A Disambiguation Algorithm for Finite Automata and Functional Transducers

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Many thanks to Cyril Allauzen for the presentation!

Motivation

- Optimization algorithms to increase efficiency of use.
- Determinization: substantial impact in text and speech and bioinformatics applications. But,
  - some finite-state transducers or weighted automata are not determinizable.
  - in some cases, the result is prohibitively large.
- Disambiguation:
  - applies to a broader set of finite-state transducers or weighted automata.
  - result can be exponentially smaller.

Disambiguation

- Unambiguous automata or transducers: no two accepting paths have the same (input) label.
- Disambiguation: algorithm returning an unambiguous automaton or transducer equivalent to the input.

Previous Work

- Disambiguation using determinization: standard determinization of finite automata, or transducer determinization (MM, 1997).
  - only for determinizable transducers.
- Construction of Schützenberger (1976), see also discussion by Sakarovitch (1998), and description in introductory chapter of Roche and Schabes (1997).
  - works for functional transducers.

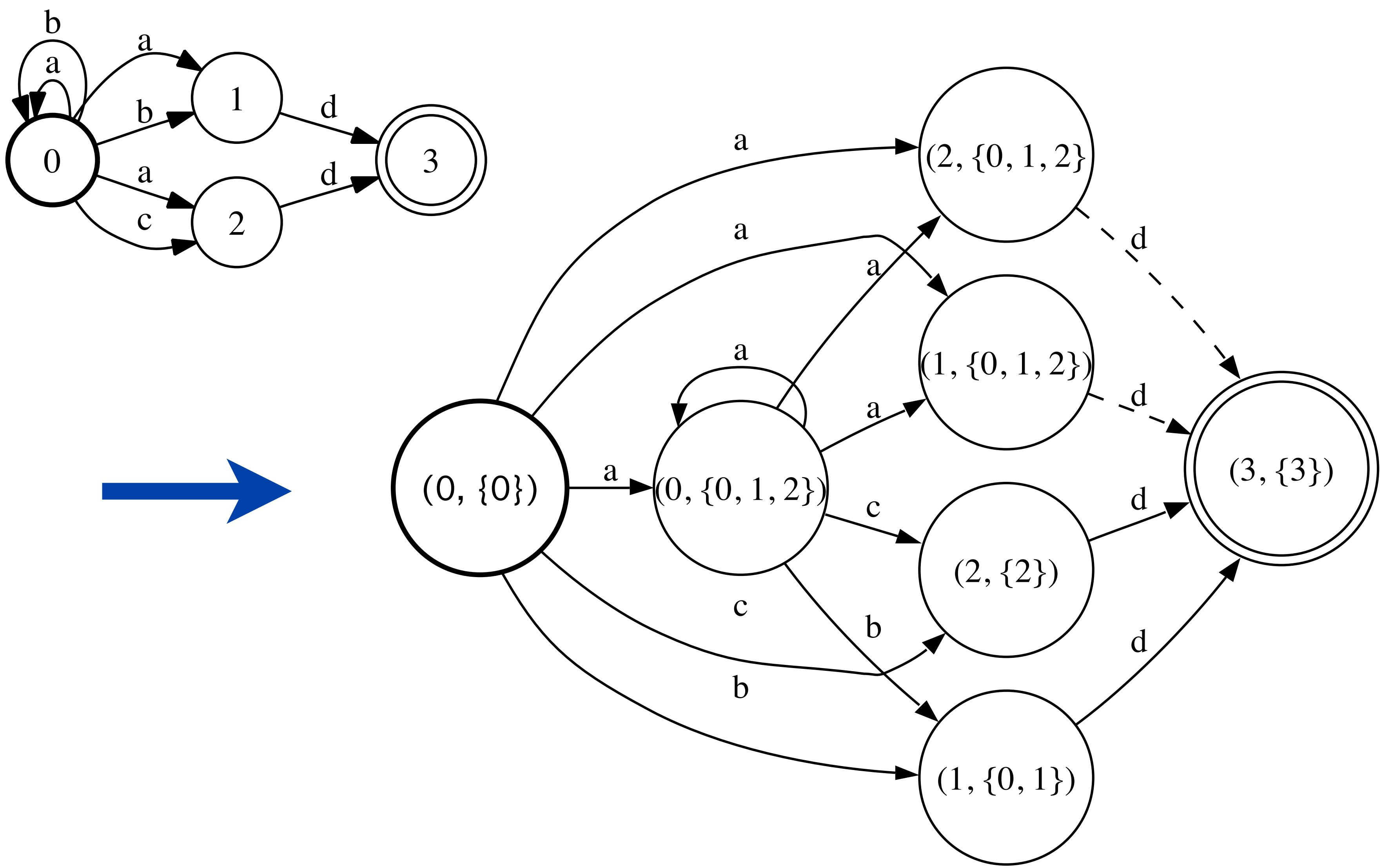
Outline

- Schützenberger's construction.
- New disambiguation algorithm.
- Extension to automata with ε-transitions.
- Disambiguation of functional transducers.

Schützenberger's Construction

- Overview: transducer T, with corresponding input automaton A.
  - compute $det(A)$.
  - compose $det(A) \circ T$, with the following rule: if two states in composition (p, s) and (q, s) admit a transition with the same label to the same state, keep only one of these transitions. Idem for finality.
  - $\longrightarrow$ the construction already requires determinization!

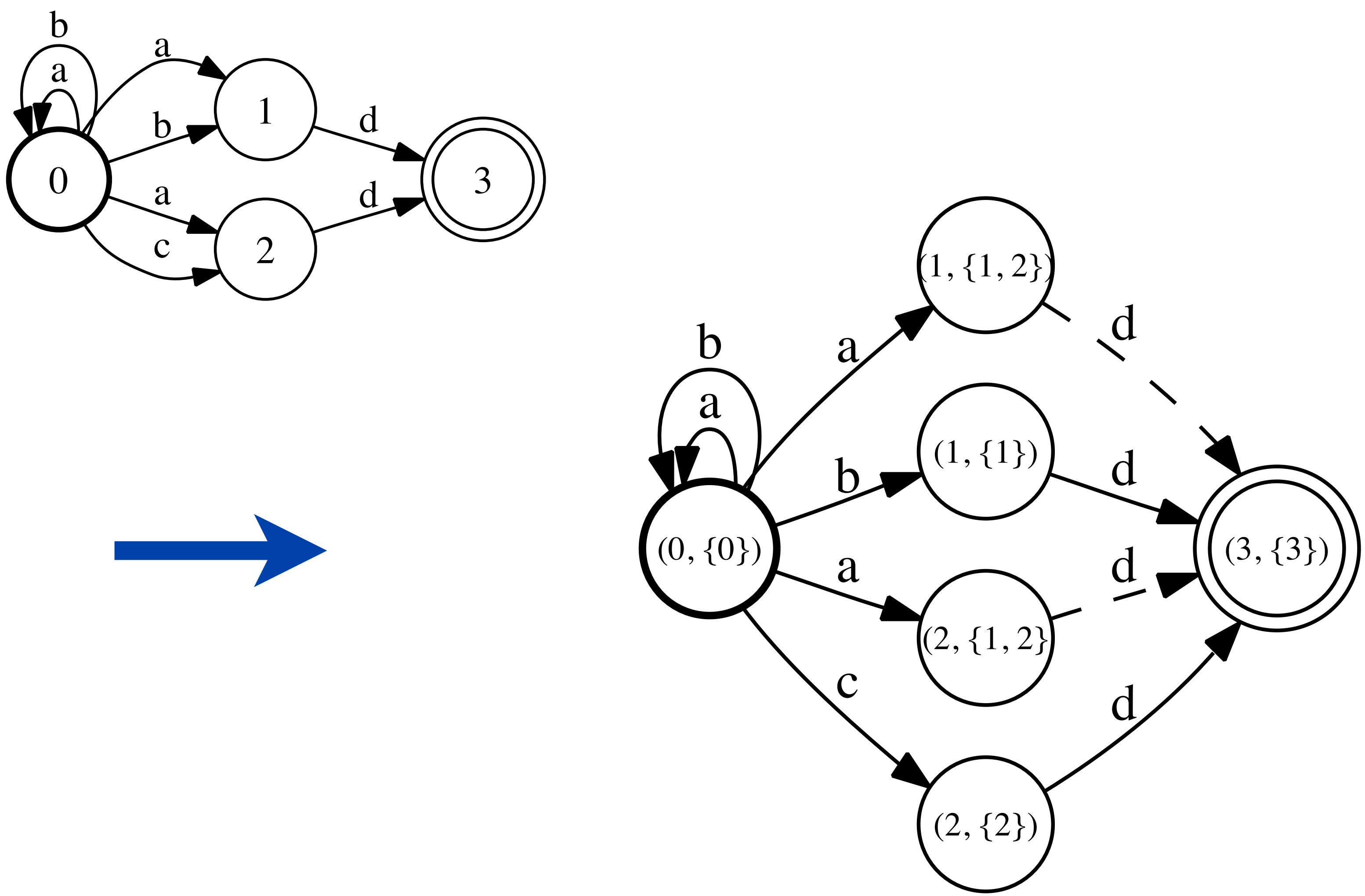
Schütz Construction - Illustration



New Disambiguation Algorithm

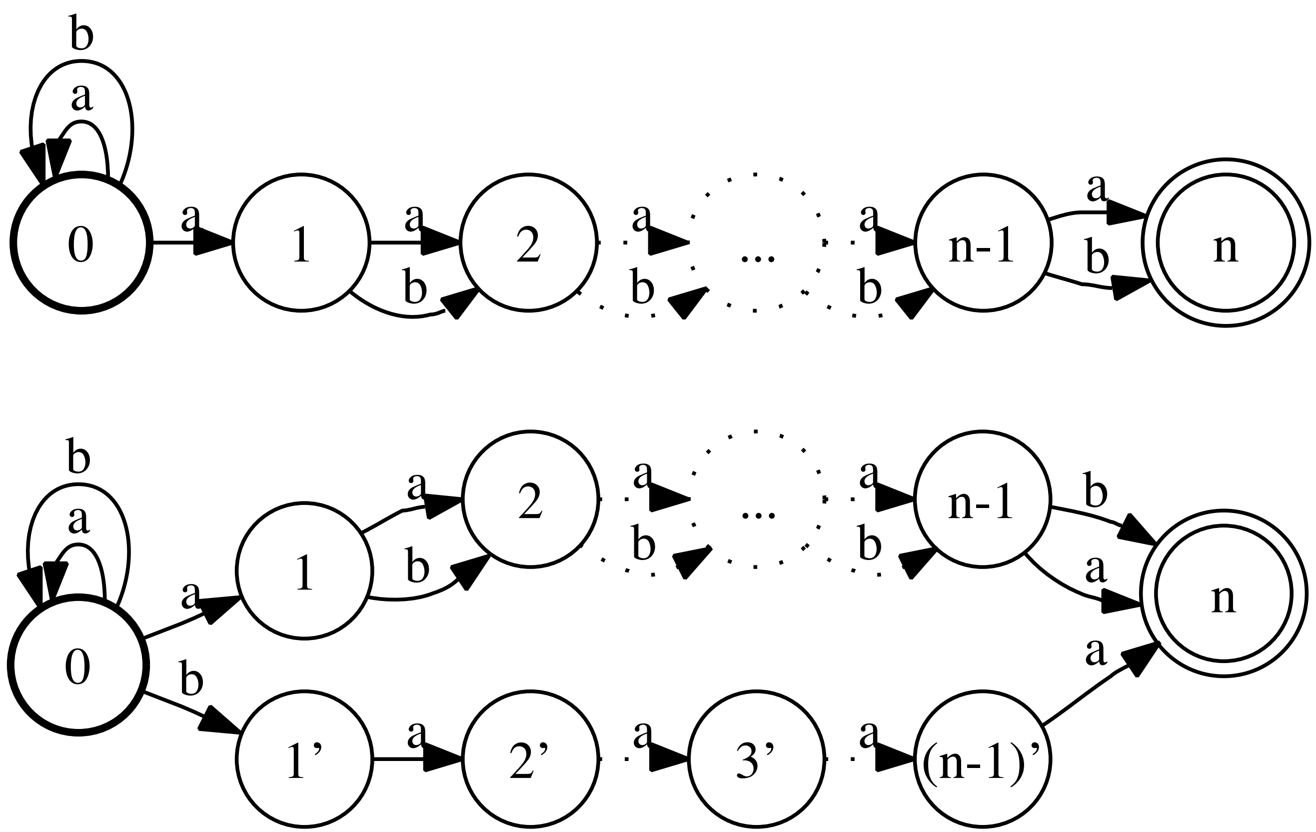
- Key ideas: for pair (p, s) with subset s.
  - no need to keep state q in s that does not admit a common future with p.
  - testing if p and q share common future can be done in constant time using $B = trim(A \circ A)$.
  - $\longrightarrow$ does not require determinization.

New Disambiguation - Illustration



New Disambiguation - Comparison

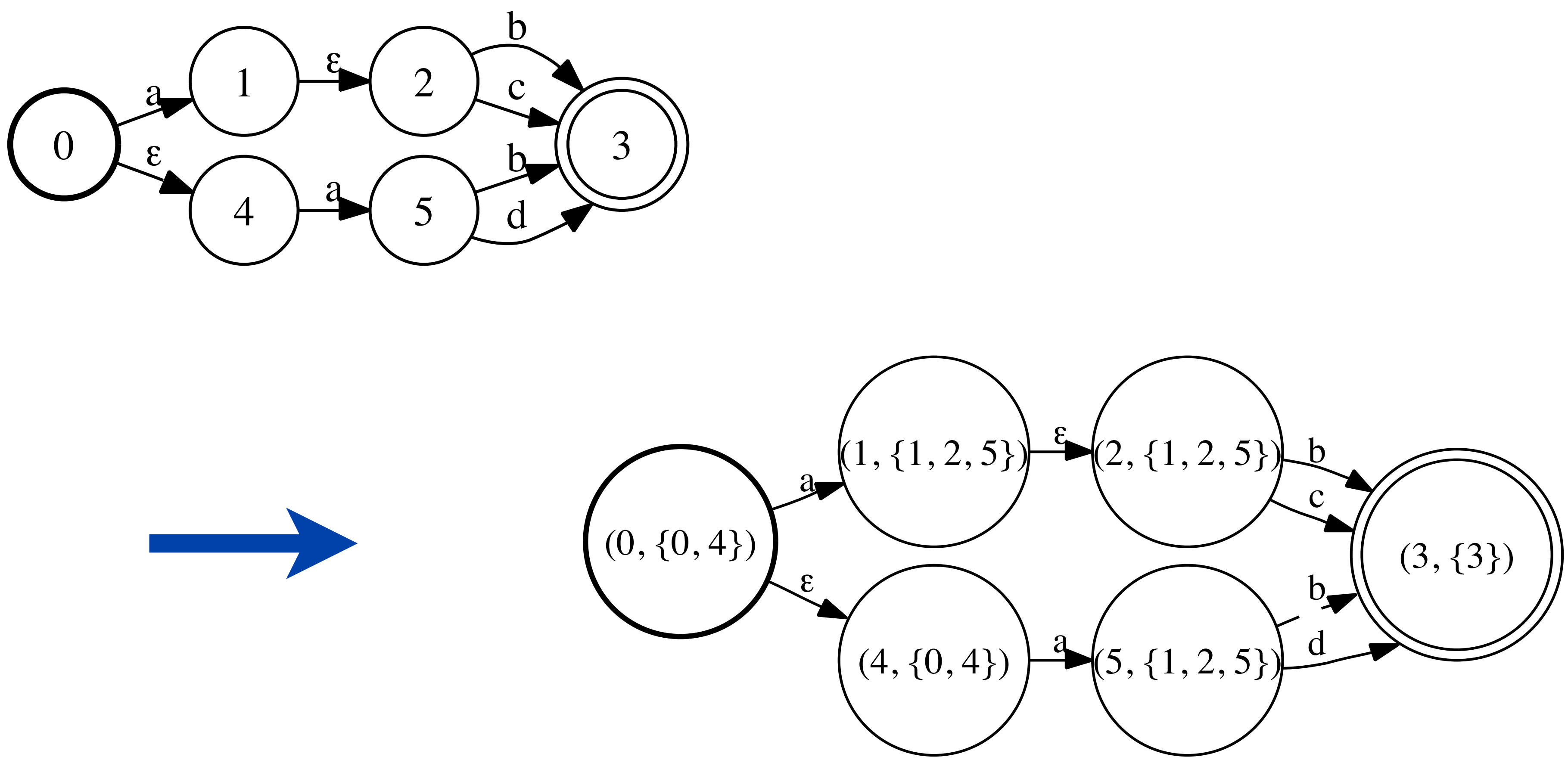
- Examples: Schütz construction: exponentially larger output because of the need for determinization. New algorithm: same automaton or linear output.



Automata with ε-transitions

- Extension:
  - define subsets using the ε-closure of the states.
  - computation of $B = trim(A \circ A)$ as in the case of automata without ε-transitions.
  - co-reachability of the states of output machine takes into account ε-transitions.

Automata with ε-transitions - Illust.



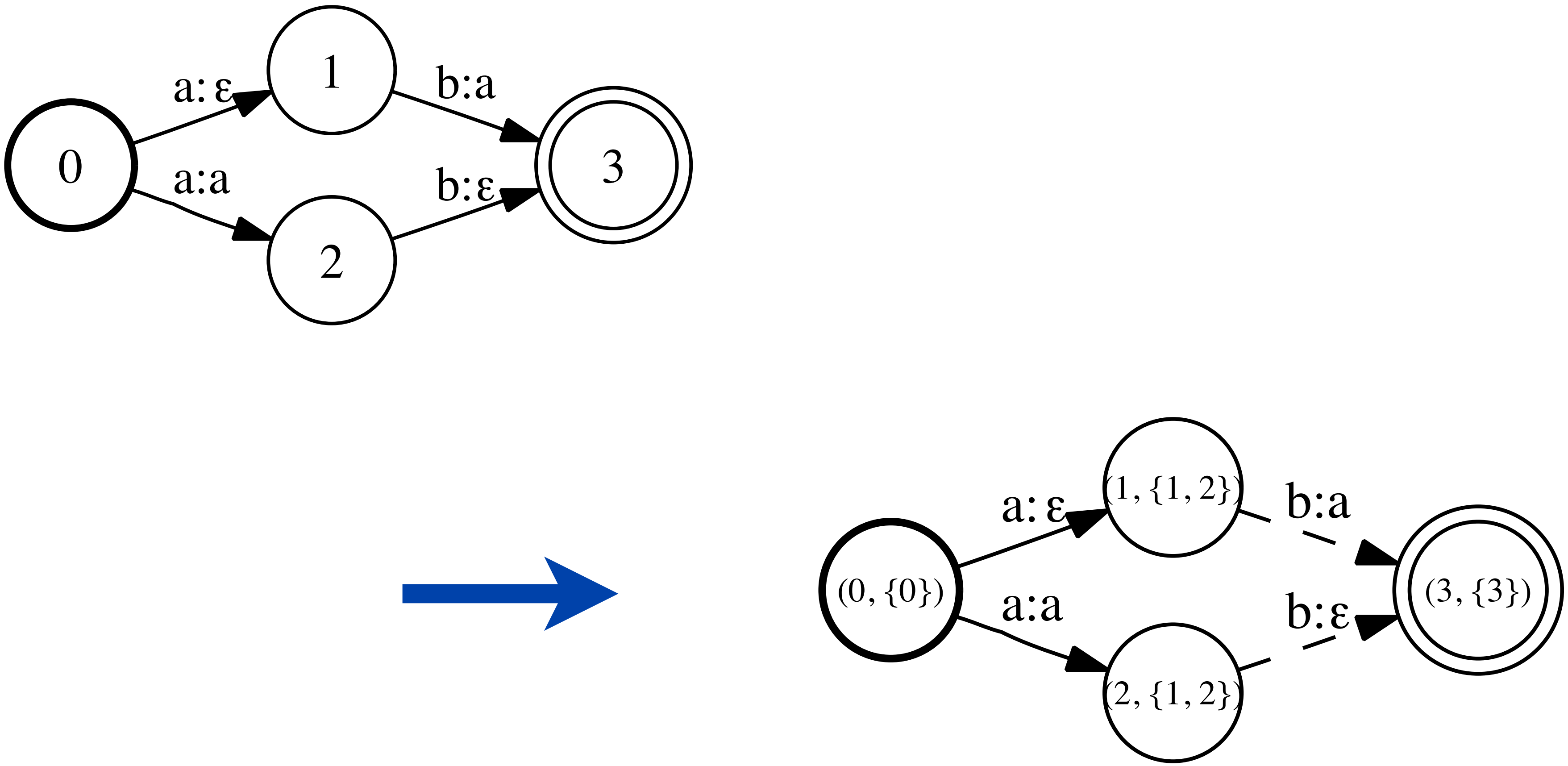
Functional Transducers

- Functional transducers: transducers representing partial functions. Thus, at most one output string for any input string.
- Theorem: functionality of transducers with output alphabet Δ can be tested in $O(|E|^2 + |\Delta|\,|Q|^2)$, see (Allauzen and MM, 2003).

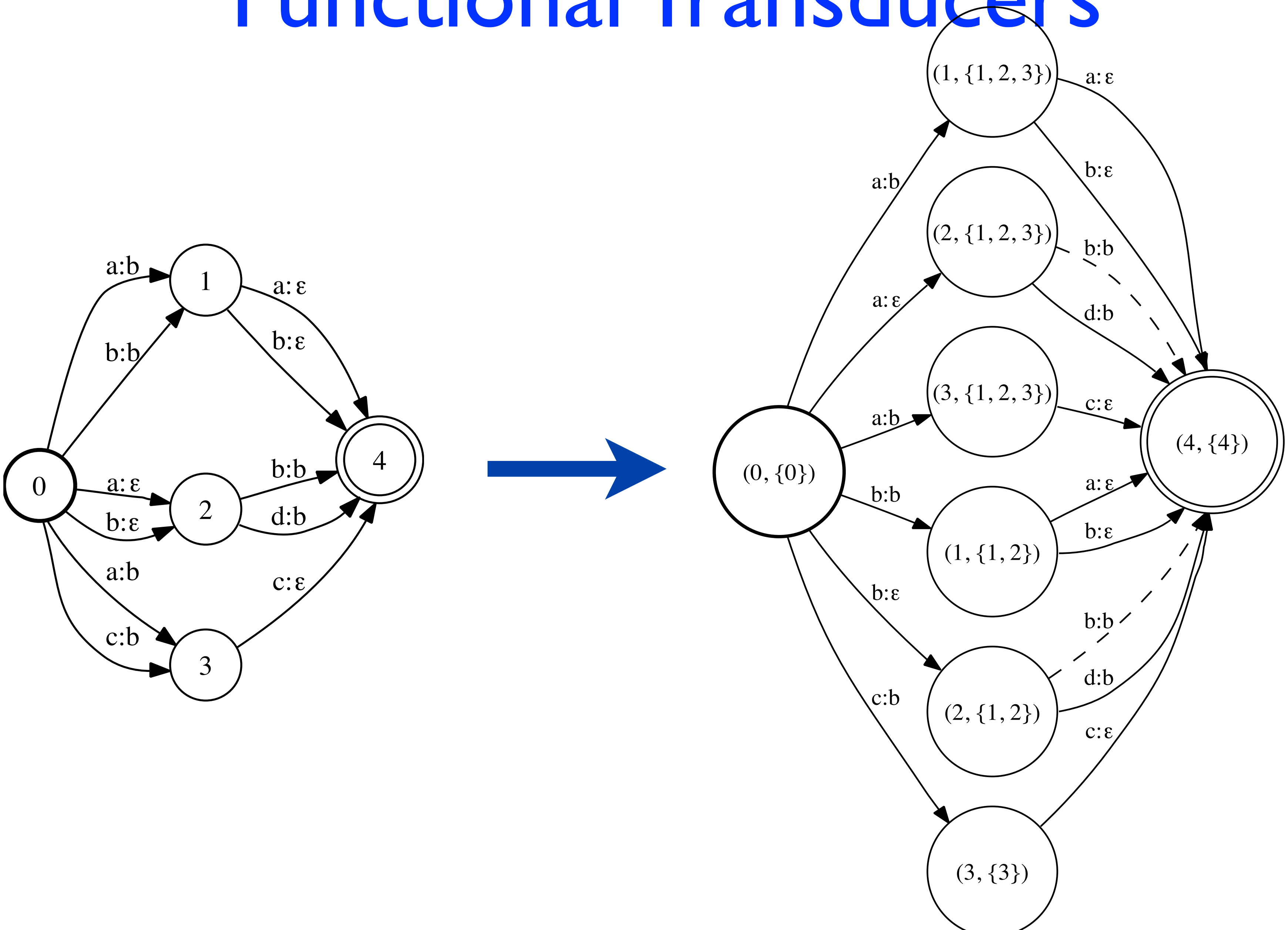
Disamb. of Functional Transducers

- Algorithm:
  - same as for automata since only input labels matter.
  - advantage over determinization: not all functional transducers are determinizable.
  - advantage over Schütz's construction: output often substantially smaller, in some cases exponentially.

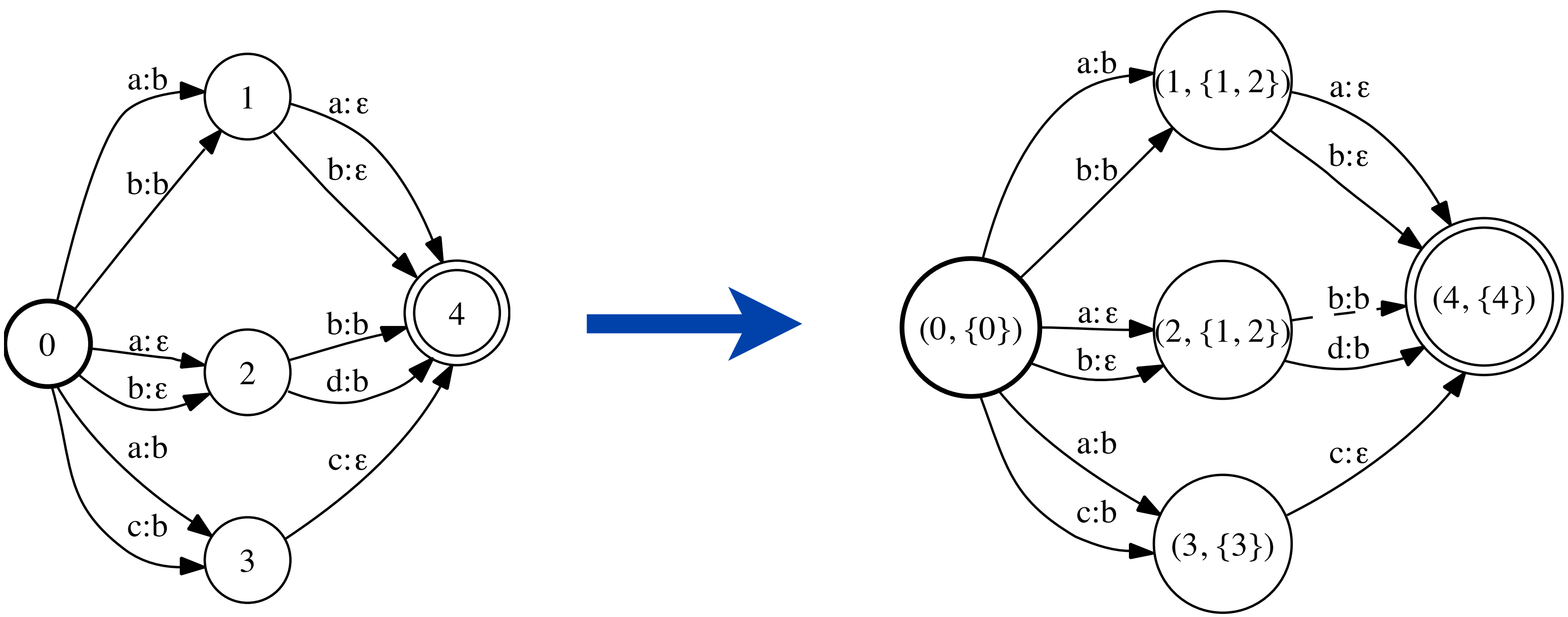
Disambiguation - Illustration



Schütz Construction - Comparison Functional Transducers



New Disambiguation - Comparison Functional Transducers



Conclusion

- Disambiguation of finite automata and functional transducers.
  - optimization algorithm with wider applicability than determinization.
  - practical importance in text and speech processing and bioinformatics applications.
- Disambiguation of broad families of weighted automata and transducers.
  - extension to be presented elsewhere.
  - theoretical analysis and guarantees.