Languages and Algorithms for Artificial Intelligence (Module 3)

Last update: 20 February 2024

Academic Year 2023 – 2024
Alma Mater Studiorum · University of Bologna

Contents

1 Introduction

Computational task Description of a problem. Computational task

Computational process Algorithm to solve a task. Computational process

Algorithm (informal) A finite description of elementary and deterministic computation steps.

1.1 Notations

Set of the first n natural numbers Given $n \in \mathbb{N}$, we have that $[n] = \{1, \ldots, n\}$.

1.1.1 Strings

Alphabet Finite set of symbols. Alphabet

String Finite, ordered, and possibly empty tuple of elements of an alphabet. String

The empty string is denoted as ε.

Strings of given length Given an alphabet S and $n \in \mathbb{N}$, we denote with S^n the set of all the strings over S of length n.

Kleene star Given an alphabet S, we denote with $S^* = \bigcup_{n=0}^{\infty} S^n$ the set of all the strings over S. Kleene star

Language Given an alphabet S, a language $\mathcal{L}$ is a subset of S^*. Language

1.1.2 Tasks encoding

Encoding Given a set A, any element $x \in A$ can be encoded into a string of the language $\{0, 1\}^*$. The encoding of x is denoted as $\llcorner x \lrcorner$ or simply x. Encoding

Task function Given two countable sets A and B representing the domain, a task can be represented as a function $f : A \to B$. Task

When not stated, A and B are implicitly encoded into $\{0, 1\}^*$.

Characteristic function Boolean function of form $f : \{0, 1\}^* \to \{0, 1\}$. Characteristic function

Given a characteristic function f, the language $\mathcal{L}_f = \{x \in \{0, 1\}^* \mid f(x) = 1\}$ can be defined.

Decision problem Given a language $\mathcal{M}$, a decision problem is the task of computing a boolean function f able to determine if a string belongs to $\mathcal{M}$ (i.e. $\mathcal{L}_f = \mathcal{M}$). Decision problem

1.1.3 Asymptotic notation

Big O A function $f : \mathbb{N} \to \mathbb{N}$ is $O(g)$ if g is an upper bound of f. Big O

$$f \in O(g) \iff \exists \bar{n} \in \mathbb{N} \text{ such that } \forall n > \bar{n}, \exists c \in \mathbb{R} : f(n) \leq c \cdot g(n)$$

Big Omega A function $f : \mathbb{N} \to \mathbb{N}$ is $\Omega(g)$ if g is a lower bound of f. Big Omega

$$f \in \Omega(g) \iff \exists \bar{n} \in \mathbb{N} \text{ such that } \forall n > \bar{n}, \exists c \in \mathbb{R} : f(n) \geq c \cdot g(n)$$

Big Theta A function $f : \mathbb{N} \to \mathbb{N}$ is $\Theta(g)$ if g is both an upper and lower bound of f. Big Theta

$$f \in \Theta(g) \iff f \in O(g) \text{ and } f \in \Omega(g)$$