

## 4.2

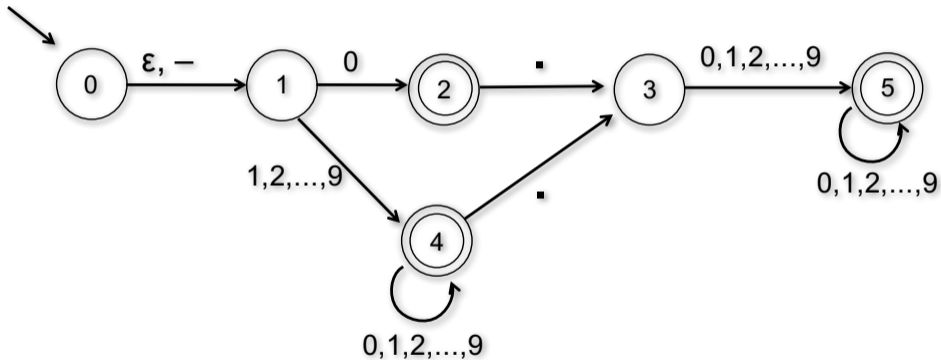
# Constructing NFAs

# DFAs and NFAs

- Every **DFA** is a **NFA** so **NFAs** are at least as powerful as **DFAs**.
- **NFAs** prove ability to “guess and verify” which simplifies design and reduces number of states
- Easy proofs of some closure properties

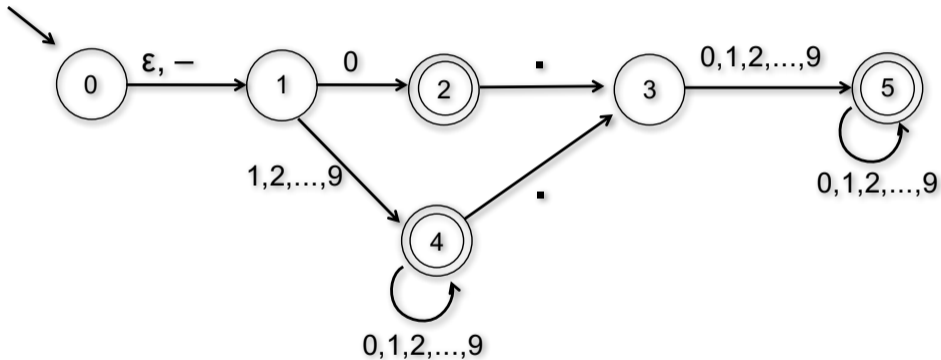
# Example

Strings that represent decimal numbers.



# Example

Strings that represent decimal numbers.



# Example

- {strings that contain CS374 as a substring}
- {strings that contain CS374 or CS473 as a substring}
- {strings that contain CS374 and CS473 as substrings}

# Example

- {strings that contain CS374 as a substring}
- {strings that contain CS374 or CS473 as a substring}
- {strings that contain CS374 and CS473 as substrings}

# Example

- {strings that contain CS374 as a substring}
- {strings that contain CS374 or CS473 as a substring}
- {strings that contain CS374 and CS473 as substrings}

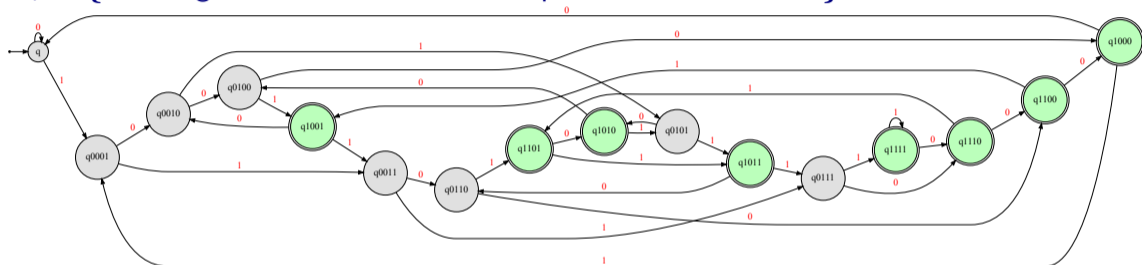
# Example

$L_k = \{\text{bitstrings that have a 1 } k \text{ positions from the end}\}$



# DFA for same task is much bigger...

$L_4 = \{\text{bitstrings that have a 1 in fourth position from the end}\}$



# A simple transformation

## Theorem

For every NFA  $N$  there is another NFA  $N'$  such that  $L(N) = L(N')$  and such that  $N'$  has the following two properties:

- $N'$  has single final state  $f$  that has no outgoing transitions
- The start state  $s$  of  $N$  is different from  $f$

# THE END

...

# (for now)