

Rigorous

Approximated Determinization
of Weighted Automata



Benjamin Aminof (Hebrew University)

Orna Kupferman (Hebrew University)

Robby Lampert (Weizmann Institute)

Israel

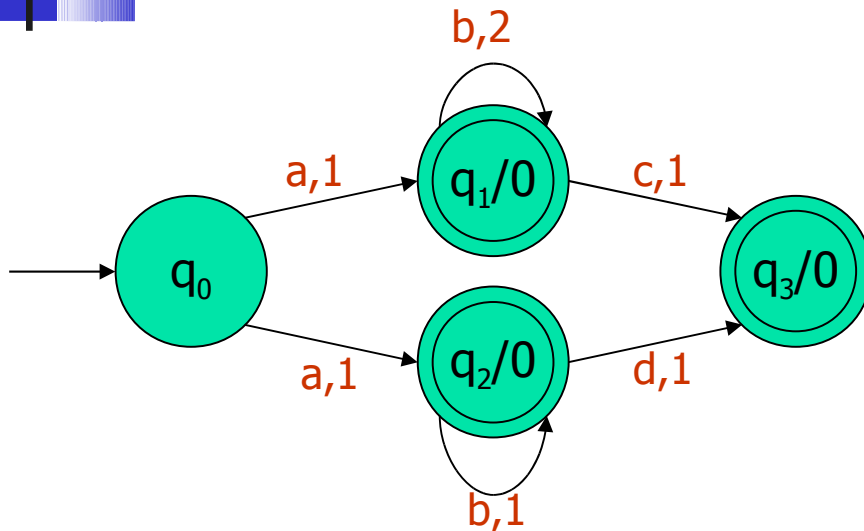


Outline

- Weighted automata
- Determinizability of weighted automata
- Mohri's determinization algorithm
- Approximated-determinization algorithm
- Correctness and termination
- Summary
- Future work

Weighted Automata (WFA)

A:



weight functions


c: transitions ! **R**


f: accepting states ! **R**

- $w=abc$ $\text{cost}(w)=(1+2+1)+0=4$
- $w=abbd$ $\text{cost}(w)=(1+1+1+1)+0=4$
- $w=abb$ $\text{cost}(w)=\min\{5,3\}=3$



Weighted Automata – language

- A **run** of A on a word $w = w_1 \dots w_n$ is a sequence $r = r_0 r_1 r_2 \dots r_n$ over Q such that $r_0 \in Q_0$ and for all $1 \leq i \leq n$, we have  .

- A run r is **accepting** $\$ r_n$ is accepting. 
(standard finite-word accepting condition)

- $L(A) = \{w : A \text{ has an accepting run on } w\}$



Weighted Automata – costs

- A cost of a run $r = r_0 r_1 r_2 \dots r_n$ is

$$\text{cost}(r) = \sum_{i=1}^n c((r_{i-1} \xrightarrow{w_i} r_i) + (r_n))$$

- defined only for accepting runs

 - A cost of a word $w = w_1 \dots w_n$ is
- $$\text{cost}(w) = \min_{\text{accepting runs } r \text{ of } A \text{ on } w} \text{cost}(r)$$
- If $w \notin L(A)$ then $\text{cost}(w) = 1$.



Weighted Automata – more

- A WFA A is **trim** if each of its states is reachable from some initial state, and has a reachable accepting state.
- A WFA A is **unambiguous (single-run)** if it has at most one accepting run on every word.

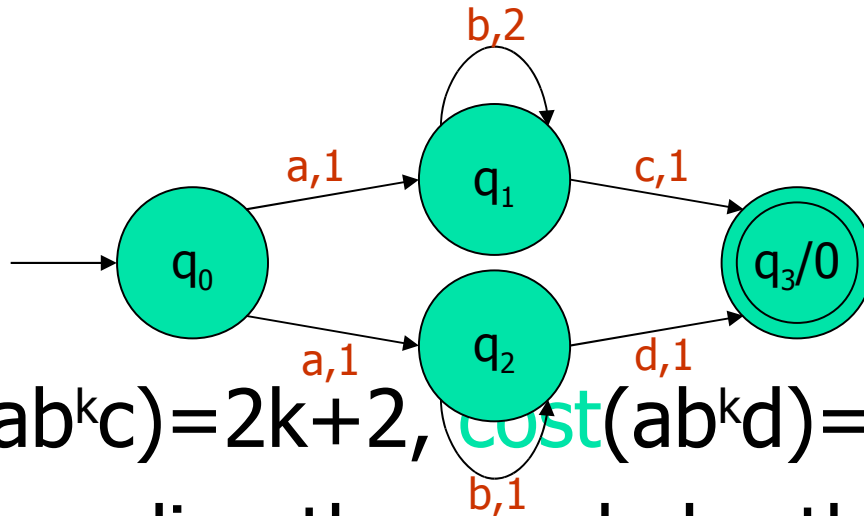


Applications of WFA

- formal verification of quantitative properties
- automatic speech recognition
- image compression
- pattern matching (widely used in computational biology)
- ...

A_1 is non-determinizable

A_1 :



- $\text{cost}(ab^k c) = 2k + 2$, $\text{cost}(ab^k d) = k + 2$
- After reading the word ab , the difference between the costs of reading c and d is k .
- For $i \neq j$, a deterministic WFA must be in different states after reading ab and ab .
- A deterministic WFA must have 1 states.



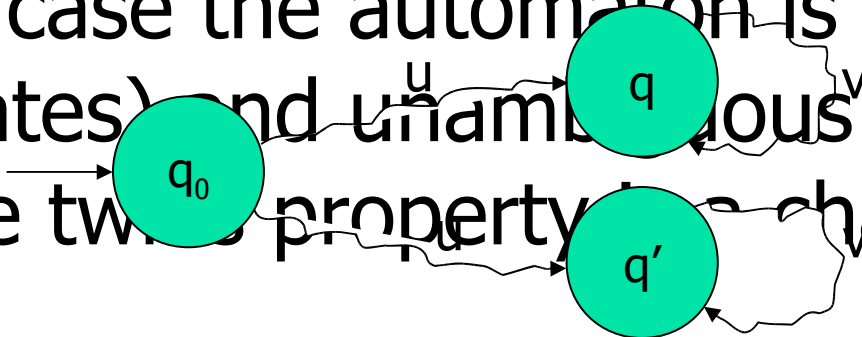
Determinizability

- Weighted automata are not necessarily determinizable.
- To decide whether a given weighted automaton is determinizable is an **open question**.
- A sufficient condition for determinizability + algorithm [Mohri '97].



A sufficient condition [Mohri '97]

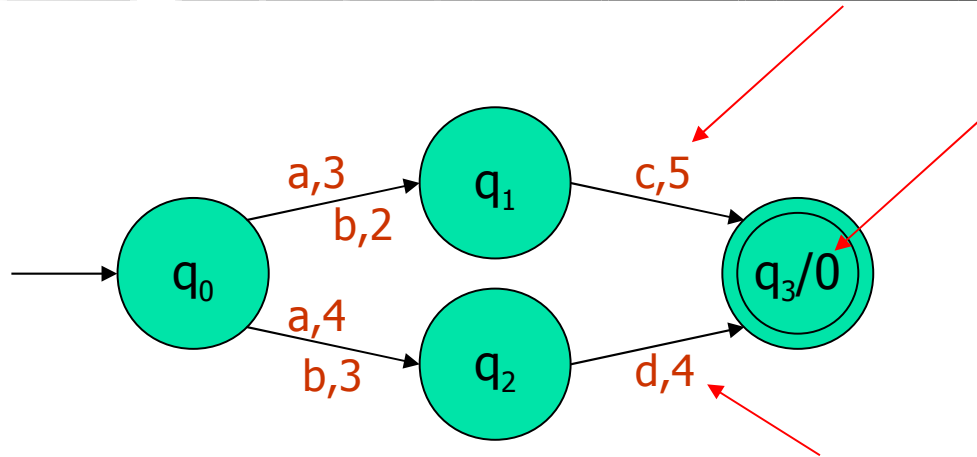
- The twins property:
For every two states $q, q' \in Q$,
and two words $u, v \in \Sigma^*$,
if $q, q' \in \delta(Q_0, u)$, $q \in \delta(q, v)$, and
then $cost(q, v, q) = cost(q', v, q') \in \delta(q', v)$,
- In case the automaton is trim (no empty states) and unambiguous (single-run), the twins property is a characterization.



Determinization algorithm

[Mohri '97] - example

A_2 :



word / cost

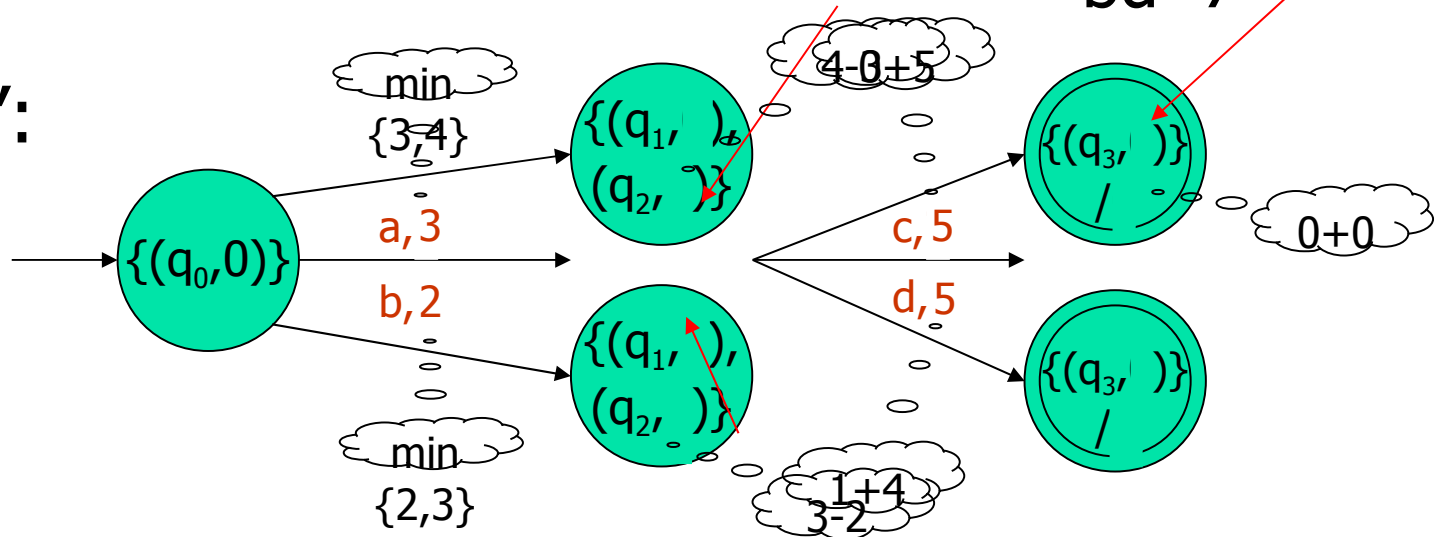
ac 8

bc 7

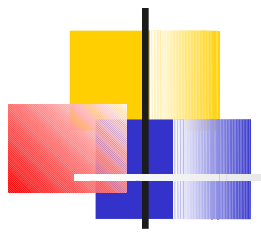
ad 8

bd 7

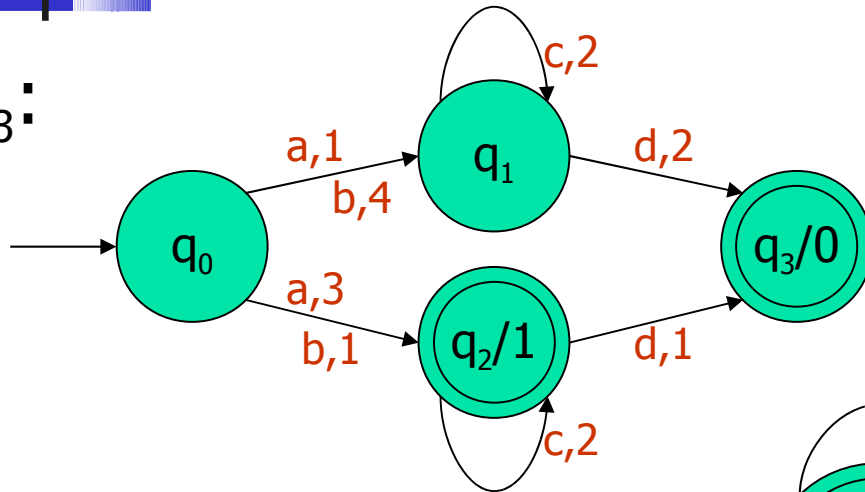
A_2' :



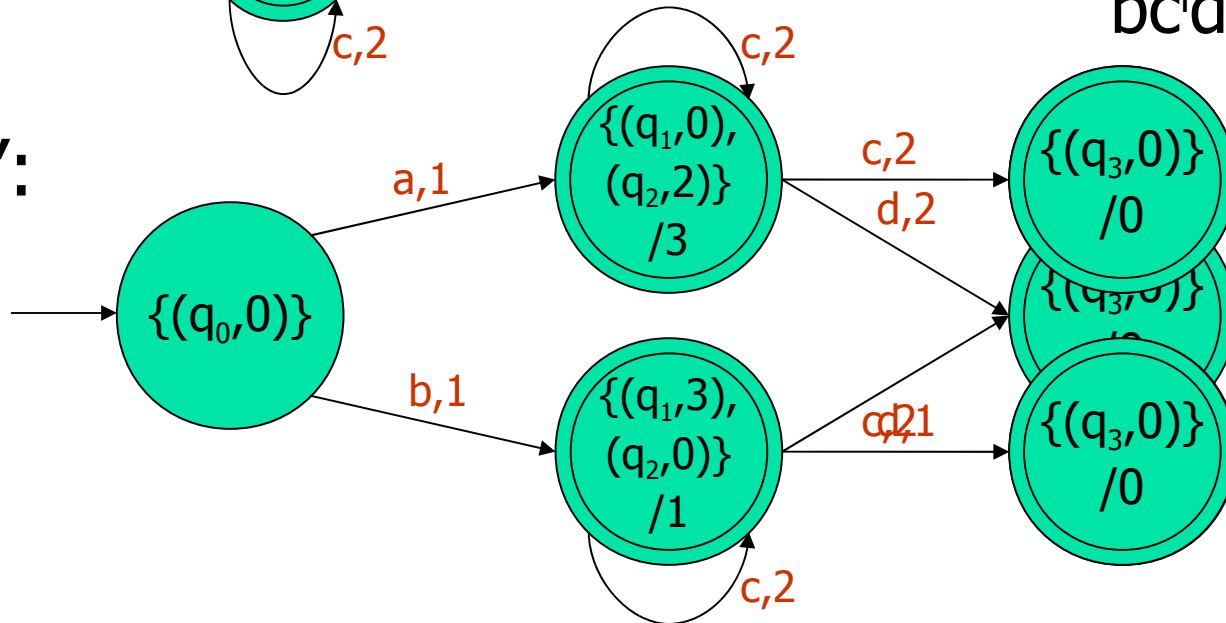
Determinization algorithm - another example



A_3 :



A_3' :



word / cost

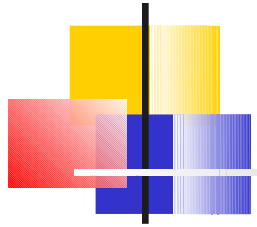
ac^i $3+2i+1$

bc^i $2+2i$

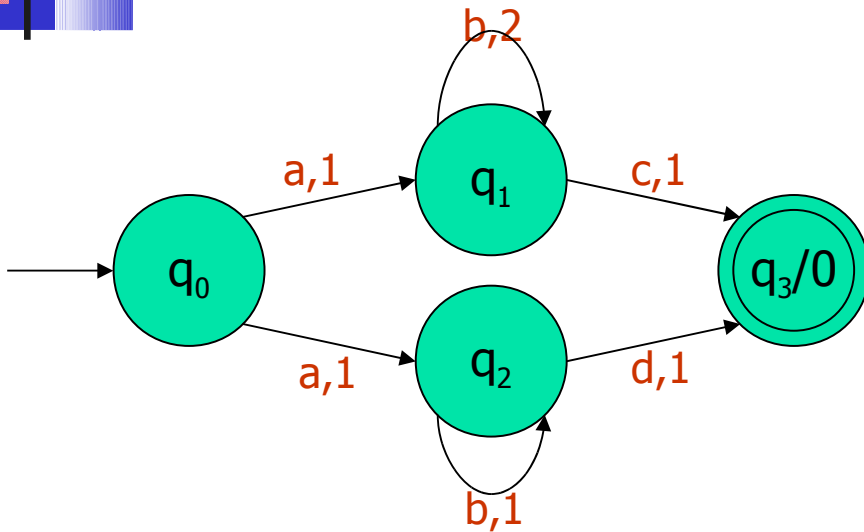
acd $3+2i$

$bcid$ $2+2i$

Determinization algorithm - non-determinizable example



A_1 :

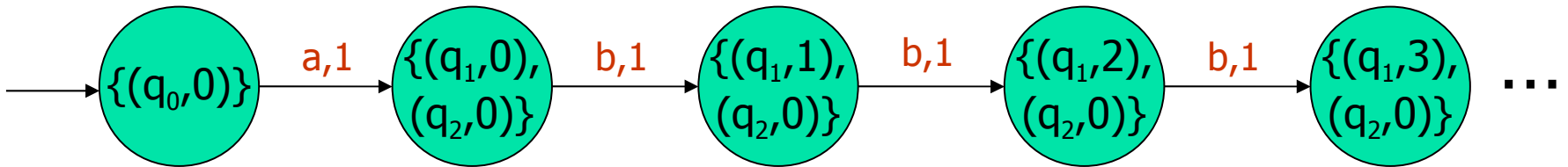


word / cost

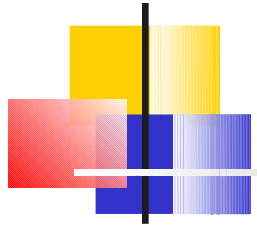
abⁱc 2+2i

abⁱd 2+i

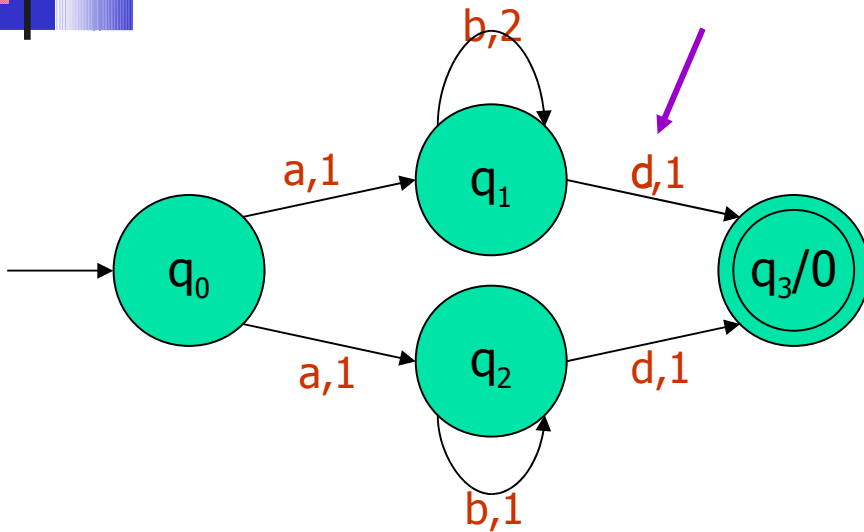
A_1' :



Determinization algorithm - a bad determinizable example



A_1 :

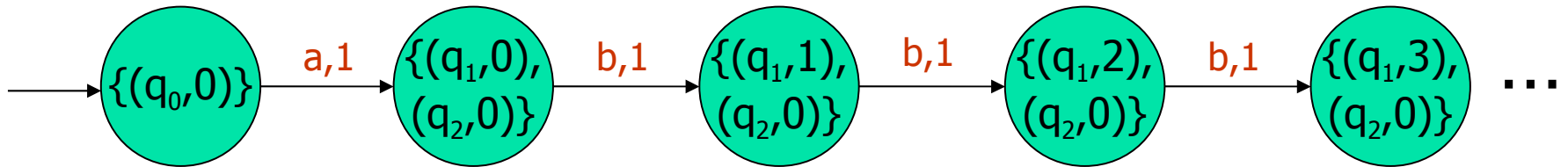


word / cost

~~abc 2+2i~~

abd 2+i

A_1' :





Mohri's algorithm - remarks

- Mohri's algorithm terminates iff the original automaton has the twins property.
- For trim and unambiguous WFAs, there is a polynomial algorithm for testing the twins property.
- There are determinizable WFAs that do not satisfy the twins property.



Approximated determinization

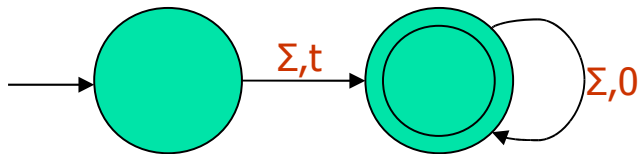
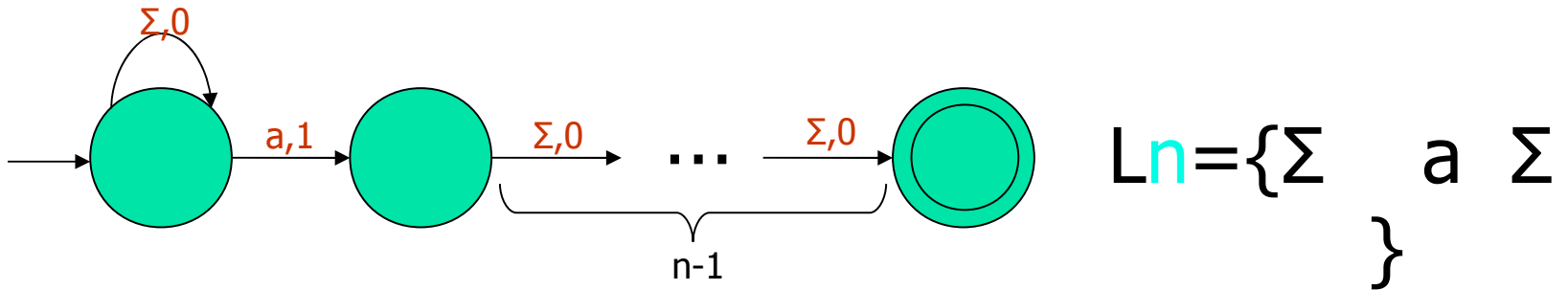
Given a WFA A and an approximation factor $t \geq 1$, construct a deterministic WFA A' , such that for every word w we have

$$\text{cost}(A, w) \leq \text{cost}(A', w) \leq t \cdot \text{cost}(A, w).$$

- When exact determinization is impossible.
- When the result of exact determinization is too large.

Succinctness

A_4 :



A deterministic equivalent requires **2** states

$$L(A_4) = \Sigma^+$$

$$\text{cost}(w) = \begin{cases} 1 & w = \varepsilon \\ t & w \in L_n \\ t & w \in \Sigma^+ \setminus L_n \end{cases}$$

A **t**-approximate deterministic?

2 states

Approx. determinization algorithm

[Buchsbau-

Giancarlo-Westbrook '01]

- Based on Mohri's algorithm.
- Relaxes the condition for unification of states – rather than requiring residuals of corresponding states to be identical, requires them to be close (within $1+\epsilon$ of the smaller one).
- No guarantees about the new costs.
- No sufficient condition for termination.

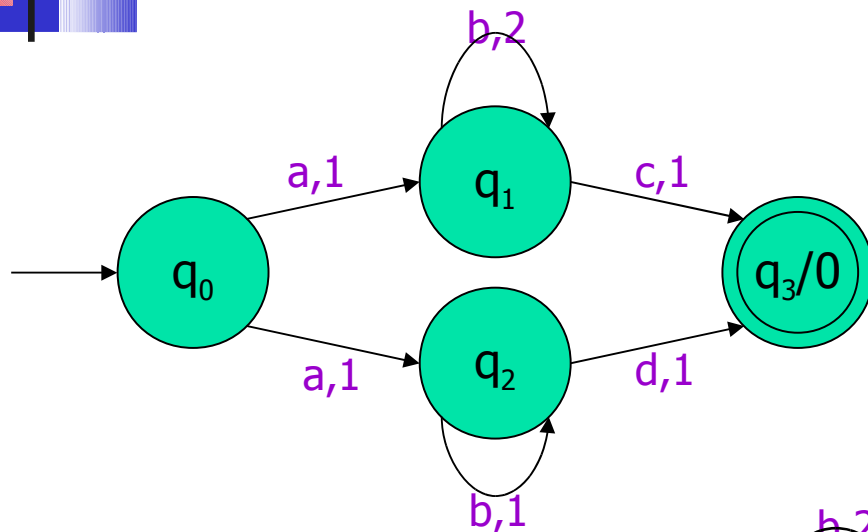


Our algorithm: t-determinization

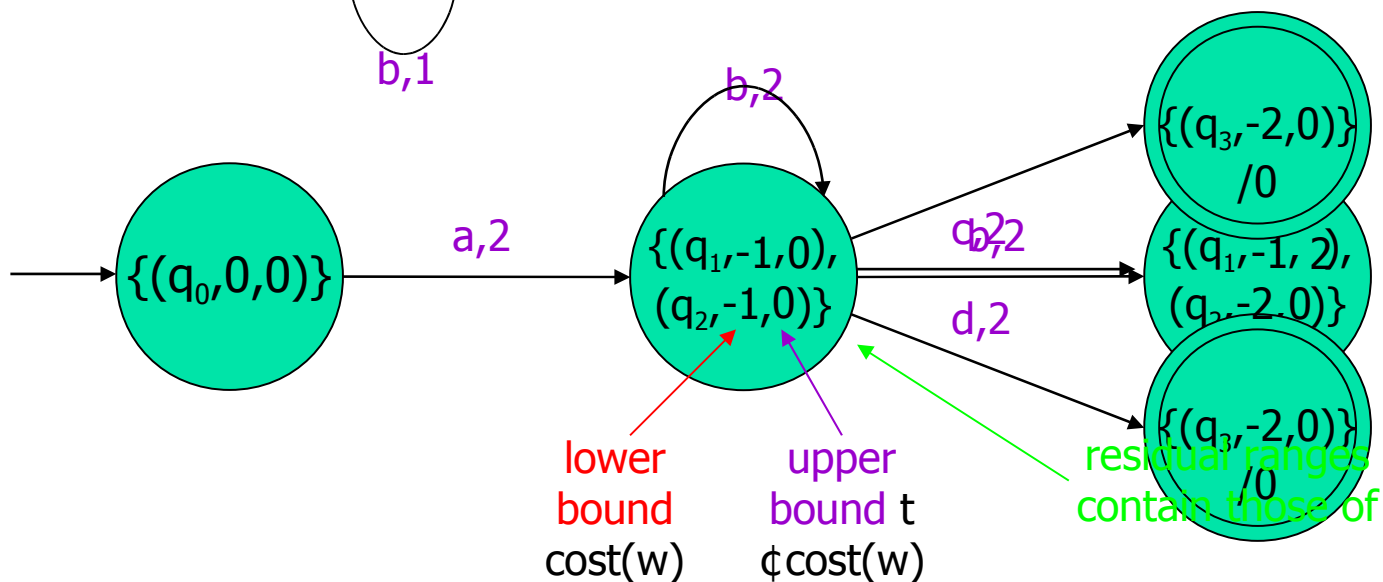
- Determinization up to a factor t
 - The new cost of any accepted word w is between $\text{cost}(w)$ and $t\text{cost}(w)$.
- differs from Mohri's algorithm
 - Weights are multiplied by t .
 - For each state in a subset we maintain a range of residues rather than one.
 - The criterion for unification of states is relaxed (they may be non-identical).

2-determinization of A_1

A_1 :

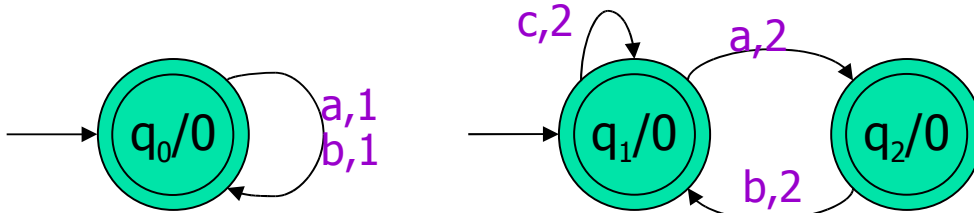


A_1' :

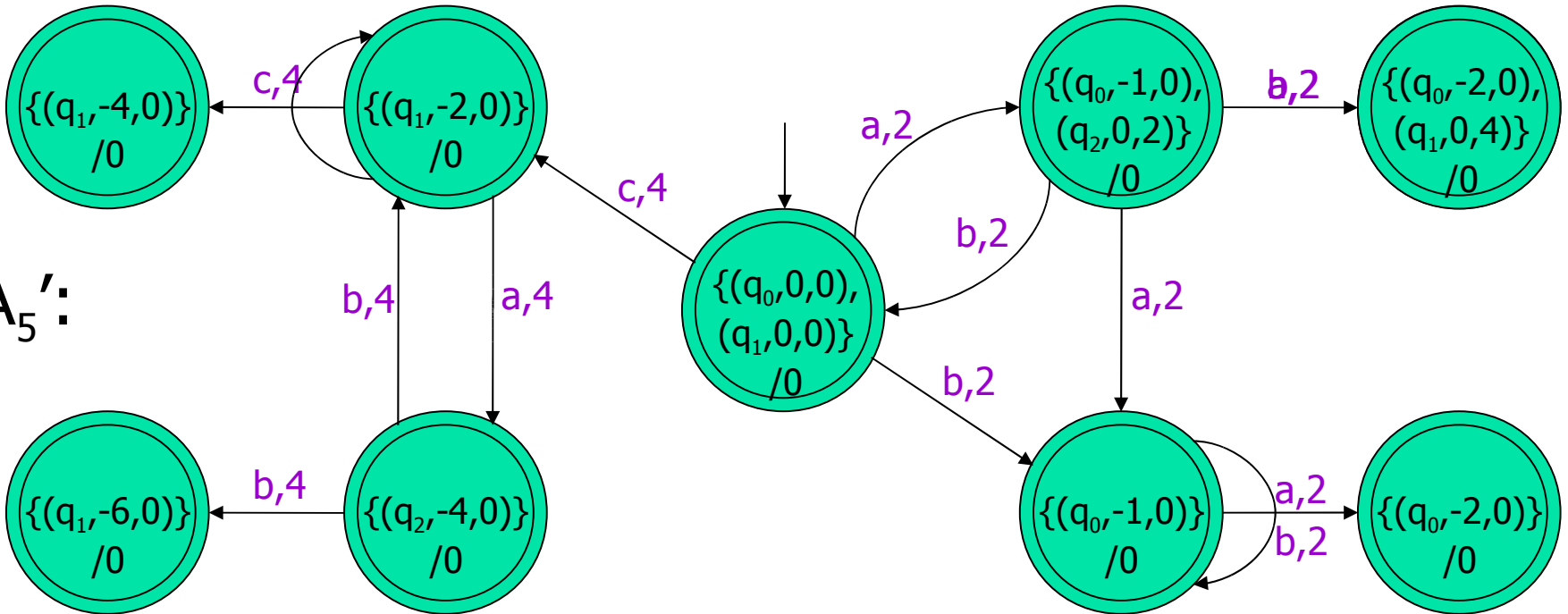


2-determinization of A_2

A_5 :



A_5' :





Correctness of the algorithm

- Thm: If the algorithm terminates on a given WFA A , with the result A' , then for every word w we have

$$\text{cost}(A,w) \leq \text{cost}(A',w) \leq t \cdot \text{cost}(A,w).$$



Termination of the algorithm

- Thm: If a WFA has the t-twins property, then the algorithm terminates on it.
 - The weights and the factor t are rational.
- Thm: For trim unambiguous WFAs, a WFA is t -determinizable iff it has the t-twins property.
- Thm: Deciding the t-twins property for trim unambiguous WFAs can be done in polynomial time.



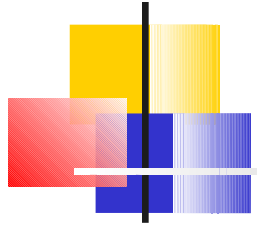
Summary

- Why approximate determinization?
 - Non-determinizable WFA
 - Equivalent deterministic is large
- t-determinization algorithm
 - Weights multiplied by t
 - Use ranges rather than single residues
 - Collapse to a state whose ranges are contained in mine
- A sufficient condition
 - The t-twins property
 - For unambiguous WFAs – characterizes determinizability
 - Decidable in polynomial time



Future work

- Generalize the termination proof to the case where the weights and the factor t are real numbers (\mathbf{R}^0).
- An algorithm to decide whether a WFA is determinizable. Alternatively – prove that it is undecidable.



Thank you!