Computer Science 351

Introduction to Deterministic Finite Automata

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #2

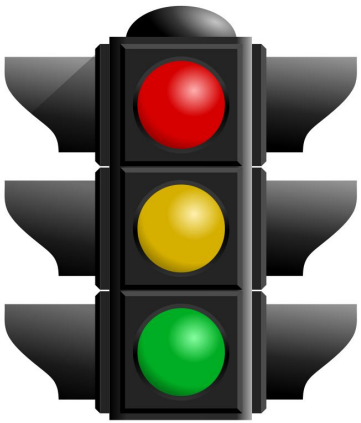
Goals for Today

Goal for Today:

- Introduction of ***deterministic finite automata*** — the simple (abstract) computational devices that will be considered at the beginning of this course

DFA's — An Example

Consider how a simple ***traffic light*** works. For now, let's just worry about the light being shone in *one* direction.

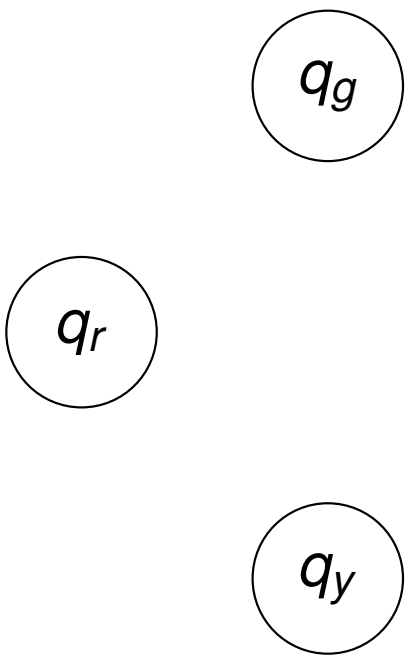


Suppose this light receives (as input) a series of *tick*'s from a timer. Let's pretend that no time — at all — is needed to process an input signal and change the light's colour from one thing to another.

DFA's — An Example

At any point, this light is in of of three ***states***:

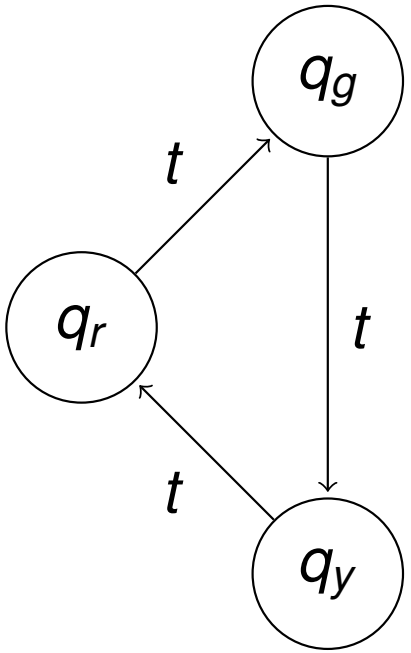
- q_r: The traffic light is currently *red*.
- q_g: The traffic light is currently *green*.
- q_y: The traffic light is currently *yellow*.



DFA's — An Example

The light receives a sequence of signals. One might call these "ticks", but we will abbreviate this to "t".

- In this simple kind of traffic night, processing a tick changes the colour's light from red to green, from green to yellow, or from yellow to red — so it would cause a change in state from q_r to g_g, from g_g to q_y, or from q_y to q_r.



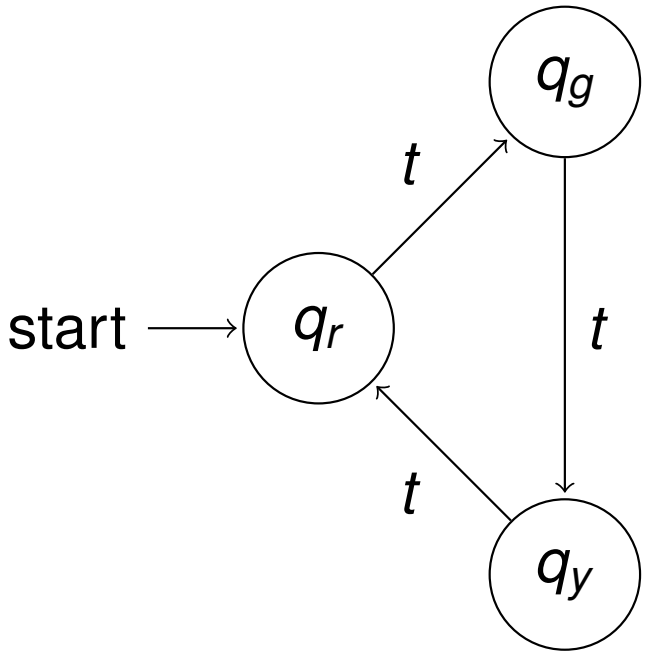
DFA's — An Example

These rules about changes can also be given using a two-dimensional table:

	t
q_r	q_g
q_g	q_y
q_y	q_r

DFA's — An Example

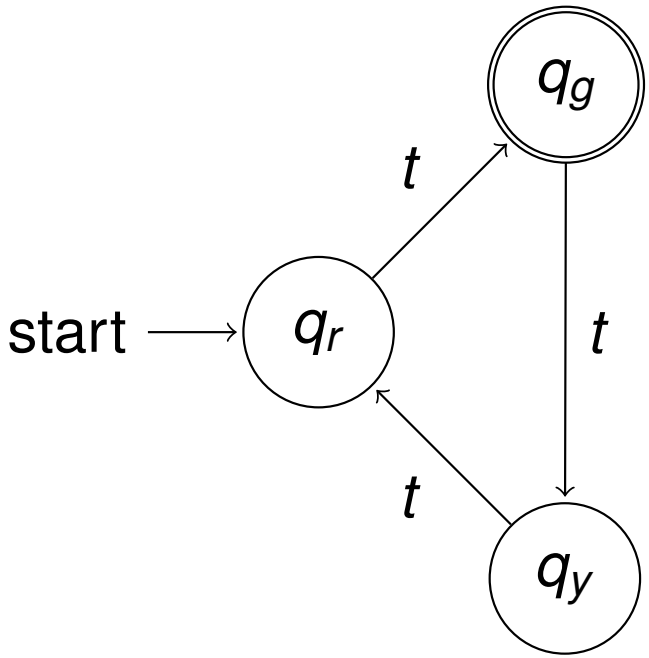
Suppose that, when it is first activated, the traffic light shows the colour red. Then its *initial state* (or *start state*) is q_r:



Now one can imagine the state reached after processing a *sequence* of signals — that is, after processing a string in $\Sigma^\star$ when $\Sigma = \{t\}$.

DFA's — An Example

If we are using this to keep track of the sequences of symbols whose processing ends with the colour of the light being green, then q_g is an *accepting state* (or *final state*).



Formal Definition

Definition: A ***deterministic finite automaton (DFA)*** is (formally modelled as) a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite, nonempty, set of ***states***;
- Σ is an ***alphabet*** such that $Q \cap \Sigma = \emptyset$;
- $\delta : Q \times \Sigma \to Q$ is a ***transition function***;
- $q_0 \in Q$ is the ***start state***; and
- $F \subseteq Q$ is the set of ***accept states***.

In particular, δ is a ***total*** function from $Q \times \Sigma$ to Q.

Example

Example: The "traffic light" DFA is (formally modelled as) a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where

- $Q = \{q_r, q_g, q_y\}$,
- $\Sigma = \{t\}$,
- $\delta : Q \times \Sigma \to Q$, such that

$$\delta(q_r, t) = q_g, \quad \delta(q_g, t) = q_y, \quad \text{and} \quad \delta(q_y, t) = q_r,$$

- q_0 is the state $q_r \in Q$, and
- $F = \{q_g\}$.

Extended Transition Function

The ***extended transition function*** is a function

$$\delta^\star : Q \times \Sigma^\star \to Q$$

describing the state that can be reached from a given state when processing a *string* — that is, *sequence* of symbols in Σ: For every state $q \in Q$ and every string $\omega \in \Sigma^\star$,

$$\delta^\star(q, \omega) = \begin{cases} q & \text{if } \omega = \lambda, \\ \delta(\delta^\star(q, \mu), \sigma) & \text{if } \omega = \mu \cdot \sigma \text{ for a string } \mu \in \Sigma^\star \\ & \text{and symbol } \sigma \in \Sigma. \end{cases}$$

Evaluating the Extended Transition Function

Two Ways to Evaluate $\delta^\star(q, \omega)$:

1. Trace execution, keeping track of states reached as each symbol in ω (read from left to right) is processed
2. Apply the recursive definition directly.

Language of a DFA

Suppose that

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

Then, for every string $\omega \in \Sigma^\star$, M ***accepts*** ω if and only if

$$\delta^\star(q_0, \omega) \in F,$$

and M ***rejects*** ω otherwise.

The ***language*** $L(M)$ of an automaton M is the set of strings $\omega \in \Sigma^\star$ such that M accepts ω.

Languages

Example: Consider the "traffic light" DFA.

- Since the start state is q_r and $F = \{q_g\}$, it can be shown that the language of this DFA is the set of input strings whose processing leaves the light with the colour green — which is

$$\{t^{3k+1} \mid k \in \mathbb{N}\} = \{t^{\ell} \mid \ell \in \mathbb{N} \text{ and } \ell \equiv 1 \bmod 3\}$$

  — where $\mathbb{N}$ is the set of *nonnegative* integers.

Regular Languages

Definition: For every alphabet Σ, a language $L \subseteq \Sigma^\star$ is a ***regular language*** if (and only if) $L = L(M)$ for some deterministic finite automaton M with alphabet Σ.

Activities

Sooner:

- Lecture presentation
- Tutorial exercise

Later:

- Design and Verification
- Variations and Applications
- Proving that a language is ***not*** regular