Automatic Speech Recognition (CS753)

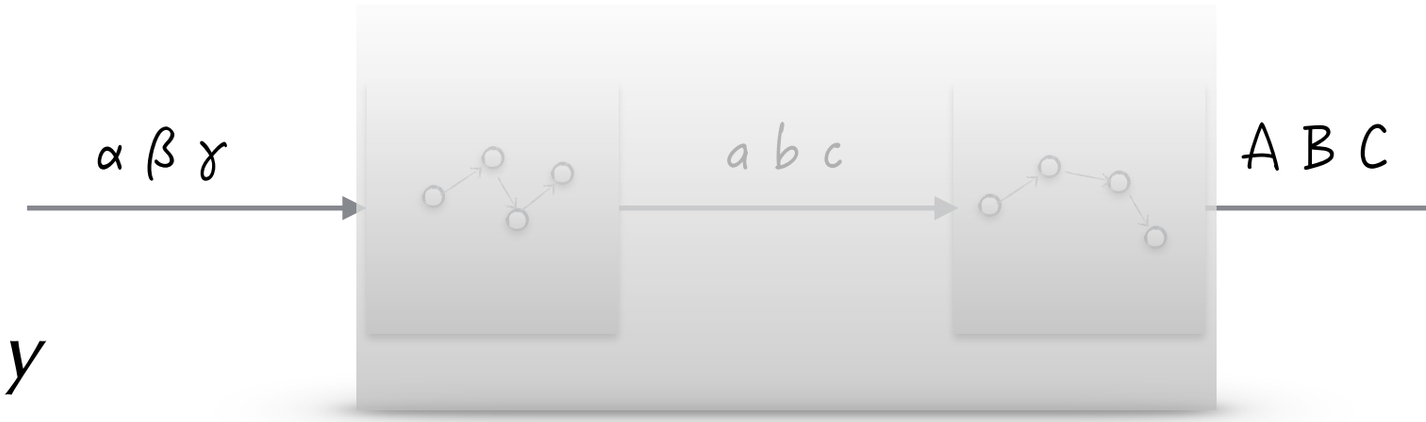
Lecture 3: WFSTs contd./WFSTs in Speech Recognition

Instructor: Preethi Jyothi

Lecture 3

Composition: Recap

- If T_1 transduces x to z,
  and T_2 transduces z to y,
  then $T_1 \circ T_2$ transduces x to y



- Note: output alphabet of $T_1 \subseteq$ input alphabet of T_2

- E.g. If T_1 removes punctuation symbols from a string, and T_2 changes uppercase letters to lowercase letters, then $T_1 \circ T_2$ brings about both changes

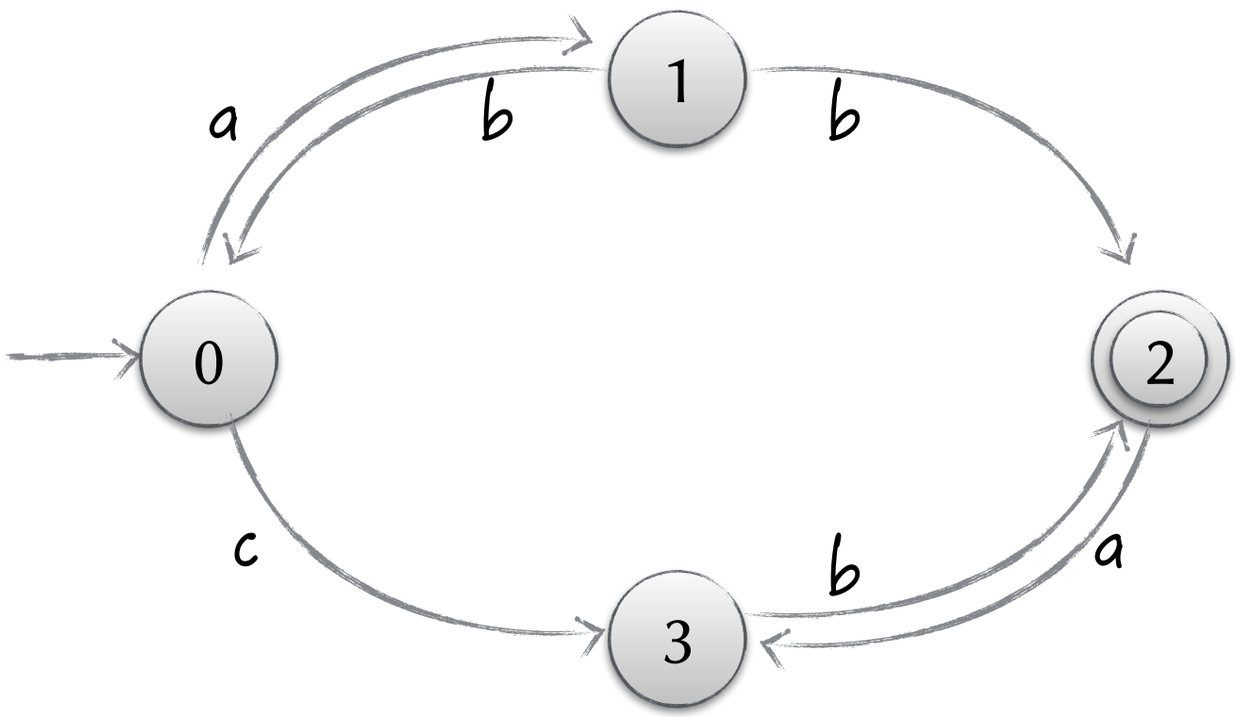
Determinization and Minimization

- WFSTs constructed using various operations (or designed by hand) may have several redundancies
  - Affects the efficiency of subsequent operations
- Determinization and minimization seek to remove redundancies
  - Determinization can expand a WFST, but makes it faster to process an input string
  - Minimization results in the smallest number of states
- Will discuss WFSAs here. Extends to WFSTs.

Deterministic FSAs

Any input sequence yields a unique path (if at all)

- An FSA is **deterministic** if:
  - Unique start state
  - No two transitions from a state share the same label
  - No epsilon labels

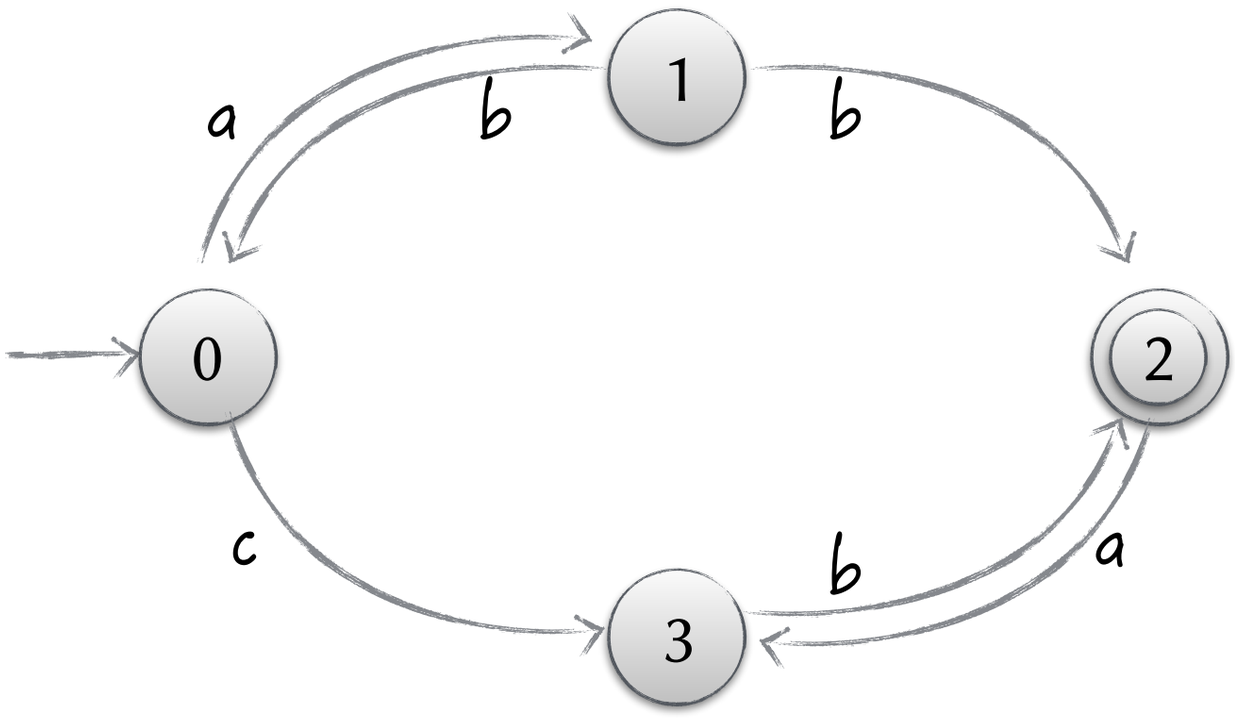


Deterministic or non-deterministic?

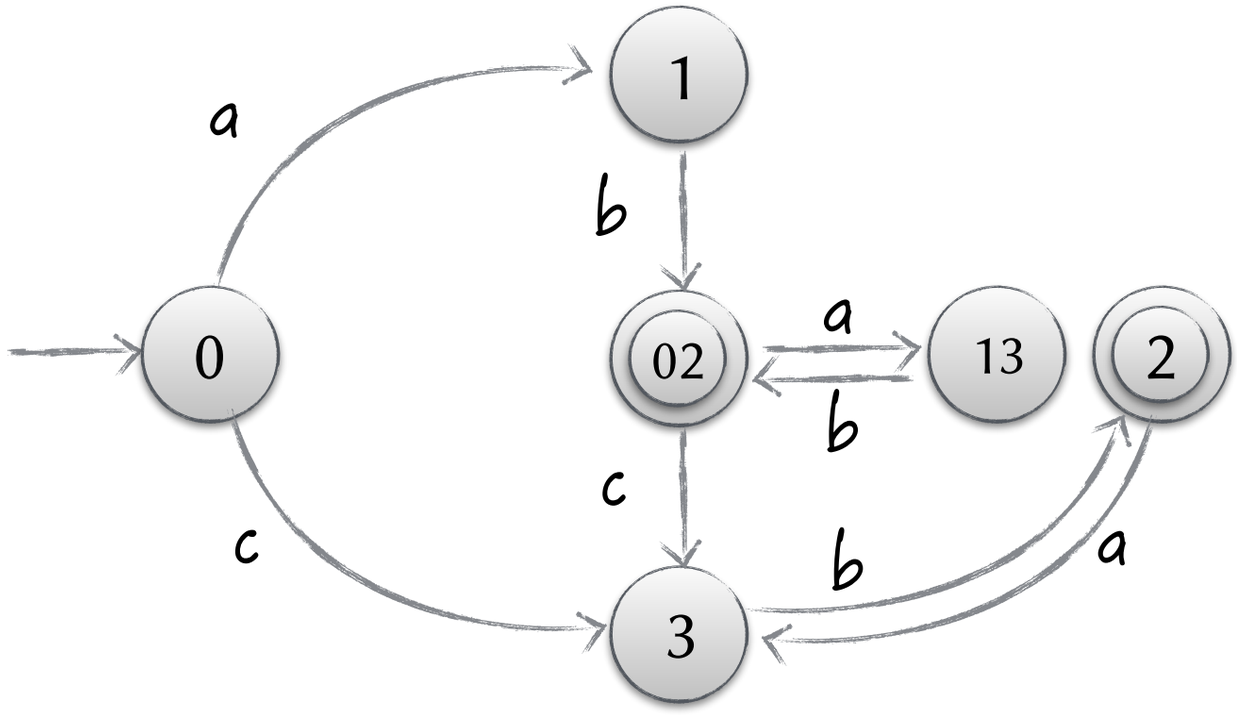
Determinization

Construct an equivalent deterministic FSA

States correspond to subsets of states in the original FSA



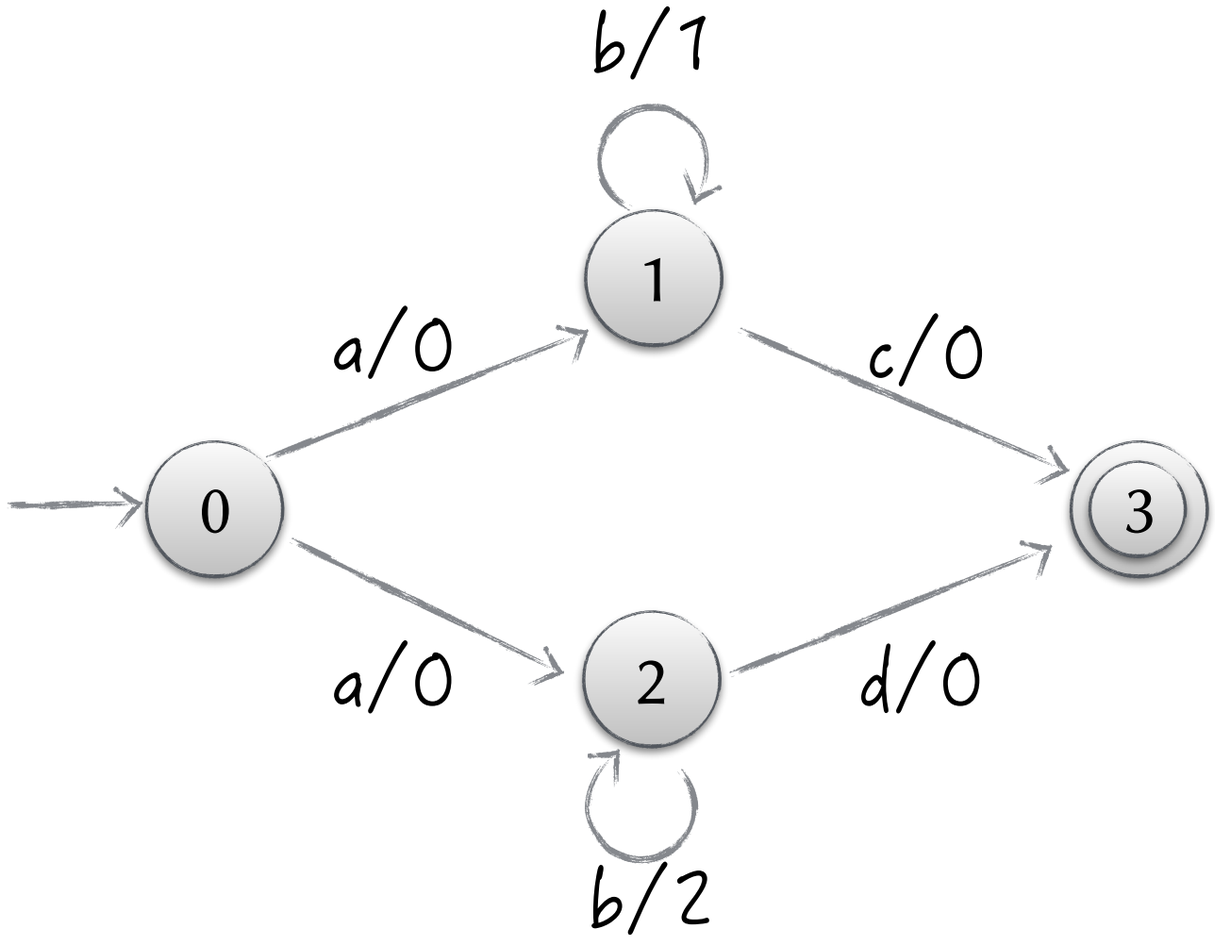
non-deterministic FSA



equivalent deterministic FSA

Determinization: Weighted FSA

Some *Weighted*-FSAs are not determinizable! [M97]



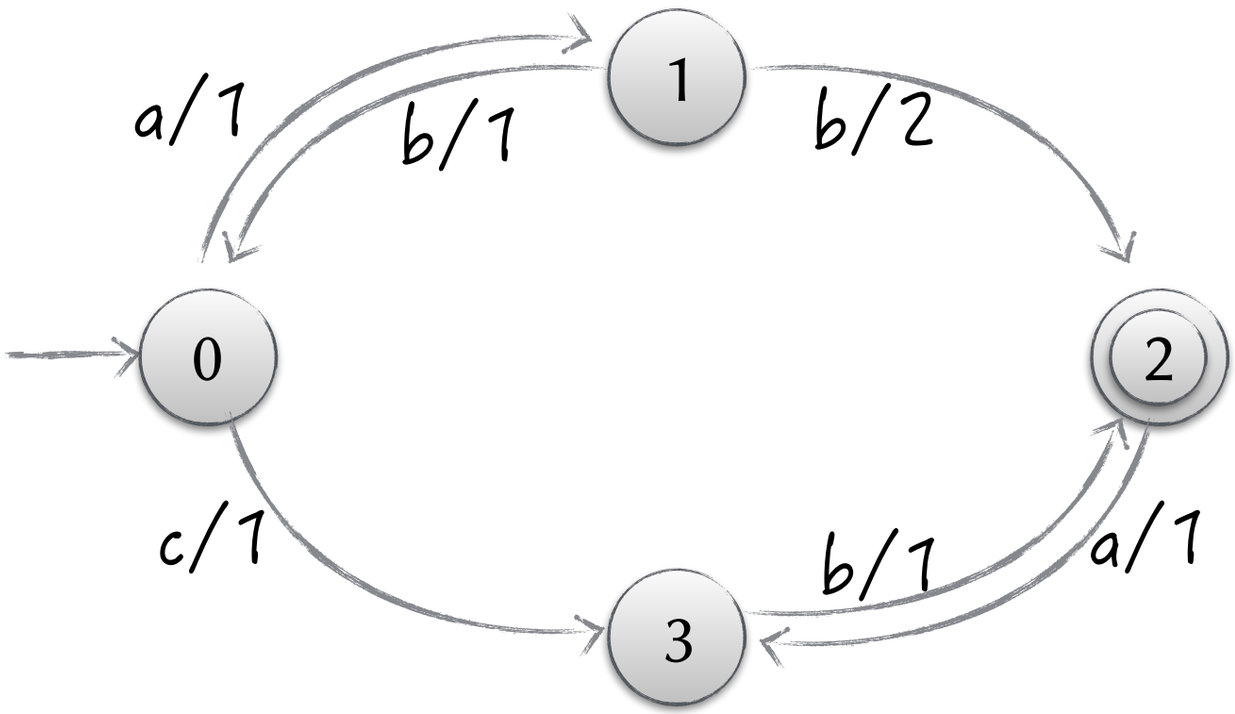
Weight of string $ab^nc = n$ and weight of $ab^nd = 2n$

After seeing ab^n an FSA can't remember n

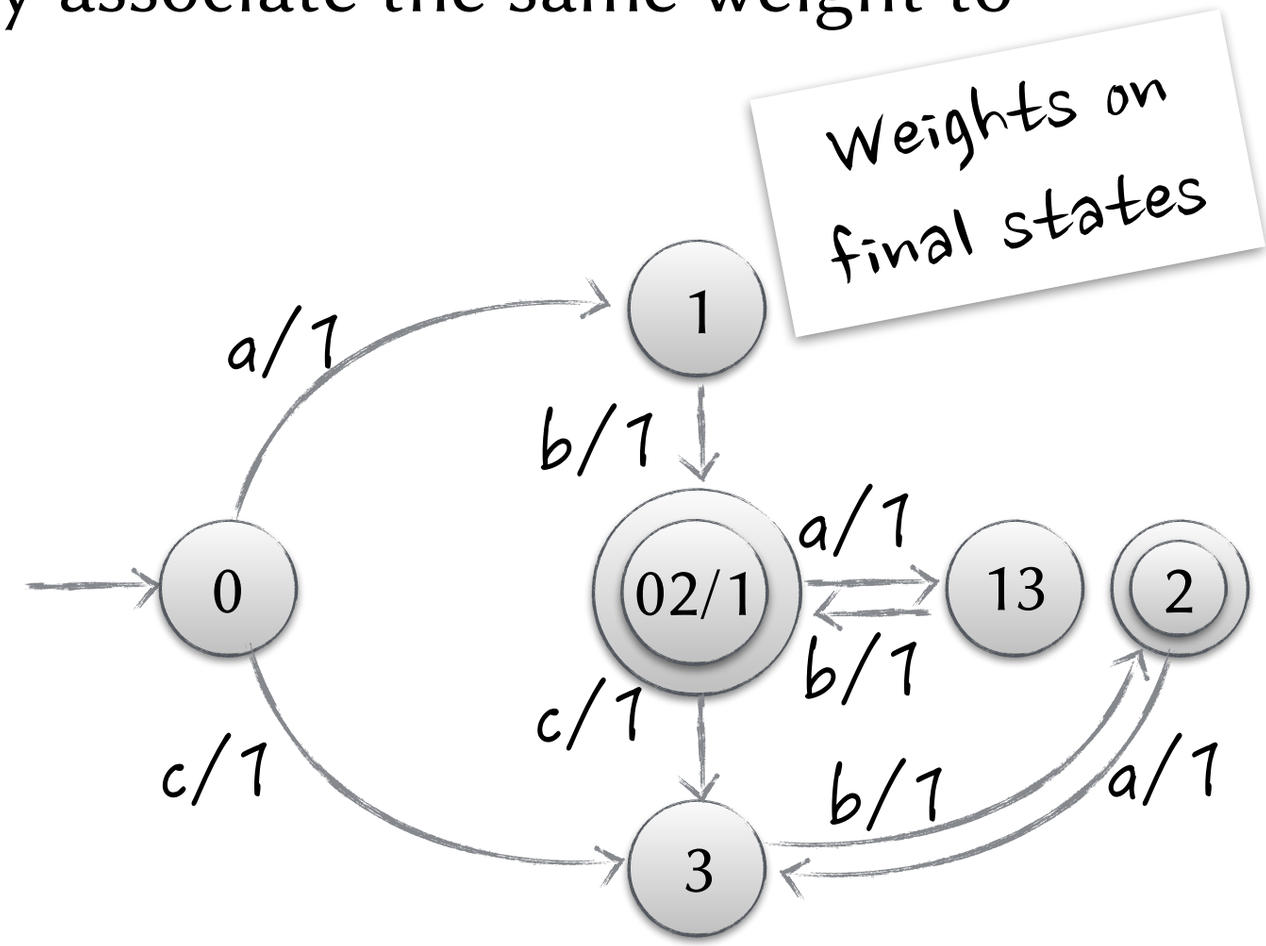
[M97] M. Mohri. Finite-State Transducers in Language and Speech Processing. Computational Linguistics, 23(2), 1997

Determinization: Weighted FSA

Two WFSAs are equivalent if they associate the same weight to each input string



non-deterministic WFSA

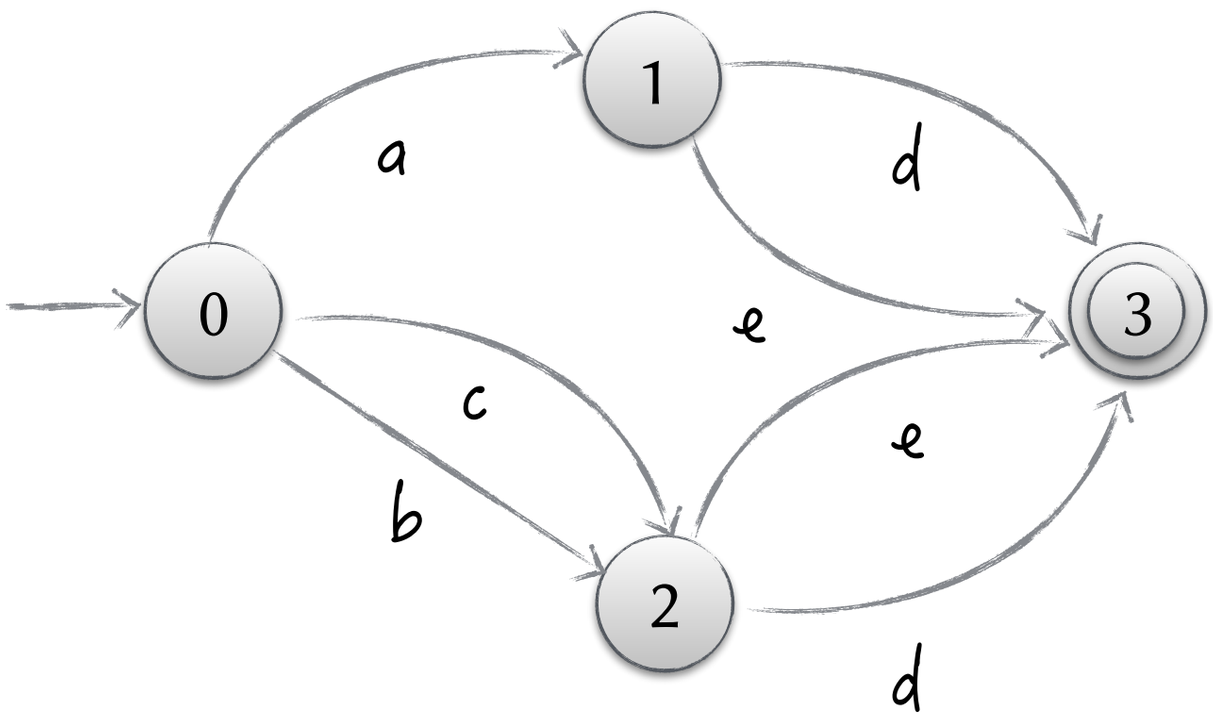


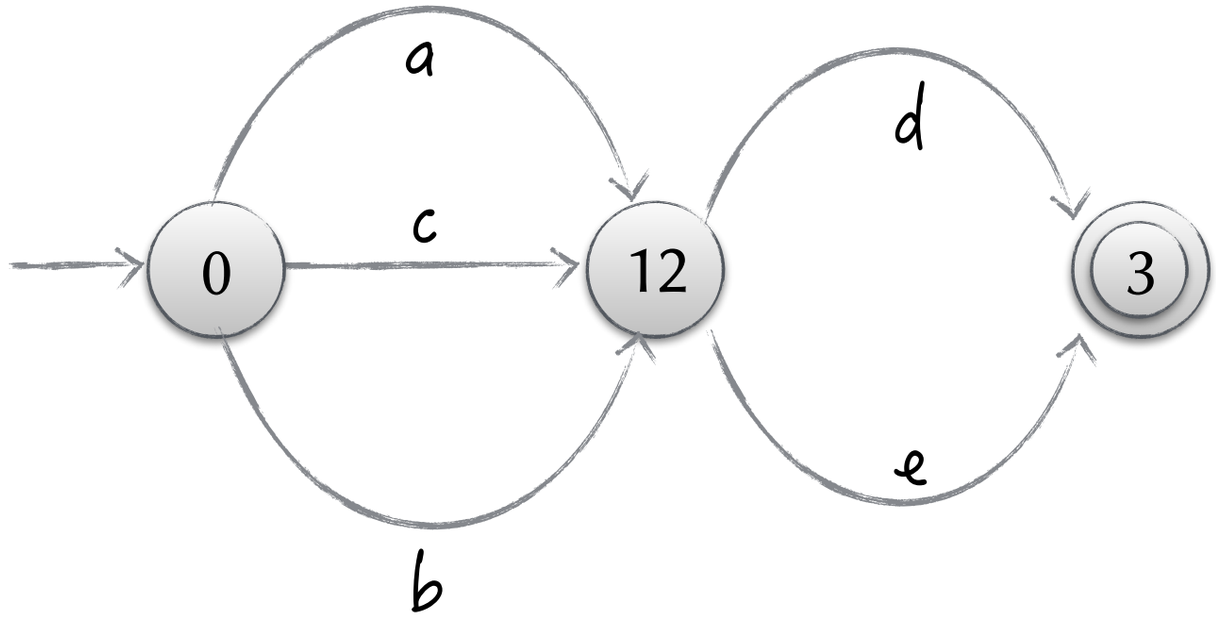
equivalent deterministic WFSA

Minimization

Minimization: find an equivalent deterministic FSA with the least number of states (and transitions)

Unweighted FSAs have a unique minimal FSA [Aho74]



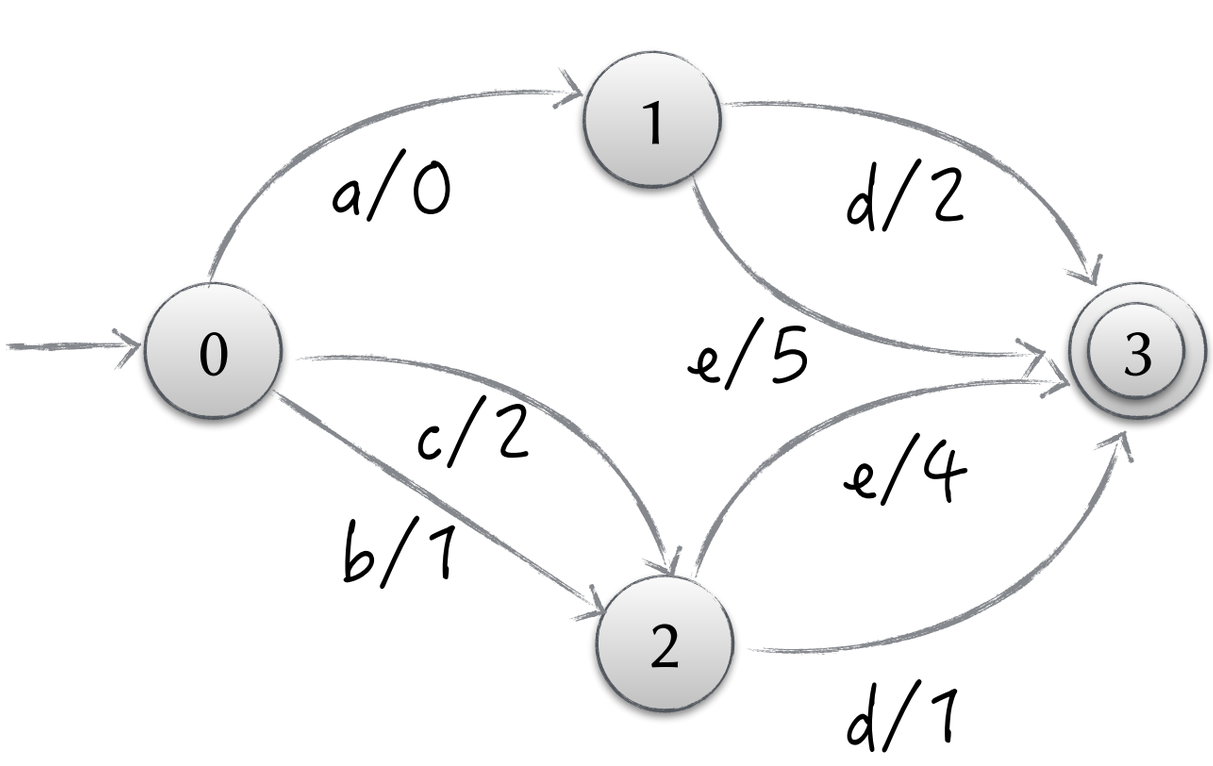


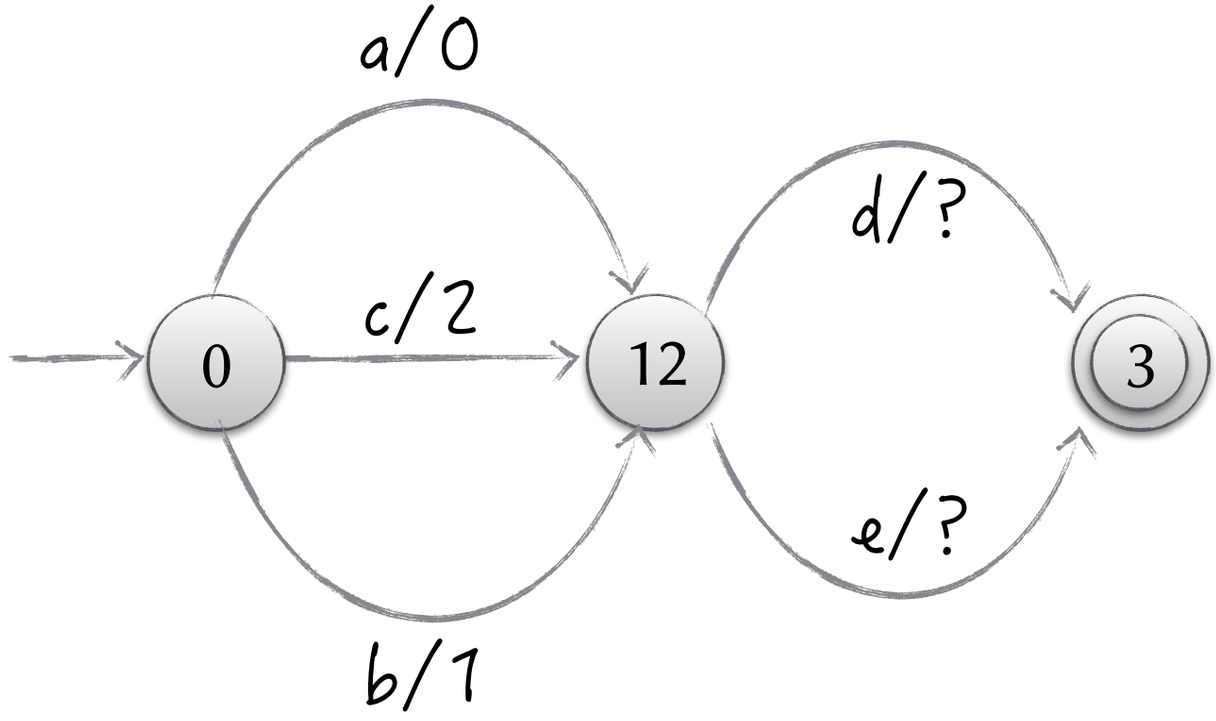
Obtained by identifying and merging *equivalent states*

Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. The design and analysis of computer algorithms. Addison Wesley, 1974.

Minimization: Weighted FSA

Two states are equivalent only if for every input string, the outcome — weight assigned to the string, if accepted — starting from the two states are the same



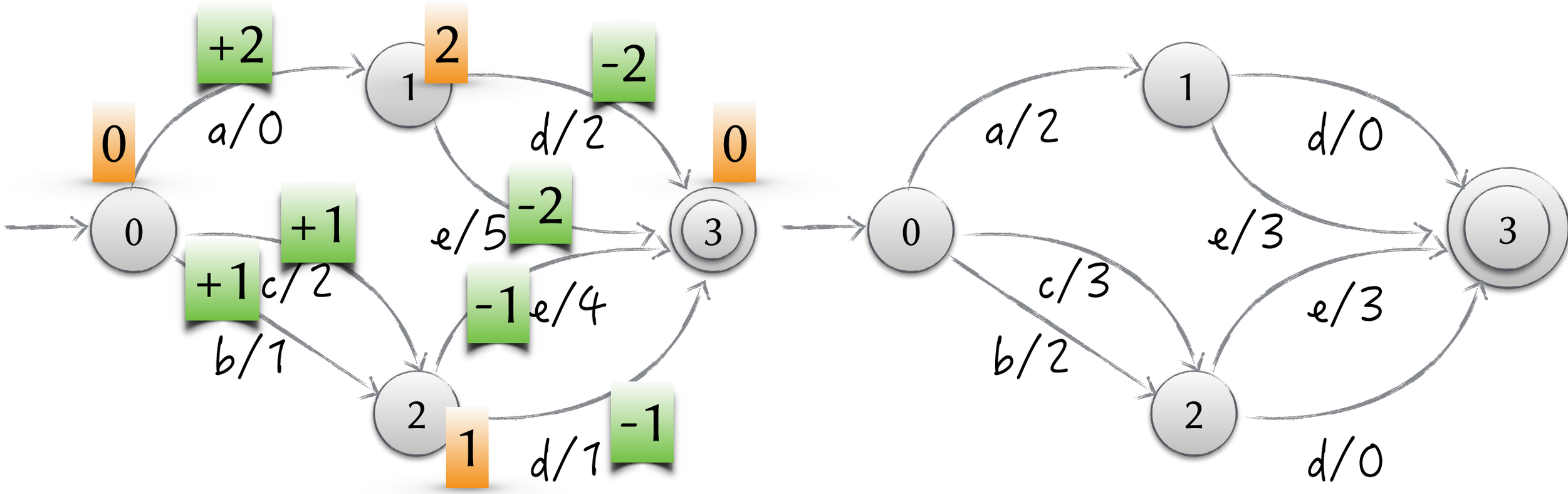


Redistribute weights before identifying equivalent states

Minimization: Weighted FSA

Reweighting OK as long as resulting WFSA is equivalent

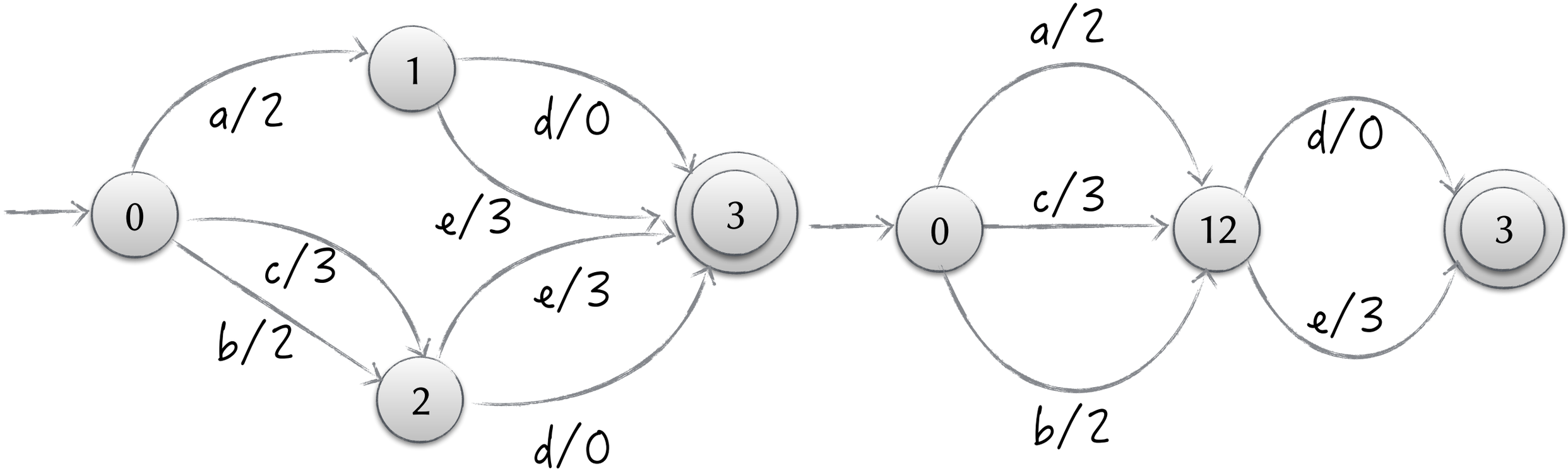
Can reweight using a "potential function" on states



"Weight pushing": Reweighting using a potential function that optimally moves weights towards the start state

Minimization: Weighted FSA

After weight-pushing, can simply apply unweighted FSA minimization (treating label/weight as label)



Guaranteed to yield a minimal WFSA (under some technical conditions required for weight-pushing)