Finite-State Automata

Recall: Any computational question framed as a language

Today: We will look at a basic "computing machine"

Problem: Check if a given word is a program keyword

int, bool, if, else, end, while

What does a tokenizer do to solve this problem?

Examines each letter of the word, takes an appropriate decision

E.g: If the word starts with 'c': not a keyword

If the word starts with 'w': see if it ends with 'hile'

If the word starts with 'i': ?

So what is the language of interest?

Before defining any language, one has to specify the alphabet

What is the alphabet here? The Roman alphabet etc. Σ

A-Za-z, 0-9, punctuation

What is the language of interest over this alphabet?

$\mathcal{L}$ = {int, bool, if, else, while, end} $\subseteq$ all strings over Σ

What are we implicitly keeping track of?

The initial subsequence seen so far

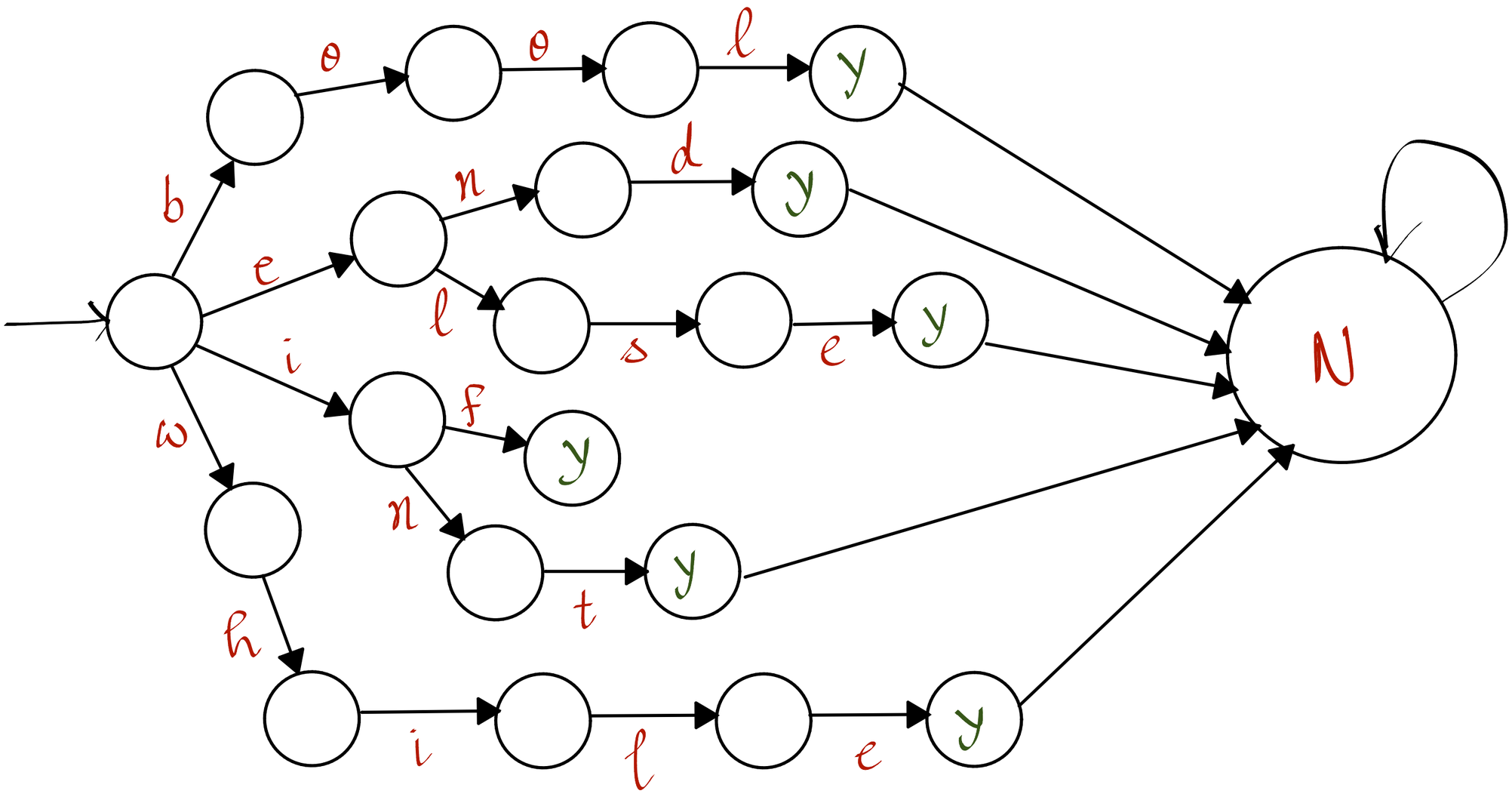
Do we care exactly what subsequence it is? No!

Only whether it matches some keyword or not.

How would we implement this as a program?

(Suppose a word is delimited by a space)

Essentially, our program has the following control flow



Each state not marked by N tracks a "candidate" initial subsequence on the way to a y

This is a Deterministic Finite Automaton recognizing L.

A state goes to exactly one other state on a letter

Number of states

Suppose we had to describe this automaton to someone

What information would we need to provide?

Alphabet

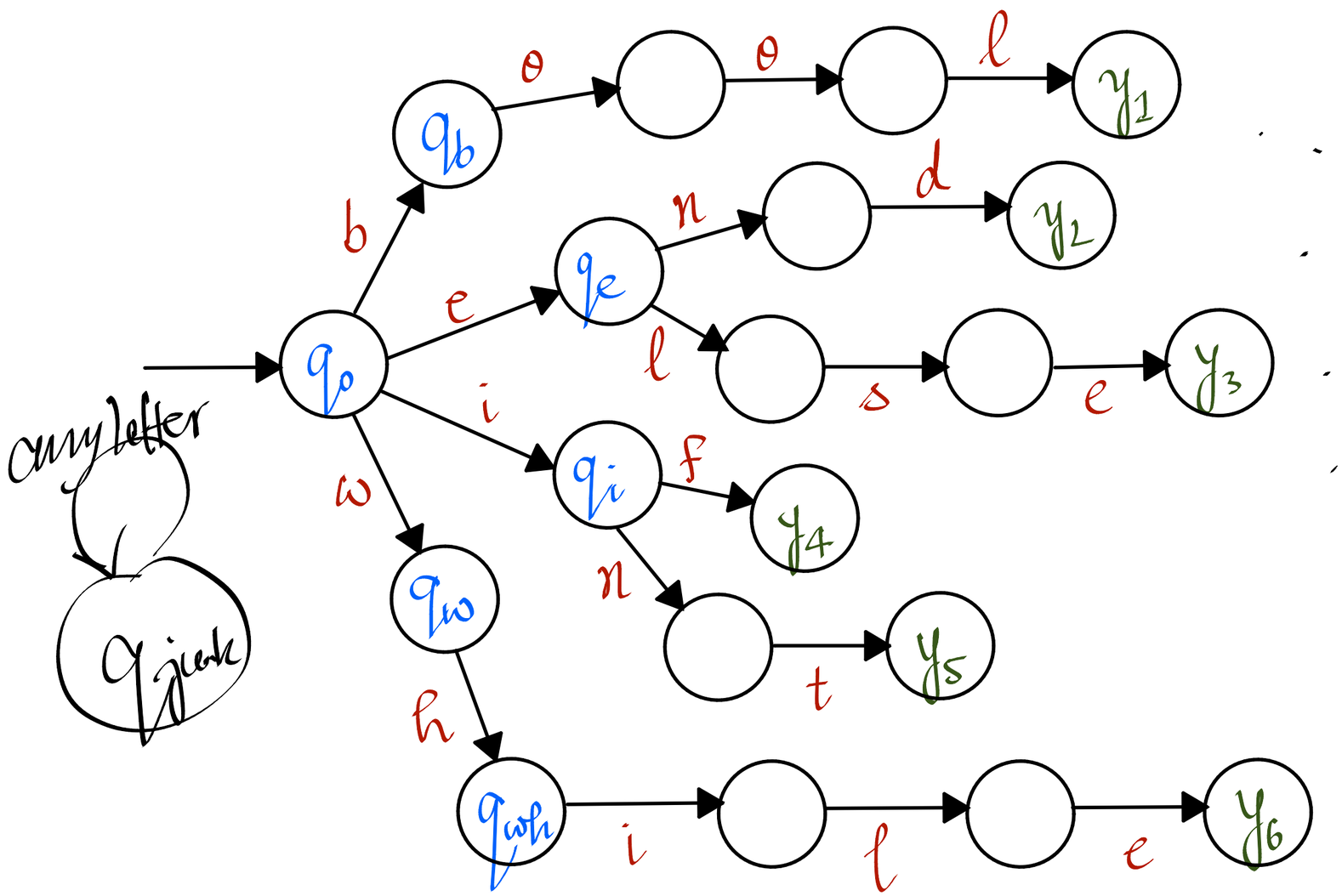
List of states

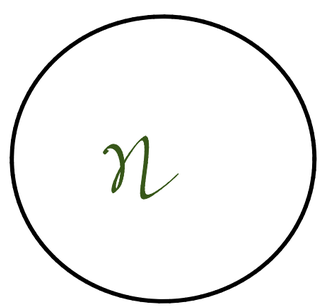
transition function

start, finish

$$M = (Q, \Sigma, \delta, q_0, F)$$

- Q: set of states
- Σ: alphabet
- δ: $\delta : Q \times \Sigma \to Q$ transition function
- q_0: $q_0 \in Q$ initial state
- F: $F \subseteq Q$ set of accepting states





$$\delta(q_w, h) = q_{wh}$$

$$\delta(q_i, l) = n$$

$$\delta(q_0, if) = y_4 \rightsquigarrow \delta(\delta(q_0, i), f) = y_4$$

Suppose we modify the problem slightly

Suppose I want to check if a given word is not a keyword

What is the language for this problem?

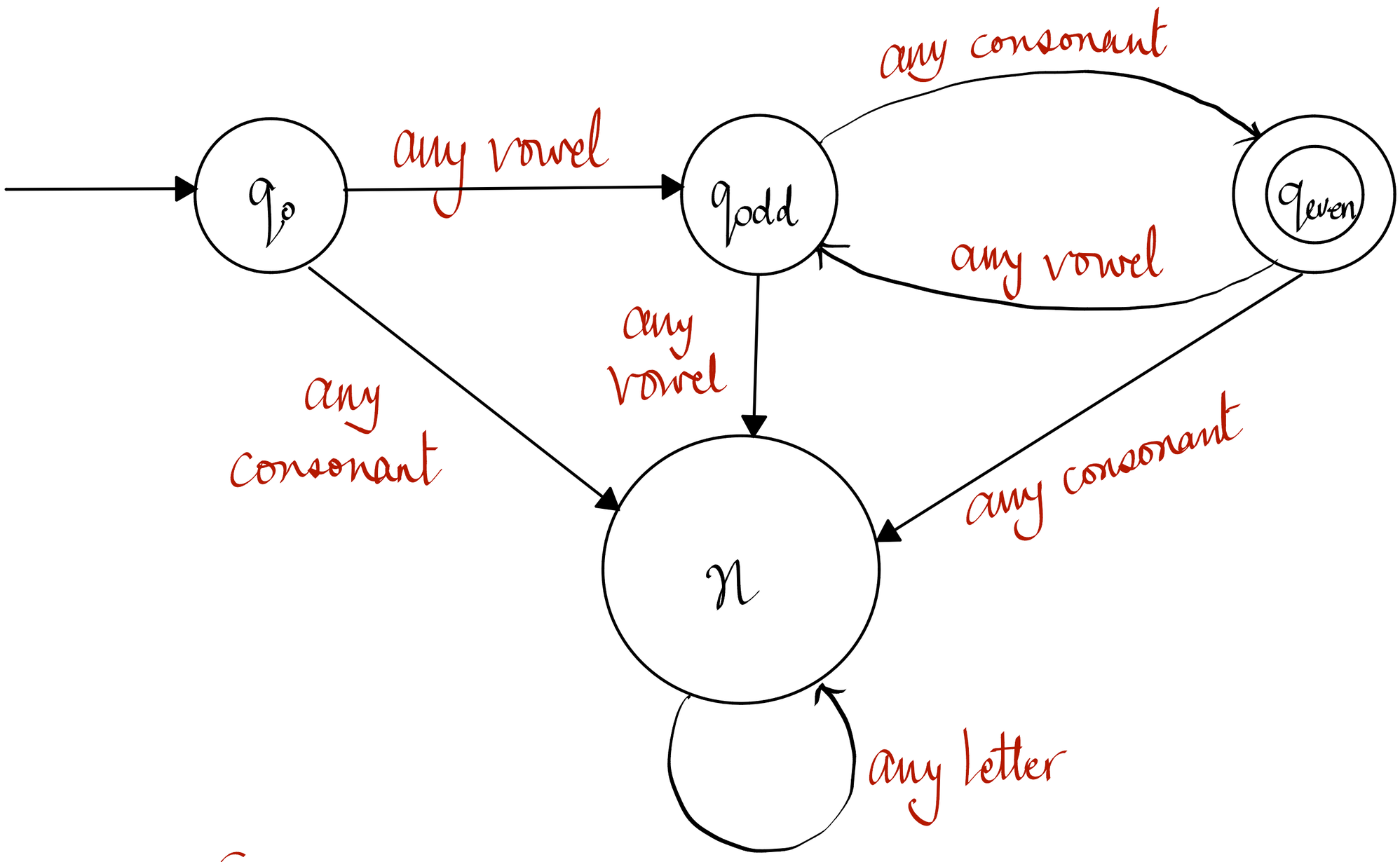
How many strings are in this language?

Is there a DFA which recognizes this new language?

$$M = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

Consider the following language:

All words of even length, with vowels at odd positions,
and consonants at non-zero even ones



$$M = (\{q_0, q_{odd}, q_{even}, n\}, \{A\text{-}Za\text{-}z\}, \delta, q_0, \{q_{even}\})$$