Nondeterministic Moore automata and Brzozowski's algorithm

G. Castiglione, A. Restivo, M. Sciortino

University of Palermo

Workshop PRIN
Varese, 5-7 Settembre 2011

Summary

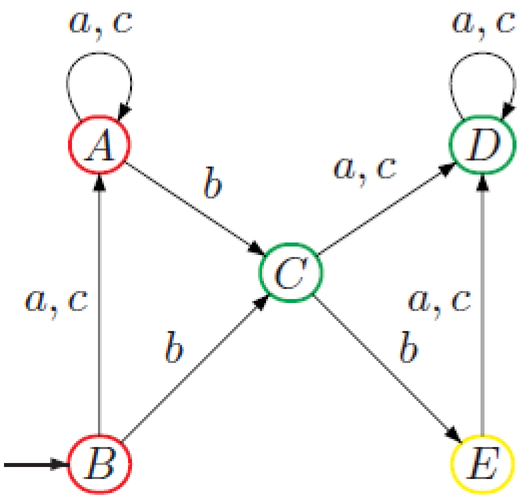
- A class of nondeterministic Moore automata
- Equivalence between nondeterministic and deterministic models
- A variant of Brzozowski's algorithm to minimize nondeterministic Moore automata.

Deterministic Moore Automaton (DMA)

- It was studied firstly by Edward Forrest Moore in 1956.
- It is a deterministic model.
- It is defined as a system $A = (\Sigma, \Gamma, Q, q_0, \delta, \lambda)$ where
  - Σ is the set of input symbols,
  - $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ is the set of output symbols (also called colors),
  - Q is the set of states,
  - q_0 is the initial state,
  - $\delta : Q \times \Sigma \to Q$ is the transition function,
  - $\lambda : Q \to \Gamma$ is the function that assigns an output (color) to the states of the automaton.

An Example of DMA

- For each input v in Σ* there exists a path in the automaton labeled by v.
- The **color** of a path π of *A* labeled by a word v is the color of the arrival state.
- Such a color is also called the **output** of v.



What about the nondeterministic model?

Non Deterministic Moore Automata

- In the literature, there exist several notions of nondeterminism for automata with output, and in particular for Moore automata, that have been introduced in specific areas and are often motivated by specific applications
  - system modeling
  - natural languages processing
  - system verification
  - machine learning
- Some references are:

[C. Cortes and M. Mohri, 2008

P. Garca, J. Ruiz, A. Cano, and G.I. Alvarez, 2005,

O. Kupferman and M.Y. Vardi, 1999,

M. Mohri, 1997, 2000

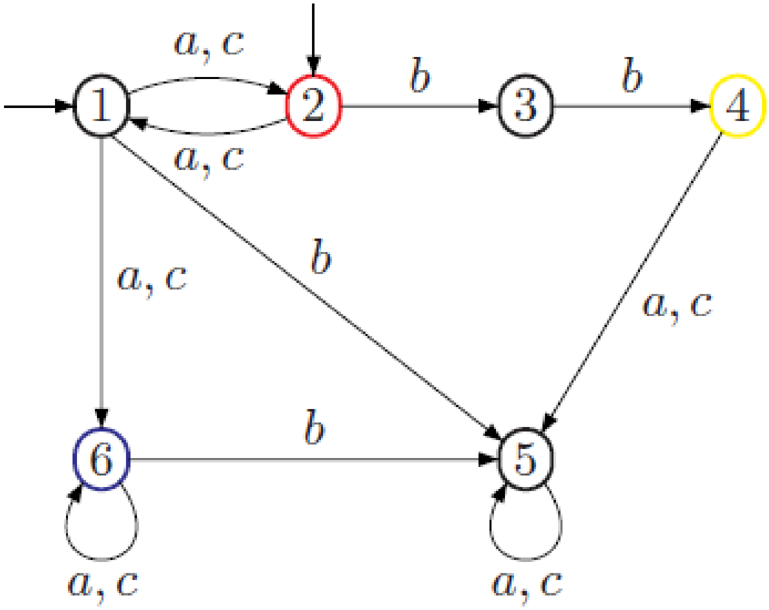
K. Takahashi, A. Fujiyoshi, and T. Kasai, 2003

B.W. Watson, 1995]

More formally...

- A nondeterministic Moore automaton (denoted by NMA) is a system $A = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ where
  - Σ is the set of input symbols,
  - $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ is the set of output symbols (also called colors),
  - Q is the set of states,
  - $I \subseteq Q$ is the set of initial states,
  - $\Delta \subseteq Q \times \Sigma \times Q$ is the set of the transitions of A,
  - $\lambda : Q \to \Gamma$ is a partial output function that assigns a color to some states of the automaton.
- *Note that in such a model both the input symbols and the output symbols could not be defined for all the transition or all the states, respectively.*

An example of NMA



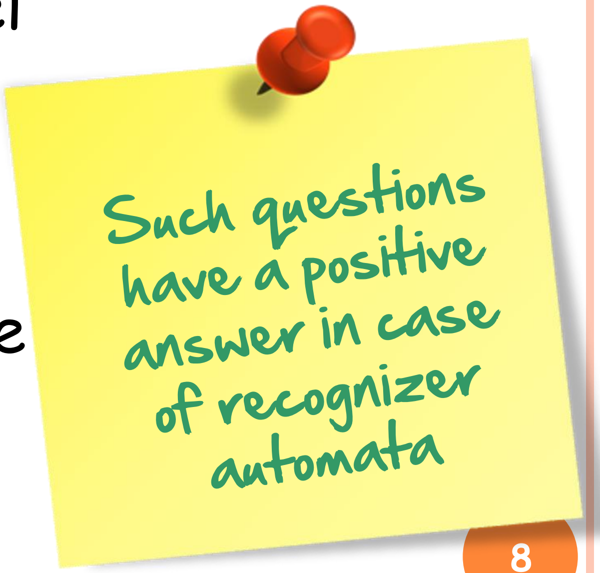
Some definitions:

- A word v is **applicable** for A if there exists at least a path π labeled by v starting from an initial state.
- To each applicable word *v* of A we could associate many paths of different color labeled by *v*.

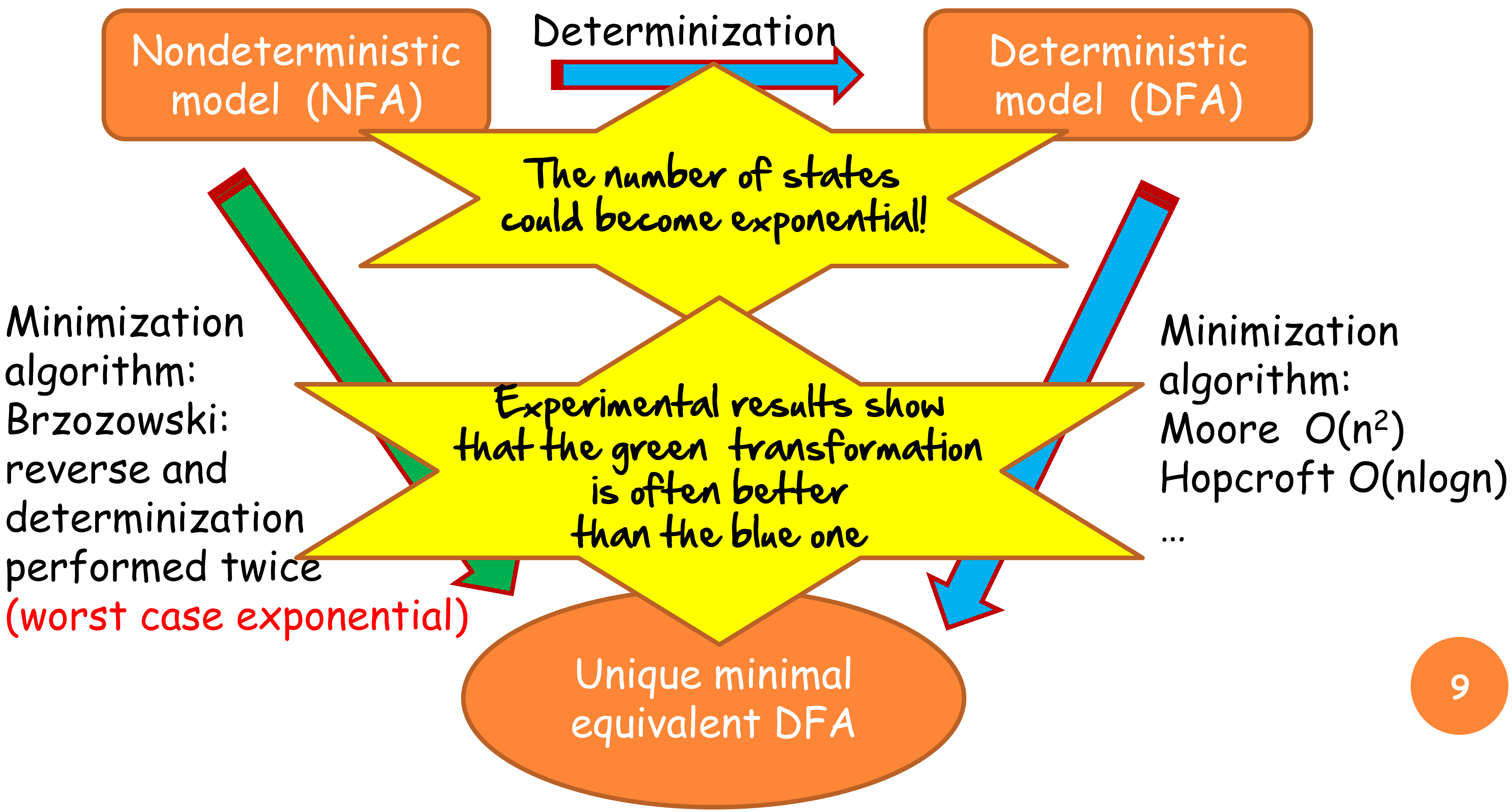
- **It is nondeterministic on the paths**
- **It is nondeterministic on the outputs**

Motivations

- We are interested in deterministic finite-state automata with output, viewed as computers of functions, not as recognizers of languages.
- A Moore automaton is a simple model of deterministic automata with output.
- Does there exist a nondeterministic model able to compute succinctly the same function?
- Is it possible to efficiently simulate such a nondeterministic model?



DETERMINISM VS NON DETERMINISM IN RECOGNIZER AUTOMATA

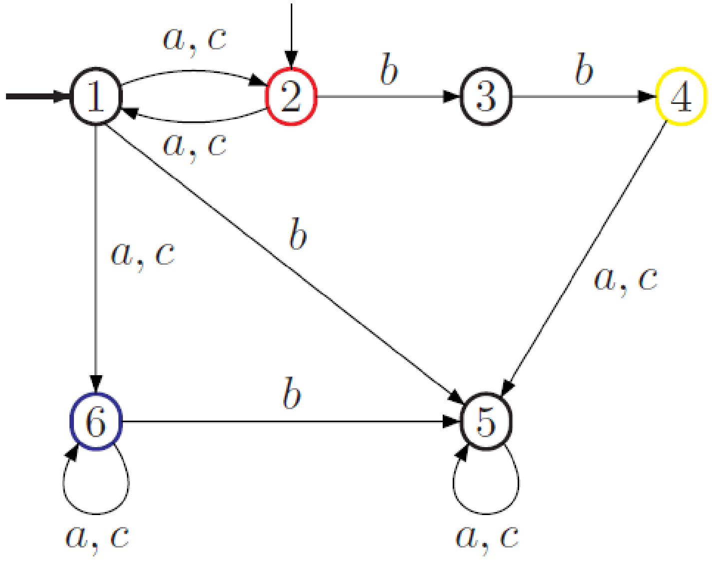


Our Goals

Searching for an appropriate nondeterministic model

Minimizing such a model by a variant of Brzozowski's algorithm

Which nondeterministic model?



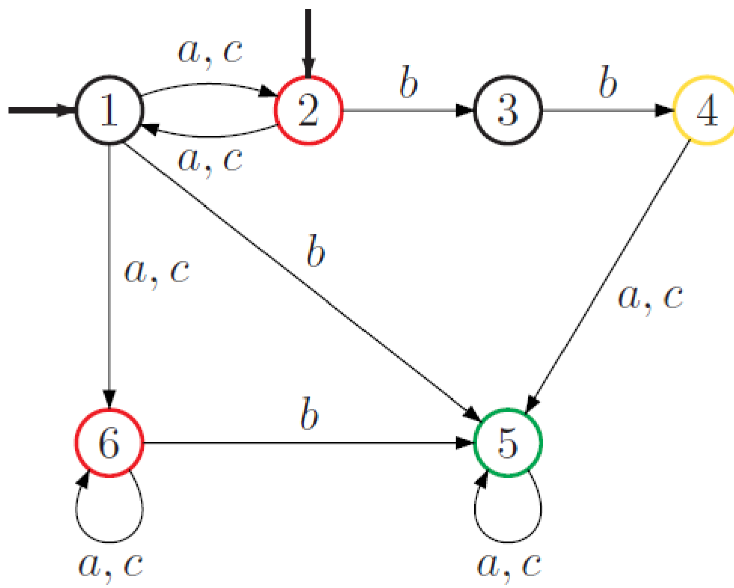
The nondeterministic Moore automaton

- is nondeterministic in respect of the paths
- is nondeterministic in respect of the outputs
- some inputs could not produce any output

We need a "determinism" on the output

Coherent NMA

- The NMA $\mathcal{A} = (\Sigma, \Gamma, Q, I, \Delta, \lambda)$ is **coherent** if
  - for each applicable word v there exists at least a colored path labeled by v
  - all the colored paths associated to v have the same color.

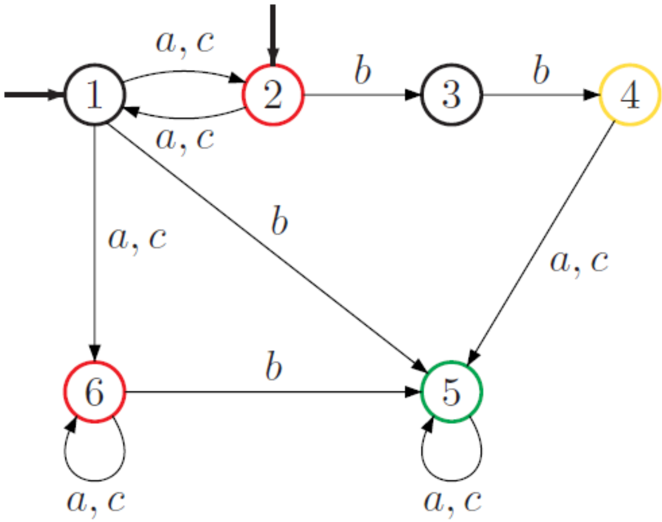


Properties of a coherent NMA

- In a coherent NMA at least one initial state must be colored and all colored initial states have the same color.

- A coherent NMA implicitly defines a partial function
  $f_A : \Sigma^* \to \Gamma$
  that to each applicable word v of A associates the color of a colored path labeled by v.

- The domain of the function is the language L(A). Such a language is partitioned into disjoint "colored" languages



$L(A) = (a + c)^*(b + bb)(a + c)^* + \epsilon$

Equivalence between coherent NMAs

- We say that two coherent NMAs A, B are **equivalent** if they compute the same functions f_A and f_B *(this implies that the induced partitions of the domain are the same)*
- A coherent NMA is **minimal** if it has minimal number of states among its equivalent ones.
  As in the case of nondeterministic recognizer automata (i.e recognizing regular languages), such a minimal nondeterministic model could be not unique.
- A DMA is a particular coherent NMA
- **There exists a unique (up to isomorphism) minimal DMA equivalent to a given DMA.**

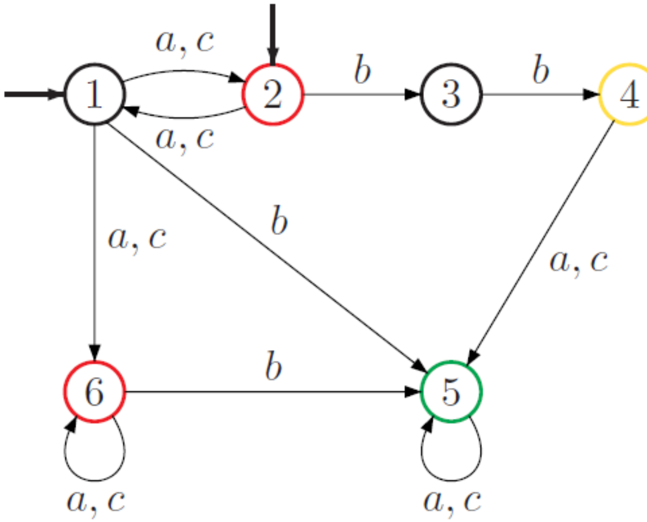
Determinization of a coherent NMA

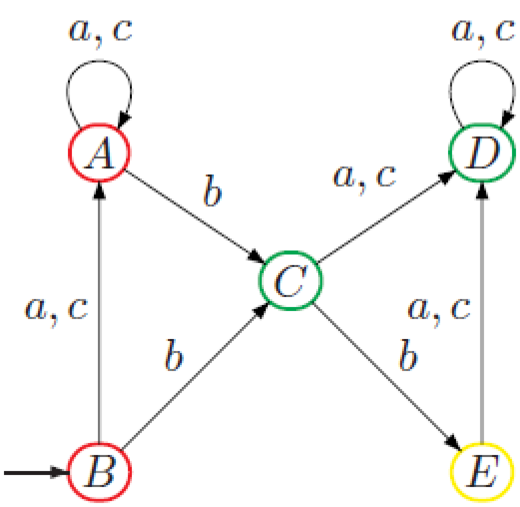
Given a NMA A we can associate the labeled colored state graph G that is obtained from A by neglecting the information about the initial states.

- It consists in a subset construction that takes as input the colored labeled graph G and the set I of initial states.
- It produces a graph in which the states are subsets of states of G accessible by the elements of I [Watson 95]
- Each subset contains at least a colored state and it cannot contain states of different color, so the following coloring function can be defined:

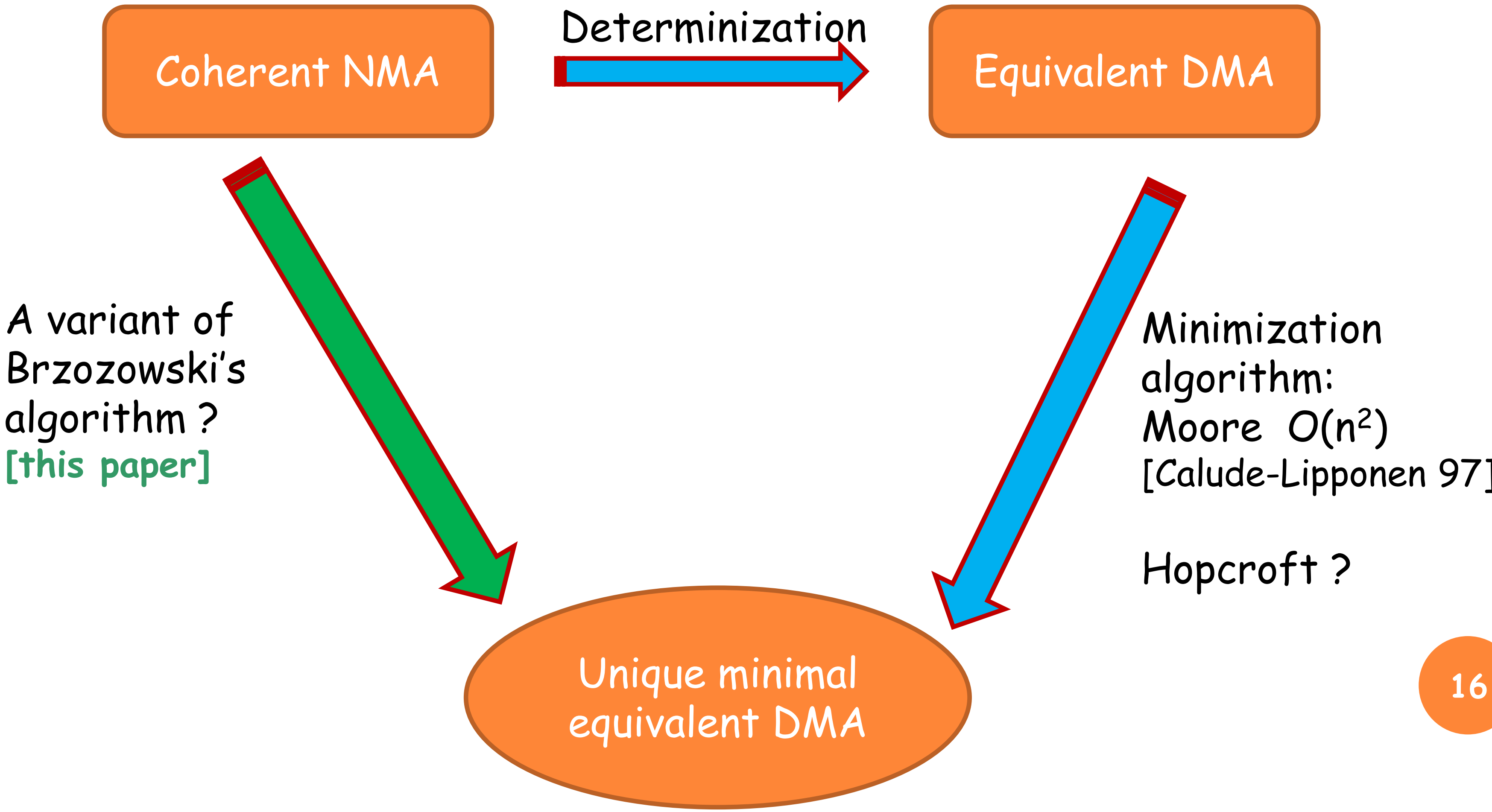
  $\lambda(p) = \gamma_i$

  if γ_i is the color of the colored states of p.
- The automaton is an equivalent DMA.



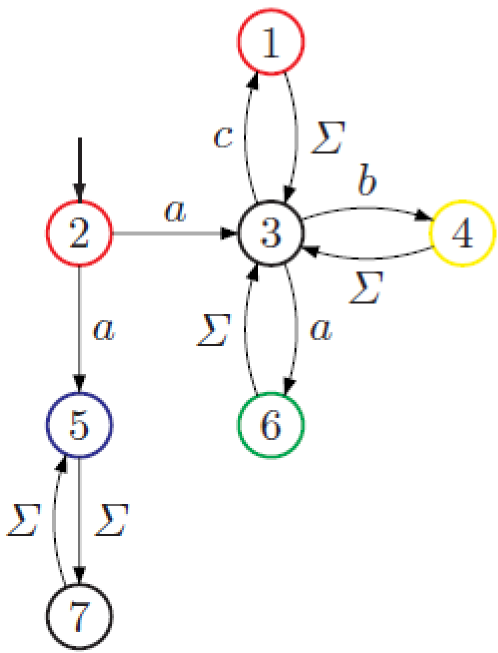


Determinism vs Non Determinism in Coherent Moore Automata

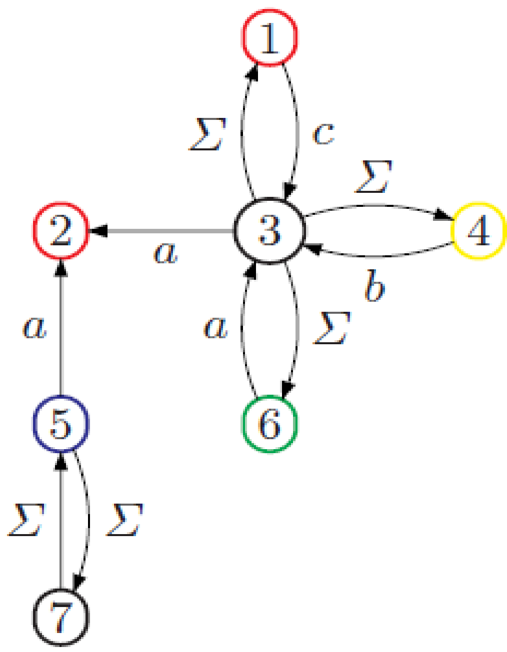


THE ALGORITHM
FIRST STEP: REVERSE

- INPUT:



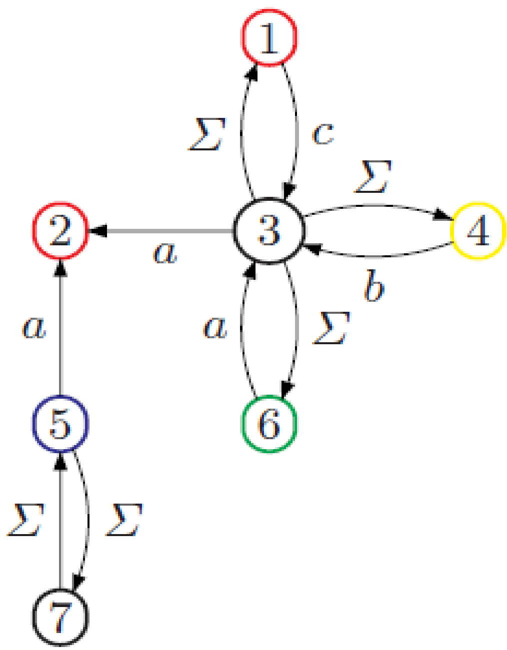
- OUTPUT



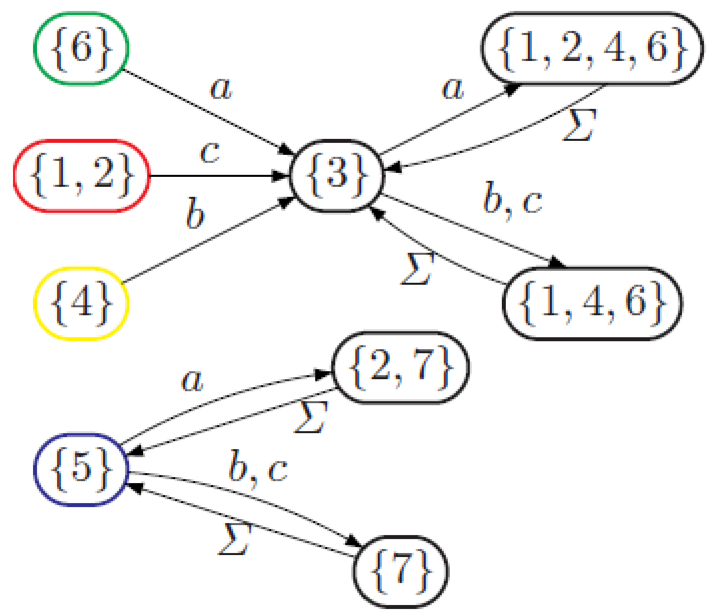
The transitions are inverted and the coloring is mantained.

THE ALGORITHM
SECOND STEP: SUBSET CONSTRUCTION

- INPUT:



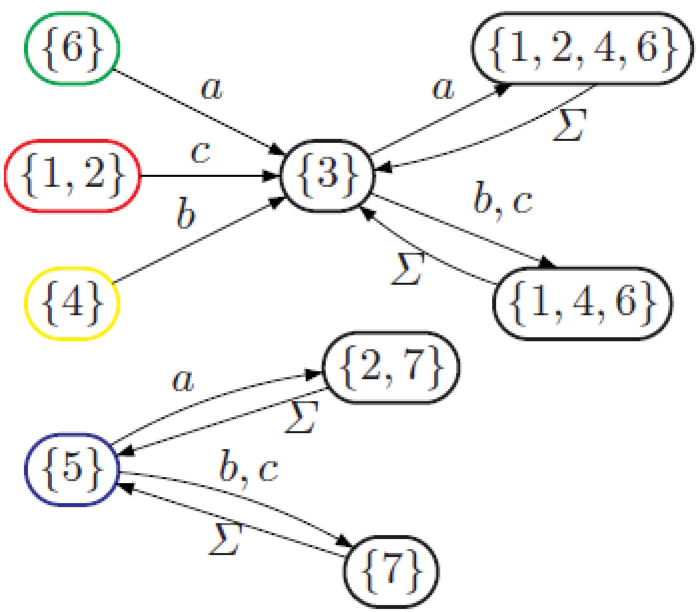
- OUTPUT



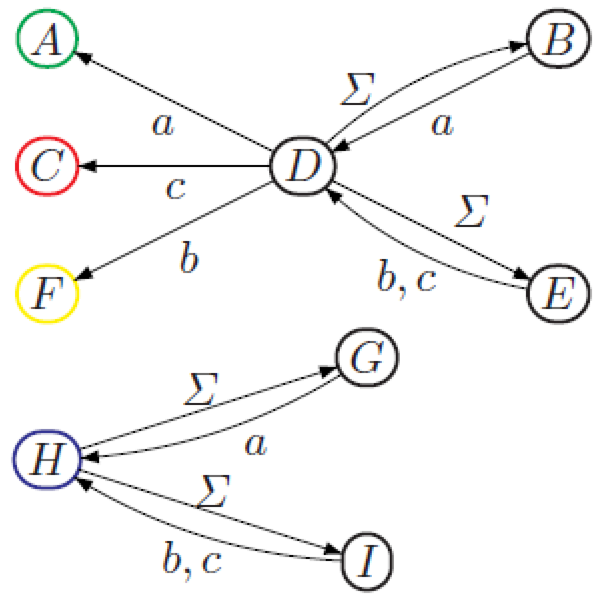
- It is a subset construction starting from the set of the subsets of the states having the same color.
- Each initial state has a color induced by its elements.

THE ALGORITHM
THIRD STEP: REVERSE

- INPUT:



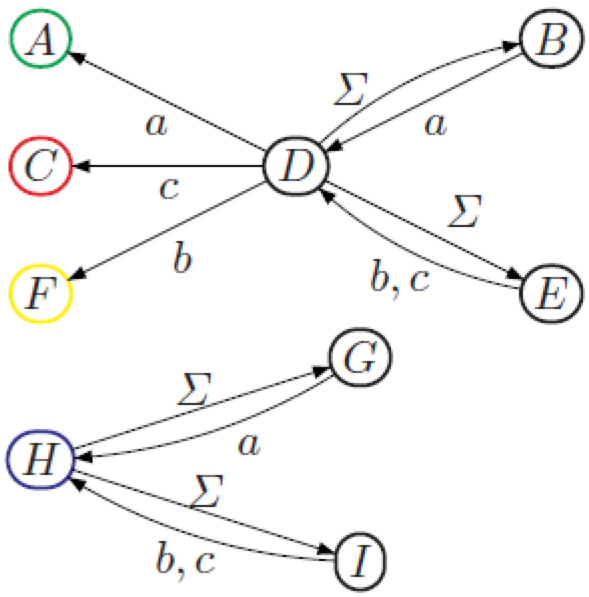
- OUTPUT



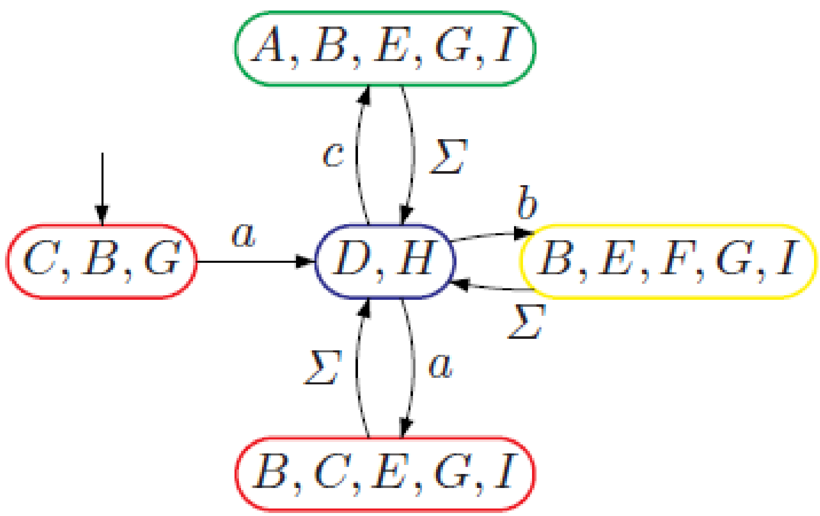
- It is the same trasformation performed at the first step.
- We need to keep the sets containing the initial states. In the example such sets are C, B, G.

THE ALGORITHM
FOURTH STEP: SUBSET CONSTRUCTION

- INPUT:



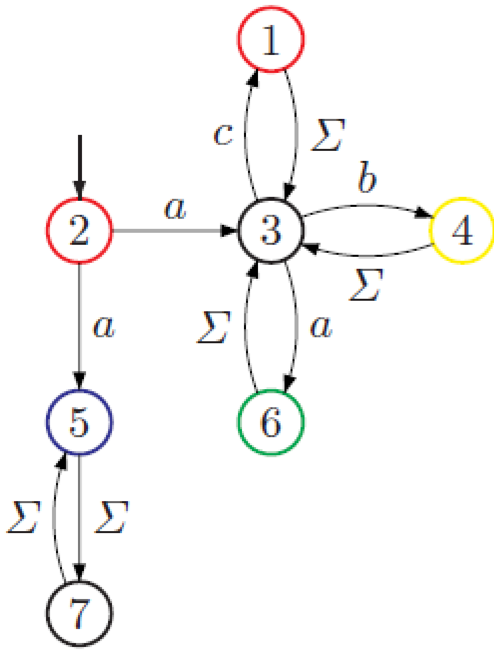
- OUTPUT



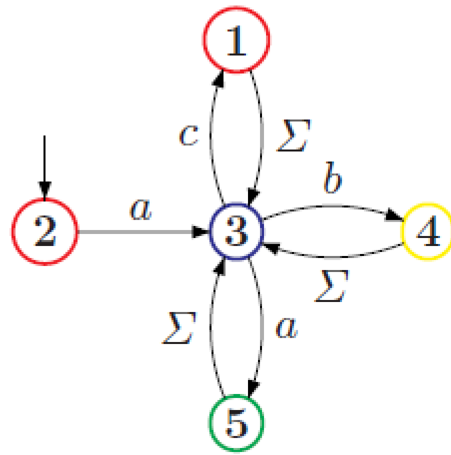
- It is a subset construction starting from the set {C,B,G}
- **Fundamental property:** each subset contains at least a colored state and it cannot contain states of different color, so the coloring function is well defined.

Minimization of a NMA

- Coherent NMA A



- DMA A_M obtained by the variant of Brzozowski's algorithm



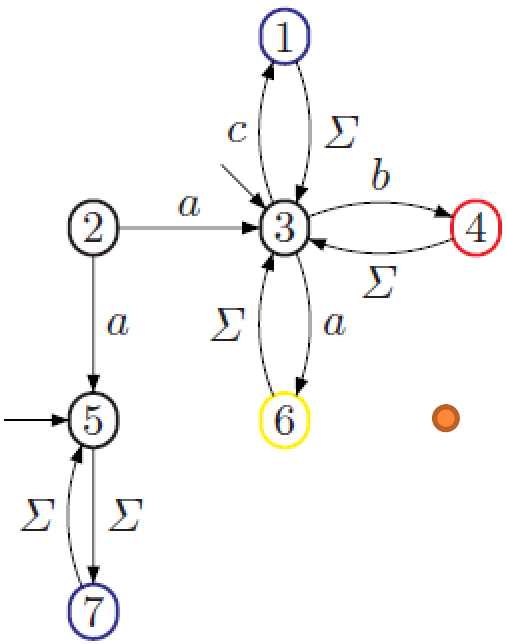
The language L(A) is partitioned into:
$L_{red} = \{w \in a\Sigma^*c \mid |w| \text{ is even}\} \cup \{\epsilon\}$
$L_{green} = \{w \in a\Sigma^*a \mid |w| \text{ is even}\}$
$L_{yellow} = \{w \in a\Sigma^*b \mid |w| \text{ is even}\}$
$L_{blue} = \{w \in a\Sigma^* \mid |w| \text{ is odd}\}$

- **Theorem**: *The deterministic Moore automaton A_M is minimal.*
- **Theorem**: *The automata A and A_M are equivalent.*

Coherent NMAs generalize the self-verifying automata

- Self verifying automata are nondeterministic recognizer automata in which the computation paths can give three types of answers: yes, no and I do not know.
- Moreover for each input string, at least one path must give answer yes or no and for the same string two paths cannot give contradictory answers.
- They have been introduced in 1997 by P. Duris, J. Hromkovic, J. Rolim, G. Schnitger
- Some papers on the topic:
  - J. Hromkovic, G. Schnitger, 2001, 2003.
  - G. Jirásková, G. Pighizzini, 2011
- This kind of nondeterminism was mainly considered in connection with randomized Las Vegas computations, but interesting also *per se.*

IF THE AUTOMATON IS NOT COHERENT...



IDEA: One could define an associative and commutative operator ⊕ on the colors of the paths with the same label

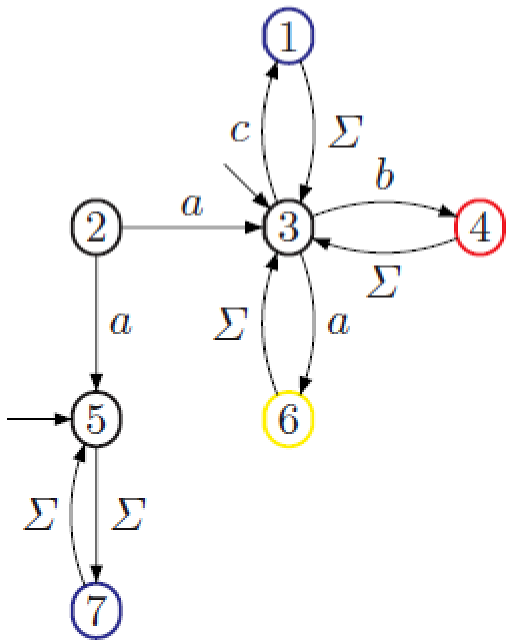
Let we use for instance the following table:
Red ⊕ Blue = Blue
Yellow ⊕ Blue = Yellow
Red ⊕ Yellow = Yellow

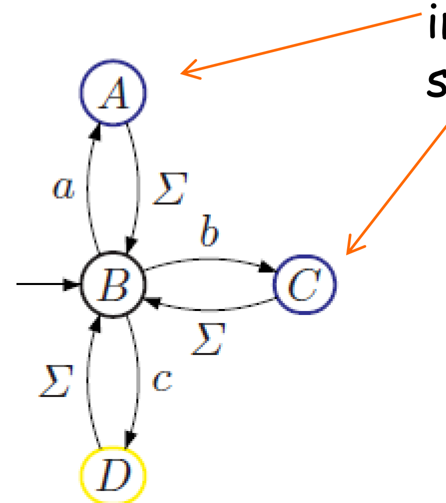
Note that the operator ⊕ is idempotent and the absence of color is treated as a neutral element $\perp$

- By using such an operator the automaton becomes able to compute functions

DOES THE ALGORITHM WORK EVEN IN THIS CASE?

- The answer is negative!





indistinguishable states

The automaton is an equivalent DMA, but it is not minimal.


THANKS FOR YOUR ATTENTION