SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

CSE211 – Formal Languages and Automata Theory

U1L10 – e-NFA to DFA Conversion

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Recap of previous class
- E-closure of a state in e-NFA
- Example for identifying e-closure
- Equivalence of FAs
- Converting e-NFA to DFA
- Eliminating epsilon from e-NFA
- Example for e-NFA to DFA conversion

Finite Automata with Epsilon-Transitions

- Epsilon-Closures (ε-closures)
  - We have to define the ε-closure to define the extended transition function for the ε-NFA
  - The "ε-closure" of a state *q* *is a set* by following all transitions out of *q* that are labeled ε
  - Formal recursive definition of **the set** ECLOSE(*q*) for *q*:
    - State *q* is in ECLOSE(*q*) (including the state itself);
    - If *p* is in ECLOSE(*q*), then all states accessible from *p* through paths of ε*'s* are also in ECLOSE(*q*)

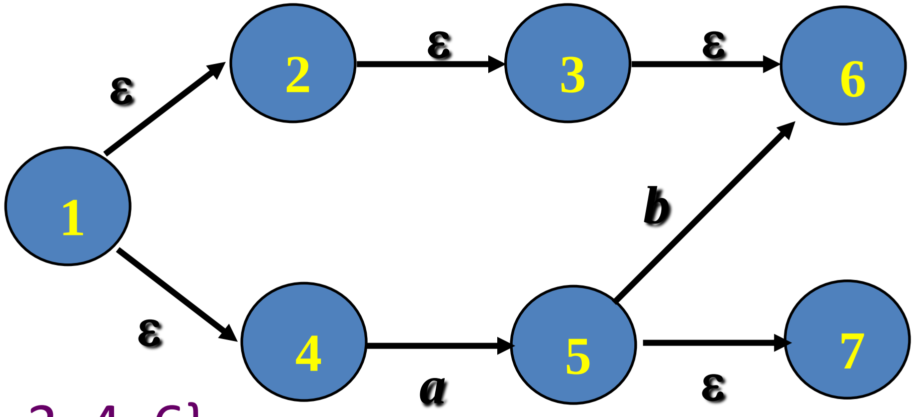
Finite Automata with Epsilon-Transitions

Epsilon-Closures

- ε-closure for a set of states S:

$$\mathrm{ECLOSE}(S) = \cup_{q \in S}\ \mathrm{ECLOSE}(q)$$

- Example 2.19



  - ECLOSE(1) = {1, 2, 3, 4, 6}
  - ECLOSE({3, 5}) = ECLOSE(3)∪ECLOSE(5) = {3, 6}∪{5, 7} = {3, 5, 6, 7}

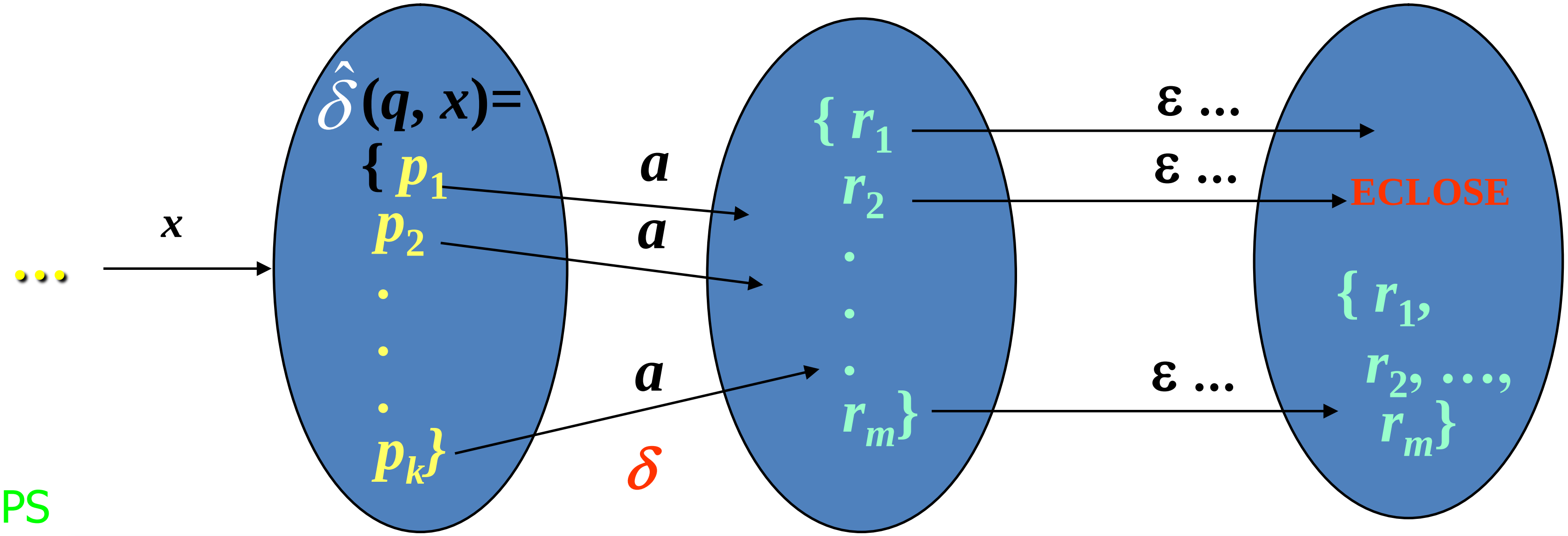
Dr.PS

Finite Automata with Epsilon-Transitions

- **Extended Transitions & Languages for ε-NFA’s**
  - Induction: if $w = xa$, then (q, w) is computed as:

    If $(q, x) = \{p_1, p_2, \ldots, p_k\}$ and $\delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$,

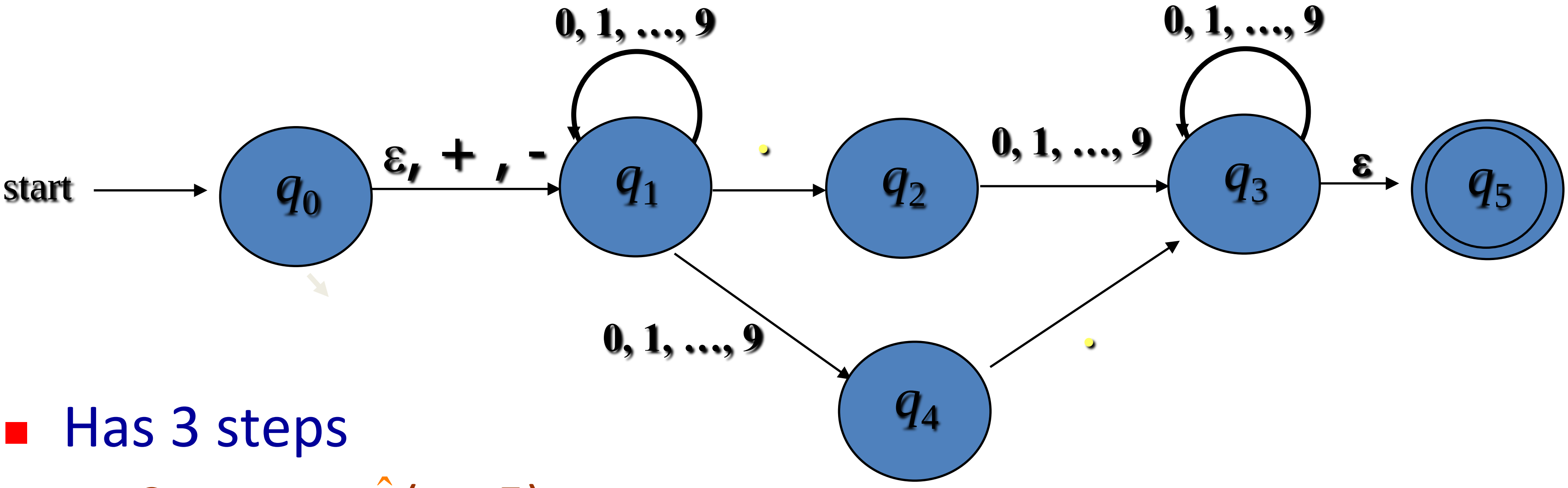
    then (q, w) = ECLOSE($\{r_1, r_2, \ldots, r_m\}$).



Dr.PS

Finite Automata with Epsilon-Transitions

- Computing $\hat{\delta}(q_0, 5.6)$ for ε-NFA

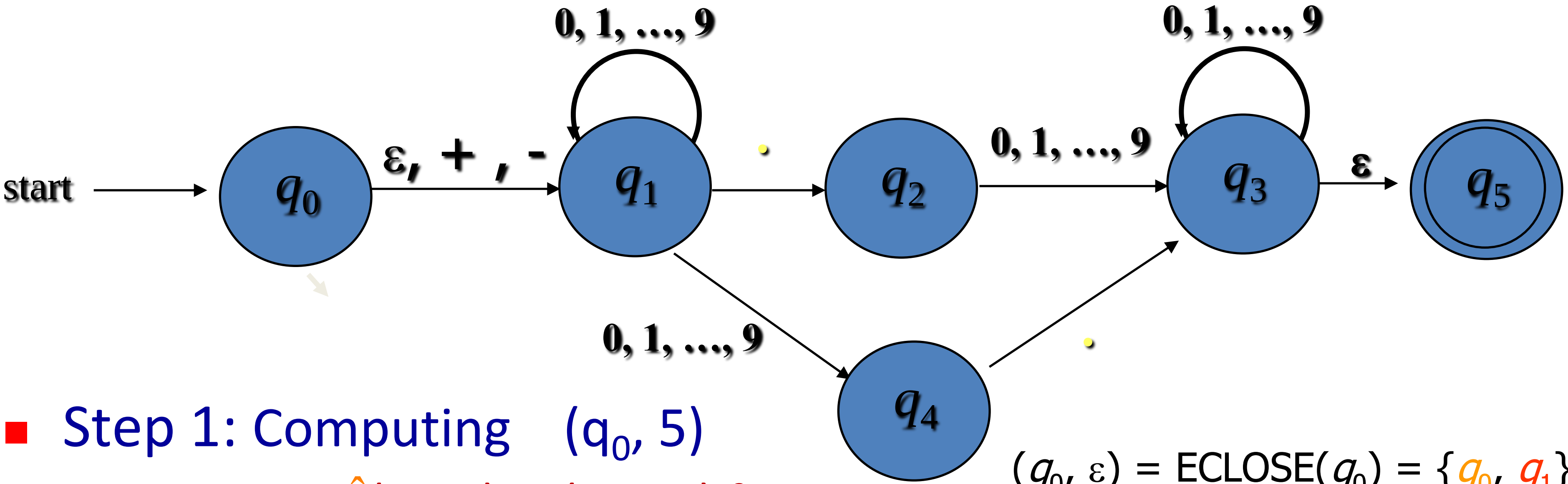


- Has 3 steps
  - Compute $\hat{\delta}(q_0, 5)$
  - Compute $\hat{\delta}(q_0, 5.)$
  - Compute $\hat{\delta}(q_0, 5.6)$

Dr.PS

Finite Automata with Epsilon-Transitions

- Computing $\hat{\delta}$ (q_0, 5.6) for ε-NFA



(q_0, ε) = ECLOSE(q_0) = {q_0, q_1}

- Step 1: Computing   (q_0, 5)
  - Compute $\hat{\delta}$(q_0, 5)=   (q_0, ε5) first:
  - $\hat{\delta}$ (q_0, ε5) = ECLOSE(δ(q_0, 5)∪δ(q_1, 5))

    = ECLOSE({q_1, q_4}) = ECLOSE({q_1})∪ECLOSE({q_4})

    = {q_1, q_4}

Converting e-NFA to DFA

- Eliminating ε-Transitions
  - The ε-transition is good for design of FA, but for implementation, they have to be eliminated
  - Given an ε-NFA, we can find an equivalent DFA (a theorem seen later).
  - Let $E = (Q_E, \Sigma, \delta_\mathrm{E}, q_0, F_E)$ be the given ε-NFA, the equivalent DFA $D = (Q_D, \Sigma, \delta_\mathrm{D}, q_D, F_D)$ is constructed as follows

Converting e-NFA to DFA

- Q_D is the set of subsets of Q_E, in which each accessible is an ε-closed subset of Q_E, i.e., are sets $S \subseteq Q_E$ such that S = ECLOSE(S).

  In other words, each ε-closed set of states, S, includes those states such that any ε-transition out of one of the states in S leads to a state that is also in S.

- q_D = ECLOSE(q_0) (initial state of D)
- $F_D = \{S \mid S \in Q_D \text{ and } S \cap F_E \neq \phi\}$

(continued in the next page)

Dr.PS

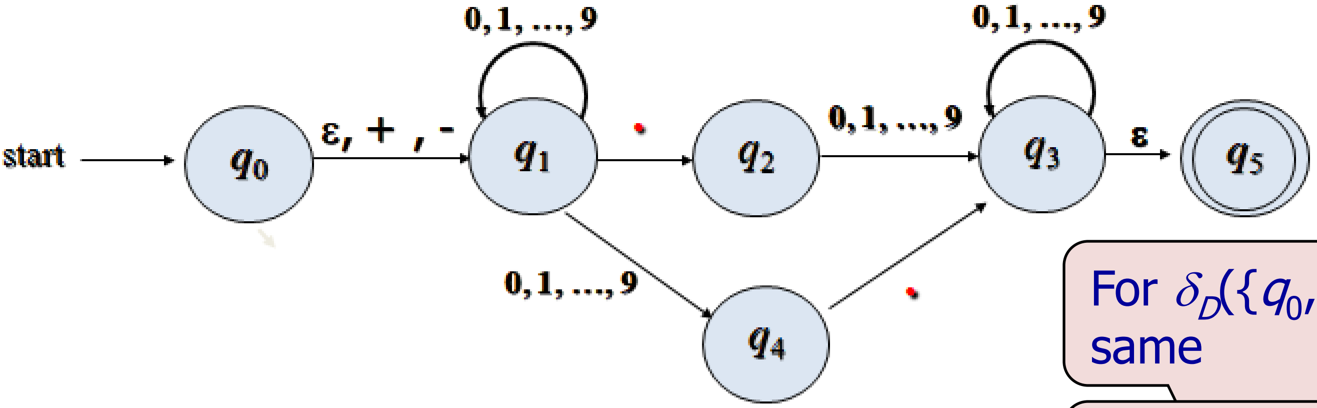
Converting e-NFA to DFA

- $\delta_D(S, a)$ is computed for each a in Σ and each S in Q_D in the following way:
  - Let $S = \{p_1, p_2, \ldots, p_k\}$
  - Compute $\delta(p_i, a)$ and let this set be $\{r_1, r_2, \ldots, r_m\}$
  - Set $\delta_D(S, a) = \text{ECLOSE}(\{r_1, r_2, \ldots, r_m\})$

$$= \text{ECLOSE}\left(\bigcup_{i=1}^{k} \delta(p_i, a)\right)$$

- Technique to create accessible states in DFA D:
  - starting from the start state q_0 of ε-NFA E, generate ECLOSE(q_0) as start state q_D of D;
  - from the generated states to derive other states.

Converting e-NFA to DFA



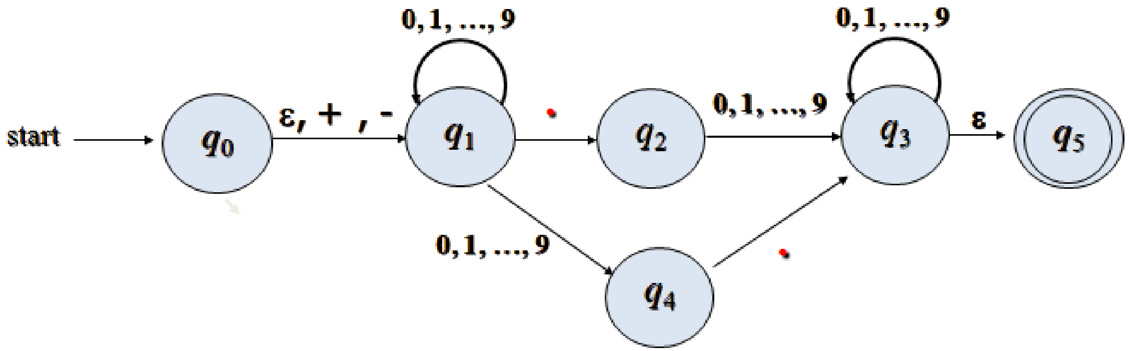
For $\delta_D(\{q_0, q_1\}, -)$ also the same

It is same for $\delta_D(\{q_0, q_1\}, 1)$ to $\delta_D(\{q_0, q_1\}, 9)$

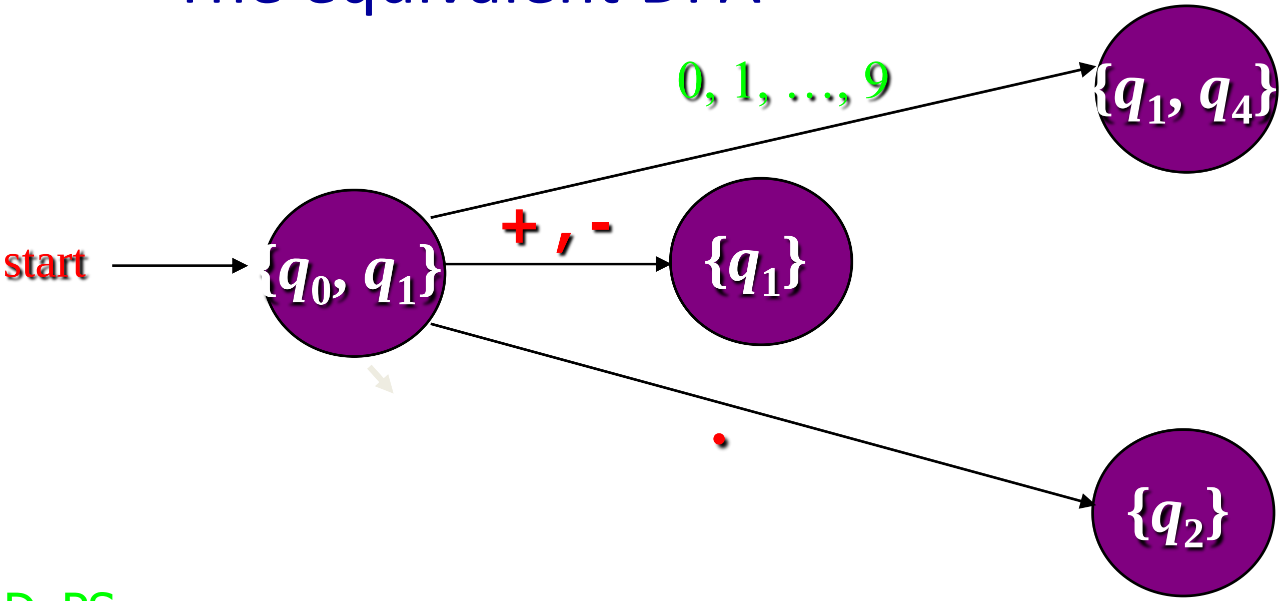
- Start state q_D = ECLOSE(q_0) = $\{q_0, q_1\}$
- $\delta_D(\{q_0, q_1\}, +)$ = ECLOSE($\delta_E(q_0, +) \cup \delta_E(q_1, +)$)
  = ECLOSE($\{q_1\} \cup \phi$) = ECLOSE($\{q_1\}$) = $\{q_1\}$, …
- $\delta_D(\{q_0, q_1\}, 0)$ = ECLOSE($\delta_E(q_0, 0) \cup \delta_E(q_1, 0)$)
  = ECLOSE ($\phi \cup \{q_1, q_4\}$) = ECLOSE($\{q_1, q_4\}$) = $\{q_1, q_4\}$, …
- $\delta_D(\{q_0, q_1\}, .)$ = ECLOSE($\delta_E(q_0, .) \cup \delta_E(q_1, .)$) = $\{q_2\}$

Dr.PS

Converting e-NFA to DFA

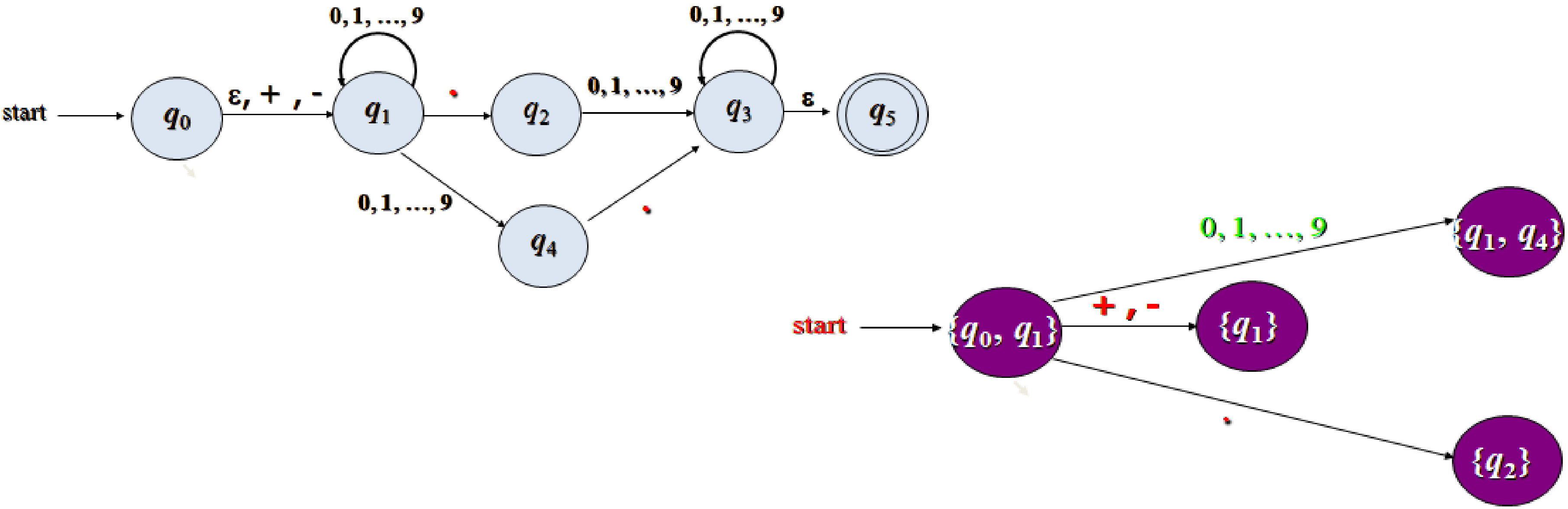


- The equivalent DFA



Dr.PS

Converting e-NFA to DFA



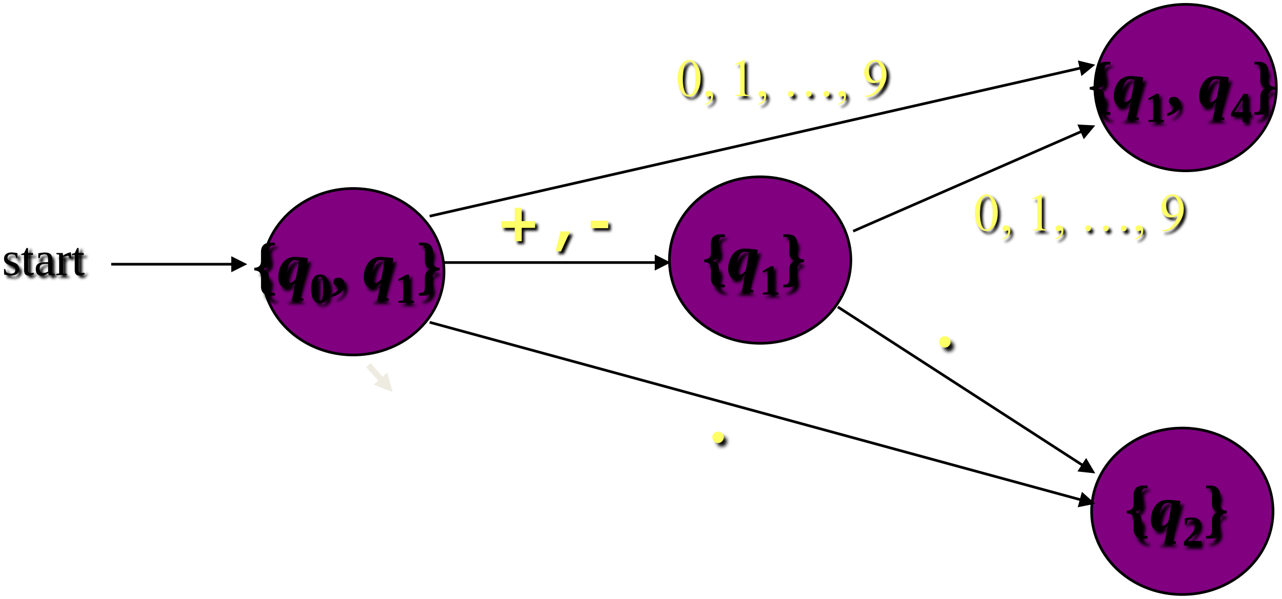
- $\delta_D(\{q_1\}, 0)$ = ECLOSE($\delta_E(q_1, 0)$) = ECLOSE($\{q_1, q_4\}$) = $\{q_1, q_4\}$…
- $\delta_D(\{q_1\}, .)$ = ECLOSE($\delta_E(q_1, .)$) = ECLOSE($\{q_2\}$) = $\{q_2\}$

Dr.PS

Converting e-NFA to DFA

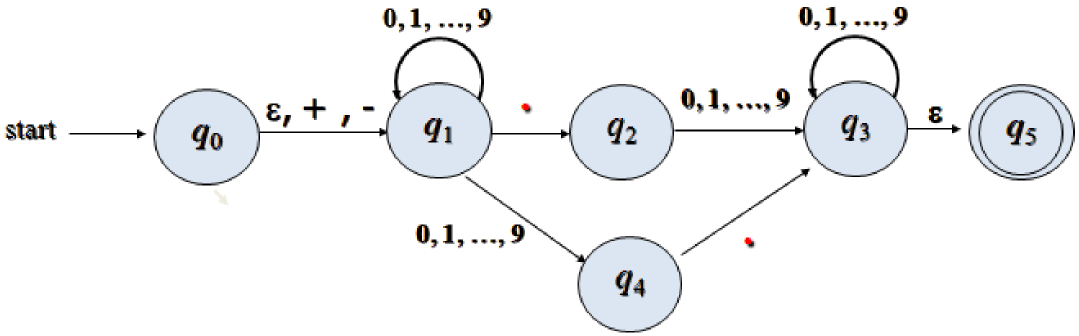
$\delta_D(\{q_1\}, 0) = \{q_1, q_4\}$…

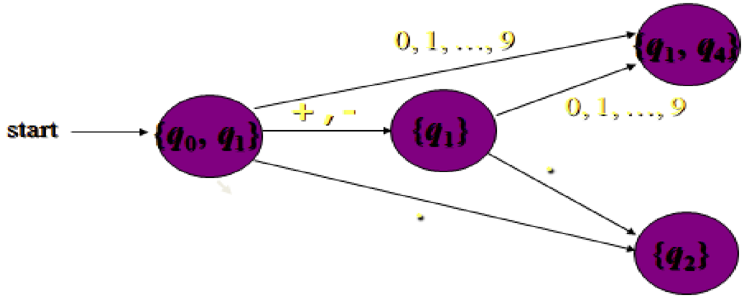
$\delta_D(\{q_1\}, .) = \{q_2\}$



Dr.PS

Converting e-NFA to DFA



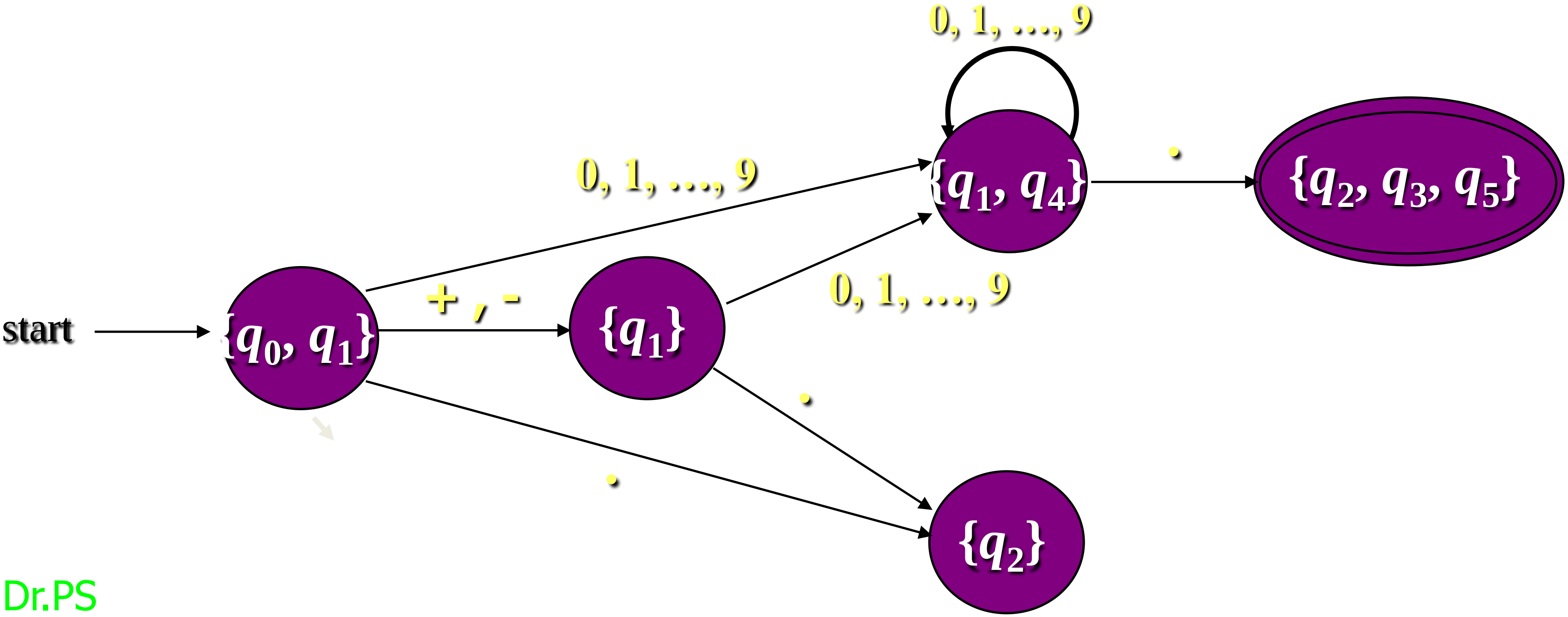


- $\delta_D(\{q_1, q_4\}, 0) = \text{ECLOSE}(\delta_E(q_1, 0) \cup \delta_E(q_4, 0))$ $= \text{ECLOSE}(\{q_1, q_4\} \cup \phi) = \{q_1, q_4\}\ldots$
- $\delta_D(\{q_1, q_4\}, .) = \text{ECLOSE}(\delta_E(q_1, .) \cup \delta_E(q_4, .)) =$ $\text{ECLOSE}(\{q_2\} \cup \{q_3\}) = \text{ECLOSE}(q_2) \cup \text{ECLOSE}\ (q_3)$ $= \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$

Converting e-NFA to DFA

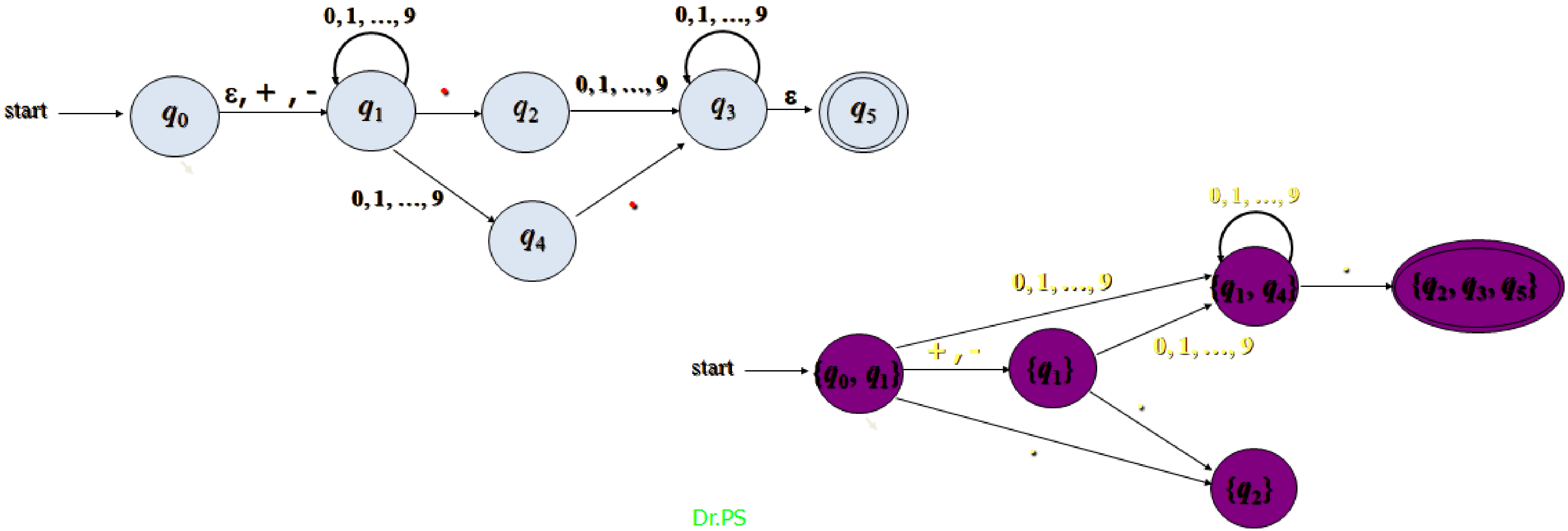
$\delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$...

$\delta_D(\{q_1, q_4\}, .) = \{q_2, q_3, q_5\}$



Dr.PS

Converting e-NFA to DFA

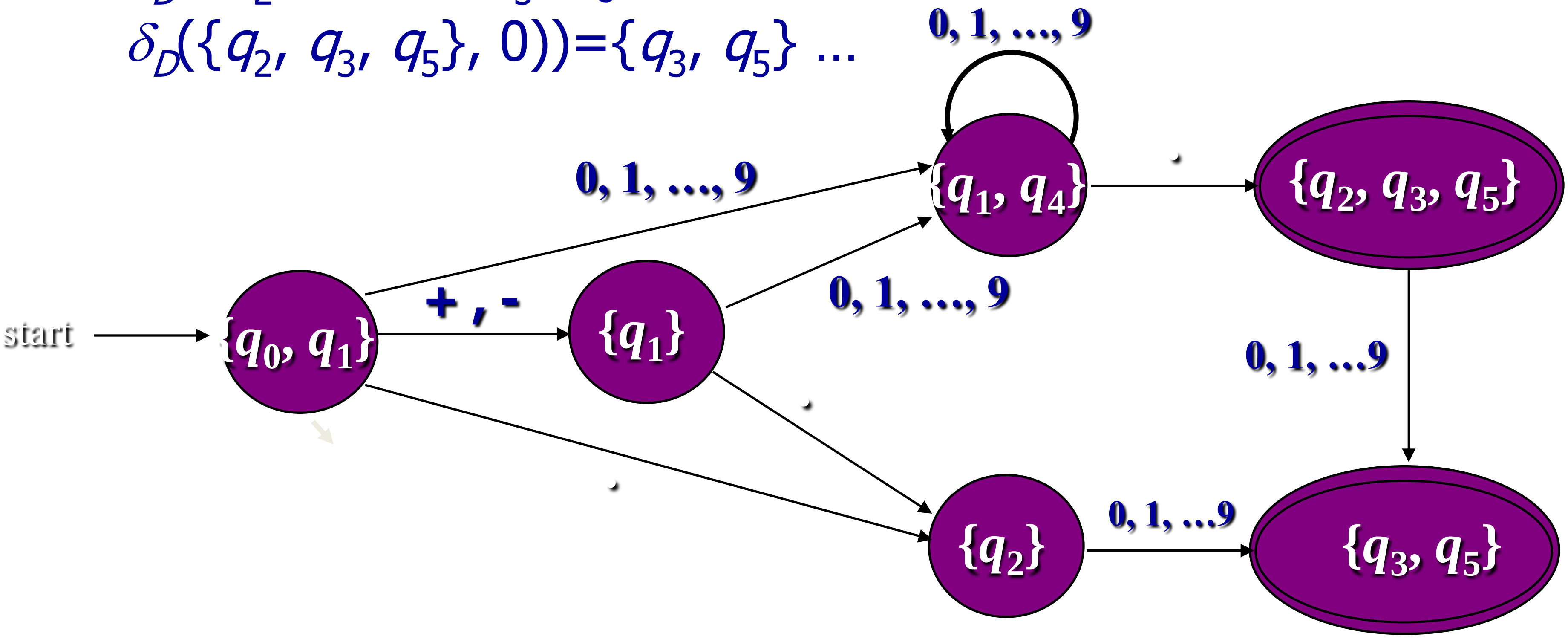


- $\delta_D(\{q_2\}, 0) = \text{ECLOSE}(\delta_E(q_2, 0)) = \text{ECLOSE}(\{q_3\}) = \{q_3, q_5\}$ …
- $\delta_D(\{q_2, q_3, q_5\}, 0) = \text{ECLOSE}(\delta_E(q_2, 0) \cup \delta_E(q_3, 0) \cup$
  $\delta_E(q_5, 0)) = \text{ECLOSE}(\{q_3\} \cup \{q_3\} \cup \phi) = \text{ECLOSE}(q_3) = \{q_3, q_5\}$ …

Converting e-NFA to DFA

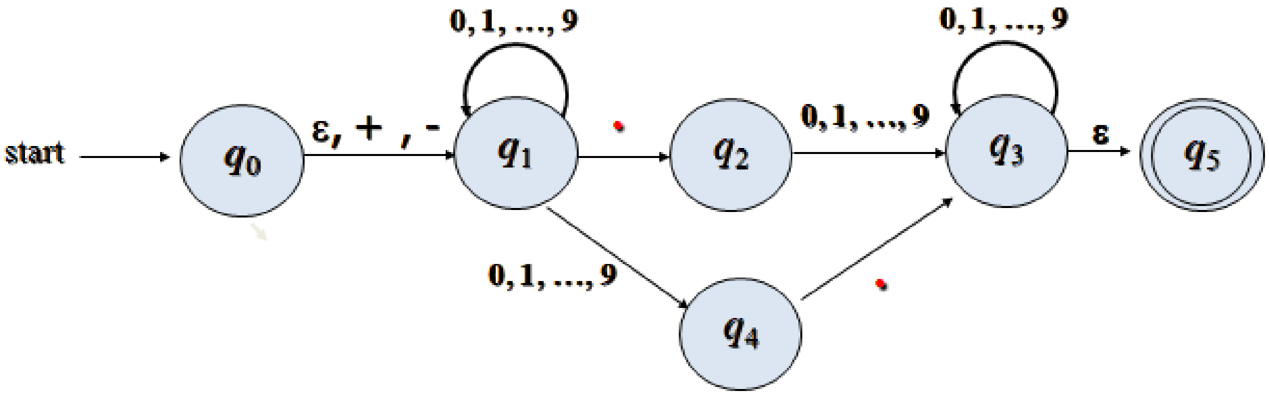
$\delta_D(\{q_2\}, 0) = \{q_3, q_5\}$ ...

$\delta_D(\{q_2, q_3, q_5\}, 0))=\{q_3, q_5\}$ ...



Dr.PS

Converting e-NFA to DFA

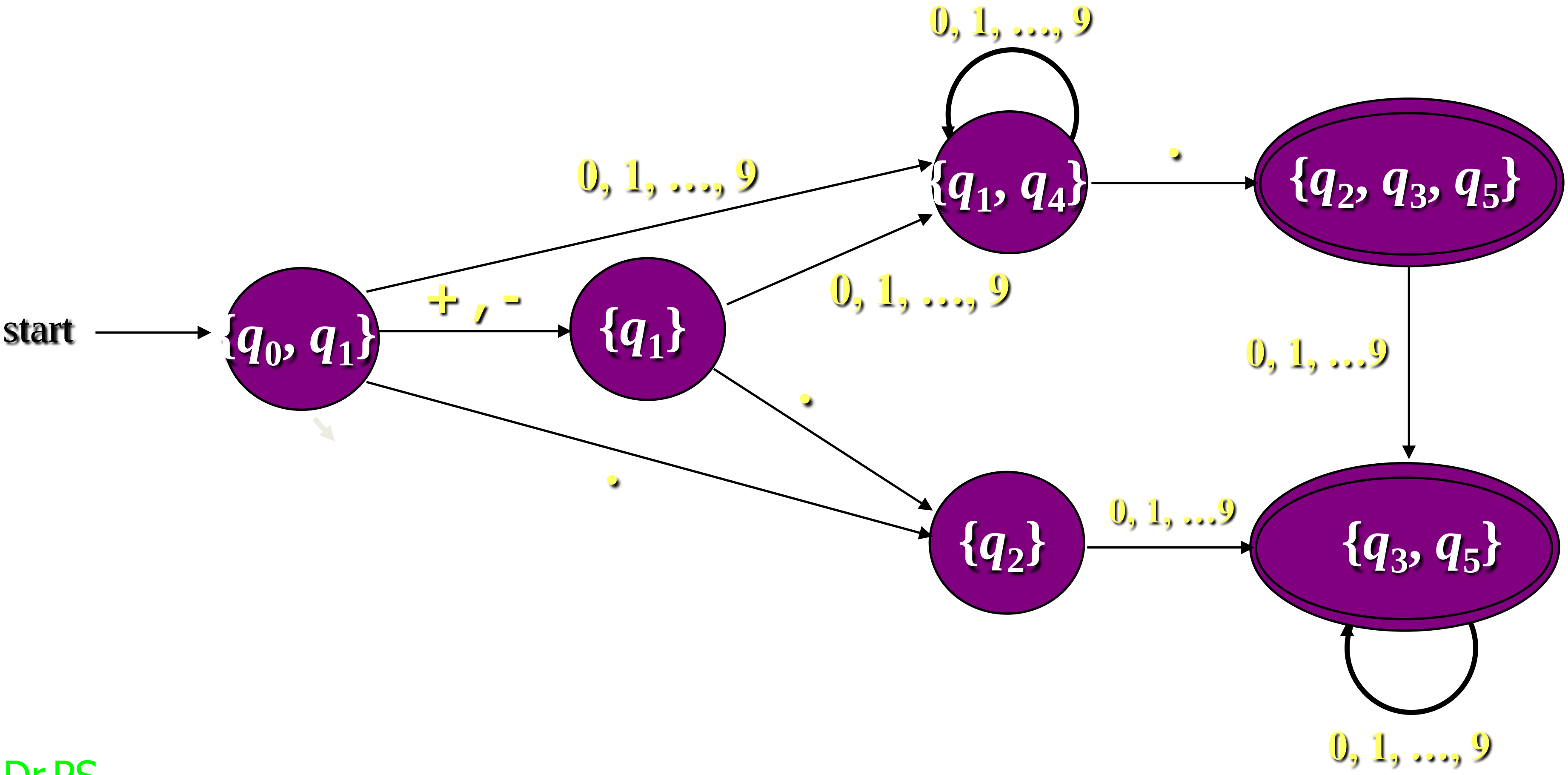


- $\delta_D(\{q_3, q_5\}, 0)$ = ECLOSE($\delta_E(q_3, 0) \cup \delta_E(q_5, 0)$) =ECLOSE($\{q_3\} \cup \phi$) =ECLOSE($q_3$)=$\{q_3, q_5\}$ …

Dr.PS

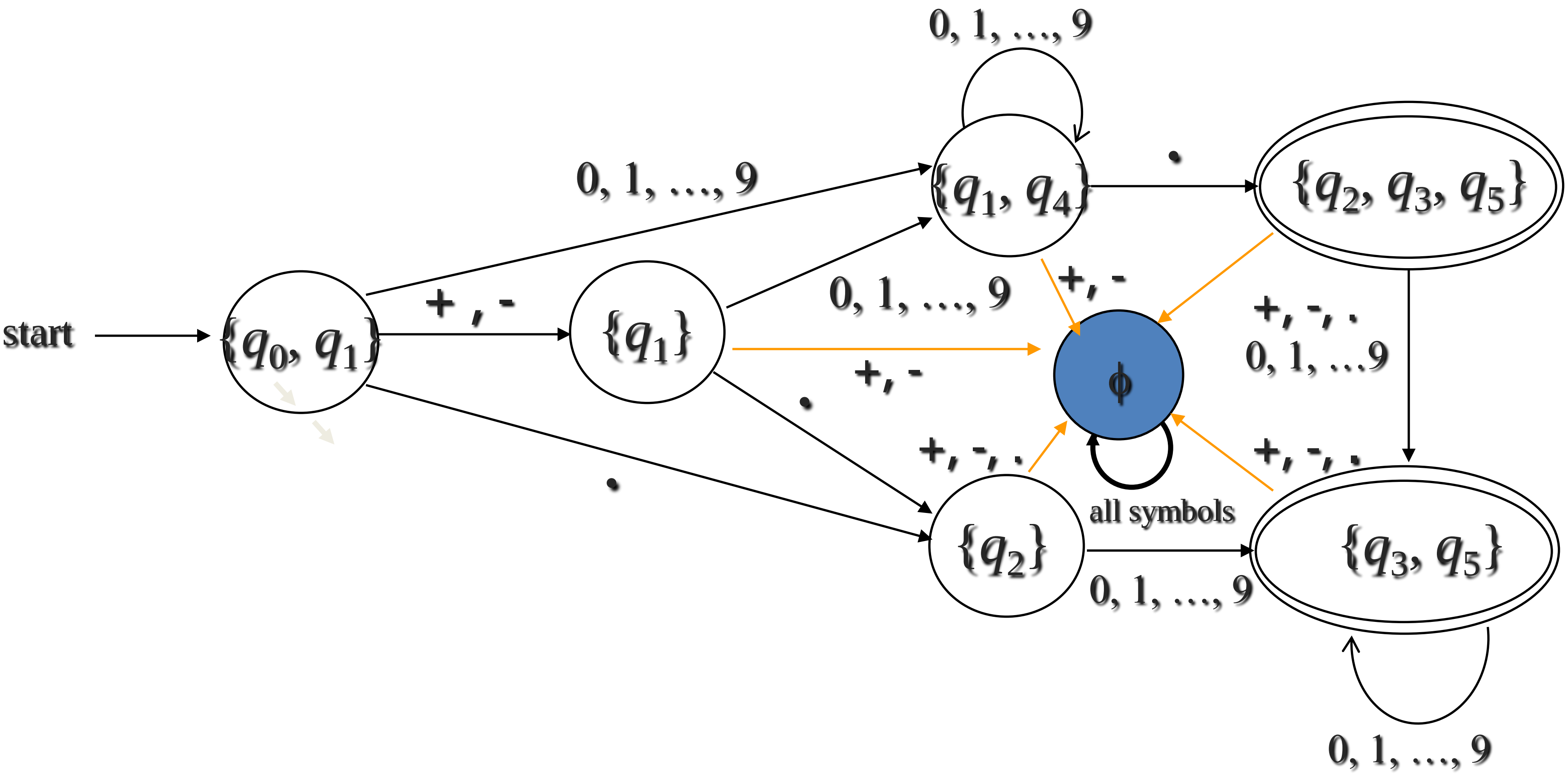
Converting e-NFA to DFA

$\delta_D(\{q_3, q_5\}, 0) = \{q_3, q_5\}$ …



Dr.PS

Final equivalent DFA is…



Dr.PS

Theorem

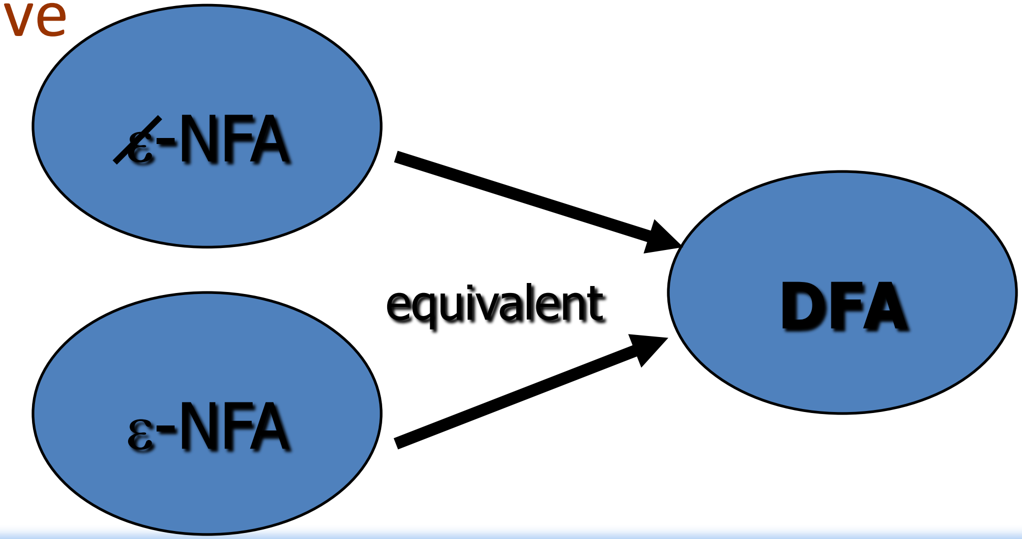
- **Theorem 2.22**
  - A language *L* accepted by some ε-NFA if and only if *L* is accepted by some DFA
  - Proof: see the textbook yourself.

Summary

- Equivalence of FAs
- Converting e-NFA to DFA
- Eliminating epsilon from e-NFA
- Example for e-NFA to DFA conversion

Review

- 3 Types of Automata
  - DFA ---good for soft/hardware implementation
    - $\delta: \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ is the transition function
  - NFA ---intermediately intuitive
    - $\delta: \mathcal{Q} \times \Sigma \rightarrow 2^{\mathcal{Q}}$ is the transition function
  - ε-NFA ---most intuitive
    - $\delta: \mathcal{Q} \times \Sigma \cup \{\varepsilon\} \rightarrow 2^{\mathcal{Q}}$ is the transition function



Summary

- What is Non-deterministic Finite Automata(NFA)?
- Examples for NFA
- Definition of NFA
- Epsilon-NFA
- E-closure of a state in e-NFA
- Example for identifying e-closure

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

Converting NFA or e-NFA to DFA

THANK YOU.