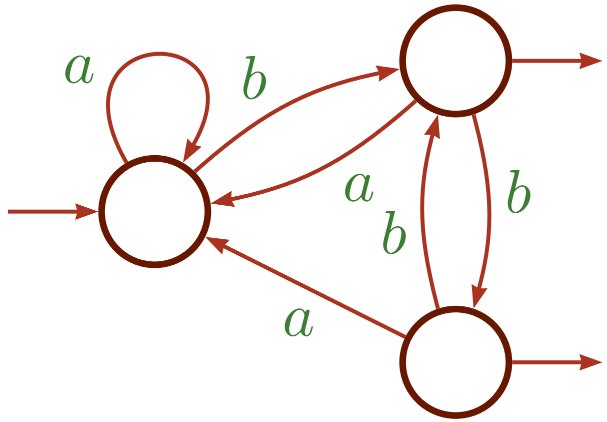
On the Size of the Universal Automaton of a Regular Language

Sylvain Lombardy

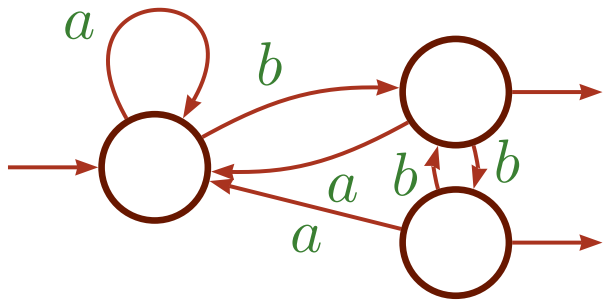
IGM - Université de Marne-la-Vallée

- Minimizing deterministic automata
  $\longrightarrow$ Merging states with the same future
  (Nerode equivalence)
  $\longrightarrow$ Hopcroft algorithm $O(n \log n)$



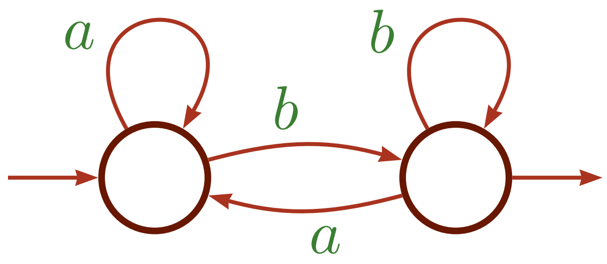
Motivation

- Minimizing deterministic automata
  $\longrightarrow$ Merging states with the same future
  (Nerode equivalence)
  $\longrightarrow$ Hopcroft algorithm $O(n \log n)$



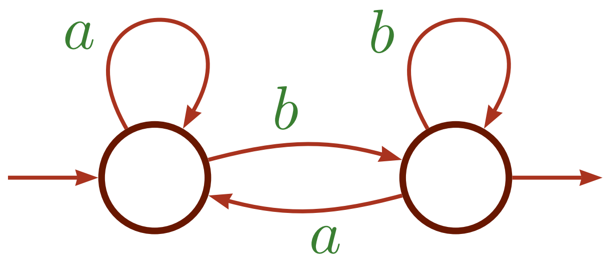
Motivation

- Minimizing deterministic automata
  $\longrightarrow$ Merging states with the same future
  (Nerode equivalence)
  $\longrightarrow$ Hopcroft algorithm $O(n \log n)$



Motivation

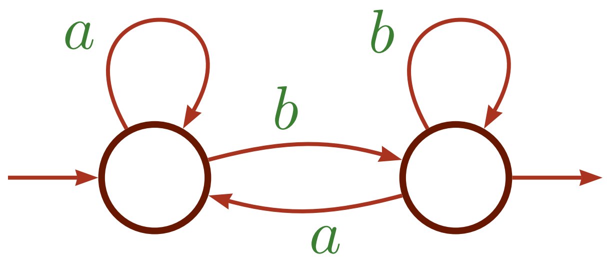
- Minimizing deterministic automata
  $\longrightarrow$ Merging states with the same future
  (Nerode equivalence)
  $\longrightarrow$ Hopcroft algorithm $O(n \log n)$



- Minimizing non deterministic automata
  $\longrightarrow$ Much harder : PSPACE-complete (Jiang - Ravikumar, 93)

Motivation

- Minimizing deterministic automata
  $\longrightarrow$ Merging states with the same future
  (Nerode equivalence)
  $\longrightarrow$ Hopcroft algorithm $O(n \log n)$



- Minimizing non deterministic automata
  $\longrightarrow$ Much harder : PSPACE-complete (Jiang - Ravikumar, 93)
  $\longrightarrow$ However merging states may reduce the size of NFAs

Motivation

- Problem 1 : How to guarantee that merging does not modify the language ?
  $\longrightarrow$ "Equivalences" (Ilie - Yu, 02)
  "Inequalities" (Champarnaud - Coulon, 04)

Motivation

- Problem 1 : How to guarantee that merging does not modify the language ?
  $\longrightarrow$ "Equivalences" (Ilie - Yu, 02)
  "Inequalities" (Champarnaud - Coulon, 04)

- Problem 2 : What is the difference between this reduce NFA and any minimal NFA for the language ?
  $\longrightarrow$ L fixed, $\exists E_L$ $\forall \mathcal{A}$ s.t. $L(\mathcal{A}) = L$
  $\|\mathcal{A}\| > E_L \Longrightarrow \mathcal{A}$ has merging states (Câmpeanu - Sântean - Yu, 04)

Motivation

- Problem 1 : How to guarantee that merging does not modify the language ?
  $\longrightarrow$ "Equivalences" (Ilie - Yu, 02)
  "Inequalities" (Champarnaud - Coulon, 04)

- Problem 2 : What is the difference between this reduce NFA and any minimal NFA for the language ?
  $\longrightarrow$ L fixed, $\exists E_L$ $\forall \mathcal{A}$ s.t. $L(\mathcal{A}) = L$
  $\|\mathcal{A}\| > E_L \Longrightarrow \mathcal{A}$ has merging states (Câmpeanu - Sântean - Yu, 04)

  $\longrightarrow$ $E_L \leqslant 2^k$ (k : size of the minimal DFA)
  (Grunsky - Kurganskyy - Potapov, 06)

Motivation

- Problem 1 : How to guarantee that merging does not modify the language ?
  $\longrightarrow$ "Equivalences" (Ilie - Yu, 02)
  "Inequalities" (Champarnaud - Coulon, 04)

- Problem 2 : What is the difference between this reduce NFA and any minimal NFA for the language ?
  $\longrightarrow$ L fixed, $\exists E_L$ $\forall \mathcal{A}$ s.t. $L(\mathcal{A}) = L$
  $\|\mathcal{A}\| > E_L \Longrightarrow \mathcal{A}$ has merging states (Câmpeanu - Sântean - Yu, 04)

  $\longrightarrow$ $E_L \leqslant 2^k$ (k : size of the minimal DFA)
  (Grunsky - Kurganskyy - Potapov, 06)
  $\longrightarrow$ $E_L \leqslant D(n)$ (n : size of a minimal NFA)

Factorizations

$\mathcal{A}$ accepts L $\qquad$ p state of $\mathcal{A}$

$$\mathsf{Past}(p) = \{w \mid \to i \xrightarrow{w} p\} \qquad \mathsf{Fut}(p) = \{w \mid p \xrightarrow{w} t \to\}$$

$\longrightarrow$ $\mathsf{Past}(p).\mathsf{Fut}(p) \subseteq L$

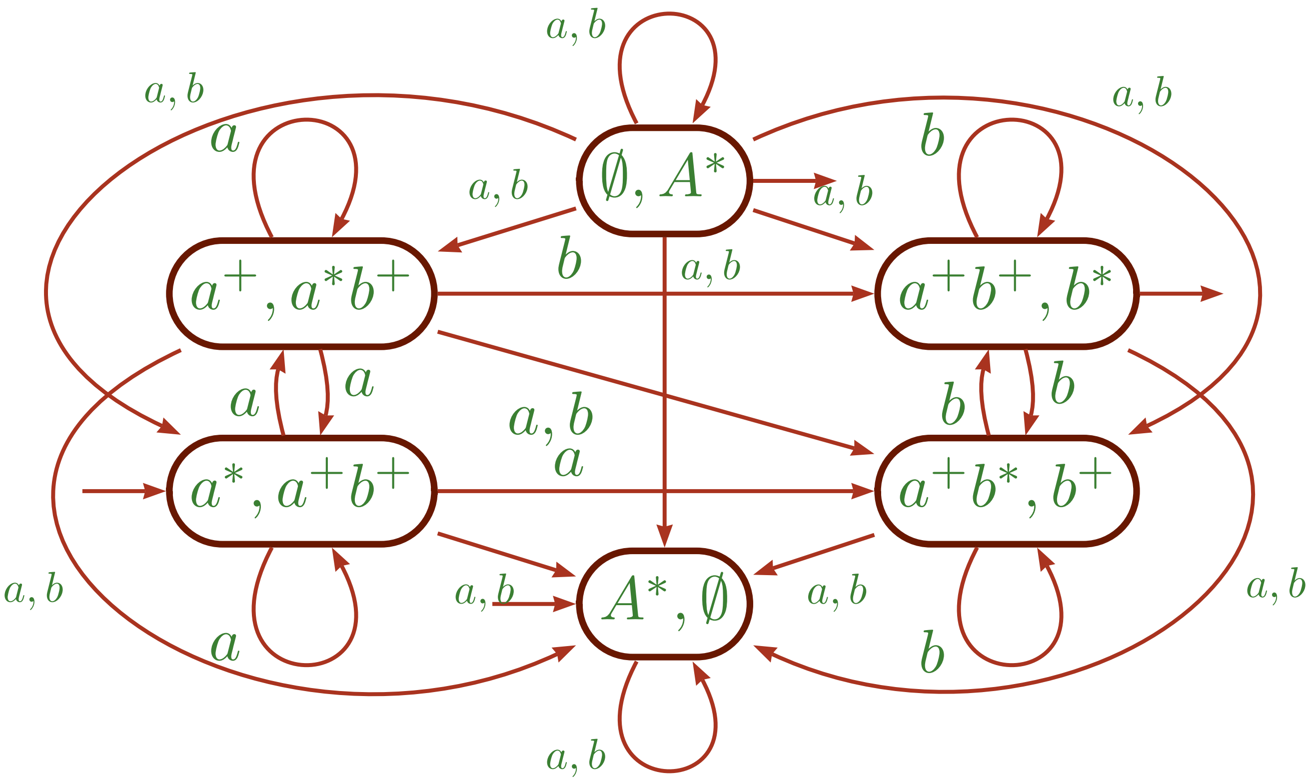
The universal automaton is an automaton s.t.
the pairs $(\mathsf{Past}(p), \mathsf{Fut}(p))$ are maximal w.r.t. L
$\longrightarrow$ Factorizations of the language (Conway,71)

Example : $L = a^+b^+$:

$$(a^+b^+, b^*) \qquad (a^+b^*, b^+) \qquad (A^*, \emptyset)$$

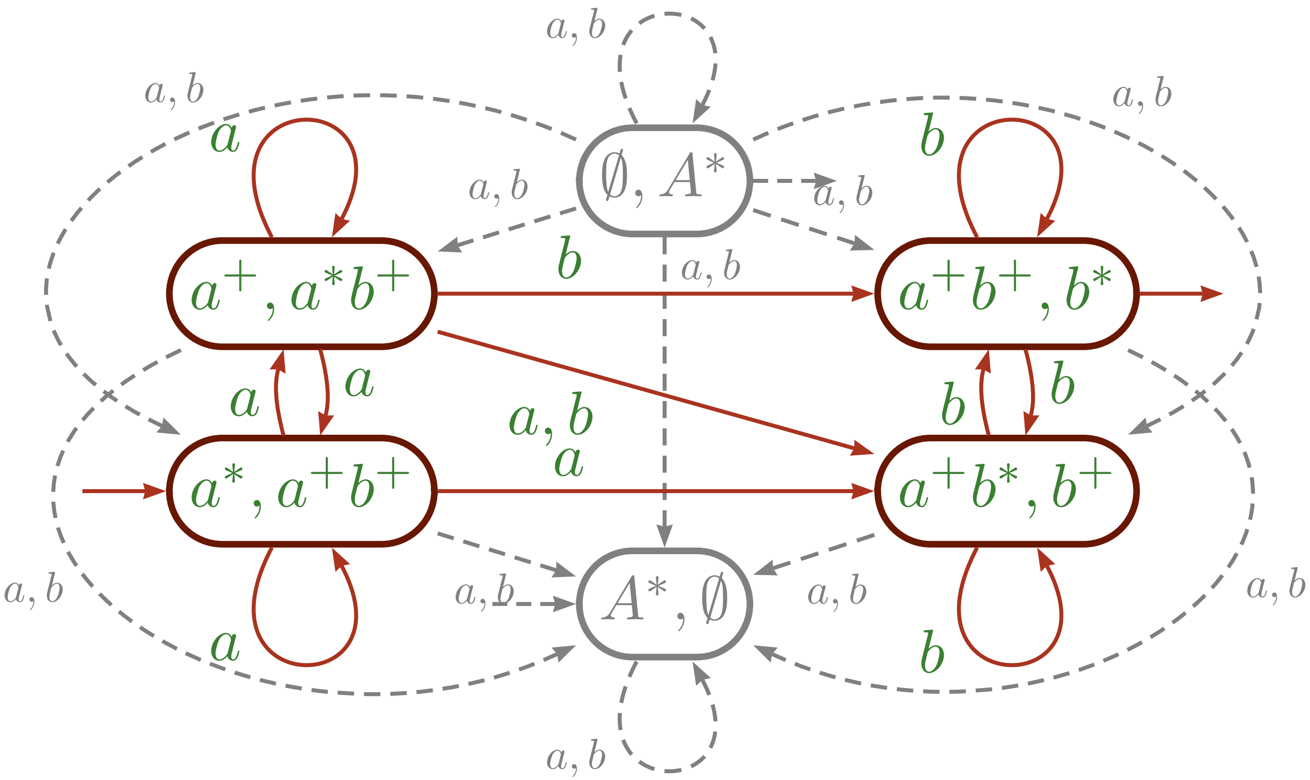
$$(a^*, a^+b^+) \qquad (a^+, a^*b^+) \qquad (\emptyset, A^*)$$

Universal automaton of L



SL - Sakarovitch, 2000

Universal automaton of L



SL - Sakarovitch, 2000

Universal automaton and reduction

Proposition : The universal automaton is reduced.

Universal automaton and reduction

Proposition : The universal automaton is reduced.

Proposition : If $L(\mathcal{A}) = L$, $\mathcal{A}$ can be reduced into a subautomaton of $\mathcal{U}_L$.

Universal automaton and reduction

Proposition : The universal automaton is reduced.

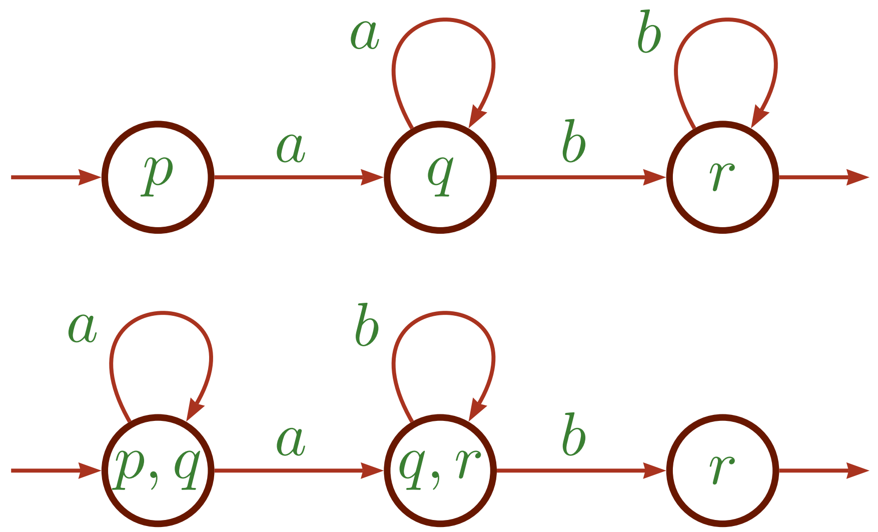
Proposition : If $L(\mathcal{A}) = L$, $\mathcal{A}$ can be reduced into a subautomaton of $\mathcal{U}_L$.

$$\begin{array}{ccc} \mathcal{A} & \longrightarrow & \mathcal{U}_L \\ p & \longmapsto & \left\{ \begin{array}{l} X_p = \{u \in A^* \mid u\mathsf{Fut}(p) \subseteq L\} \\ Y_p = \{v \in A^* \mid X_p v \subseteq L\} \end{array} \right. \end{array}$$

Construction of the universal automaton

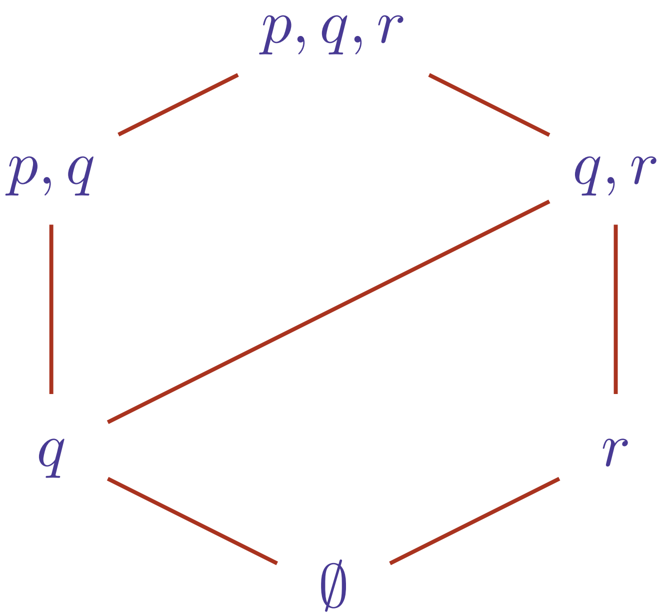
The co-determinized automaton of $\mathcal{A}$ is obtained from $\mathcal{A}$ by a “backward” determinization.

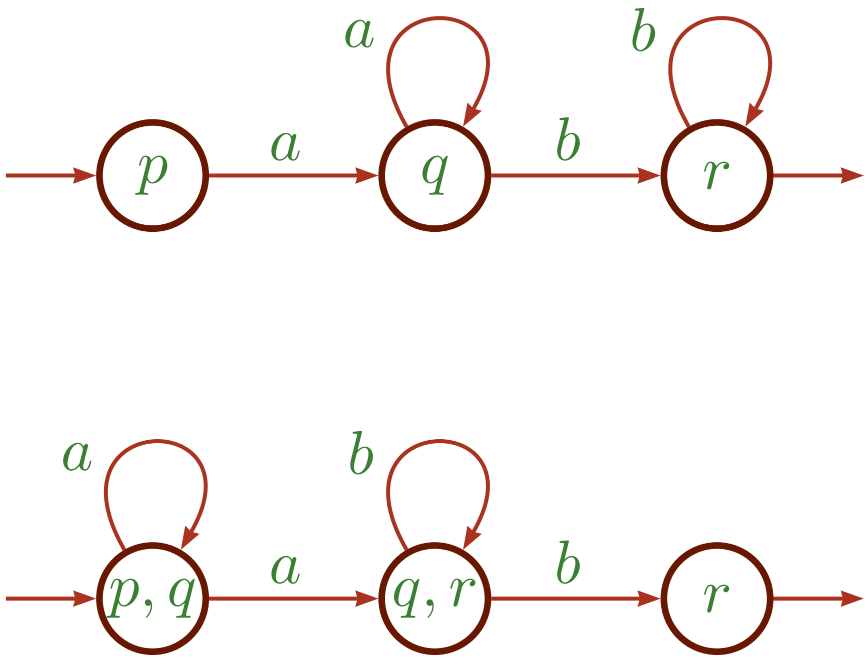
Example :



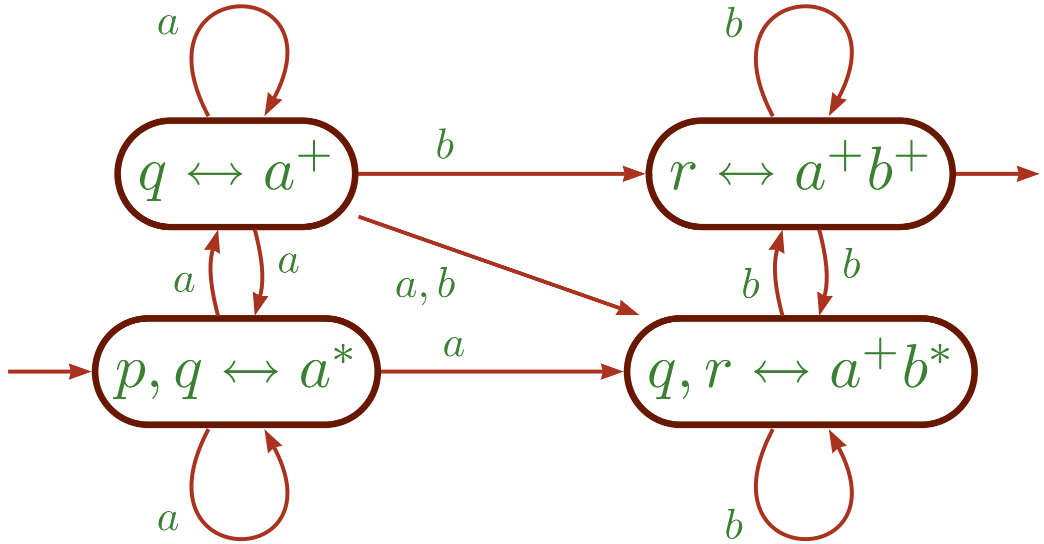
Construction of the universal automaton

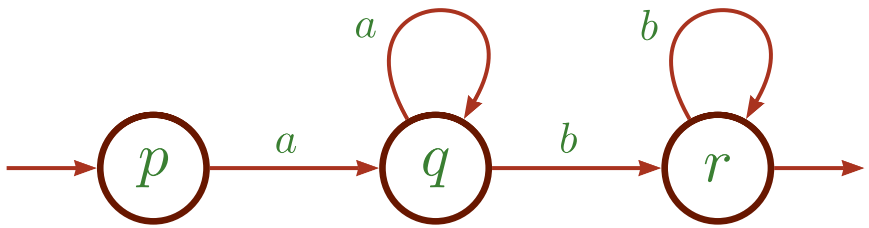
Theorem : (ICALP 2002) The set of states of the universal automaton of L is obtained from the minimal automaton $\mathcal{A}$ of L by computing the intersection of states of the co-determinized automaton of $\mathcal{A}$

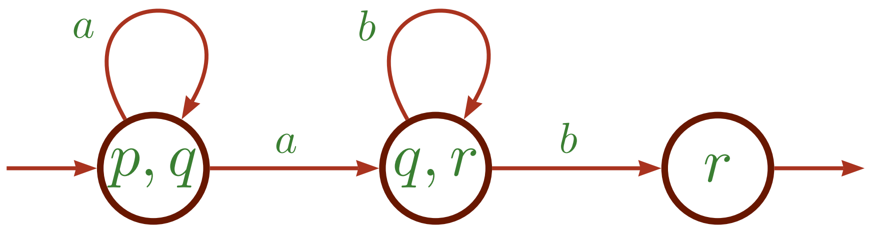




Construction of the universal automaton

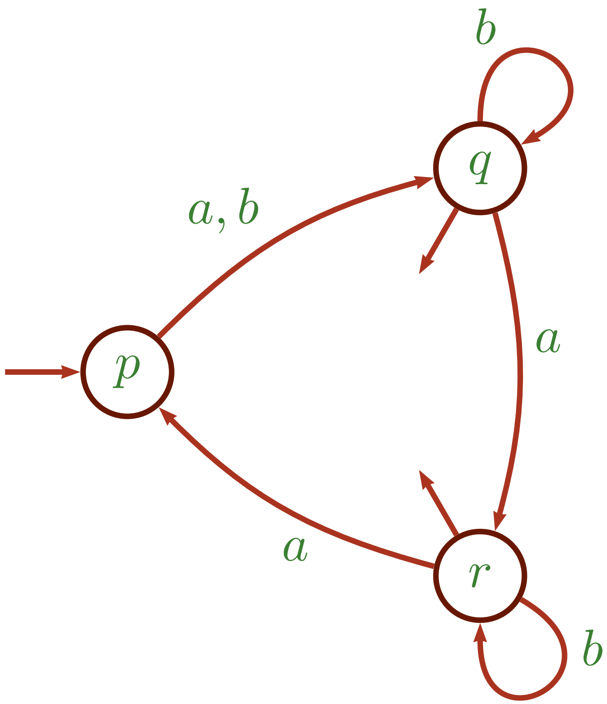






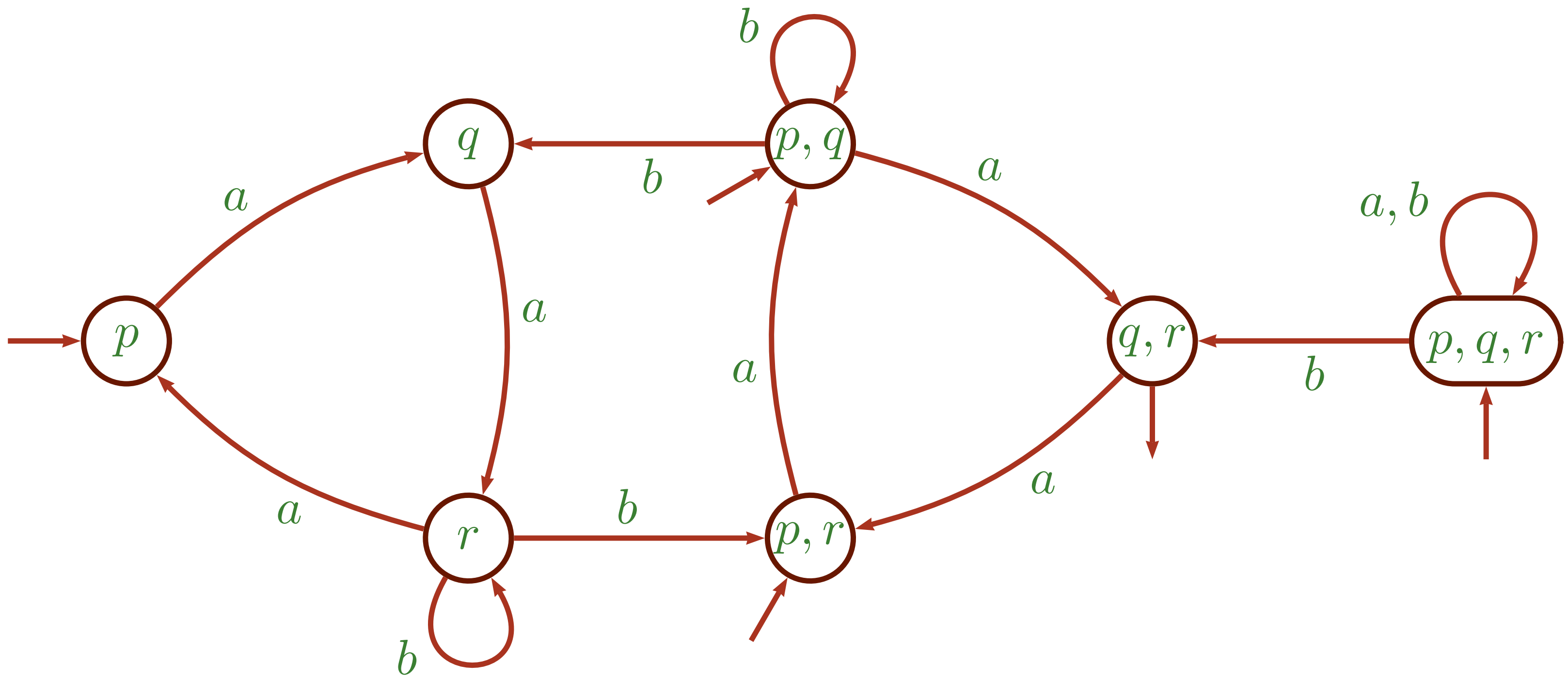
Bound w.r.t minimal DFA

Proposition : For every n, there exists a deterministic automaton with n states that recognizes a language for which the universal automaton has 2^n states.



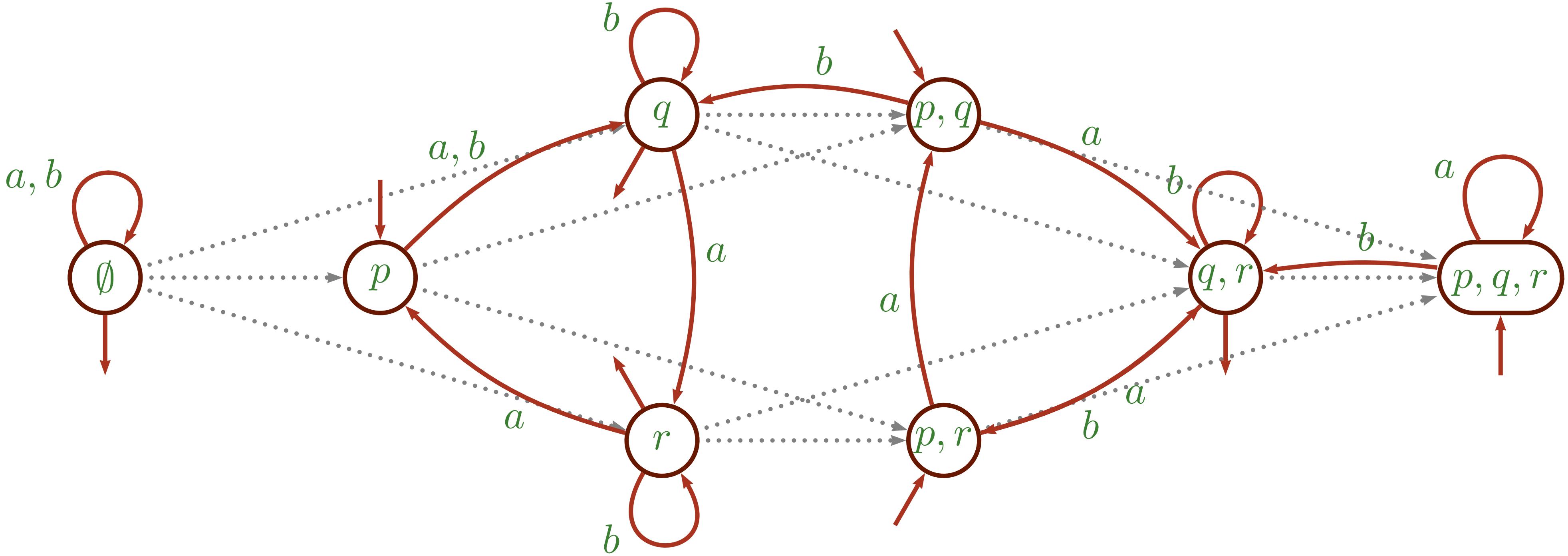
Bound w.r.t minimal DFA

Proposition : For every n, there exists a deterministic automaton with n states that recognizes a language for which the universal automaton has 2^n states.

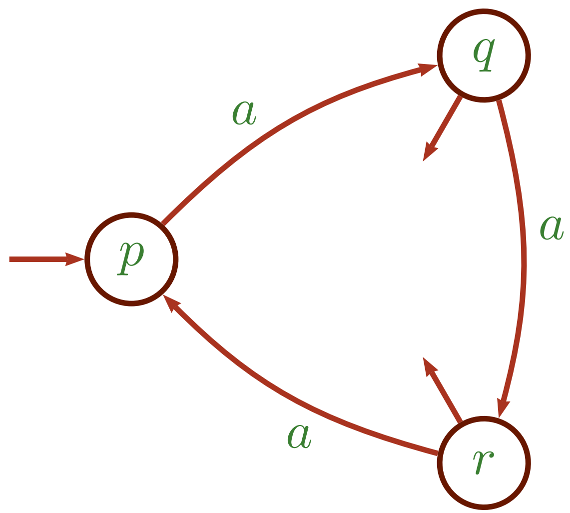


Bound w.r.t minimal DFA

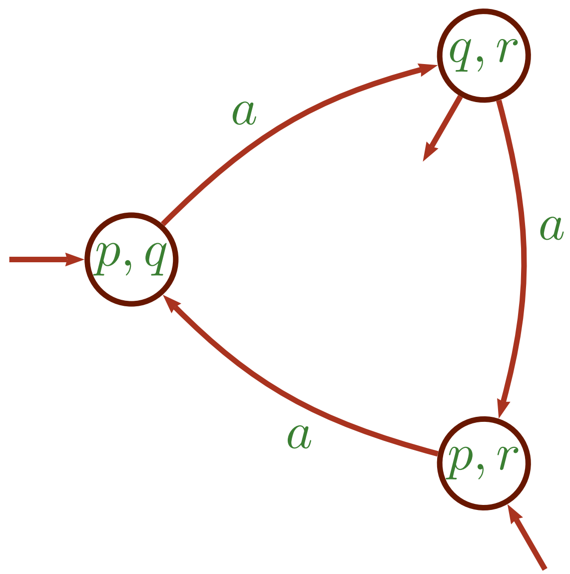
Proposition : For every n, there exists a deterministic automaton with n states that recognizes a language for which the universal automaton has 2^n states.



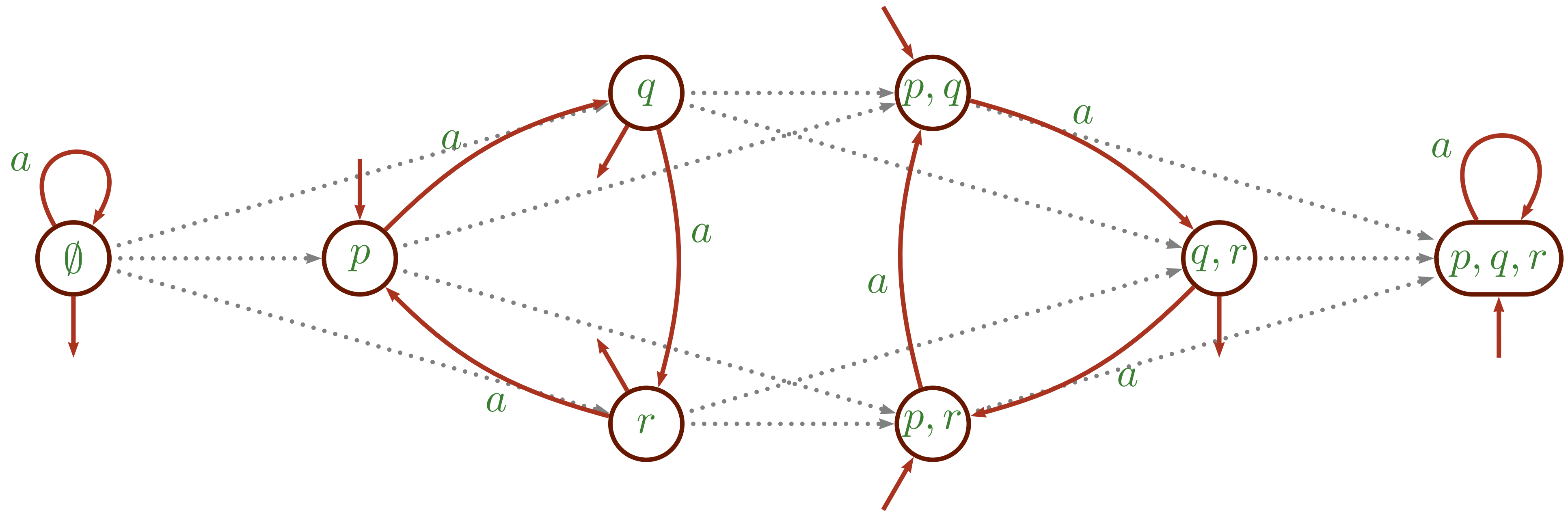
Unary alphabet



Unary alphabet



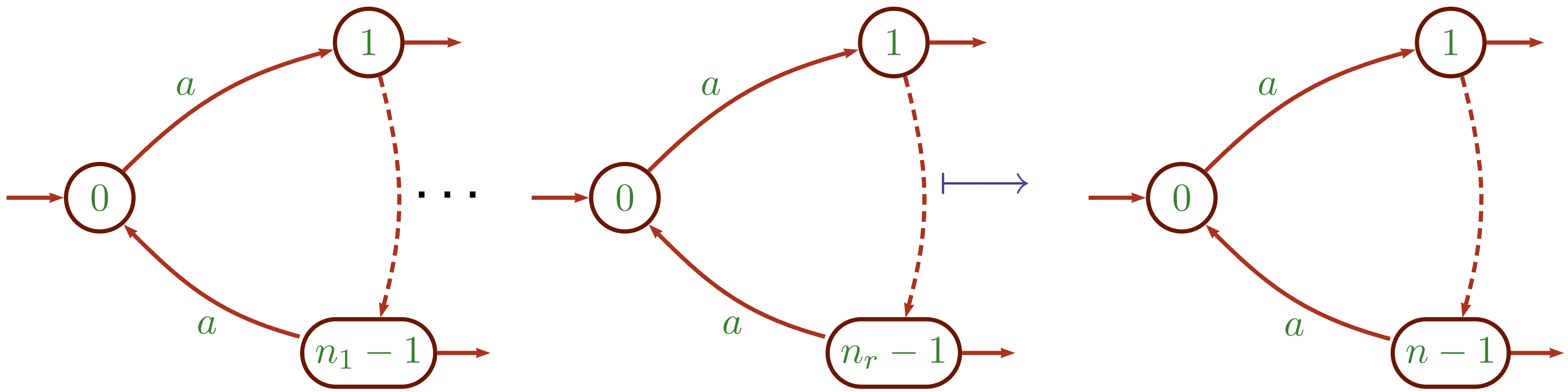
Unary alphabet



From a NFA : unary alphabet

Construction of the universal automaton : first determinization (at most $k = G(n)$ states) then construction of the universal automaton (2^k states).
$\longrightarrow$ Upper bound : $2^{G(n)}$

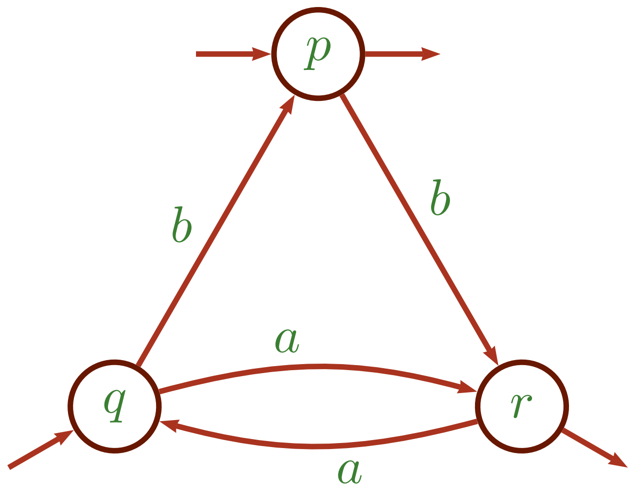
$$G(n) = \max\{\text{lcm}(n_1, \ldots, n_r) \mid \sum n_i = n\}$$



$\longrightarrow$ The upper bound is tight.

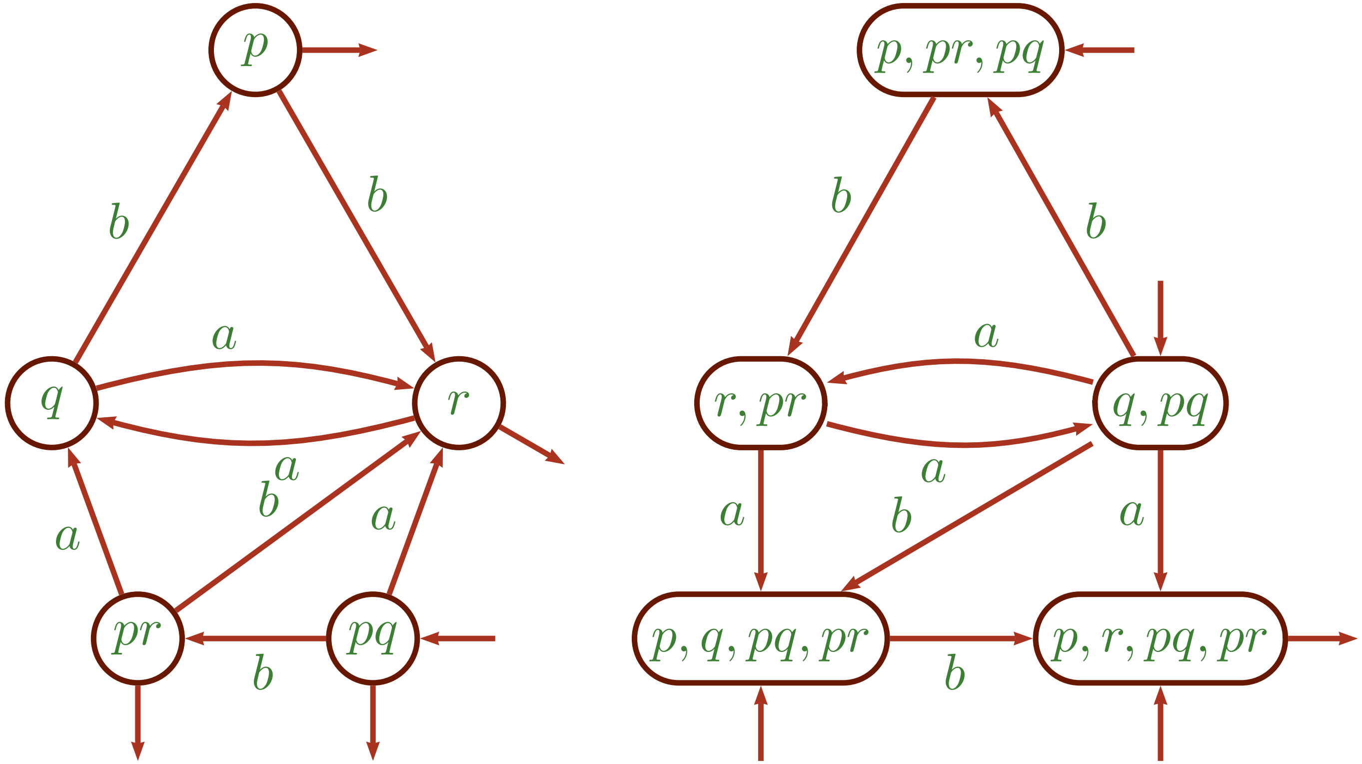
From a NFA

Construction of the universal automaton : first determinization (at most $k = 2^n - 1$ states) then construction of the universal automaton (2^k states).
$\longrightarrow$ Upper bound : 2^{2^n-1}



From a NFA

Construction of the universal automaton : first determinization (at most $k = 2^n - 1$ states) then construction of the universal automaton (2^k states).
$\longrightarrow$ Upper bound : $2^{2^n - 1}$



Characterization of the states

Proposition : $\mathcal{D}$ determinized of $\mathcal{A}$, $\mathcal{C}$ co-determinized of $\mathcal{D}$.
Let $P \subset R$ be two states of $\mathcal{D}$.
If P belongs to a state $\mathcal{X}$ of $\mathcal{C}$, so does R.
$\longrightarrow$ States of $\mathcal{C}$ are upperset.

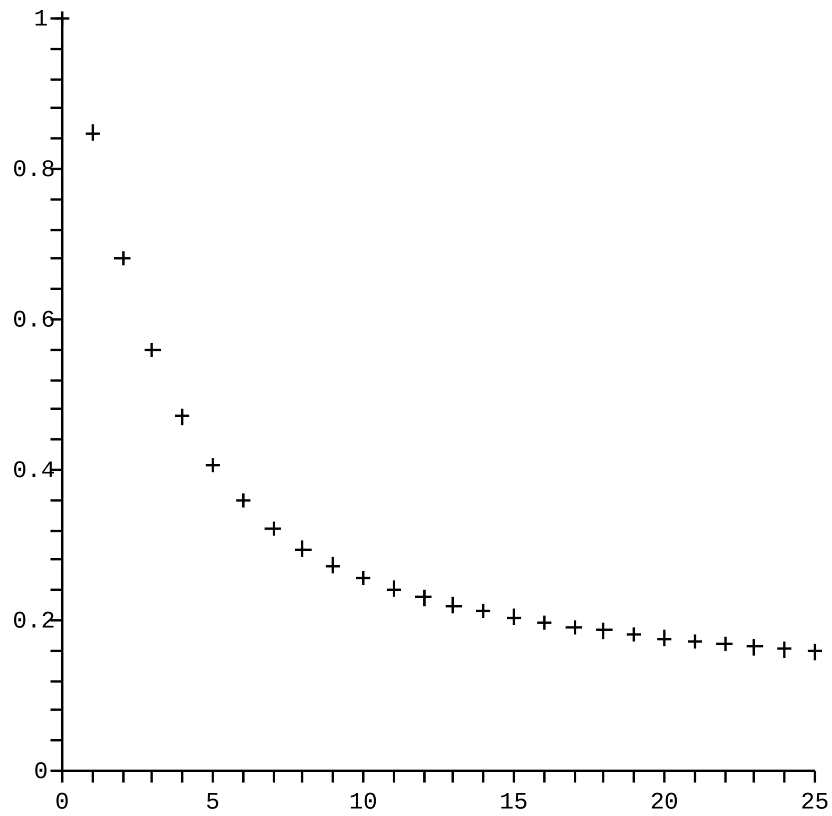
Lemma : The uppersets are closed under intersection.

Theorem : The states of the universal automaton are uppersets.

Theorem : The universal automaton of a language recognized by a NFA with n states has at most $D(n)$ states (n-th Dedekind number).

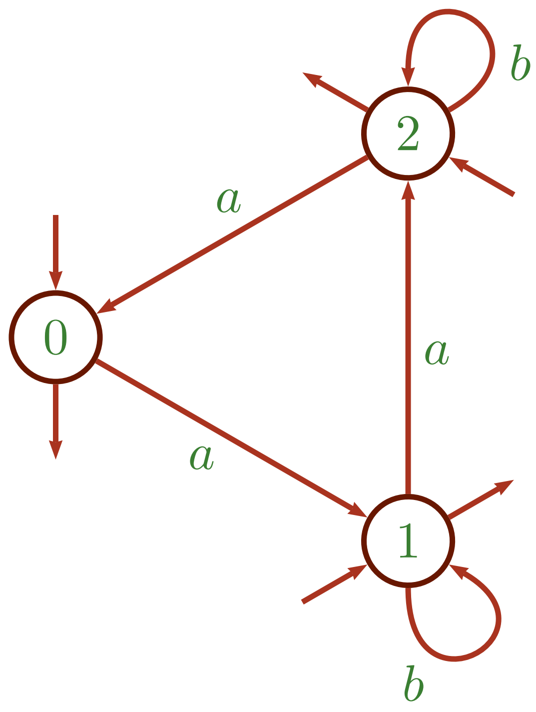
D(n) ?

n	$D(n)$
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788

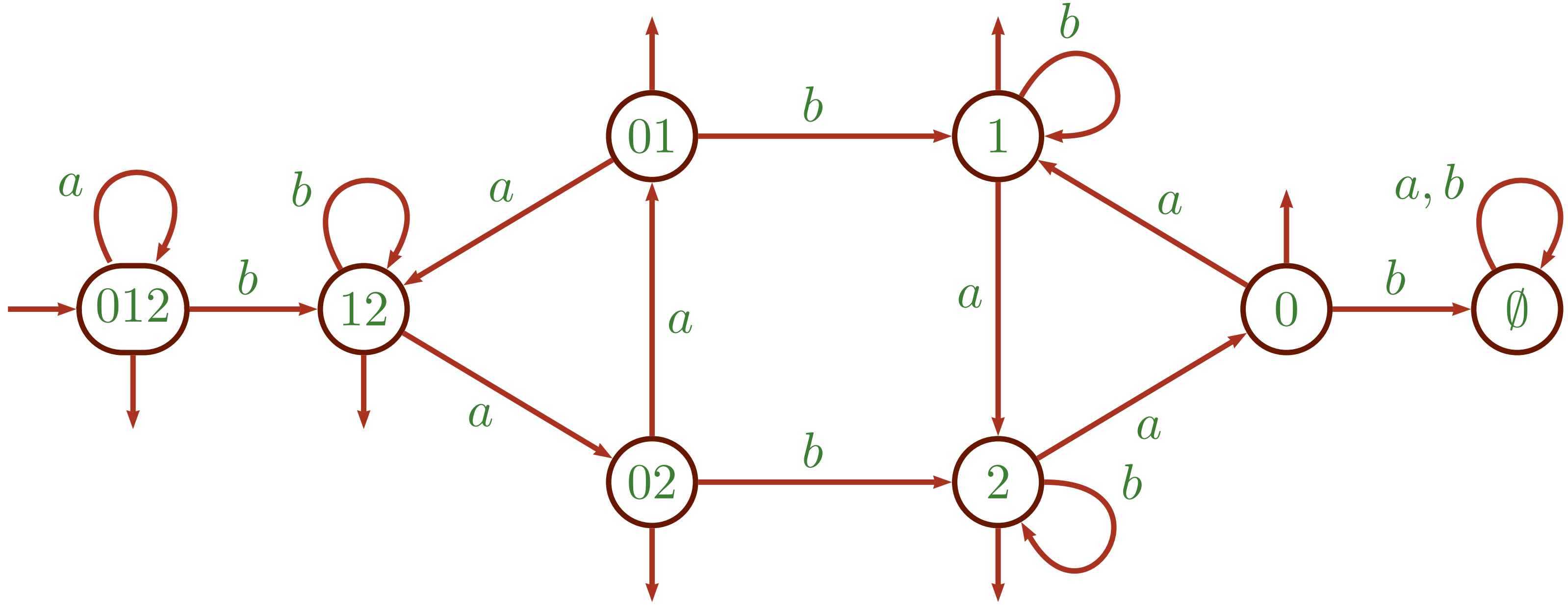


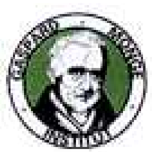
$$n \longmapsto \frac{\log_2 D(n)}{2^n}$$

Lower bound

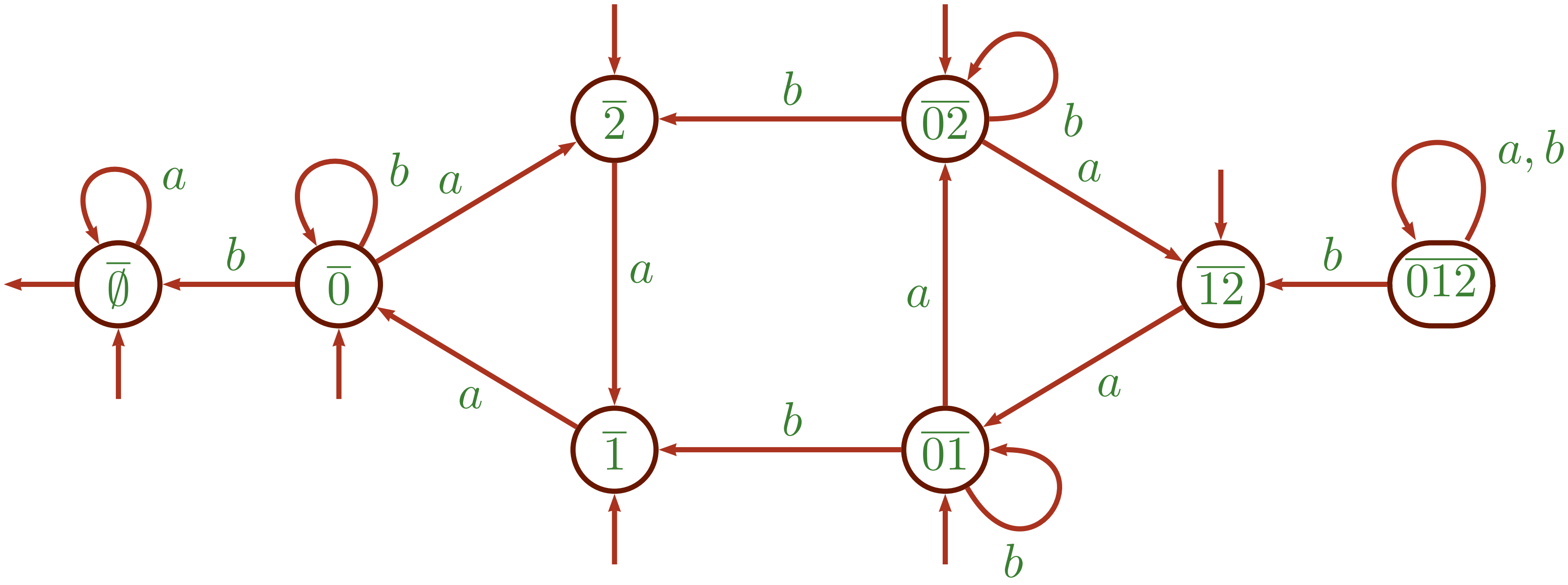


Lower bound

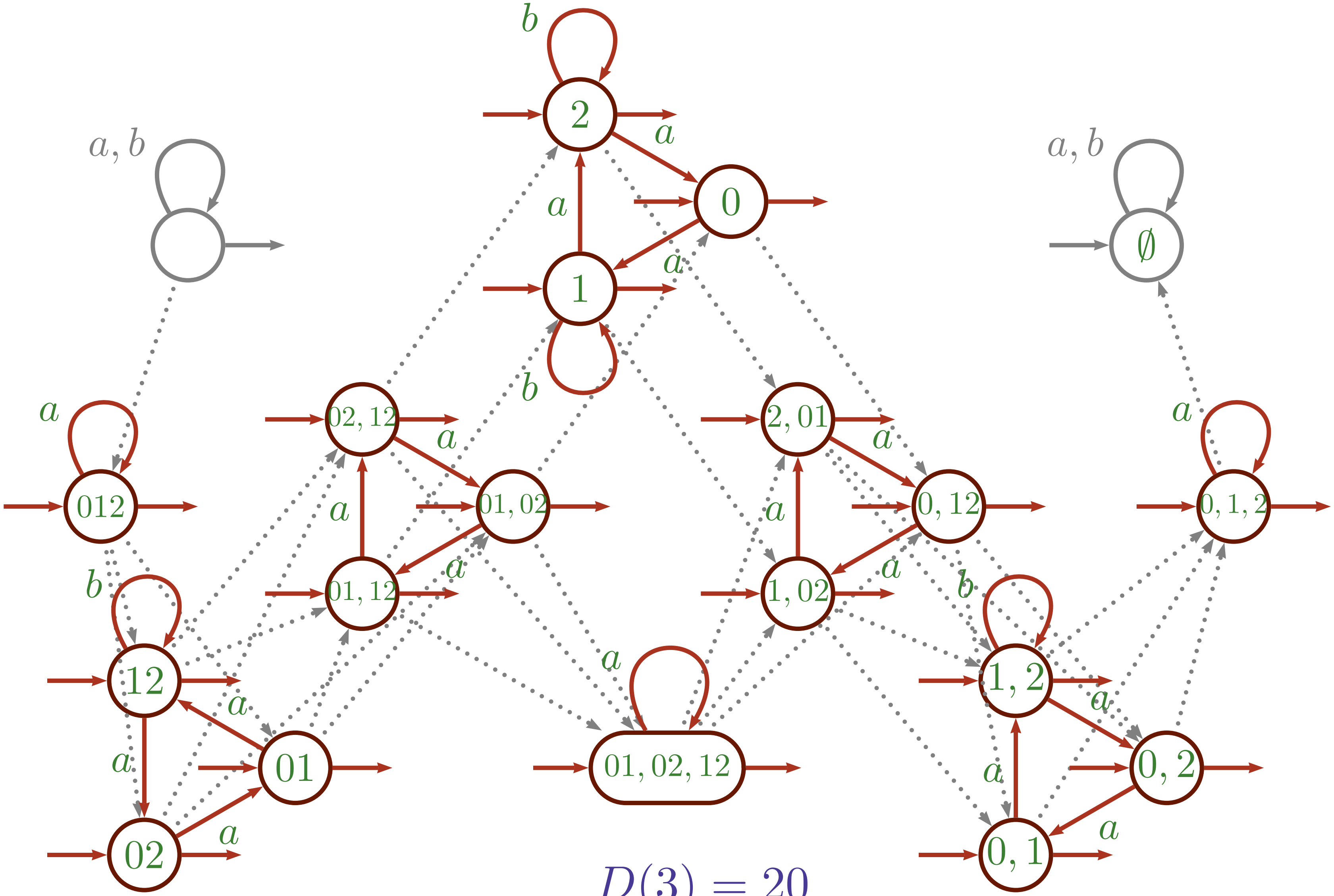




Lower bound



Lower bound



$D(3) = 20$

Thank you for your attention