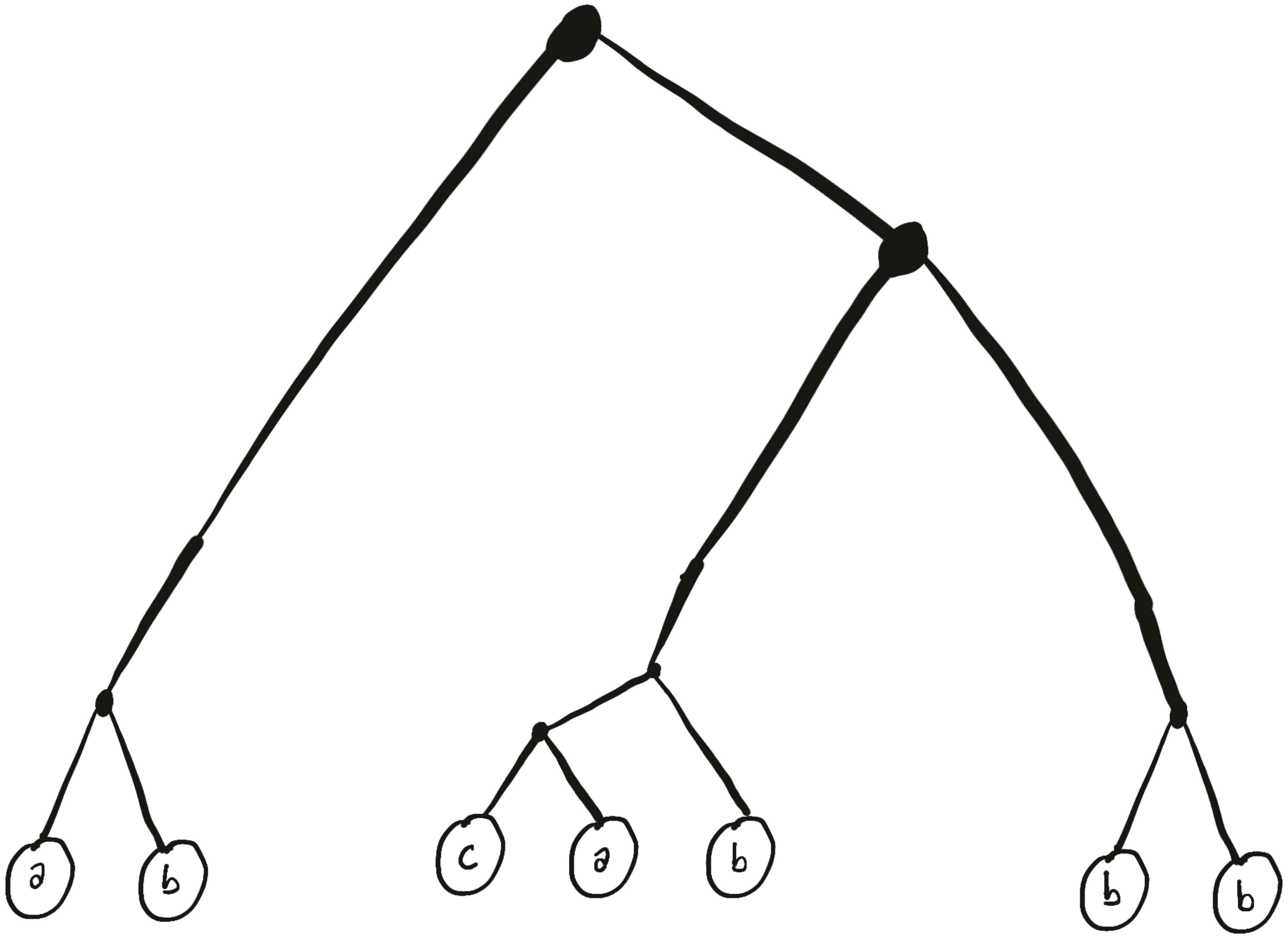
Finite lists   Labelled orders

Terms



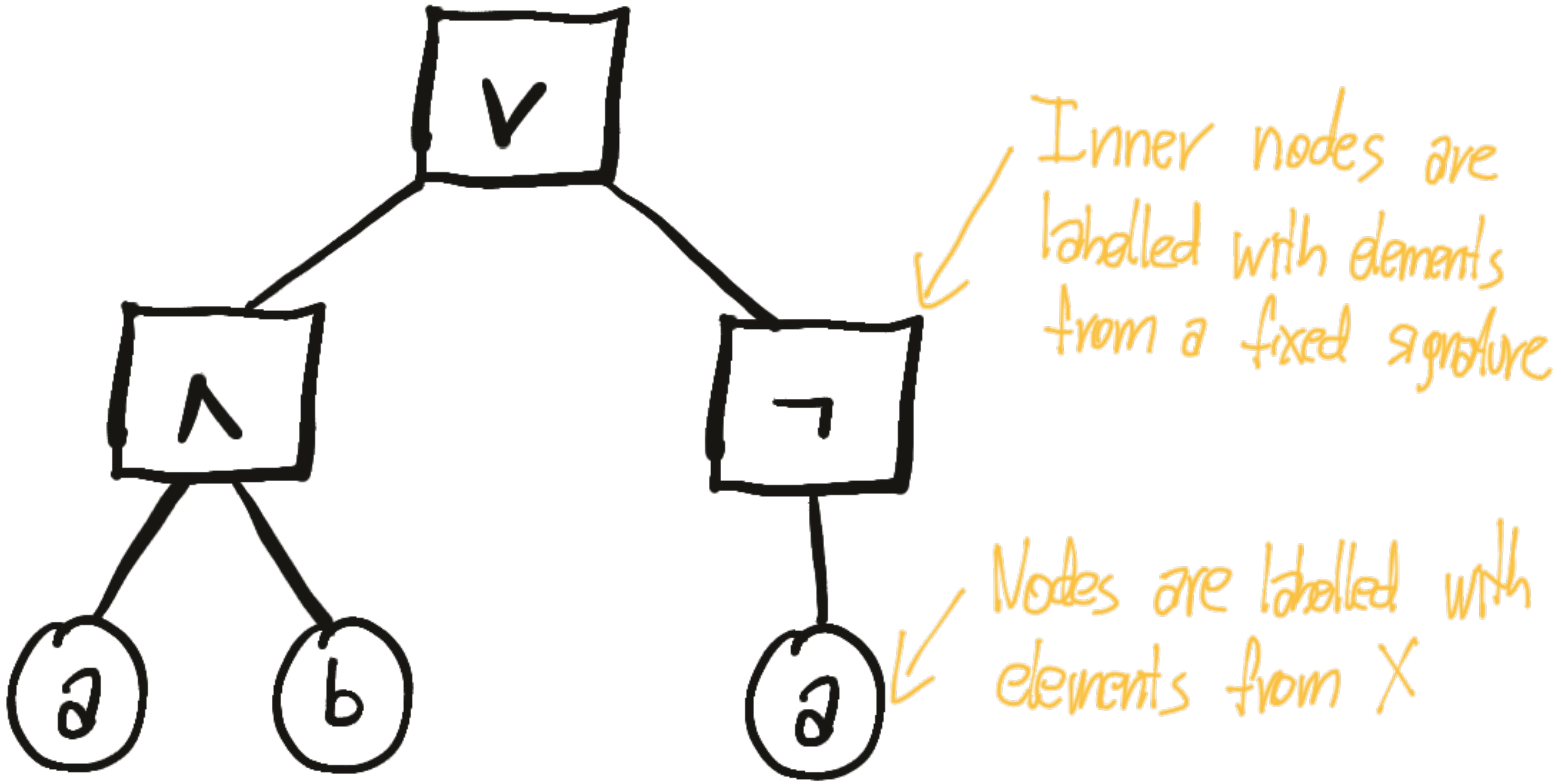
Finite lists **Labelled orders**

Terms



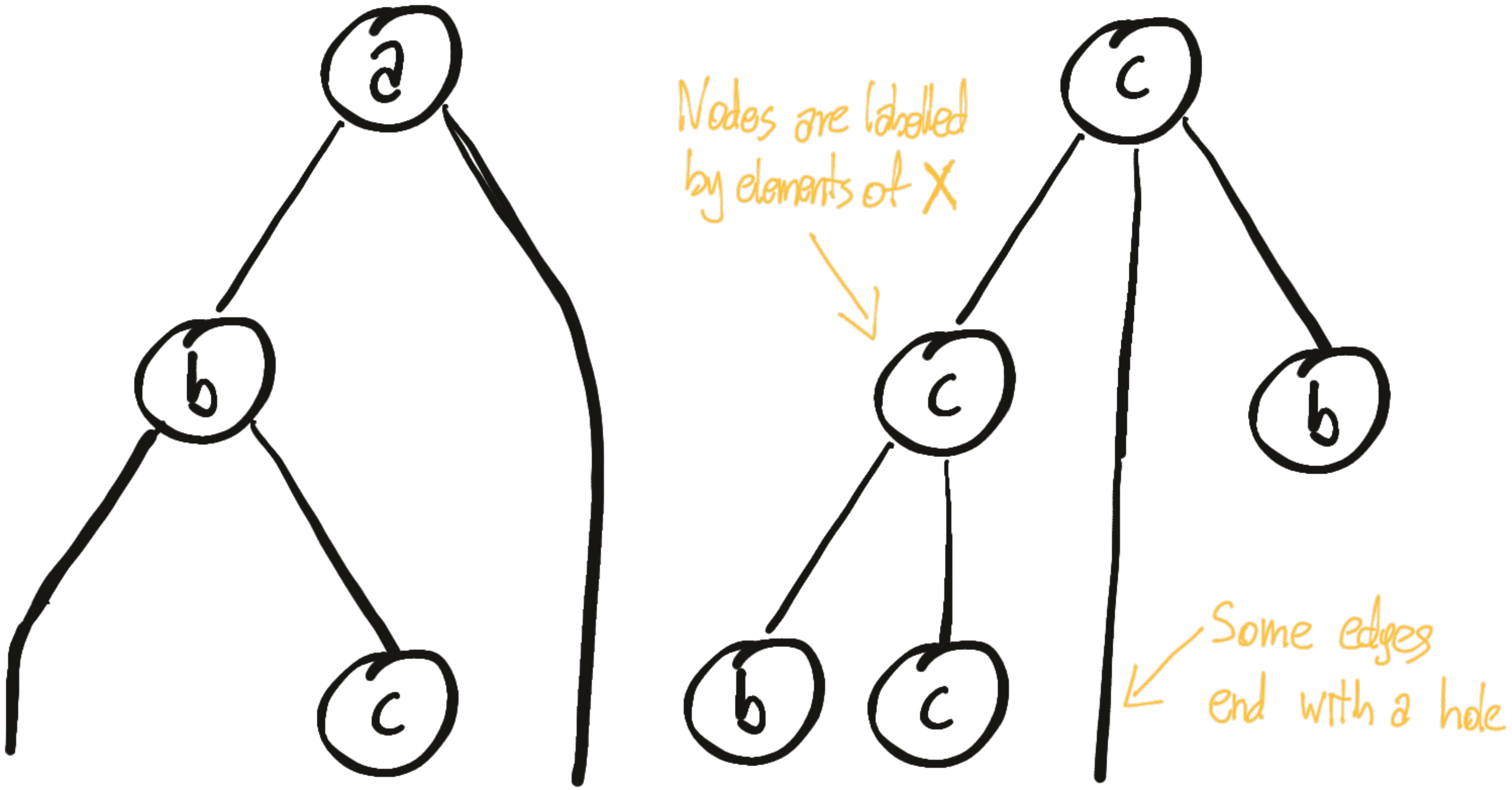
Finite lists   Labelled orders

Terms



Finite lists   Labelled orders

Forests with ports

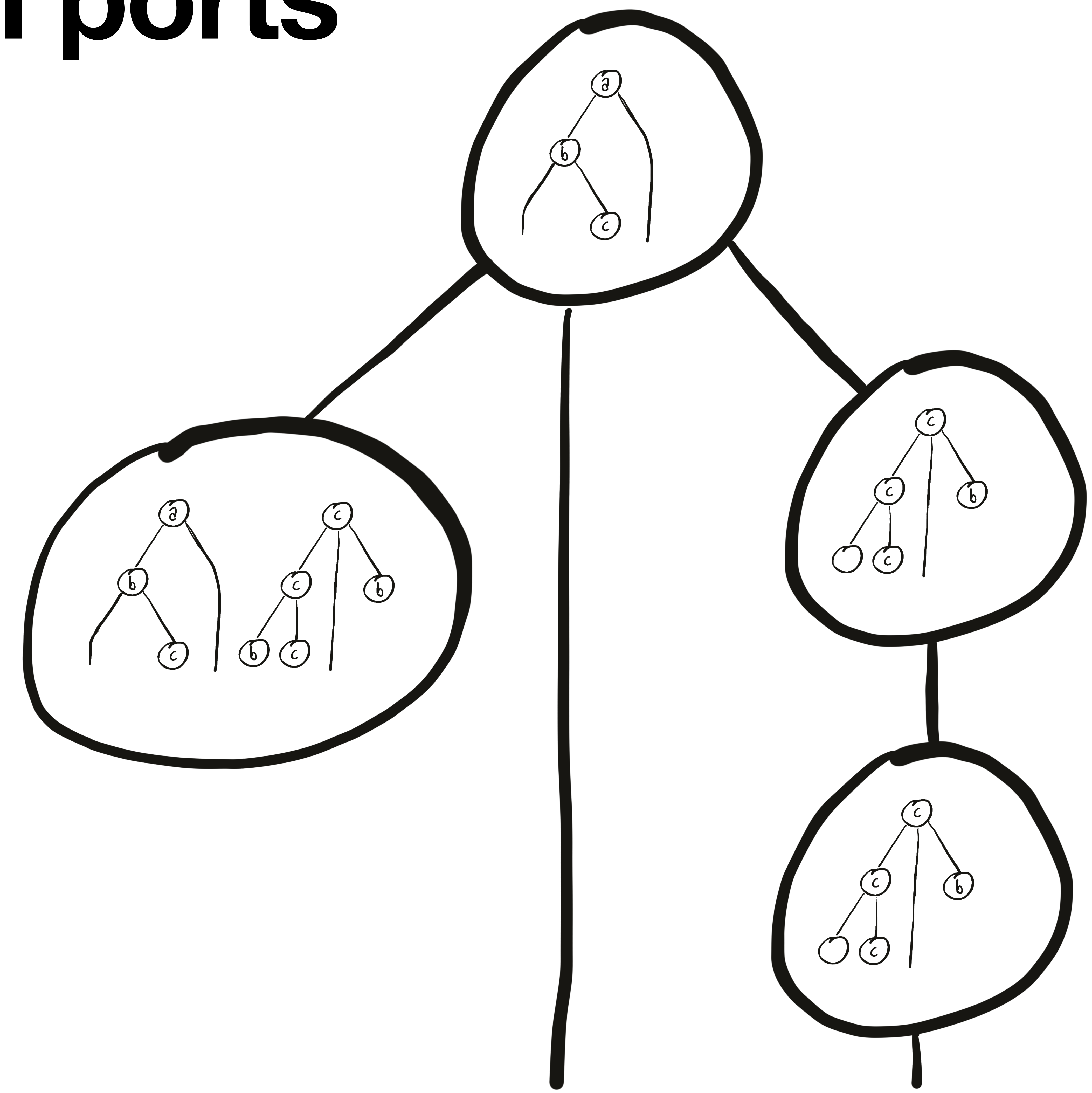


Finite lists   Labelled orders   Terms

Forests with ports

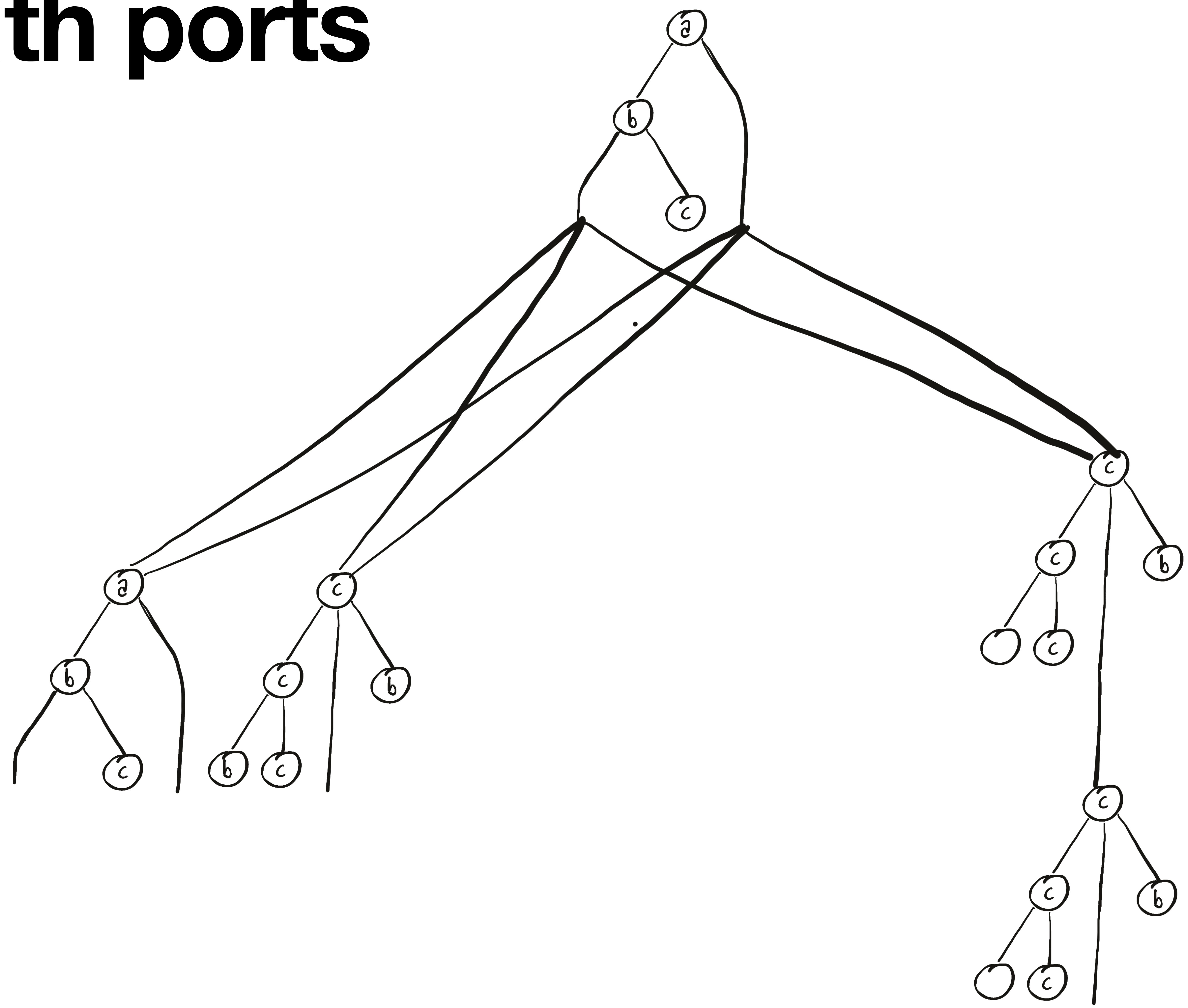
$$\partial \xmapsto{\eta} \partial$$

Forests with ports



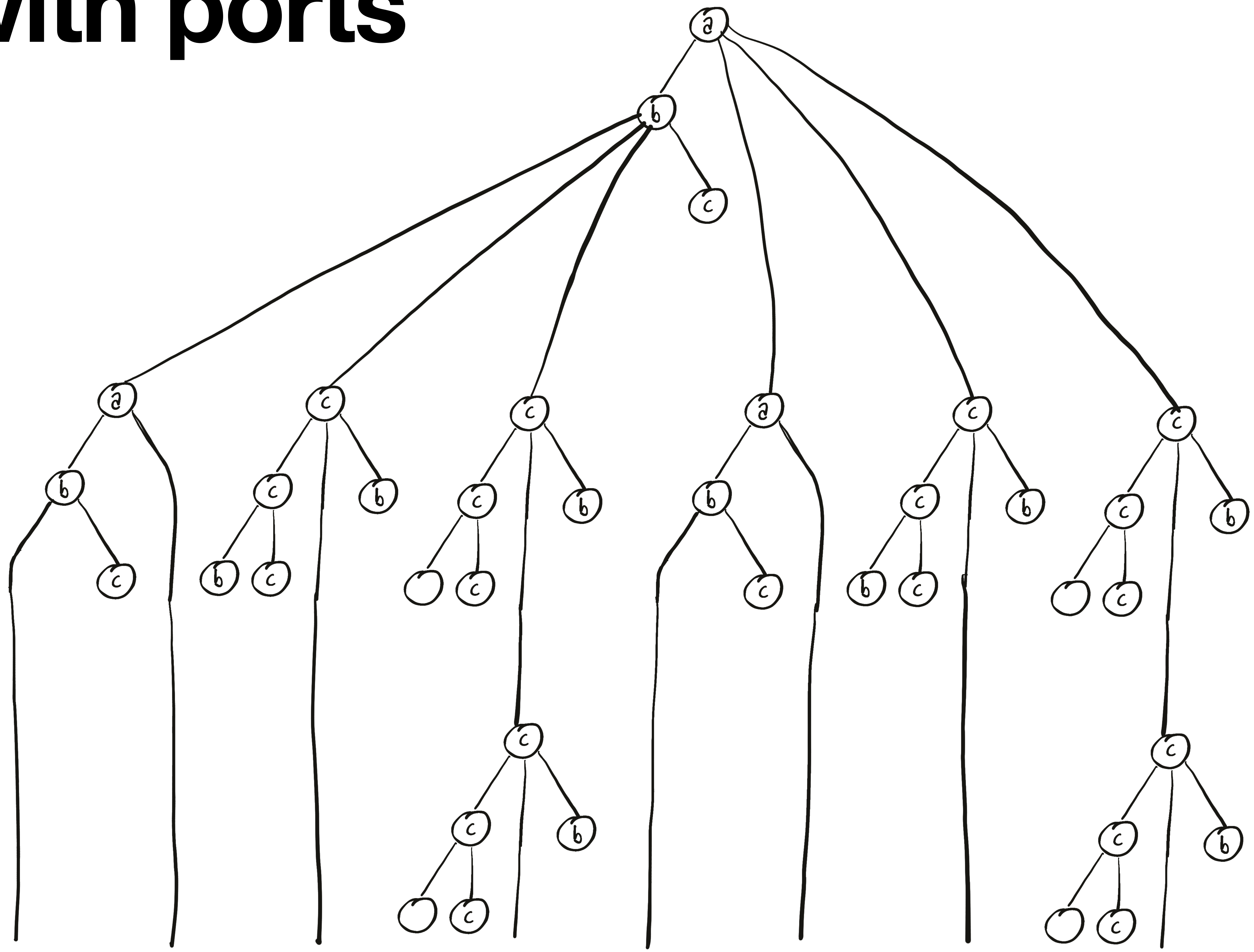
Finite lists Labelled orders Terms

Forests with ports



Finite lists   Labelled orders   Terms

Forests with ports



Finite lists   Labelled orders   Terms

Finite lists **Labelled orders** **Terms** **Forests with ports**

Monoids: alternative defintion

$$(A, \texttt{prod} : A^* \to A)$$

Eilenberg-Moore algebras

$$(A, \mathtt{prod} : MA \to A)$$

Eilenberg-Moore algebras

$$(A, \texttt{prod} : MA \to A)$$

$$\begin{tikzcd}
MMA \arrow[r, "\mu"] \arrow[d, "M\texttt{prod}"'] & MA \arrow[d, "\texttt{prod}"] \\
MA \arrow[r, "\texttt{prod}"] & A
\end{tikzcd}$$

Eilenberg-Moore algebras

$$(A, \texttt{prod} : MA \to A)$$

$$\begin{tikzcd}
A \arrow[r, "\eta"] \arrow[dr, "\mathrm{id}"'] & MA \arrow[d, "\texttt{prod}"] \\
& A
\end{tikzcd}$$

Recognisable languages over a monad

A $\quad h : \Sigma \to A$ $\quad f : A \to \{\text{Yes, No}\}$

$$M\Sigma \xrightarrow{Mh} MA \xrightarrow{\texttt{prod}} A \xrightarrow{f} \{Yes, No\}$$

$$L \subseteq M\Sigma$$

Recognisable languages over a monad

$$A \qquad h : \Sigma \to A \qquad f : A \to \{\text{Yes, No}\}$$

$$\underbrace{M\Sigma \xrightarrow{Mh} MA \xrightarrow{\texttt{prod}} A}_{h^* : M\Sigma \to A} \xrightarrow{f} \{Yes, No\}$$

$$L \subseteq M\Sigma$$

Finite lists **Labelled orders** **Terms** **Forests with ports**

Finite lists   **Labelled orders**   **Terms**   **Forests with ports**

Regular languages

Finite lists **Labelled orders** **Terms** **Forests with ports**

Regular languages

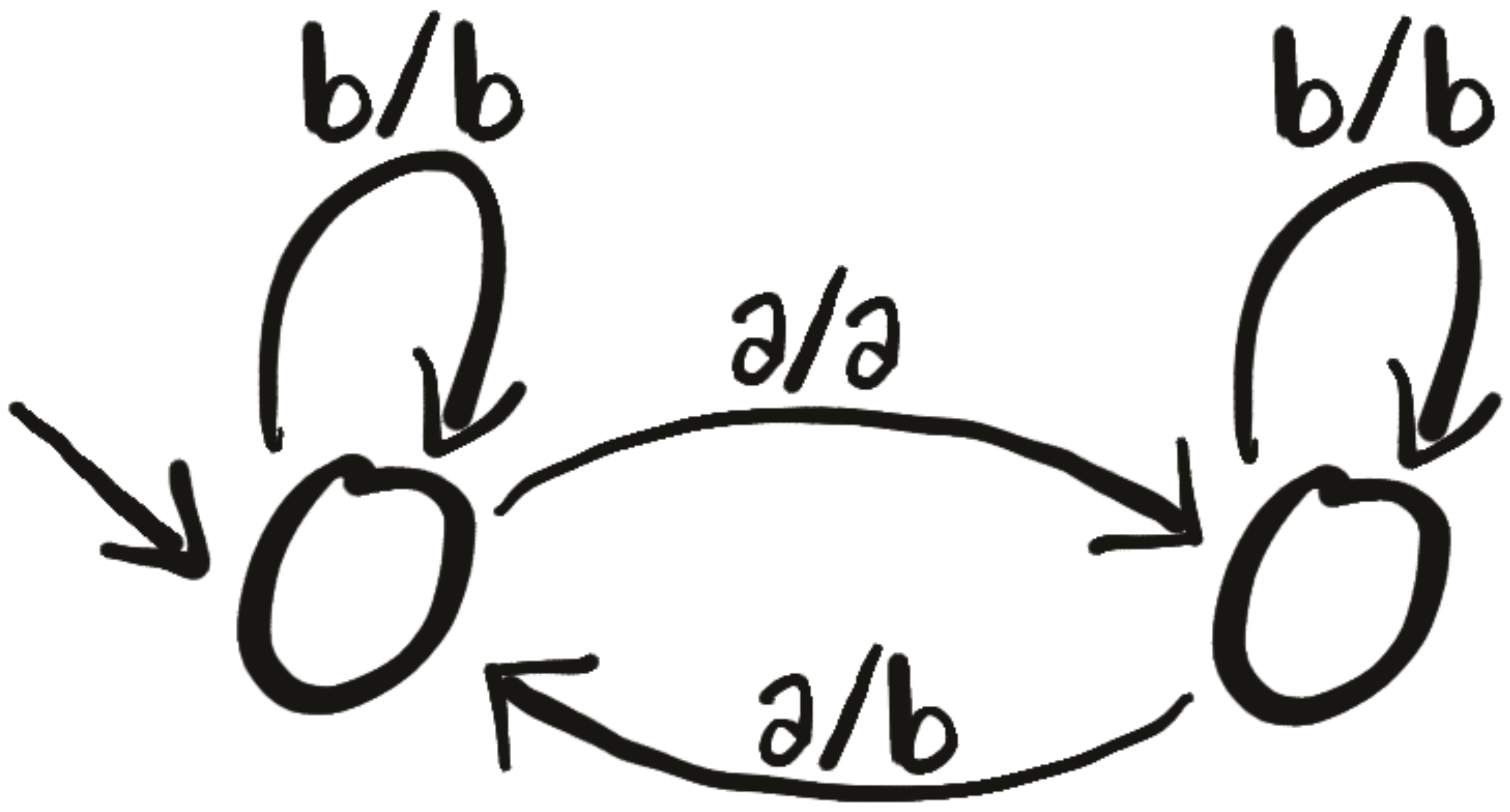
On ω-words:

ω-regular languages

Finite lists	Labelled orders	Terms	Forests with ports
Regular languages	On ω-words: ω-regular languages		On trees: Regular tree languages

Mealy machines, monads, and comonads

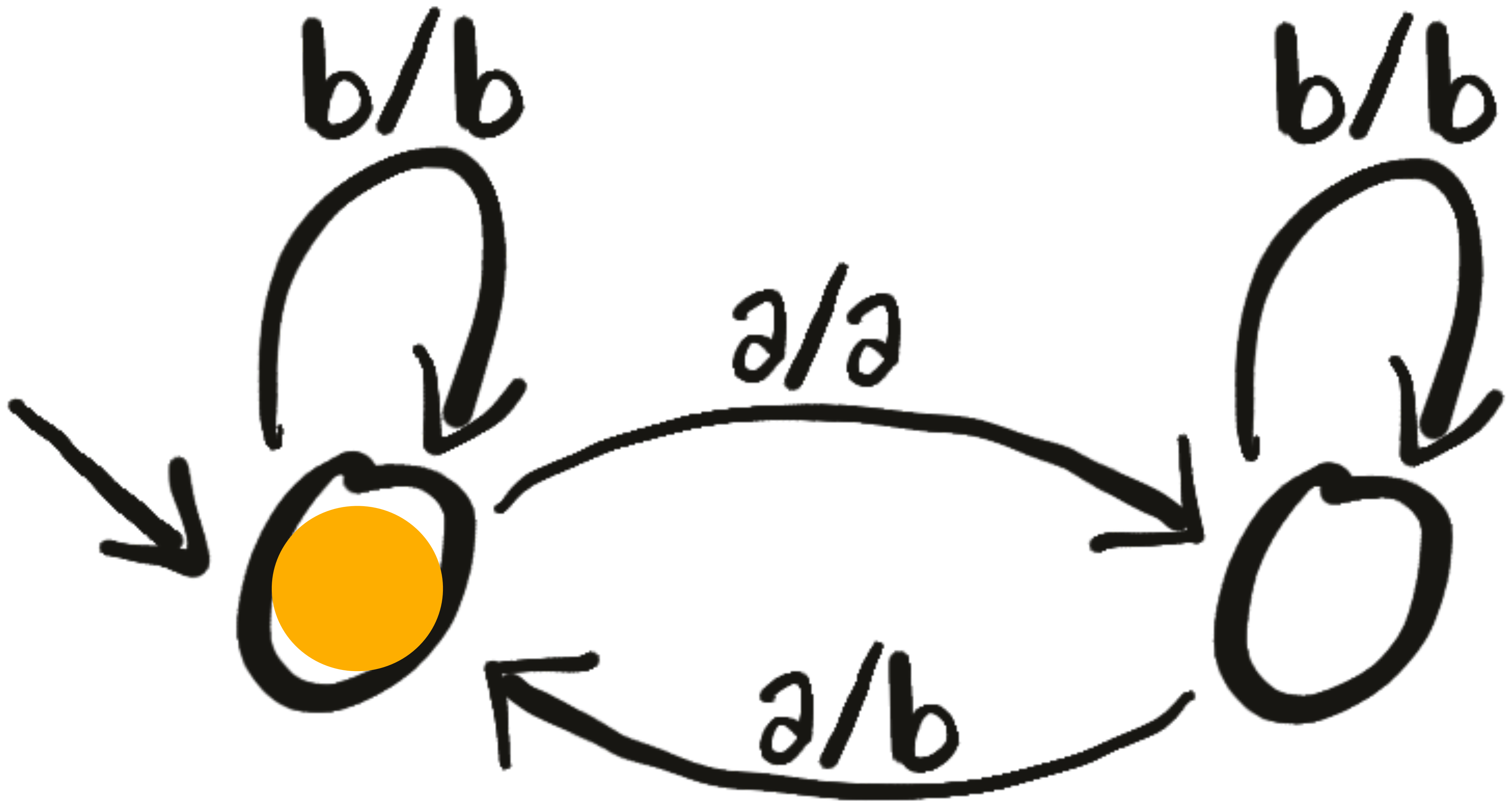
Mealy machine



$$t \in \{a, b\}^* \to \{a, b\}^*$$

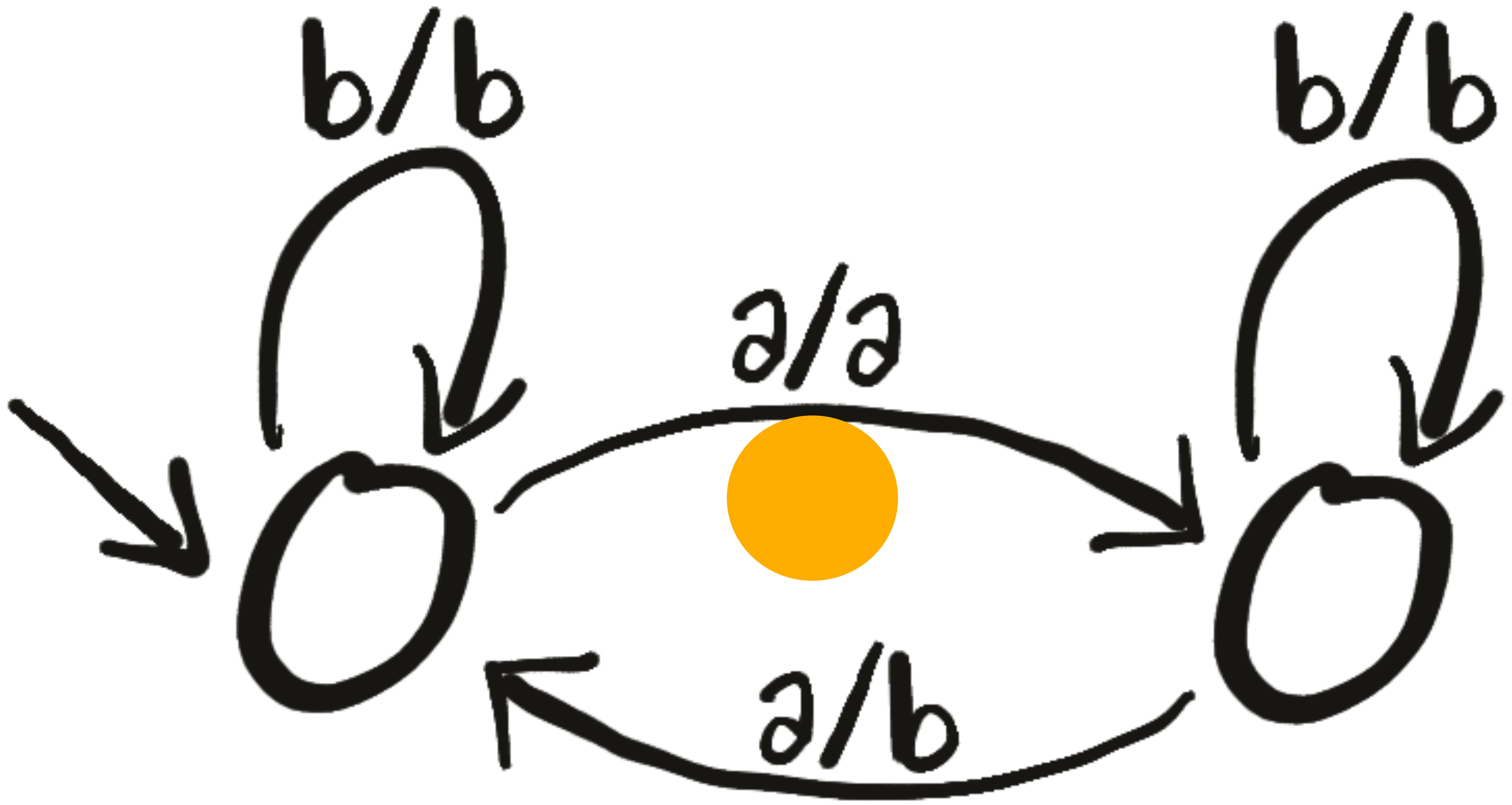
Replace every other a with b

Mealy machine



a a b b a a

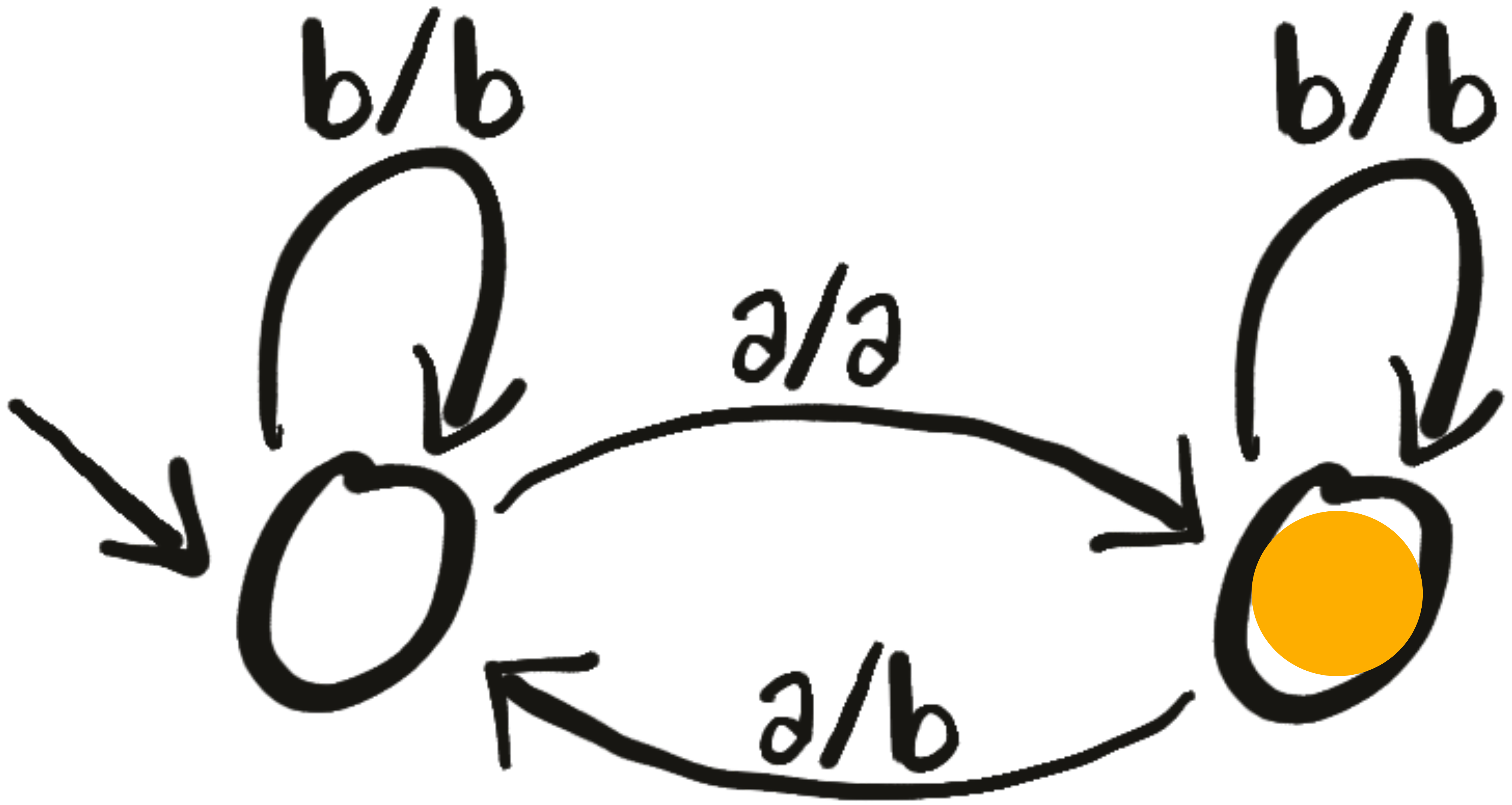
Mealy machine



a a b b a a

a

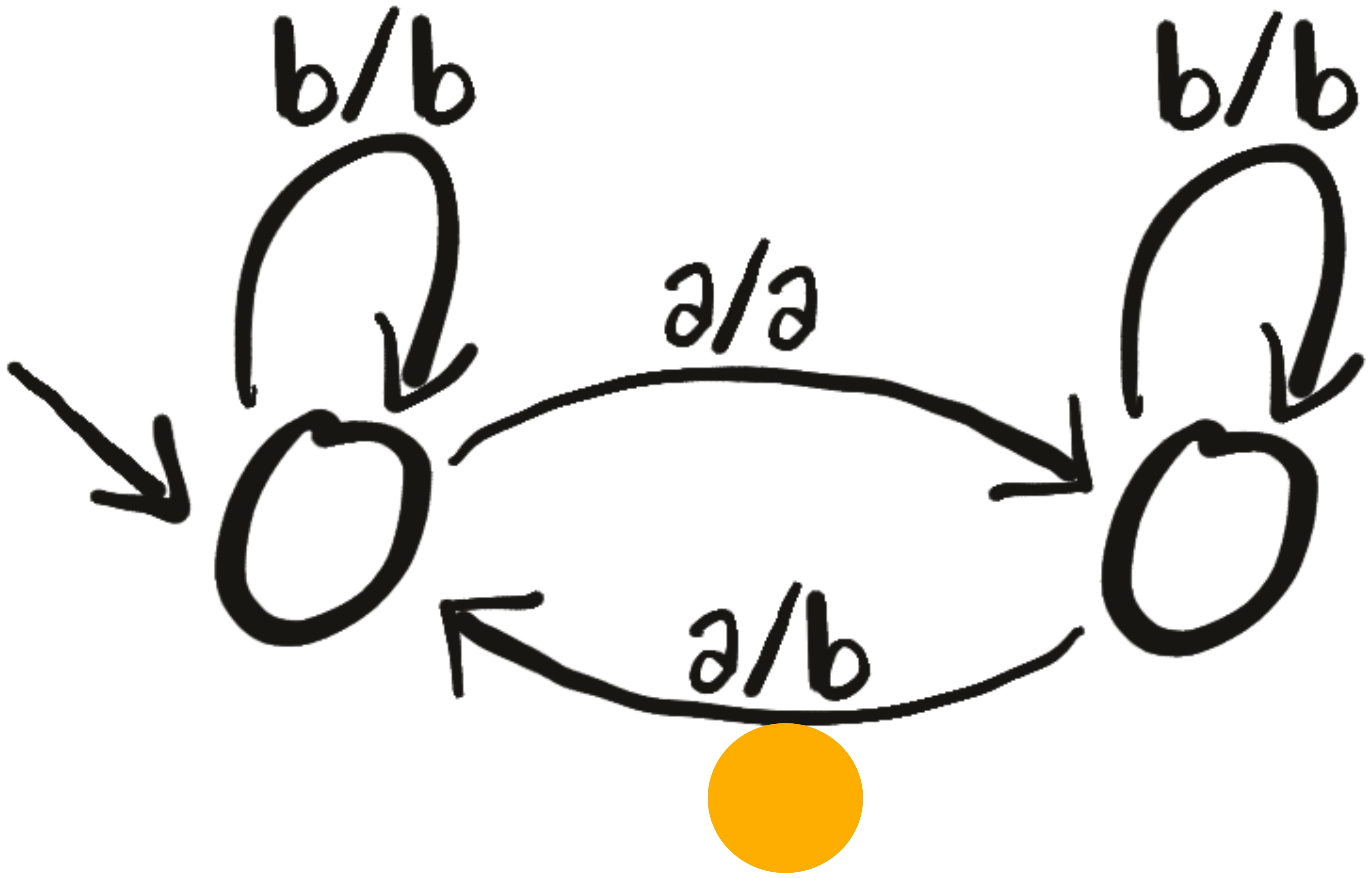
Mealy machine



a a b b a a

a

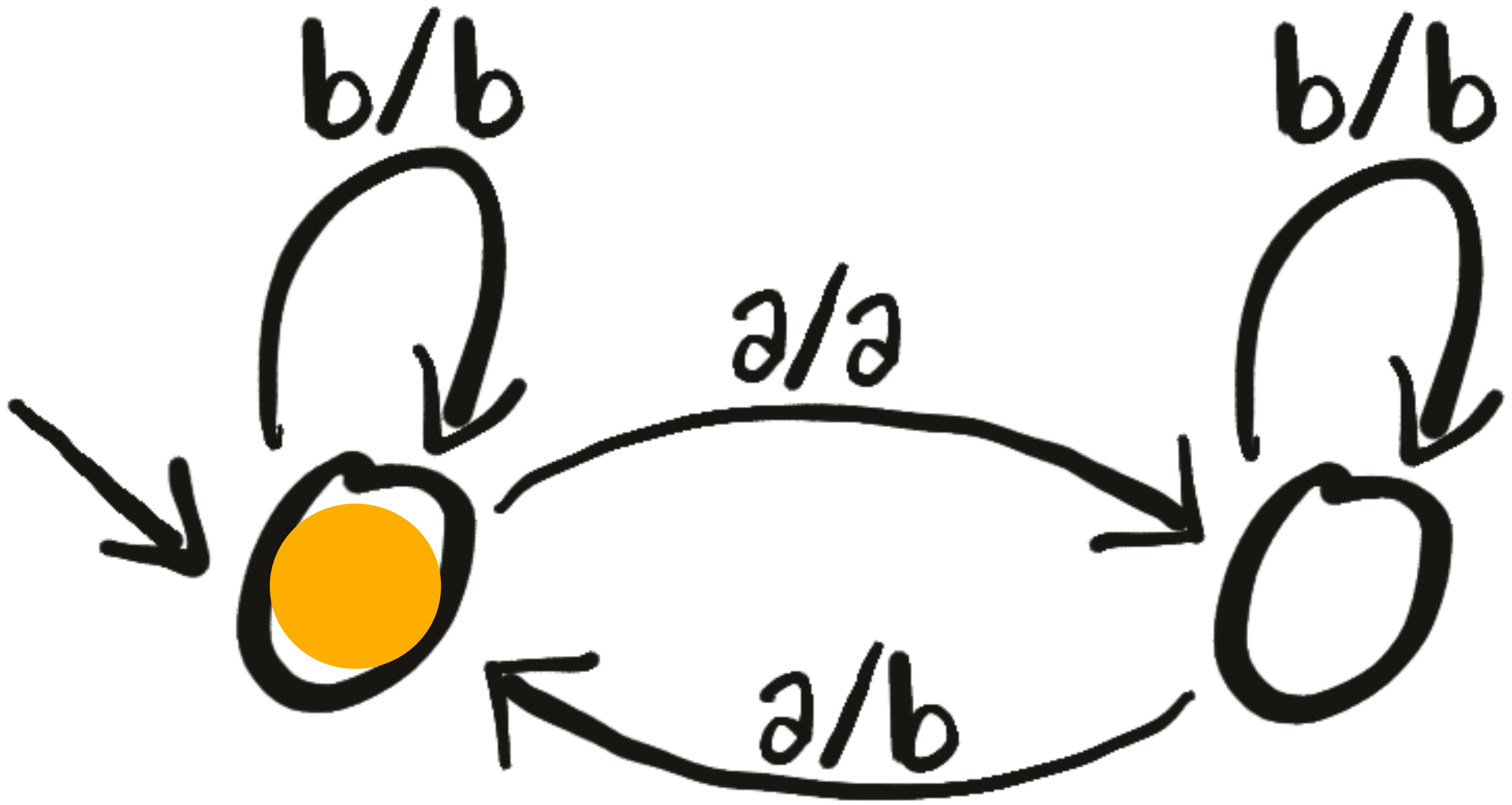
Mealy machine



a a b b a a

a b

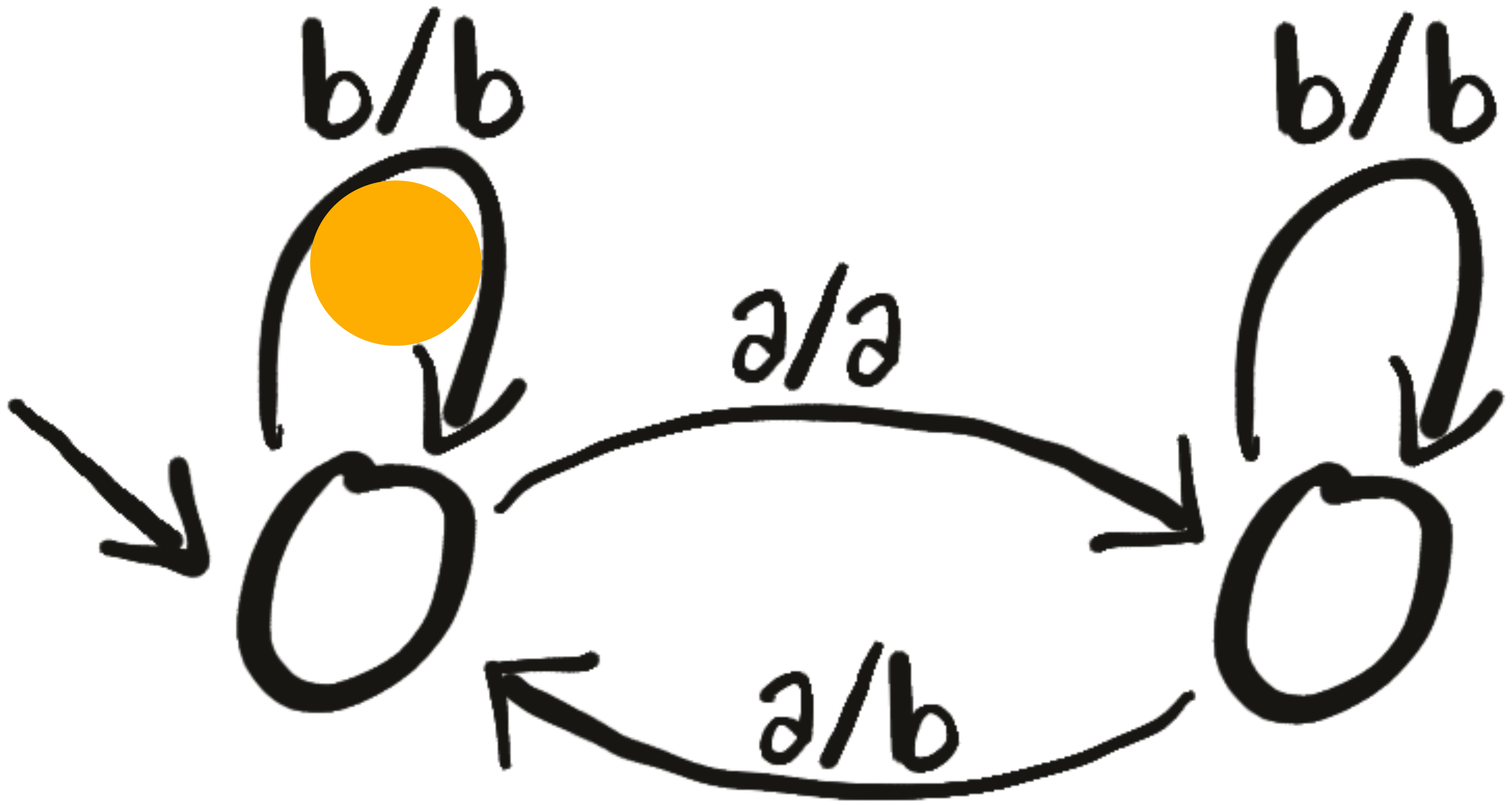
Mealy machine



a a b b a a

a b

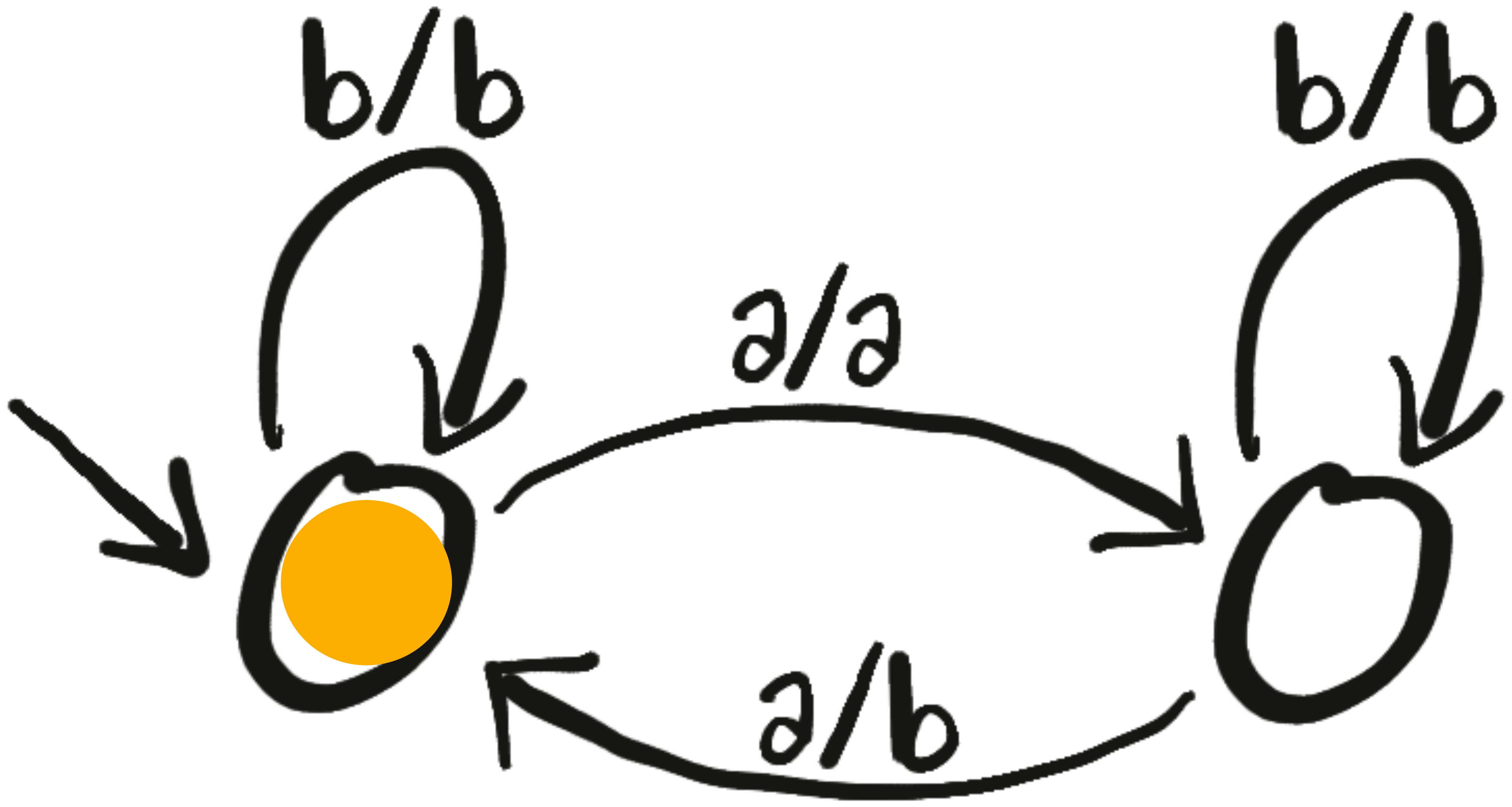
Mealy machine



a a b b a a

a b b

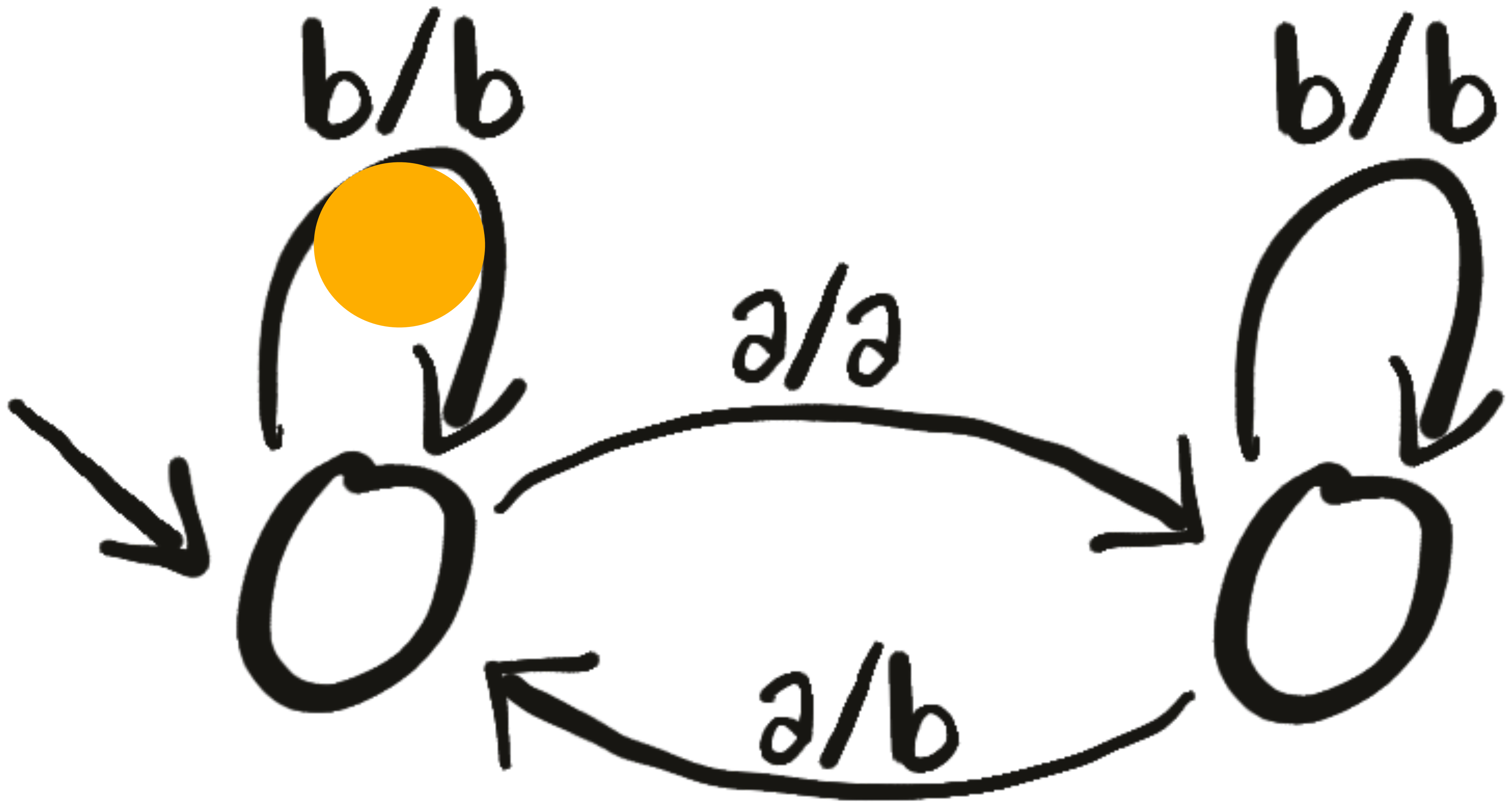
Mealy machine



a a b b a a

a b b

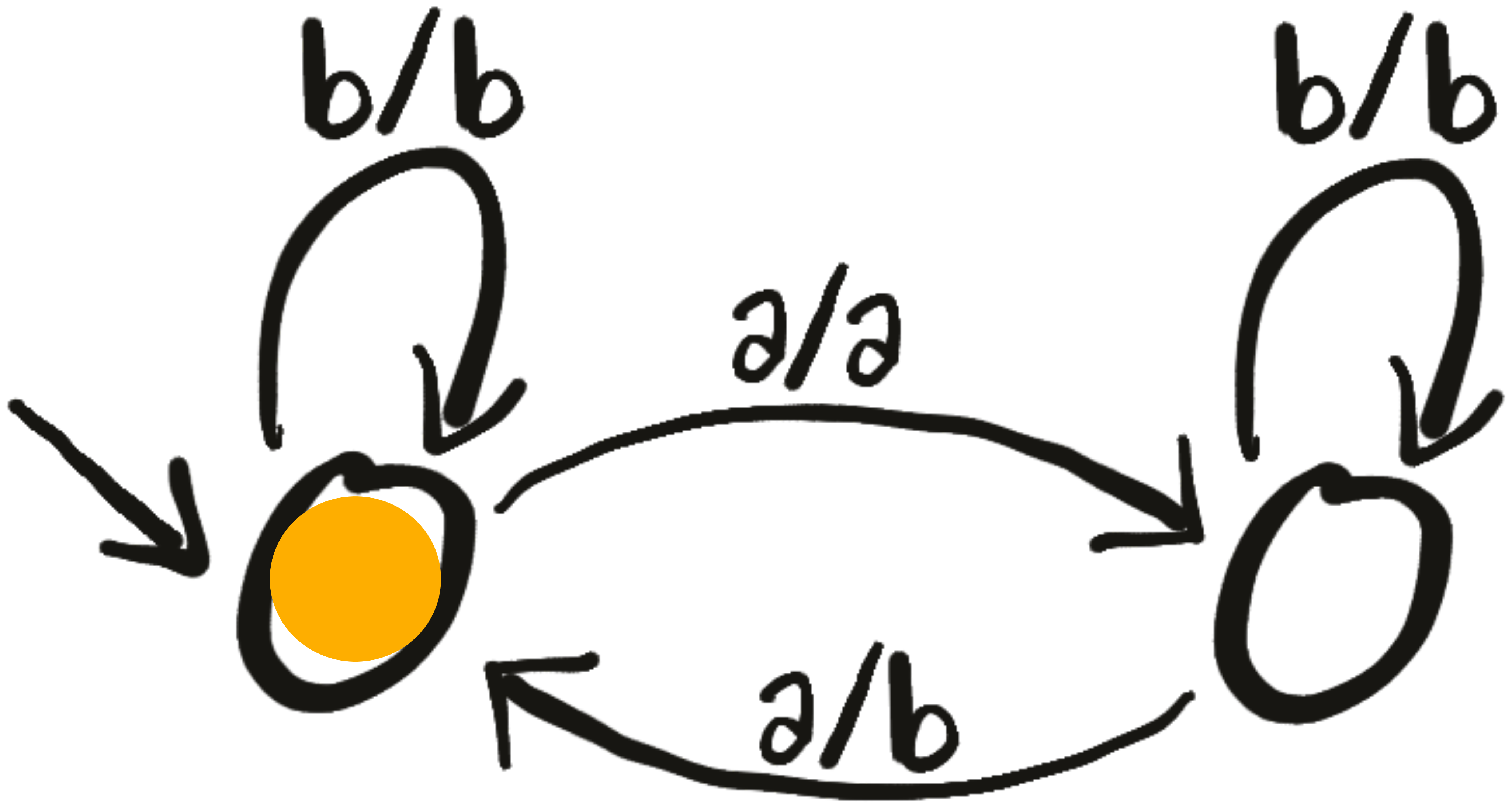
Mealy machine



a a b b a a

a b b b

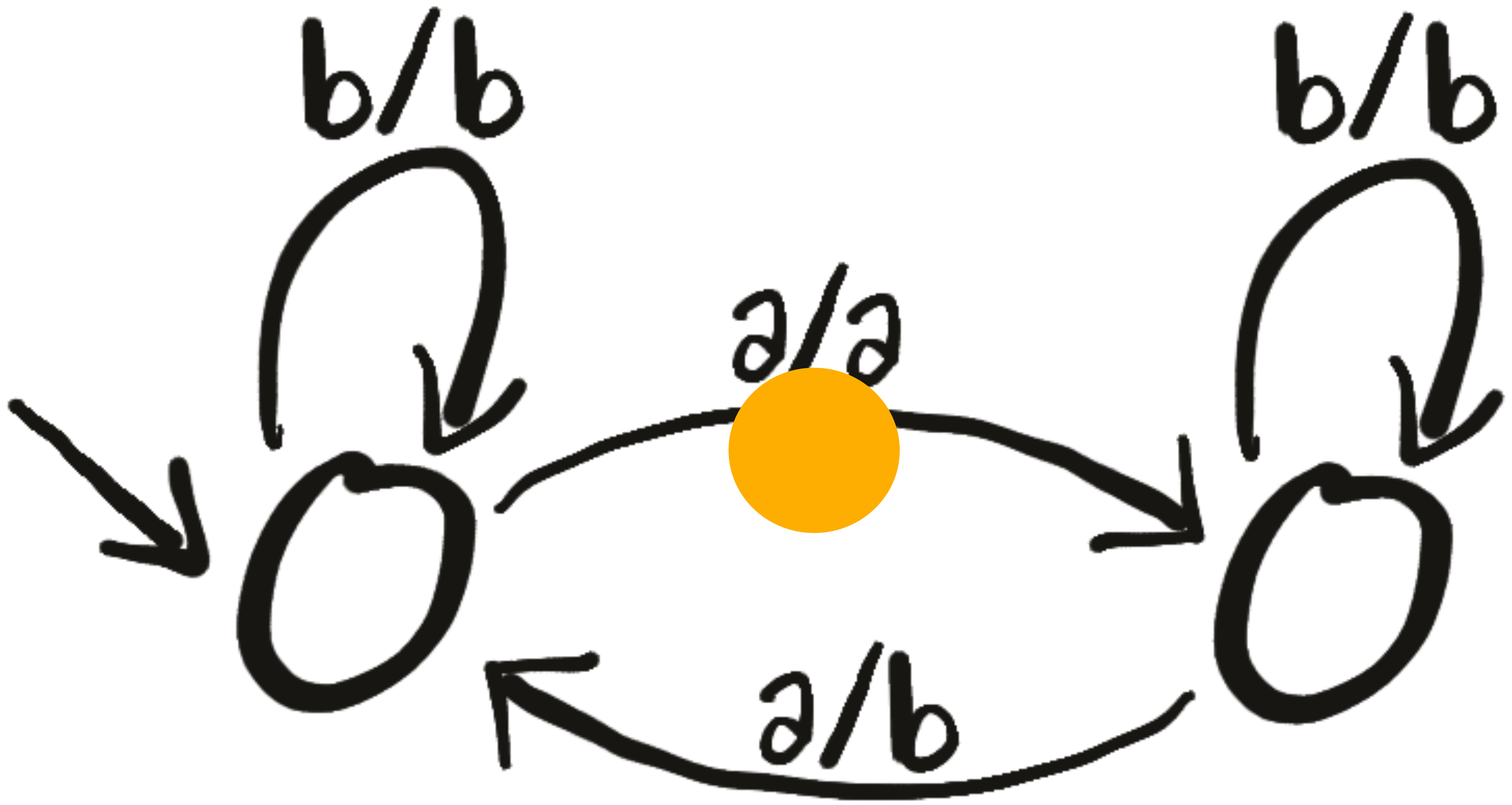
Mealy machine



a a b b a a

a b b b

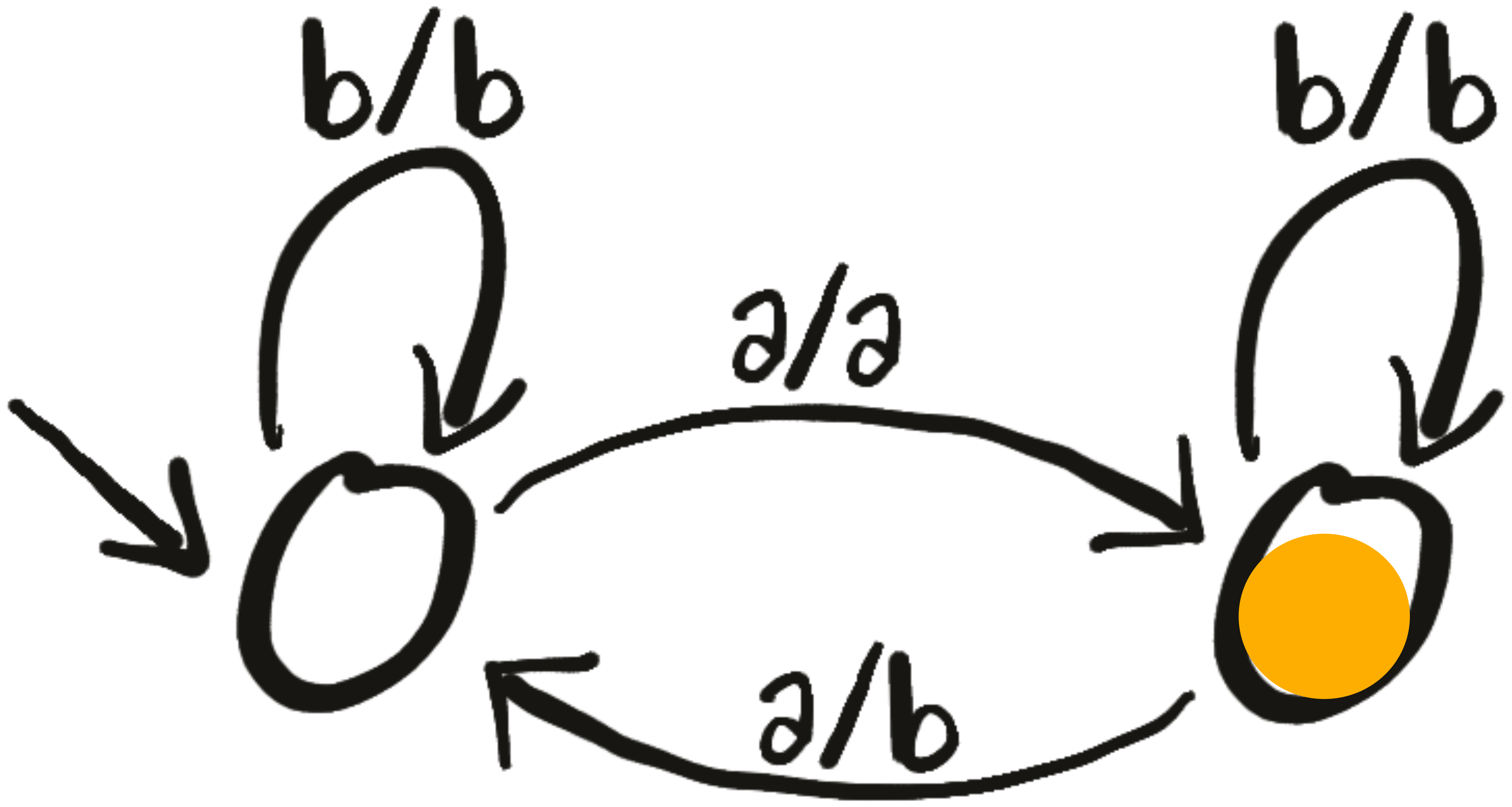
Mealy machine



a a b b a a

a b b b a

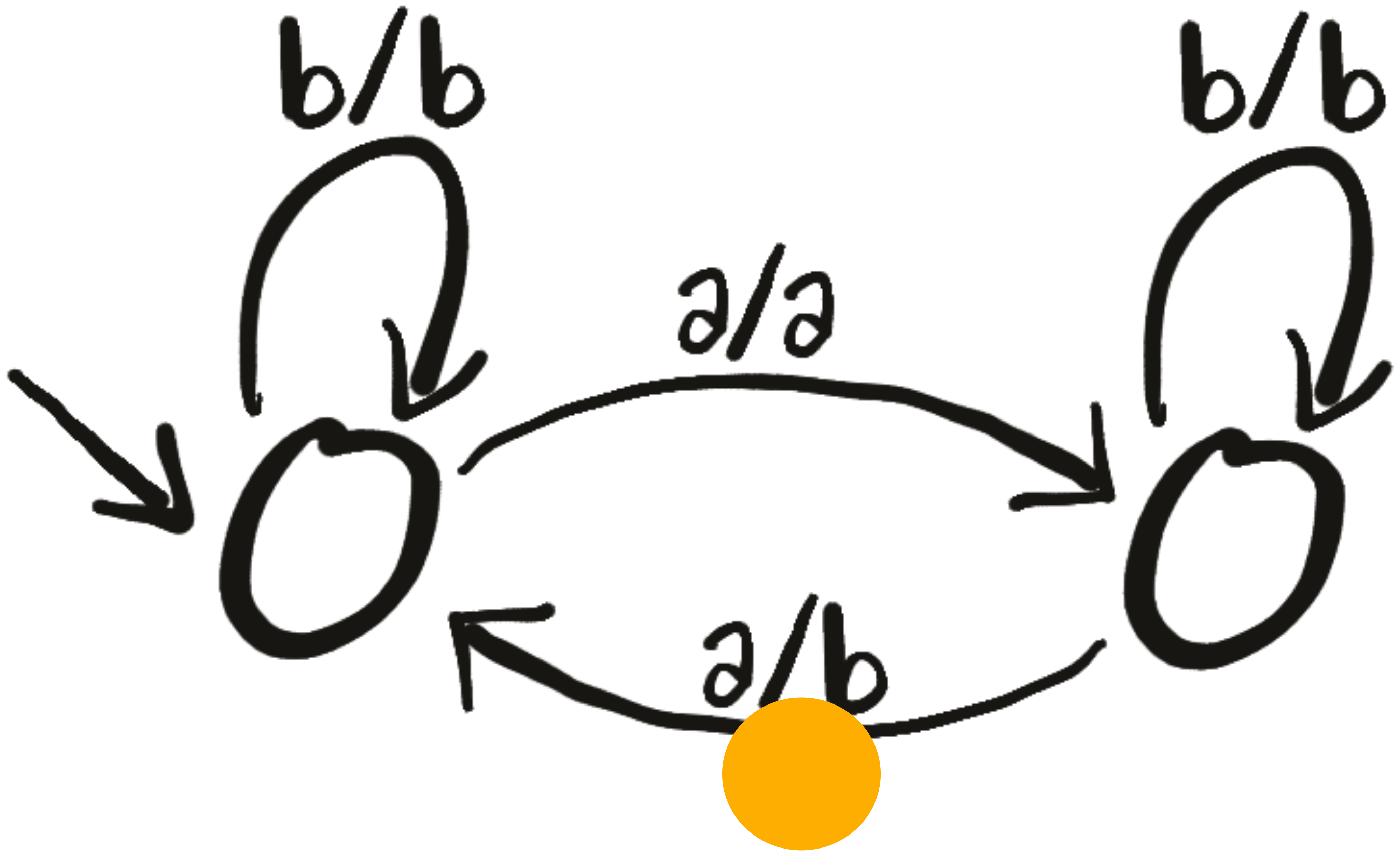
Mealy machine



a a b b a a

a b b b a

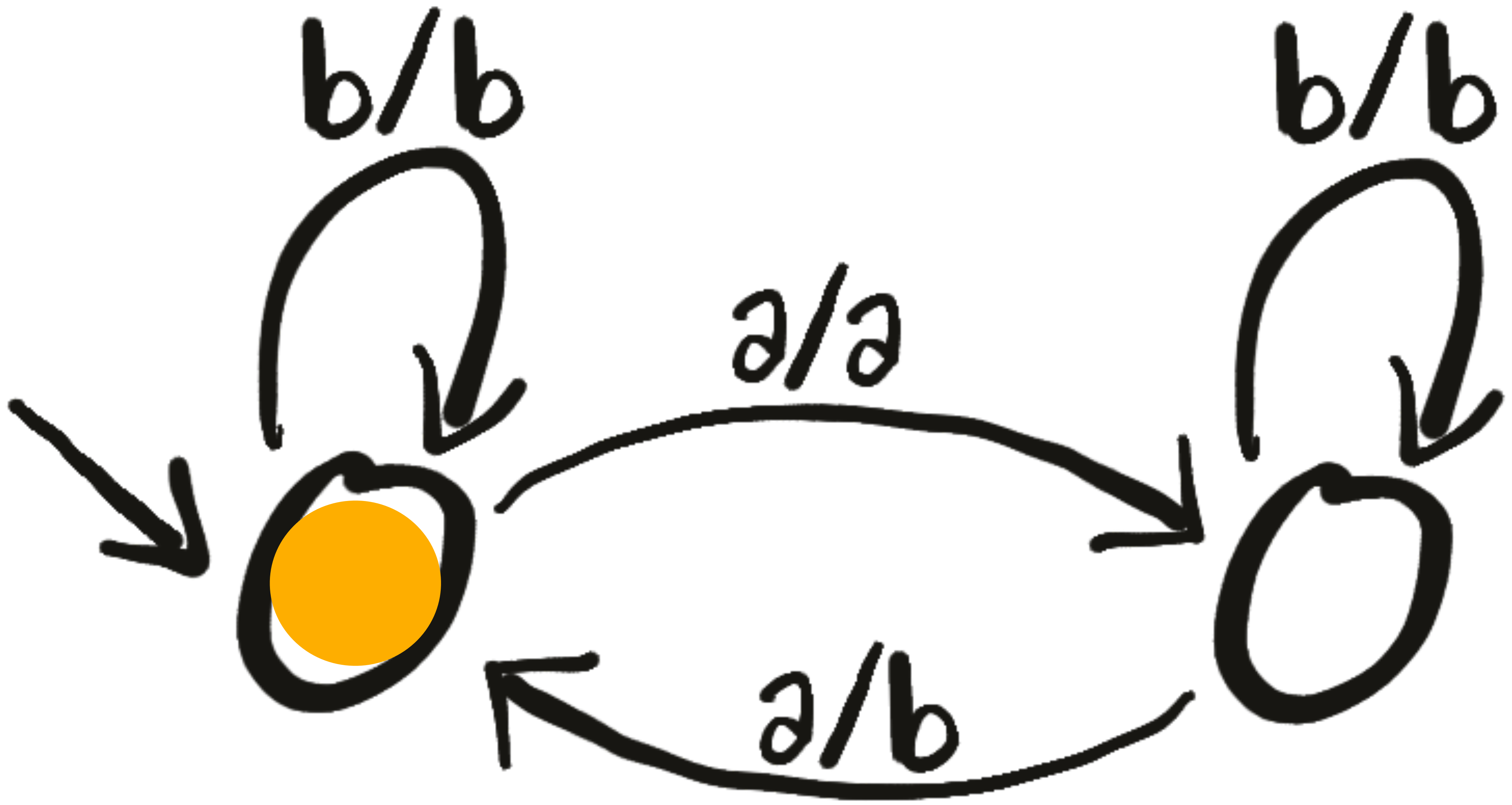
Mealy machine



a a b b a a

a b b b a b

Mealy machine



a a b b a a

a b b b a b

Finite monoids and Mealy machines

$$A \qquad h : \Sigma \to A \qquad \lambda : A \to \Gamma$$

Finite monoids and Mealy machines

$$A \qquad h : \Sigma \to A \qquad \lambda : A \to \Gamma$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{MMh} (A^*)^* \xrightarrow{M\,\texttt{prod}} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

$$A \qquad h : \Sigma \to A \qquad \lambda : A \to \Gamma$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} \underbrace{(\Sigma^*)^* \xrightarrow{MMh} (A^*)^* \xrightarrow{M\,\texttt{prod}} A^*}_{Mh^*} \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

$$A = \mathbb{Z}_2 \times \{a, b\}$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

$$A = \mathbb{Z}_2 \times \{a, b\}$$

$$(p_1,\ l_1) \cdot (p_2, l_2) = (p_1 + p_2, l_2)$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

$$h(a) = (1, a)$$

$$h(b) = (0, b)$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

$$\lambda(0, a) = b \qquad \lambda(0, b) = b$$

$$\lambda(1, a) = a \qquad \lambda(1, b) = b$$

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[a b a b]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[[a] [a b] [a b a] [a b a b]]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[(1, a) (1, b) (0, a) (0, b)]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[a (1, b) (0, a) (0, b)]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[a b (0, a) (0, b)]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[a b b (0, b)]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

Replace every other a with b

[a b b b]

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^* \xrightarrow{Mh^*} A^* \xrightarrow{M\lambda} \Gamma^*$$

Finite monoids and Mealy machines

$$\Sigma^* \xrightarrow{\texttt{prefixes}} (\Sigma^*)^*$$

Comonads

$$(M,\ \epsilon_X : MX \to X,\ \delta_X : MX \to MMX)$$

Comonads

$$(M,\ \epsilon_X : MX \to X,\ \delta_X : MX \to MMX)$$

$$\begin{tikzcd}
MX \arrow[r, "\delta_X"] \arrow[d, "\delta_X"] & MMX \arrow[d, "M\delta_X"] \\
MMX \arrow[r, "\delta_{MX}"] & MMMX
\end{tikzcd}$$

Comonads

$$(M,\ \epsilon_X : MX \to X,\ \delta_X : MX \to MMX)$$

$$\begin{tikzcd}
MMX \arrow[r, "\epsilon_{MX}"] \arrow[d, "M\epsilon_X"] \arrow[dr, "\mathrm{id}"] & MX \arrow[d, "\delta_X"] \\
MX \arrow[r, "\delta_X"'] & MMX
\end{tikzcd}$$

Monad, comonad, and a transducer

$$A \qquad\qquad h : \Sigma \to A \qquad\qquad \lambda : A \to \Gamma$$

Monad, comonad, and a transducer

$$A \qquad h : \Sigma \to A \qquad \lambda : A \to \Gamma$$

$$M\Sigma \xrightarrow{\delta} MM\Sigma \xrightarrow{Mh^*} MA \xrightarrow{M\lambda} M\Gamma$$

Monad, comonad, and a transducer

$$A \qquad h : \Sigma \to A \qquad \lambda : A \to \Gamma$$

$$M\Sigma \xrightarrow{\delta} MM\Sigma \xrightarrow{Mh^*} MA \xrightarrow{M\lambda} M\Gamma$$

$$h* = Mh; \texttt{prod}$$

Non-empty lists

Non-empty lists

$$\epsilon([a, b, c, d]) = d$$

Non-empty lists

$$\epsilon([a, b, c, d]) = d$$

$$\delta([a, b, c, d]) = \big[[a], [a, b], [a, b, c], [a, b, c, d]\big]$$

Non-empty lists

$$\epsilon([a, b, c, d]) = d$$

$$\delta([a, b, c, d]) = \big[[a], [a, b], [a, b, c], [a, b, c, d]\big]$$

Mealy machines

Non-empty lists (right-to-left)

Non-empty lists (right-to-left)

$$\epsilon([a, b, c, d]) = a$$

Non-empty lists (right-to-left)

$$\epsilon([a, b, c, d]) = a$$

$$\delta([a, b, c, d]) = \Big[[a, b, c, d], [b, c, d], [c, d], [d]\Big]$$

Non-empty lists (right-to-left)

$$\epsilon([a, b, c, d]) = a$$

$$\delta([a, b, c, d]) = \big[[a, b, c, d], [b, c, d], [c, d], [d]\big]$$

right-to-left Mealy machines

Lists with an underlined element

Lists with an underlined element

$$\epsilon([a, b, \underline{c}, d]) = c$$

Lists with an underlined element

$$\epsilon([a, b, \underline{c}, d]) = c$$

$$\delta([a, b, \underline{c}, d]) = \Big[[\underline{a}, b, c, d], [a, \underline{b}, c, d], \underline{[a, b, \underline{c}, d]}, [a, b, c, \underline{d}], \Big]$$

Lists with an underlined element (monad)

Lists with an underlined element (monad)

$$\eta(a) = [\underline{a}]$$

Lists with an underlined element (monad)

$$\eta(a) = [\underline{a}]$$

$$\mu\left(\left[[\underline{a}, b], \underline{[c, d, \underline{e}]}, [\underline{f}, g]\right]\right) = [a, b, c, d, \underline{e}, f, g]$$

Lists with an underlined element (monad)

$$\eta(a) = [\underline{a}]$$

$$\mu\left(\left[[\underline{a}, b], \underline{[c, d, \underline{e}]}, [\underline{f}, g]\right]\right) = [a, b, c, d, \underline{e}, f, g]$$

letter-to-letter rational functions

Letter to letter rational functions

Letter to letter rational functions

Replace the first letter with a copy of the last letter

Letter to letter rational functions

Replace the first letter with a copy of the last letter

a a b b

Letter to letter rational functions

Replace the first letter with a copy of the last letter

a a b b

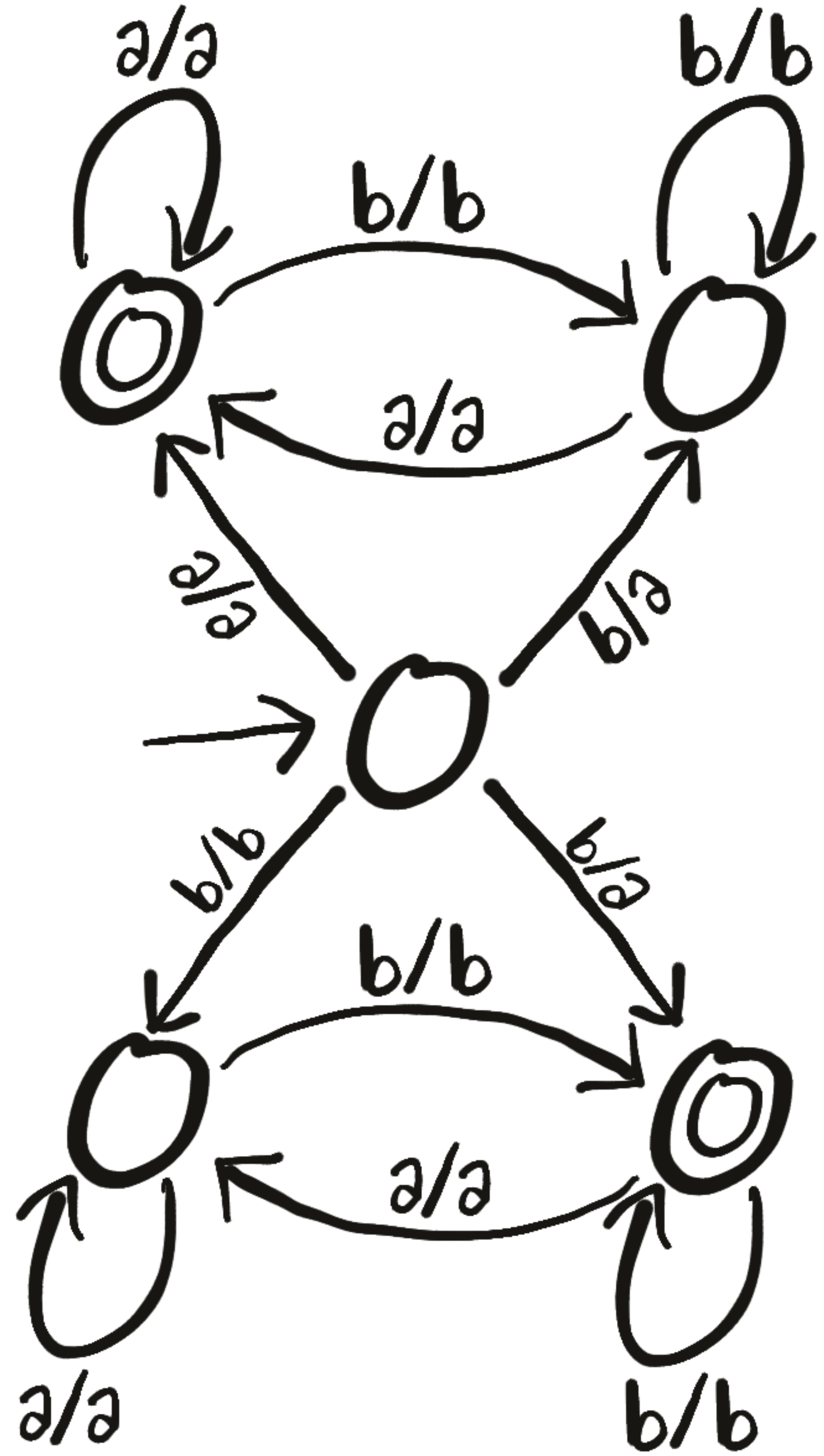
b a b b

Unambiguous (nondeterministic) Mealy machines

Replace the first letter with a copy of the last letter

Unambiguous (nondeterministic) Mealy machines

Replace the first letter with a copy of the last letter



Algebras for lists with an underlined element

$$\underline{A} = \underset{\text{prefix}}{A} \times \underset{\text{underlined}}{A} \times \underset{\text{suffix}}{A}$$

$$[(p_1, x_1, s_1), \ldots, \underline{(p_i, x_i, s_i)}, \ldots, (p_n, x_n, s_i)] \in M\underline{A}$$

Algebras for lists with an underlined element

$$\underline{A} = \underset{\text{prefix}}{A} \times \underset{\text{underlined}}{A} \times \underset{\text{suffix}}{A}$$

$$[(p_1, x_1, s_1), \ldots, \underline{(p_i, x_i, s_i)}, \ldots, (p_n, x_n, s_i)] \in M\underline{A}$$

Algebras for lists with an underlined element

$$\underline{A} = \underset{\text{prefix}}{A} \times \underset{\text{underlined}}{A} \times \underset{\text{suffix}}{A}$$

$$[(p_1, x_1, s_1), \ldots, (p_i, x_i, s_i), \ldots, (p_n, x_n, s_i)] \in M\underline{A}$$

$$\downarrow$$

$$(a_1 \cdot \ldots \cdot a_{i-1} \cdot p_i, \;\; x_i, \;\; s_i \cdot a_{i+1} \cdot \ldots \cdot a_n)$$

where $a_j = p_j \cdot x_j \cdot s_j$

Transducers for lists with an underlined element

$$\underline{A} = \underbrace{A}_{\text{prefix}} \times \underbrace{A}_{\text{underlined}} \times \underbrace{A}_{\text{suffix}}$$

$$h : \Sigma \to \underline{A} \qquad \lambda : \underbrace{A}_{\text{prefix}} \times \underbrace{A}_{\text{current letter}} \times \underbrace{A}_{\text{suffix}} \to \Gamma$$

Transducers for lists with an underlined element

$$\underline{A} = \underbrace{A^*}_{\text{prefix}} \times \underbrace{A}_{\text{underlined}} \times \underbrace{A^*}_{\text{suffix}}$$

$$h : \Sigma \to \underline{A} \qquad \lambda : \underbrace{A^*}_{\text{prefix}} \times \underbrace{A}_{\text{current letter}} \times \underbrace{A^*}_{\text{suffix}} \to \Gamma$$

Transducers for lists with an underlined element

$$\underline{A} = \underbrace{A^*}_{\text{prefix}} \times \underbrace{A}_{\text{underlined}} \times \underbrace{A^*}_{\text{suffix}}$$

$$h : \Sigma \to \underline{A} \qquad \lambda : \underbrace{A^*}_{\text{prefix}} \times \underbrace{A}_{\text{current letter}} \times \underbrace{A^*}_{\text{suffix}} \to \Gamma$$

Eilenberg bimachine

Three M's for finite words

M	Expressive Power

Three M's for finite words

M	Expressive Power
Non-empty lists with prefixes	Mealy machines

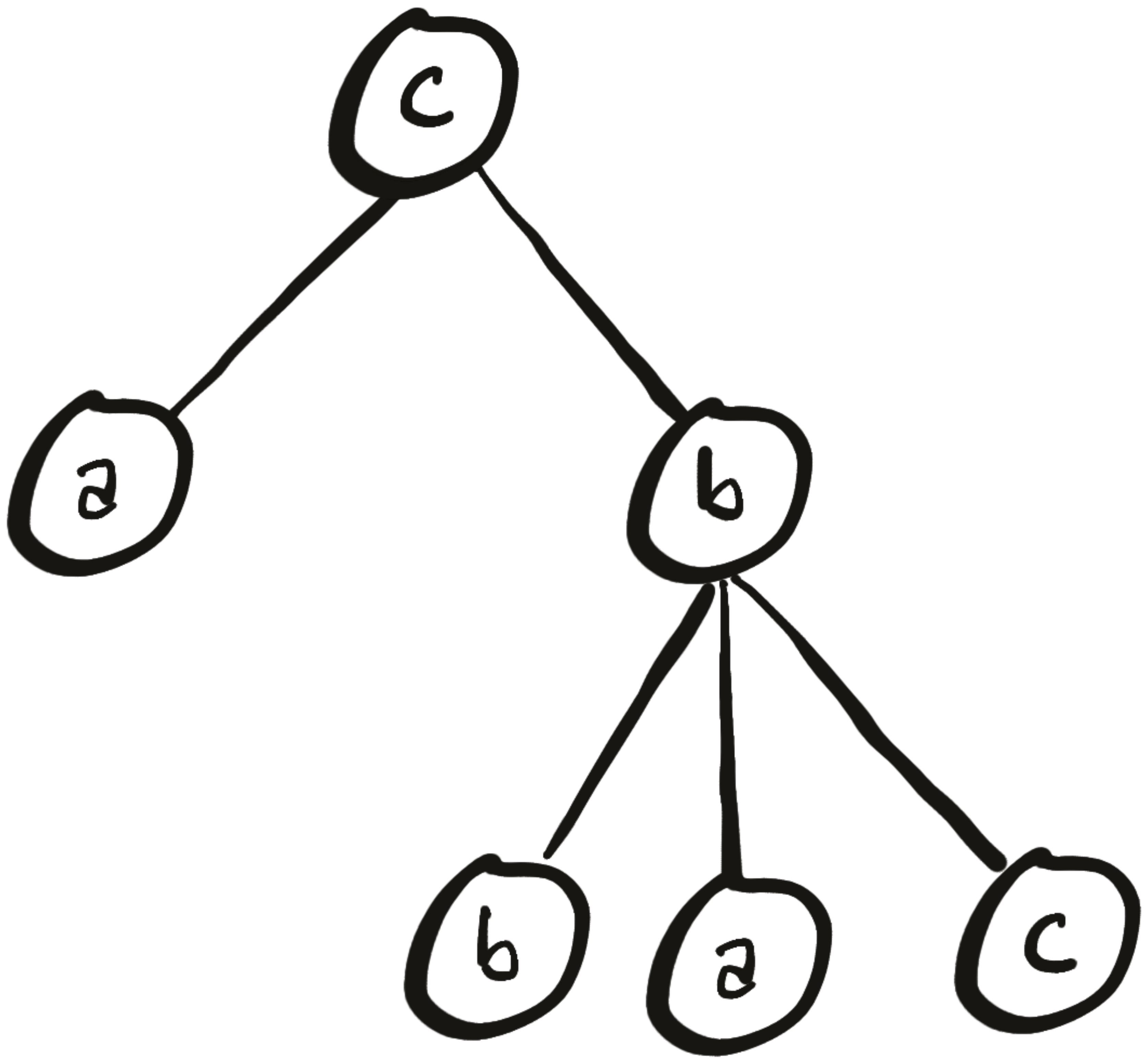
Three M's for finite words

M	Expressive Power
Non-empty lists with prefixes	Mealy machines
Non-empty lists with suffixes	Right-to-left Mealy machines

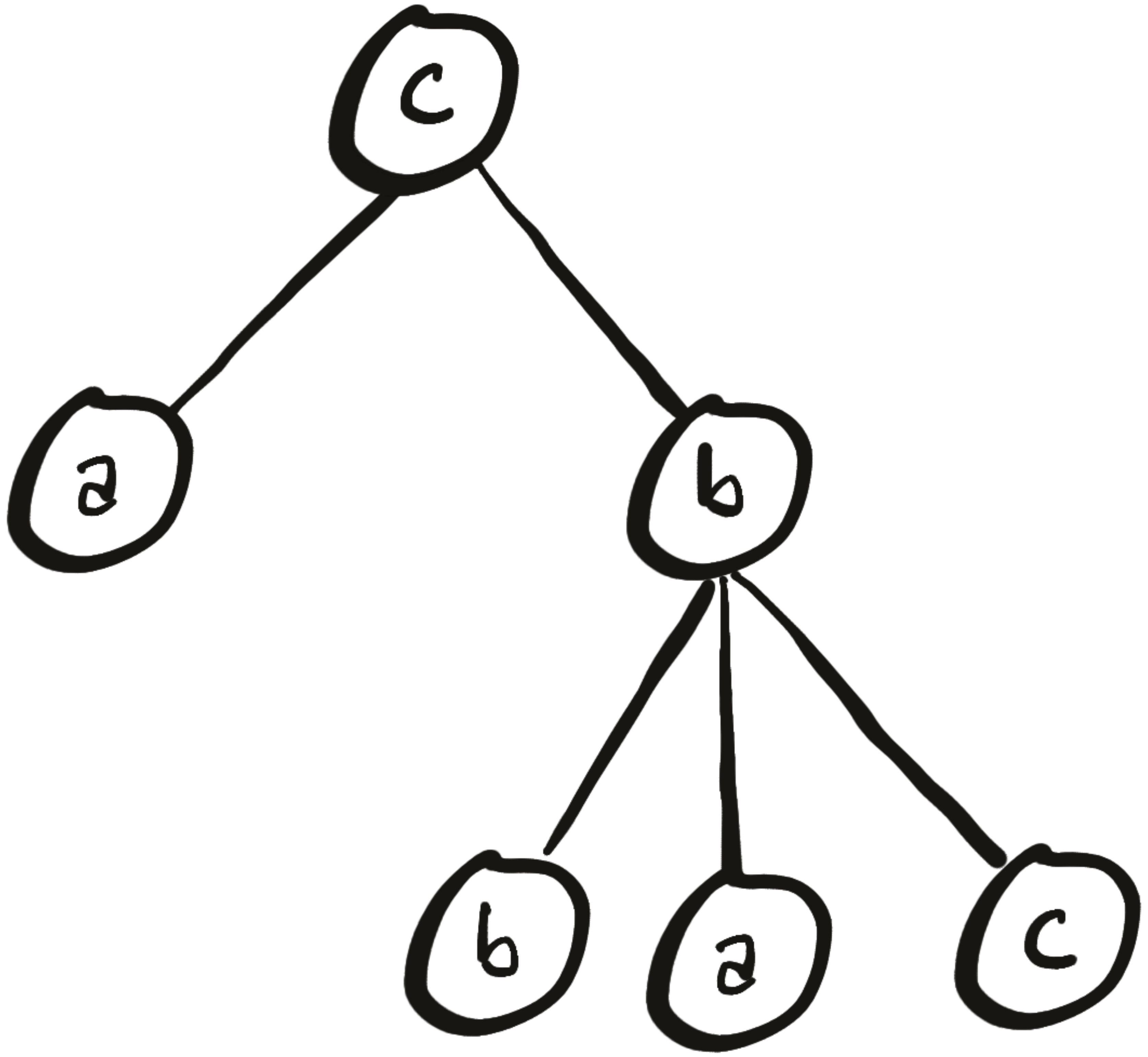
Three M's for finite words

M	Expressive Power
Non-empty lists with prefixes	Mealy machines
Non-empty lists with suffixes	Right-to-left Mealy machines
Lists with an underlined element	Rational letter-to-letter functions

Non-empty trees

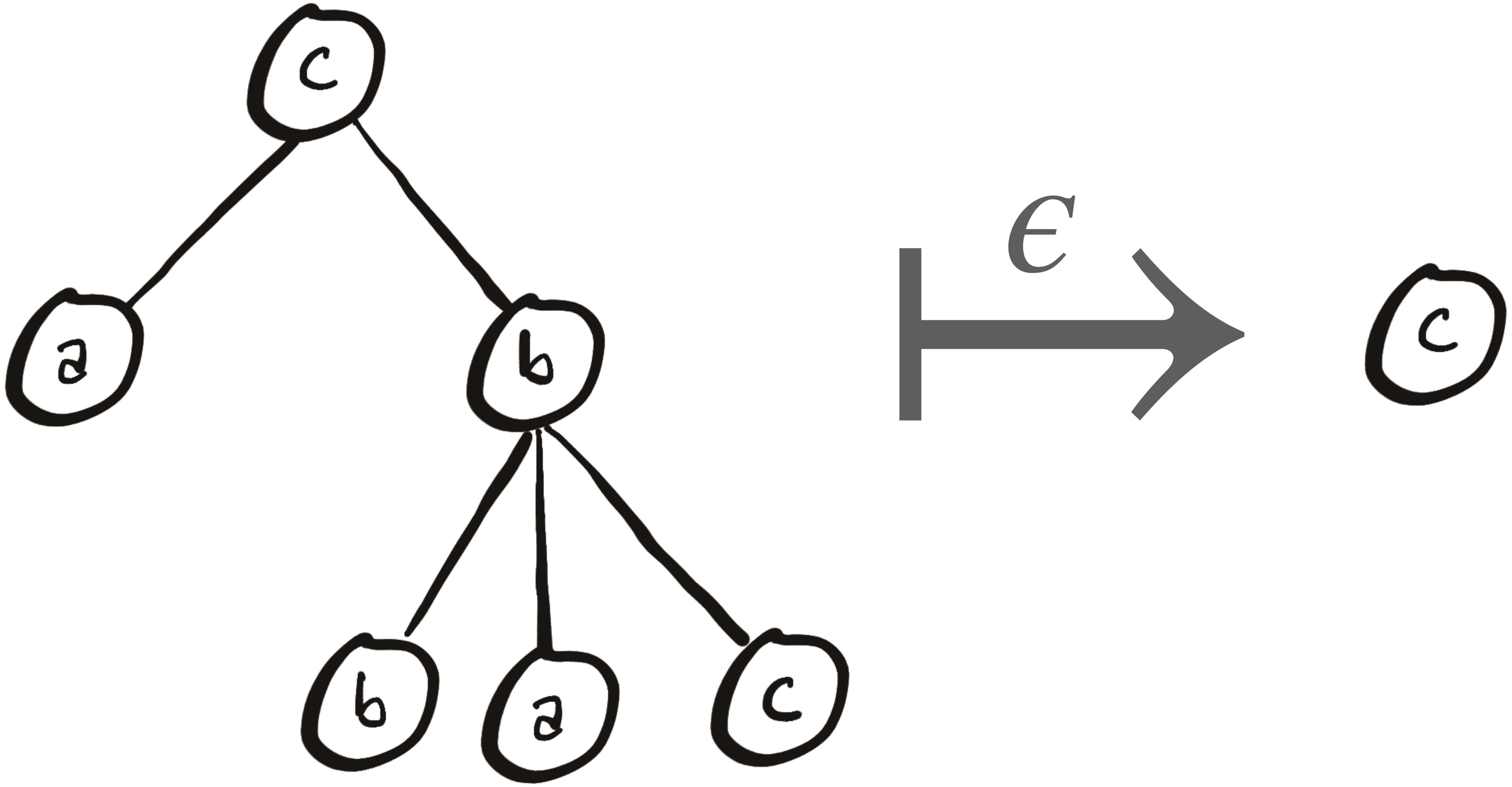


Non-empty trees

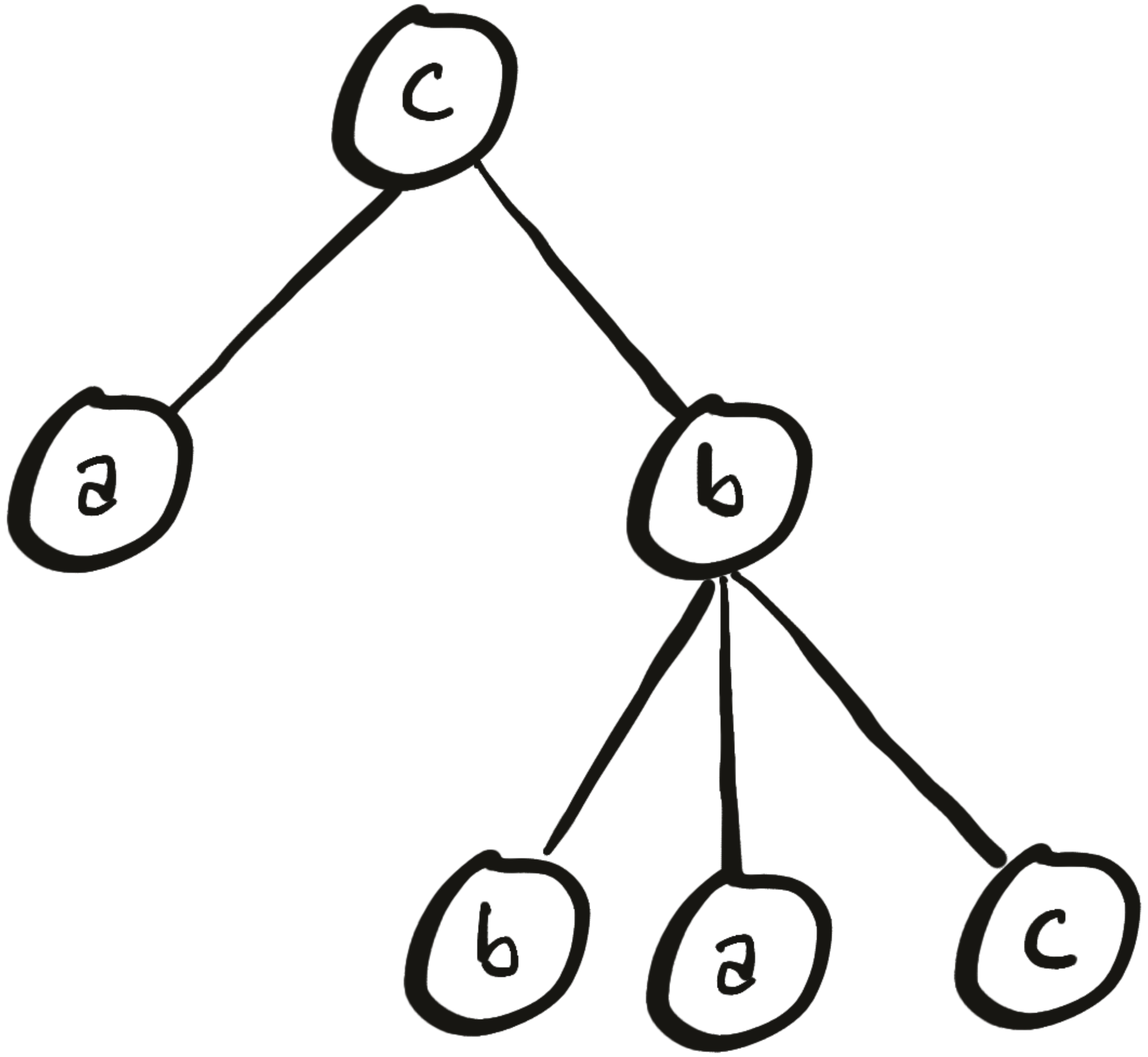


$$\xmapsto{\epsilon}$$

Non-empty trees

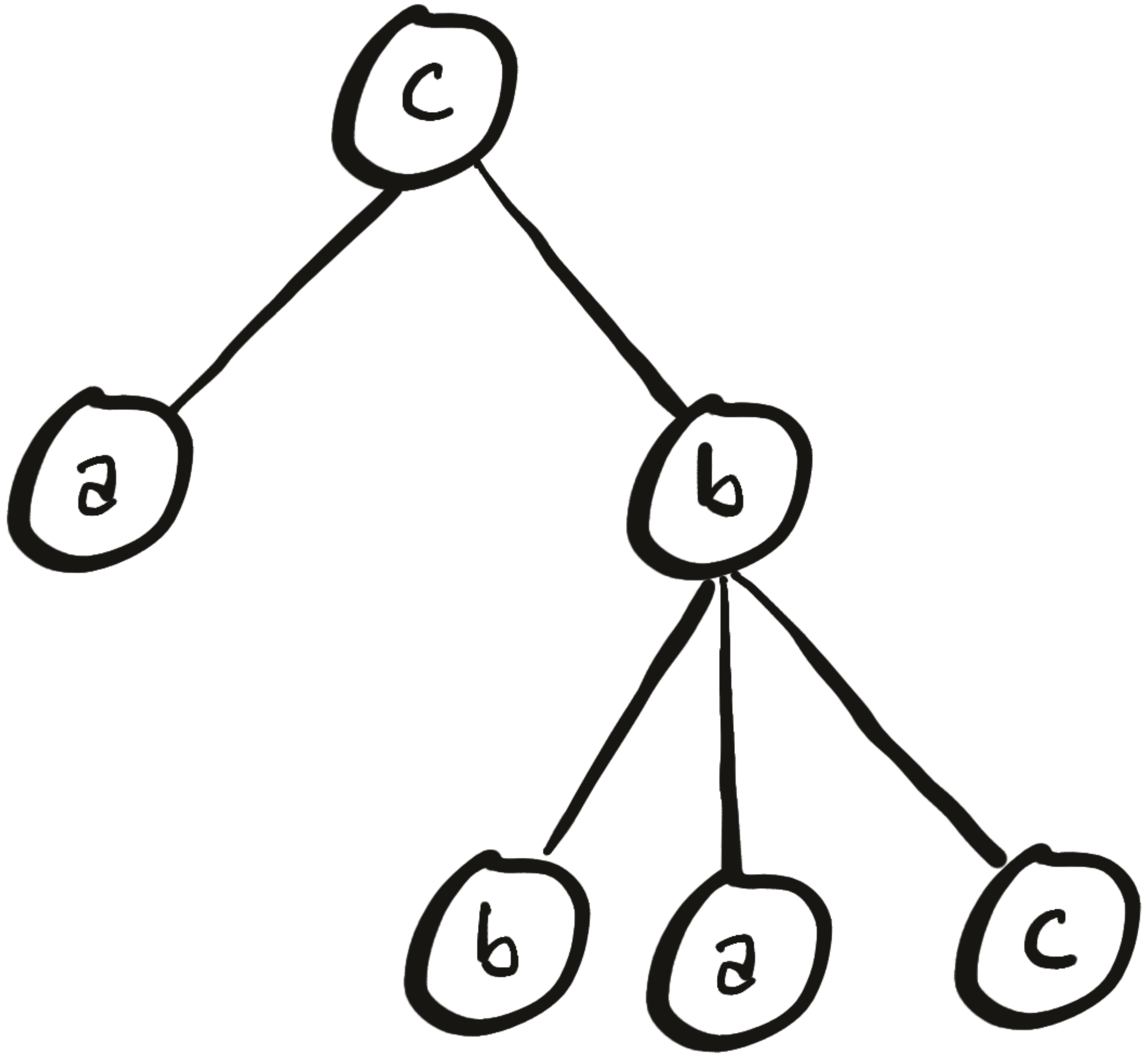


Non-empty trees



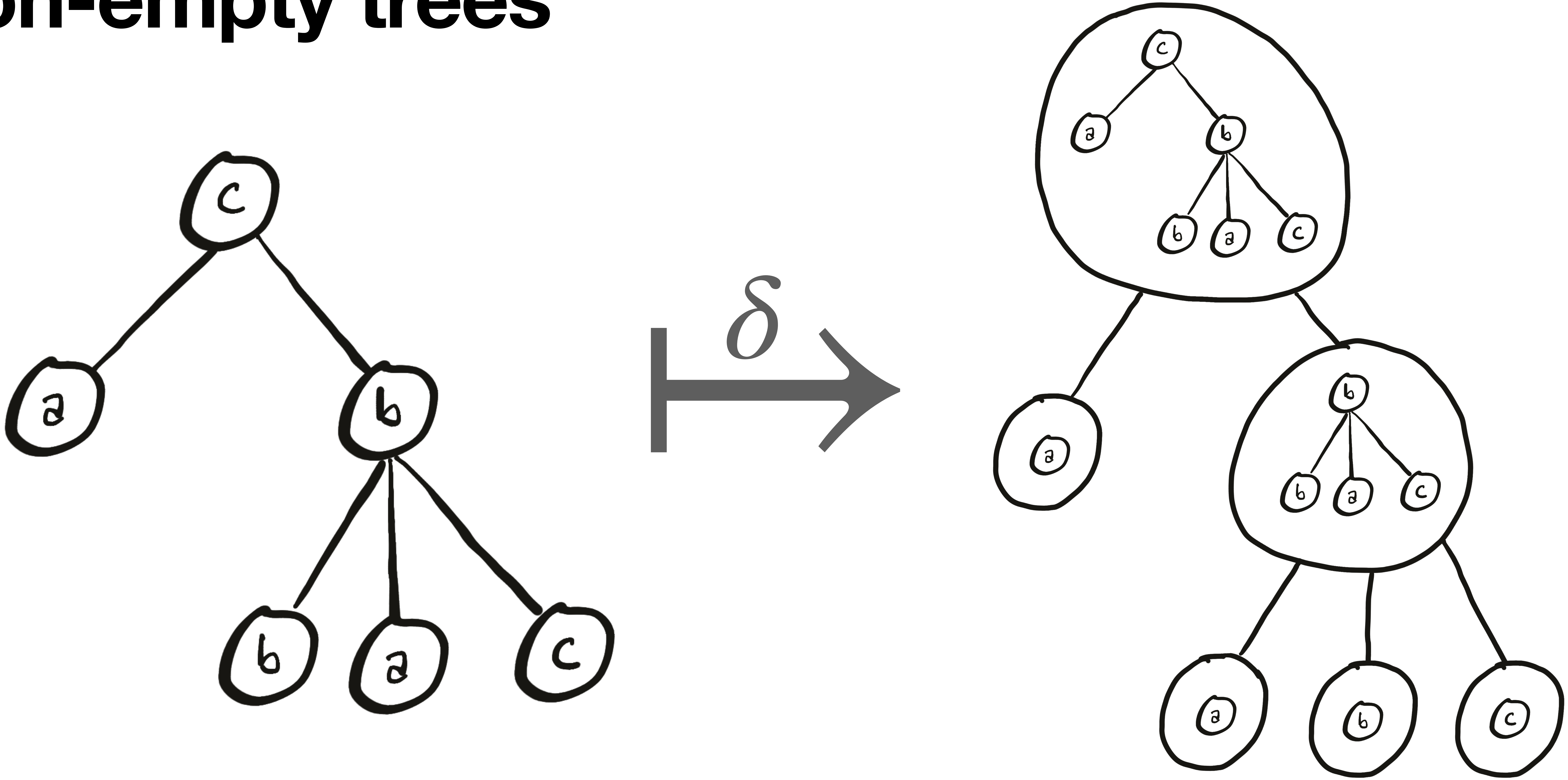
$$\xmapsto{\epsilon} c$$

Non-empty trees



$$\xmapsto{\delta}$$

Non-empty trees



Non-empty trees

Non-empty trees

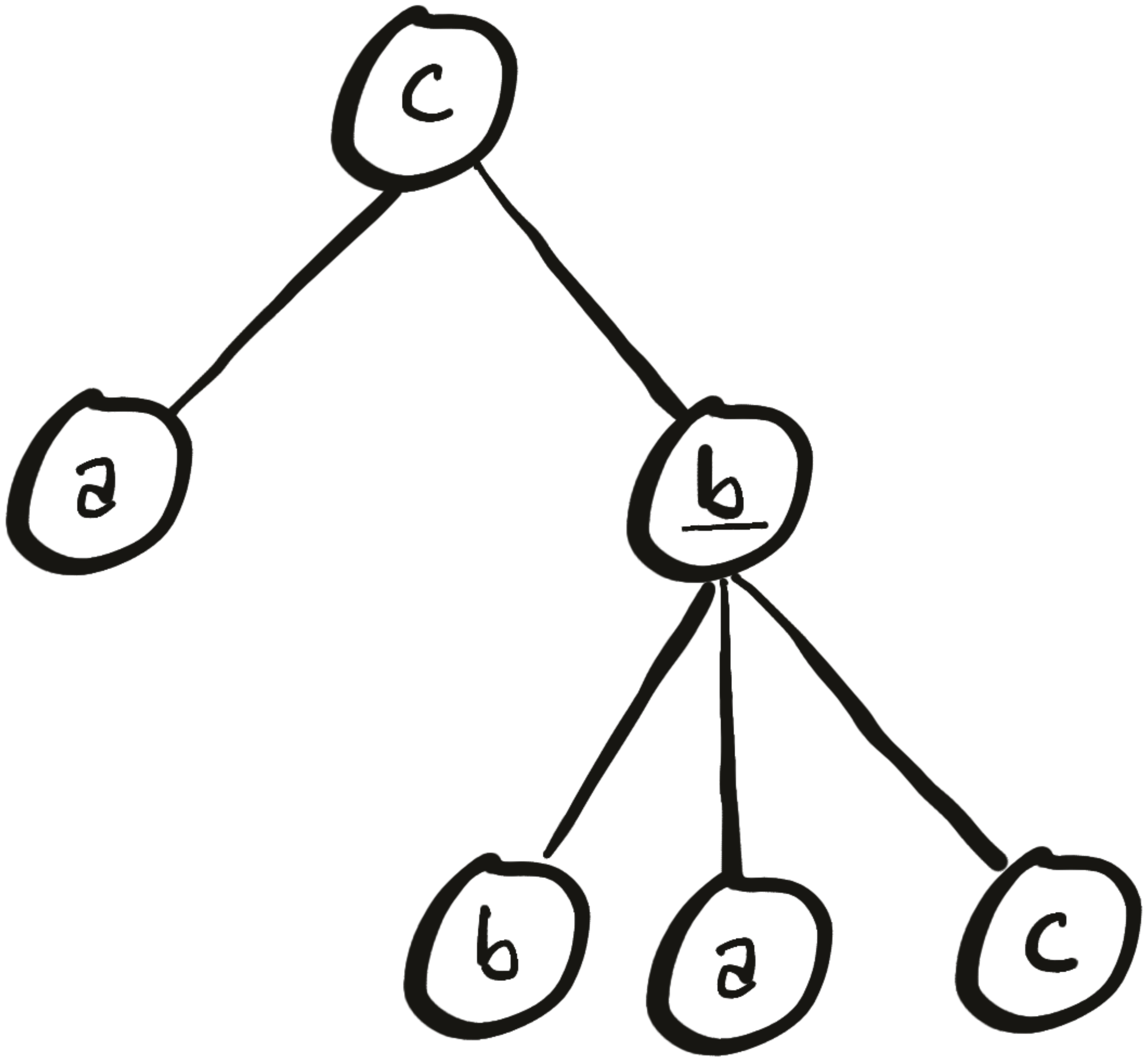
Bottom up Mealy machines
on trees

Non-empty trees

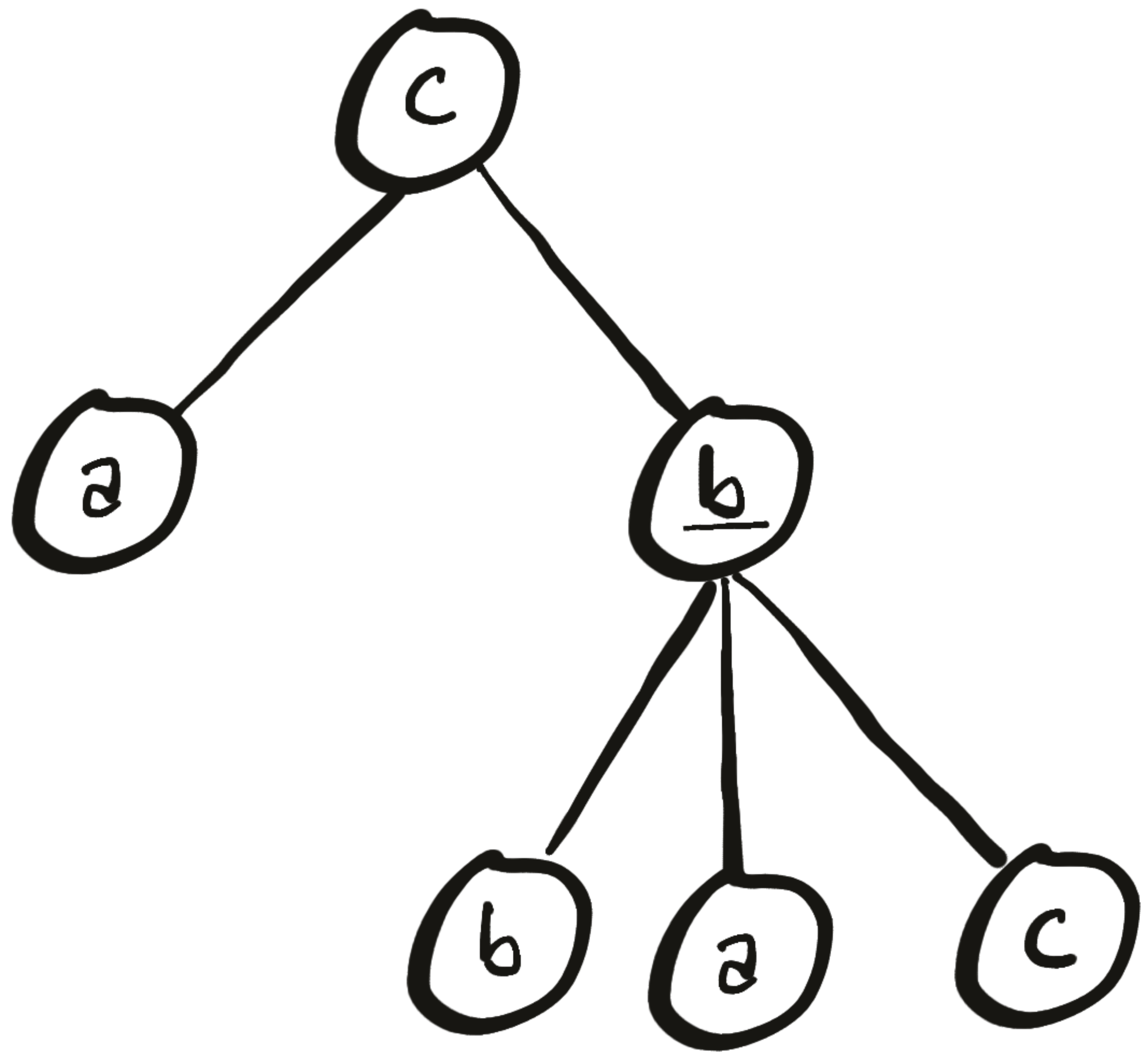
Bottom up Mealy machines
on trees

?

Trees with an underlined element

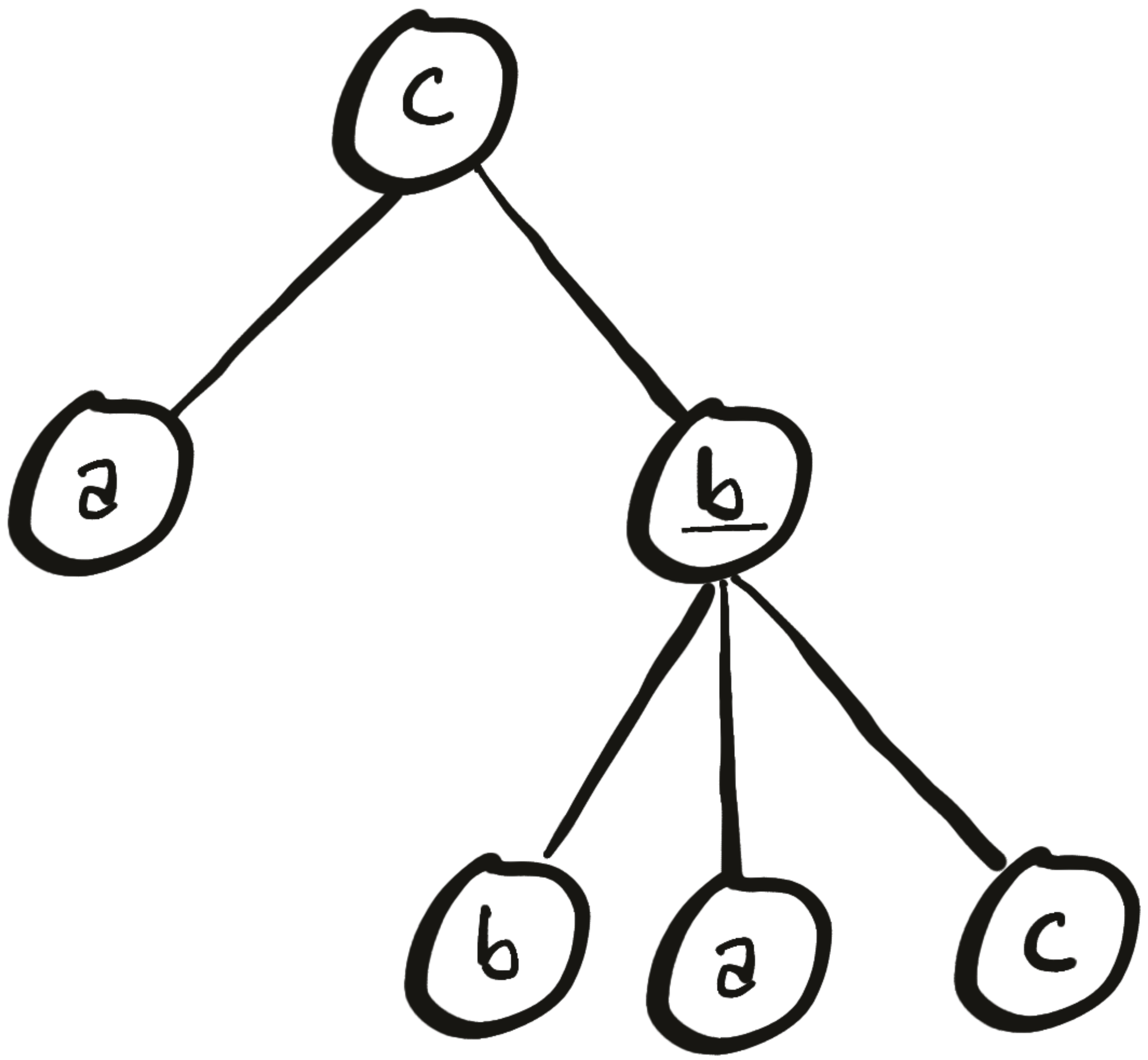


Trees with an underlined element



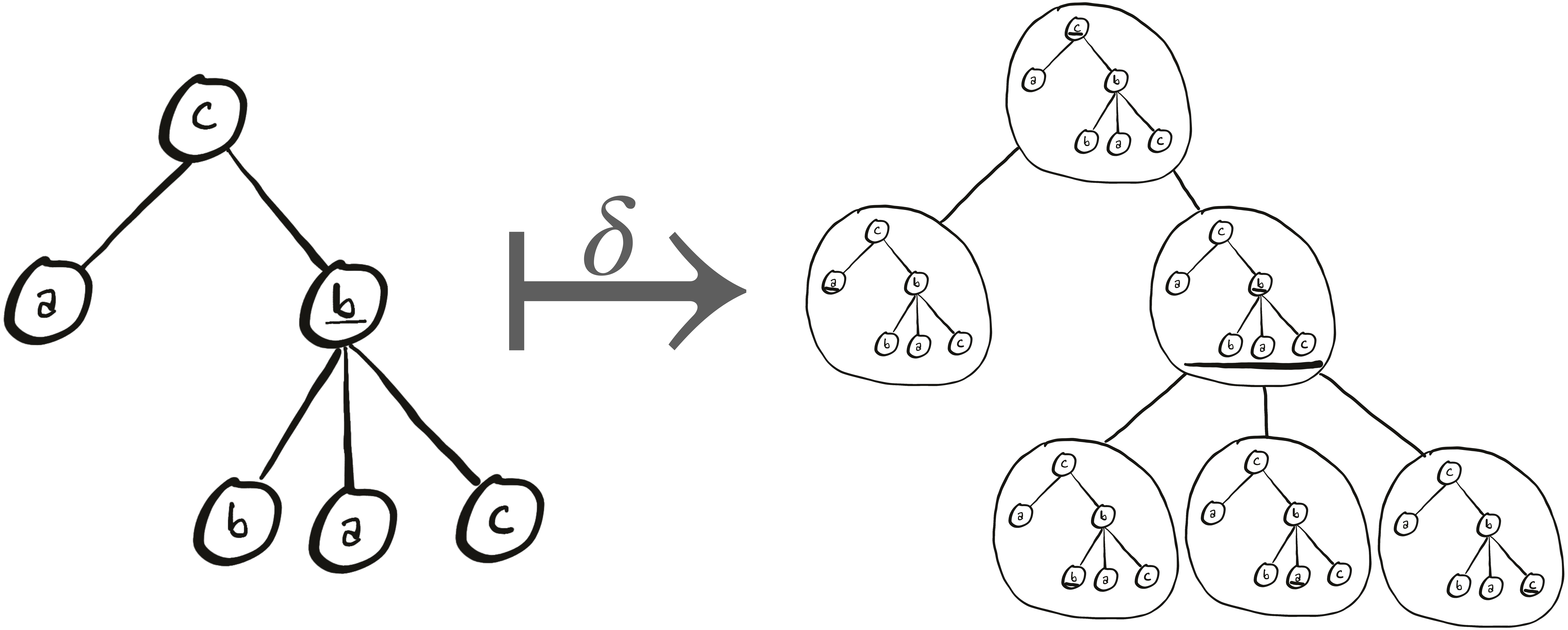
$\overset{\epsilon}{\longmapsto}$

Trees with an underlined element

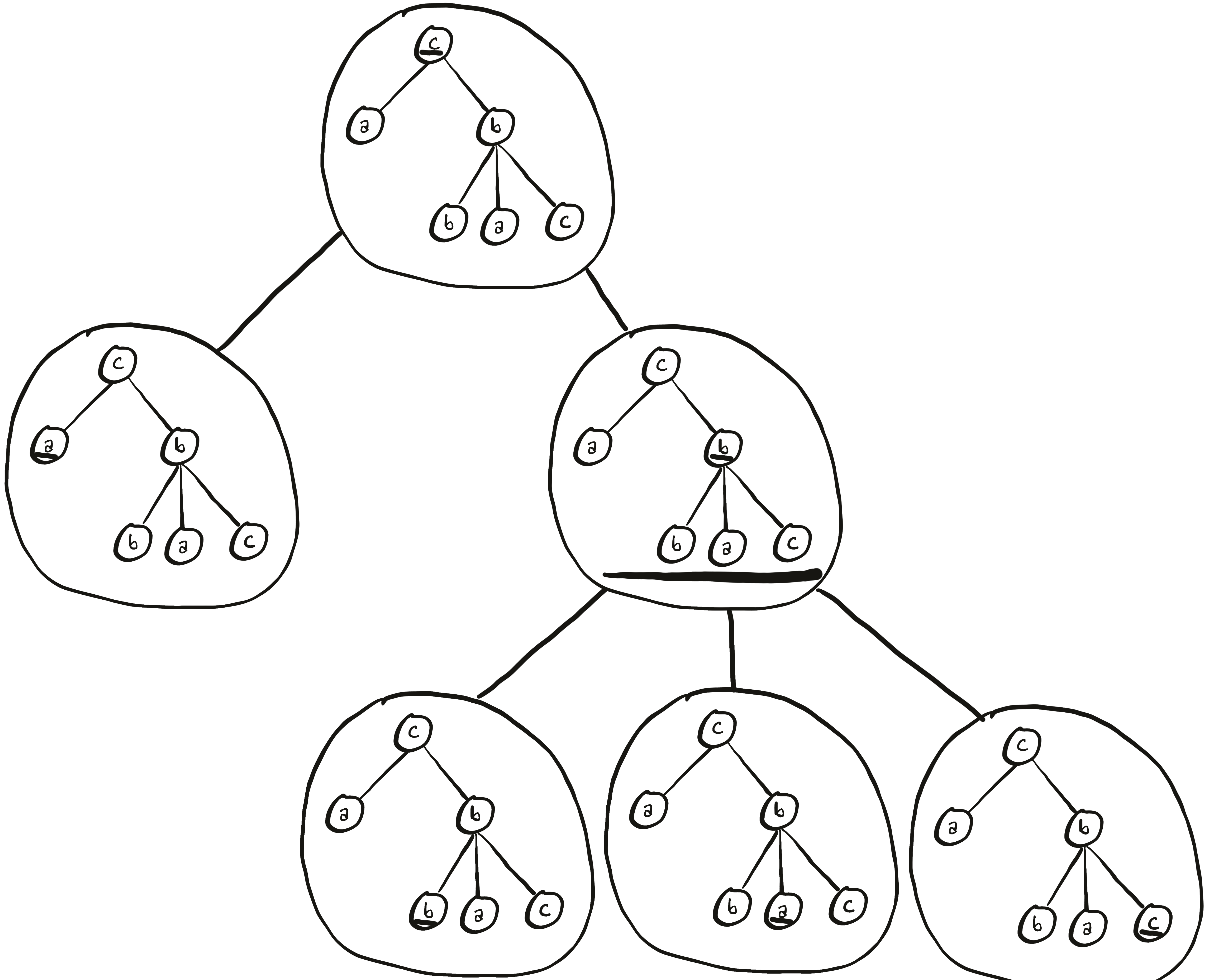


$$\xmapsto{\epsilon} \; b$$

Trees with an underlined element



Trees with an underlined element



Trees with an underlined element

Trees with an underlined element

Rational functions
on trees

Trees with an underlined element

Rational functions
on trees

?

Other examples

Other examples

- Labelled orders with a maximal element

Other examples

- Labelled orders with a maximal element
- Labelled orders with an underlined element

Other examples

- Labelled orders with a maximal element
- Labelled orders with an underlined element
- Terms with an underlined leaf

Compositions

Composition

$$\begin{tikzcd}
M\Sigma \arrow[r, "\delta"] \arrow[rrrrdddd, dashed] & MM\Sigma \arrow[r, "MMh_1"] & MMA_1 \arrow[r, "M\,prod_1"] & MA_1 \arrow[r, "M\lambda_1"] & M\Gamma \arrow[d, "\delta"] \\
 & & & & MM\Gamma \arrow[d, "MMh_2"] \\
 & & & & MMA_2 \arrow[d, "M\,prod_2"] \\
 & & & & MA_2 \arrow[d, "M\lambda_2"] \\
 & & & & M\Delta
\end{tikzcd}$$

Composition

$$
\begin{tikzcd}
M\Sigma \arrow[r, "\delta"] \arrow[ddddrrrr, dashed, "{S_3, h_3, \lambda_3}"'] & MM\Sigma \arrow[r, "MMh_1"] & MMA_1 \arrow[r, "M\,prod_1"] & MA_1 \arrow[r, "M\lambda_1"] & M\Gamma \arrow[d, "\delta"] \\
 & & & & MM\Gamma \arrow[d, "MMh_2"] \\
 & & & & MMA_2 \arrow[d, "M\,prod_2"] \\
 & & & & MA_2 \arrow[d, "M\lambda_2"] \\
 & & & & M\Delta
\end{tikzcd}
$$

Composition

Composition

- M has to be strong: $\mathtt{strenght} : X \times MY \to M(X \times Y)$

Composition

- M has to be strong: $\texttt{strenght} : X \times MY \to M(X \times Y)$
- M has to have: $\texttt{set} : MX \times X \to MX$

Composition

- M has to be strong: $\texttt{strenght} : X \times MY \to M(X \times Y)$
- M has to have: $\texttt{set} : MX \times X \to MX$
- All those structures have to be compatible

Set structure

$$\texttt{set} : MX \times X \to MX$$

Based on Haskell's lenses:

https://www.schoolofhaskell.com/school/to-infinity-and-beyond/pick-of-the-week/a-little-lens-starter-tutorial

Set structure

$$\texttt{set} : MX \times X \to MX$$

$$\text{set}([a, b, \underline{c}, d],\ f\ ) =$$

Set structure

$$\texttt{set} : MX \times X \to MX$$

$$\text{set}([a, b, \underline{c}, d],\ f\ \ ) = \quad [a, b, \underline{\quad\quad}, d]$$

Set structure

$$\texttt{set} : MX \times X \to MX$$

$$\text{set}([a, b, \underline{c}, d], \ f \ ) = \quad [a, b, \underline{\ f\ }, d]$$

Based on Haskell's lenses:

https://www.schoolofhaskell.com/school/to-infinity-and-beyond/pick-of-the-week/a-little-lens-starter-tutorial

Set structure

$$\texttt{set} : MX \times X \to MX$$

$$\begin{tikzcd} MB \arrow[r, "id \times \varepsilon"] \arrow[dr, equal] & MB \times B \arrow[d, "set"] \\ & MB \end{tikzcd}$$

get-set

$$\begin{tikzcd} MB \times B \arrow[r, "set"] \arrow[dr, "\pi_2"'] & MB \arrow[d, "get"] \\ & B \end{tikzcd}$$

get-set

$$\begin{tikzcd} (MB \times B) \times B \arrow[r, "set \times id"] \arrow[d, "\pi_1 \times id"'] & MB \times B \arrow[d, "set"] \\ MB \times B \arrow[r, "set"'] & MB \end{tikzcd}$$

set-set

Based on Haskell's lenses:

https://www.schoolofhaskell.com/school/to-infinity-and-beyond/pick-of-the-week/a-little-lens-starter-tutorial

Contexts

Let A be M-algebra:

Contexts

Let A be M-algebra:

Every element of MA corresponds to a function A^A

Contexts

Let A be M-algebra:

Every element of MA corresponds to a function A^A

$$MA \times A \xrightarrow{\texttt{set}} MA \xrightarrow{\texttt{prod}} A$$

Contexts

Let A be M-algebra:

Every element of MA corresponds to a function A^A

$$\Lambda\,(MA \times A \xrightarrow{\texttt{set}} MA \xrightarrow{\texttt{prod}} A)$$

Contexts

Let A be M-algebra:

Every element of MA corresponds to a function A^A

$$\Lambda\,(MA \times A \xrightarrow{\texttt{set}} MA \xrightarrow{\texttt{prod}} A)$$

The set of contexts is closed under compositions

Non empty lists

$$A^A \simeq A$$

Non empty lists

$$A^A \simeq A$$

Every context is of the following form:

$$x \mapsto t \cdot x$$

for some $t \in A$

Non empty lists

If A is finite:

A is a group $\Leftrightarrow$ All possible contexts are permutations

Lists with an underlined element

$$\underline{A} = A \times A \times A$$

$$A^A \simeq A^2$$

Every context is of the following form:

$$(p, x, s) \mapsto (t_1 \cdot p,\ x,\ s \cdot t_2)$$

for some $t_1, t_2 \in A$

M-wreath product

$$A_1 \qquad A_2$$

Wreath product

$$A_1 \wr_M A_2$$

Wreath product

$$A_1 \wr_M A_2 = A_1 \times (A_1^{A_1} \to A_2)$$

Wreath product

$$A_1 \wr_M A_2 = A_1 \times (A_1^{A_1} \to A_2)$$

Non-empty lists

$$A_1 \times (A_1 \to A_2)$$

Wreath product

$$A_1 \wr_M A_2 = A_1 \times (A_1^{A_1} \to A_2)$$

Non-empty lists

$$A_1 \times (A_1 \to A_2)$$

Lists with an underline

$$A_1 \times (A_1^2 \to A_2)$$

Wreath product

$$A_1 \wr_M A_2 \quad = A_1 \times (A_1^{A_1} \to A_2)$$

Non-empty lists

$$A_1 \times (A_1 \to A_2)$$

Lists with an underline

$$A_1 \times (A_1^2 \to A_2)$$

Thank you!