COMP2012/G52LAC
Languages and Computation
Lecture 2

Deterministic Finite Automata (DFA)

Henrik Nilsson

University of Nottingham

Recap: Formal Languages

The terms ***language*** and ***word*** are used in a strict technical sense in this course:

- A ***language*** is a (possibly infinite) set of words.
- A ***word*** is a ***finite*** sequence (or string) of symbols.

ϵ denotes the ***empty word***, the sequence of zero symbols.

The term ***string*** is often used interchangeably with the term ***word***.

Recap: Symbols and Alphabets

A symbol can be anything, but has to come from an ***alphabet*** Σ which is a ***finite*** set.

A common (and important) instance is $\Sigma = \{0, 1\}$.

ϵ, the empty word, is ***never*** a symbol of an alphabet.

Recap: Examples of Languages

Some examples of languages:

- The set $\{0010, 00000000, \epsilon\}$ is a language over $\Sigma = \{0, 1\}$.
  This is an example of a ***finite*** language.
- The set of words with odd length over $\Sigma = \{1\}$. (Finite or infinite?)
- The set of words that contain the same number of 0s and 1s is a language over $\Sigma = \{0, 1\}$. (Finite or infinite?)

All Words Over an Alphabet (1)

Given an alphabet Σ we define the set Σ^* as set of words (or sequences) over Σ:

- The empty word $\epsilon \in \Sigma^*$.
- given a symbol $x \in \Sigma$ and a word $w \in \Sigma^*$, $xw \in \Sigma^*$.
- These are all elements in Σ^*.

This is called an ***inductive definition***.

Is Σ^* always non-empty? Always infinite?

Recap: Language Membership

Fundamental question for a language L: $w \in L$?

- L finite: Easy! (Enumerate L and check)
- L infinite: ?

We need:

- A ***finite*** (and preferably concise) formal ***description*** of L.
- An algorithmic ***method to decide*** if $w \in L$ given a suitable description.

Various approaches to achieve this will be key a theme throughout the module.

Formal Definition of DFA

Formally, a ***Deterministic Finite Automaton*** or ***DFA*** is defined by a 5-tuple

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

where

Q	:	***Finite*** set of States
Σ	:	Alphabet (finite set of symbols)
$\delta \in Q \times \Sigma \to Q$	:	Transition Function
$q_0 \in Q$	:	Initial or Start State
$F \subseteq Q$	:	Accepting (or Final) States

Extended Transition Function

The ***Extended Transition Function*** is defined on a state and a ***word*** (string of symbols) instead of on a single symbol.

For a DFA $A = (Q, \Sigma, \delta, q_0, F)$, the extended transition function is defined by:

$$\begin{aligned} \hat{\delta} &\in Q \times \Sigma^* \to Q \\ \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, xw) &= \hat{\delta}(\delta(q, x), w) \end{aligned}$$

where $q \in Q$, $x \in \Sigma$, $w \in \Sigma^*$.

Language of a DFA

The ***language*** $L(A)$ defined by a DFA A is the set or words ***accepted*** by the DFA. For a DFA

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

the language is defined by

$$L(A) = \{\ w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\ \}$$