Chapter 2
Abstraction in Different Disciplines

"Were it not for the ability to construct useful abstractions, intelligent agents would be completely swamped by the real world"

[Russel and Norvig, 2010]

The notion of abstraction has been used, formally or informally, in a large variety of disciplines, including Mathematics, Cognitive Science, Artificial Intelligence, Art, Philosophy, Complex Systems, and Computer Science [473].

In this section we outline the notions of abstraction used in some selected domains. We will try to synthesize the nature of each one, in an effort to make it comparable with similar ones from other contexts. Given the number and variety of fields interested by the notion of abstraction, it is far from us the intent to provide an exhaustive treatment of the subject. On the contrary, we focus on abstraction intended (at least potentially) as a computational process.

2.1 Philosophy

Abstraction, either overtly or in disguise, is at the heart of most philosophical systems. However, according to Rosen [457],[1] the "*abstract/concrete distinction has a curious status in contemporary Philosophy. It is widely agreed that the distinction is of fundamental importance. But there is no standard account of how the distinction is to be explained*." Clearly, the ability to classify objects as abstract or concrete strictly depends on the very definition of *abstraction*, which, apparently, is no more easy to find in Philosophy than elsewhere.

---

[1] http://plato.stanford.edu/entries/abstract-objects/

L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*,
DOI: 10.1007/978-1-4614-7052-6_2, © Springer Science+Business Media New York 2013

One of the first attempts to pin down the idea of abstraction has been made in the Greek Philosophy, most notably by Plato, who proposed a distinction between the *forms* or *ideas* (abstract, ideal entities that capture the essence of things) and the *objects* in the world (which are instantiations of those ideas) [420]. According to him, abstraction is simple: ideas do not exist in the world, they do not have substance or spatial/temporal localization, but their instantiations do. In this approach we may recognize the basic reflex of associating abstraction with being far from the sensible world, and of capturing the "essence" of things; however, Plato's ideas have still their own kind of existence in some other realm, like "idols in a cavern", from where they shape the reality and have *causal* power.

The foundation of abstract reasoning was set later on by Aristotle, who perfected the symbolic methods of reasoning, and whose views dogmatically entered the whole body of Medieval Philosophy. According to Aristotle, there are three types of abstraction:

- **Physical abstraction**—Concrete objects are deprived of their specific attributes but keep their material nature. For instance, starting from the physical reality of an individual man, the physical, universal characteristics of all men can be apprehended.
- **Mathematical abstraction**—Sensory characteristics of embodied objects are ignored, and only the intelligible ones are kept.
- **Metaphysical abstraction**—Entities are considered as disembodied, leaving apart any connotation linked to their realizations. Metaphysics starts not from things, but from the idea of things (*res* or *aliquid*) and tries to discover the essence contained in that idea.

In Philosophy the idea of abstraction has been mainly related to two aspects of reasoning: on the one hand, generalization, intended as a process that reduces the information content of a concept or an observable phenomenon, typically in order to retain only information which is relevant for a particular purpose. Abstraction, thus, results in the reduction of a complex idea to a simpler concept, which allows the understanding of a variety of specific scenarios in terms of basic ideas.

On the other hand, abstraction has been investigated in connection with the very nature or essence of things, specifically in order to ascertain their epistemological or ontological status. Abstract things are sometimes defined as those things that do not exist in reality, do not have a spatio/temporal dimension, and are causally inert. On the opposite, a physical object is concrete because it is a particular individual that is located at a particular place and time.

Originally, the "abstract/concrete" distinction was a distinction between words or terms. Traditionally, grammar distinguishes the abstract noun "whiteness" from the concrete noun "white" without implying that this linguistic contrast corresponds to a metaphysical distinction. In the seventeenth century this grammatical distinction was transposed to the domain of ideas. Locke supported the existence of abstraction [339], recognizing the ability to abstract as the quality that distinguishes humans from animals and makes language possible. Locke speaks of the general idea of a triangle which is "*neither oblique nor rectangle, neither equilateral nor scalenon, but all and none of these at once*".

Locke's conception of an abstract idea, as one that is formed from concrete ideas by the omission of distinguishing details, was immediately rejected by Berkeley, and then by Hume. Berkeley argued that the concept of an abstract idea is incoherent because it requires both the inclusion and the exclusion of one and the same property [53]. An abstract idea would have to be general and precise at the same time, general enough to include all instances of a concept, yet precise enough to exclude all non-instances. The modern empiricism of Hume and Berkely refuses that the mind can attain knowledge of the universals through the generalization process. The mind does not perform any abstraction, but, on the contrary, selects a particular and makes, out of it, a template of all particular occurrences that are the only possible realities.

For Kant there is no doubt that all our knowledge begins with experience [280], i.e., it has a concrete origin. Nevertheless, by no means it follows that everything derives from experience. For, on the contrary, it is possible that our knowledge is a compound of sensory impressions (*phenomena*) and of something that the faculty of cognition supplies from itself *a priori* (*noumena*). By the term "knowledge *a priori*", therefore, Kant means something that does not come from the sensory input, and that is independent from all experiences. Opposed to this is "empirical knowledge", which is possible to obtain only *a posteriori*, namely through experience. Knowledge *a priori* is either *pure* or *impure*. Pure *a priori* knowledge is not mixed up with any empirical element. Even though not set by Kant himself in these terms, the counterposition between *a priori* (or pure) knowledge and *a posteriori* or empirical knowledge mirrors the dichotomy between abstract and concrete knowledge. From this point of view, abstraction is not (directly) related to generalization or concept formation, but represents some sort of *a priori* category of human thinking.

The Kantian Illuminism, with its predilection for the intellect, was strongly criticized by Hegel [240], who considered it as the philosophical abstraction of everything, both real and phenomenological. According to Hegel, the philosophers of his time had so abstracted the physical world that nothing was left. Hegel rejected this line of reasoning, concluding in contrast that "*What is real is rational—what is rational is real*". He set out to reverse this trend, moving away from the abstract and toward the concrete. Hegel viewed the phenomenological world (what can be sensed by humans or manmade instruments) and the conceptual (thoughts and ideas) as equal parts to existence. Hegel thought that abstraction inherently leads to the isolation of parts from the whole. Eventually, abstraction leads to the point where physical items and phenomenological concepts have no value.

Abstraction plays a central role also in Marx's philosophy. By criticizing Hegel, Marx claims that his own method starts from the "real concrete" (the world) and proceeds through "abstraction" (intellectual activity) to the "thought concrete" (the whole present in the mind) [355]. In one sense, the role Marx gives to abstraction is the simple recognition of the fact that all thinking about reality begins by breaking it down into manageable parts. Reality may be in one piece when lived, but to be thought about and communicated it must be parceled out. We "see" only some of what lies in front of us, "hear" only part of the noises in our vicinity; in each case, a focus is established, and a kind of boundary set within our perceptions, distinguishing what is relevant from what is not. Likewise, in thinking about any subject, we focus on only some of its qualities and relations. The mental activity involved in establishing such boundaries, whether conscious or unconscious, is the process of *abstraction*.

A complication in grasping Marx's notion of abstraction arises from the fact that Marx uses the term in four different senses. First, and most important, it refers to the mental activity of subdividing the world into the mental constructs with which we think about it, which is the process that we have been describing. Second, it refers to the results of this process, the actual parts into which reality has been apportioned. That is to say, for Marx, as for Hegel before him, "abstraction" functions as a noun as well as a verb, the noun referring to what the verb has brought into being. But Marx also uses "abstraction" in a third sense, where it refers to a kind of particularly ill fitting mental constructs. Whether because they are too narrow, or take in too little, or focus too exclusively on appearances, these constructs do not allow an adequate grasp of their subject matter. Taken in this third sense, abstractions are the basic unit of ideology, the inescapable ideational result of living and working in an alienated society. "Freedom", for example, is said to be such an abstraction whenever we remove the real individual from "*the conditions of existence within which these individuals enter into contact*" [356]. Omitting the conditions that make freedom possible makes "freedom" a distorted and obfuscated notion.

Finally, Marx uses the term "abstraction" in a fourth sense, where it refers to a particular organization of elements in the real world (having to do with the functioning of capitalism). Abstractions in this fourth sense exist in the world and not, as in the case with the other three, in the mind. In these abstractions, certain spatial and temporal boundaries and connections stand out, just as others are obscure or invisible, making what is in practice inseparable to appear separate. It is in this way that commodities, value, money, capital are likely to be misconstrued from the start. Marx labels these objective results of capitalist functioning "real abstractions", and it is to these abstractions that he refers to when he says that in capitalist society "*people are governed by abstractions*" [356]. As a conclusion, we can say that Marx's abstractions are not things but rather *processes*. These processes are also, of necessity, systemic relations. Consequently, each process acts as an aspect, or subordinate part, of other processes, grasped as clusters of relations.

In today's Philosophy the abstract/concrete distinction aims at marking a line in the domain of objects. An important contribution was given by Frege [181].[2] Frege's way of drawing this distinction is an instance of what Lewis calls the *Way of Negation* [329]. Abstract objects are defined as those that lack certain features possessed by paradigmatic concrete things. Contemporary supporters of the Way of Negation modify now Frege's criterion by requiring that abstract objects be non-spatial and/or causally inefficacious. Thus, an abstract entity can be defined as a non-spatial (or non-spatio/temporal), causally inert thing.

The most important alternative to the Way of Negation is what Lewis calls the *Way of Abstraction* [329]. According to the tradition in philosophical Psychology,

[2] See also Sect. 2.3.

abstraction is a specific mental process in which new ideas or conceptions are formed by considering several objects or ideas and omitting the features that distinguish them. Nothing in this tradition requires that ideas formed in this way represent or correspond to a distinctive class of objects. But it might be maintained that the distinction between abstract and concrete objects should be explained by reference to the psychological process of abstraction or something like it. The simplest version of this strategy would be to say that an object is abstract if it is (or might be) the referent of an abstract idea, i.e., an idea formed by abstraction.

Starting from an observation by Frege, Wright [568] and Hale [230] have developed a "formal" account of abstraction. Frege points out that terms that refer to abstract entities are often formed by means of functional expressions, for instance, the *direction of a line*, the *number of books*. When such a function $f(a)$ can be defined, there is typically an equation of the form:

$$f(a) = f(b) \quad \textit{if} \text{ and only } \textit{if} \ R(a, b), \tag{2.1}$$

where R is an equivalence relation.[3] For example,

$$\textit{direction}(a) = \textit{direction}(b) \quad \textit{if} \text{ and only } \textit{if} \ a \text{ and } b \text{ are parallel} \tag{2.2}$$

These equations are called *abstraction principles*,[4] and appear to have a special meaning: in fact, they are not exactly definitions of the functional expression that occurs on the left-hand side, but they hold in virtue of the meaning of that expression. To understand the term "direction" requires to know that "the direction of a" and "the direction of b" refer to the same entity if and only if the lines a and b are parallel. Moreover, the equivalence relation that appears on the right-hand side of the equation comes semantically *before* the functional expression on the left-hand side [403]. Mastery of the concept of "direction" presupposes mastery of the concept of parallelism, but not vice versa. In fact, the direction is what a set of parallel lines have in common.

An in depth discussion of the concrete/abstract distinction in Philosophy, with a historical perspective, is provided by Laycock [321].[5] He starts by considering the two dichotomies "concrete versus abstract", and "universal versus particular", which are commonly presented as being mutually exclusive and jointly exhaustive categories of objects. He claims that "*the abstract/concrete, universal/particular … distinctions are all prima facie different distinctions, and to thus conflate them can only be an invitation to further confusion*". For this reason he suggests that the first step to clarify the issues involved with the dichotomies is to investigate the relationship *between* them.

Regarding the dichotomy of concrete and abstract objects, he notices that "*this last seems particularly difficult. On the one hand, the use of the term "object" in this*

[3] An equivalence relation is a relation that is reflexive, symmetric and transitive.

[4] See Chap. 4.

[5] http://plato.stanford.edu/entries/object/

context strongly suggests a contrast between two general ontic categories. On the other hand, though, the adjective abstract is closely cognate with the noun "abstraction", which might suggest "a product of the mind", or perhaps even "unreal" or "non-existent" ...". This dichotomy has "*at least two prominent but widely divergent interpretations. On the one hand, there is an ontic interpretation, and there is a purely semantic or non-objectual interpretation, on the other hand. Construed as ontic, the concrete/abstract dichotomy is commonly taken to simply coincide with that of universal and particular*." This interpretation has been adopted, for instance, by Quine [437]. On the contrary, the semantic interpretation of the dichotomy was accepted by Mill [373] and applied to *names*: "*A concrete name is a name which stands for a thing; an abstract name is a name which stands for an attribute of a thing*."

According to Barsalou and Wiemer-Hastings [37], concrete and abstract concepts differ in their focus on situational contexts: concrete concepts focus on specific objects and their properties in situations, whereas abstract concepts focus on events and introspective properties.

Once the distinction between concrete and abstract has been introduced, it is a small step ahead to think of varying degrees of abstraction, organized into a hierarchy. The study of the reality on different *levels* has been the object of various kinds of "levelism", from epistemological to ontological. Even though some of the past hierarchical organizations of reality seem obsolete, Floridi claimed recently [175] that the epistemological one is tenable, and proposed a "theory of the levels of abstraction". At the basis of this theory there is the notion of "observable". Given a system to be analyzed, an *observable* is a variable whose domain is specified, together with the feature of the system that the variable represents.[6] Defining an observable in a system corresponds to a focalization on some specific aspect of the system itself, obtaining, as a result, a *simplification*. It is important to note that an observable is properly defined only with respect to its context and use. A *level of abstraction* (LoA) is nothing else than a finite and non-empty set of observables. Different levels of abstraction for the same system are appropriate for different goals. Each level "sees" the system under a specific perspective. The definition of a level of abstraction is only the first step in the analysis of a system. In fact, taken in isolation, each observable might take on values that are incompatible with those assumed by some others. Then, Floridi introduces a predicate over the observables, which is true only if the values assumed by the observables correspond to a feasible *behavior* of the system. A LoA with associated a behavior is a *moderated LoA*.

As previously said, different LoAs correspond to different views of a system. It is thus important to establish relations among them. To this end, Floridi introduces the concept of *Gradient of Abstraction* (GoA), which is a finite set $\{L_i | 1 \leqslant i \leqslant n\}$ of moderated LoAs, and a set of relations relating the observables belonging to pairs of LoAs. A GoA can be *disjoint* or *nested*. Informally, a disjoint GoA is a collection of unrelated LoAs, whereas a nested one contains a set of LoAs that are refinements one of another.

[6] An observable does not necessarily correspond to a physically measurable entity, because the system under analysis may be a conceptual one.

The use of LoAs in building up models of a system is called the *Method of Abstraction*. It basically consists in clearly specifying the LoAs at which a system is analyzed. A LoA is linked with the amount of information derivable from the corresponding analysis: coarse models provide less information than fine-grained ones.

Methods of analysis similar to the LoAs have been proposed by Newell [398] and Simon [490], with the "ontological" *Levels of Organization* (LoO), and by Benjamin et al. [46], with the "epistemological" *Levels of Explanation*. One of the best known layered analysis of an information processing system is provided by Marr [352], who proposed the *three-level hypothesis*, namely, a system can be analyzed at the following levels:

- ***Computational level***—This level consists of a description of "*the abstract computational theory of the device, in which the performance of the device is characterized as a mapping from one kind of information structures to another. The abstract properties of this mapping are defined precisely, and its appropriateness and adequacy for the task at hand are demonstrated*".
- ***Algorithmic level***—This level consists of the description of "*the algorithm, and of the choice of representation for the input and output and the algorithm to be used to transform one into the other*".
- ***Implementation level***—At this level it is possible to discern "*the details of how the algorithm and representation are realized physically*".

The three levels are supposed to be related by a one-to-many mapping: for any computational problem there may be several algorithms for solving it, and any algorithm may be implemented on different machines and in different languages.

The theory of the LoAs has been used by Abbott [2] to show that software is *externalized thought*. Assuming that (a) consciousness is implemented in the brain as a LoA, (b) we all experience "*to have and idea*", and (c) we are aware of having an idea, Abbott claims that a computer scientist is able to turn this idea into a reality that works by itself in the world (once written, a program works by itself when run on a computer). This type of relation between an abstract idea and a concrete implementation differentiates computer scientists from engineers, who, on the contrary, externalize their idea into material objects that act in the physical world through the human intervention.

Generalization and abstraction in Engineering are discussed by de Vries [127] in a recent paper. For de Vries abstraction is "*abstaining from certain aspects of reality in order to get a deeper understanding of the remaining aspects*." The elimination of specificities from an observed behavior leads to generalization, because the observation can be extended to other situations as well. Another mechanism that produces generalization is *idealization*, intended as "*replacing a complicated detail of reality by a simplified version of that detail.*" Again, simplification allows knowledge elaborated for the simplified version to be applicable for a larger set of cases than the original one. In summary, both abstraction and idealization are means to obtain generalization; the difference between the two is that while abstraction does not change the description of the reality, but simply limit it (leaving aside some aspects it provides a description which is precise, but reduced), idealization describes reality in a (slightly) different way than it is (approximating some aspects, it provides imprecise knowledge).

An approach similar in spirit to Marr's has been described by Pylyshyn [436], who suggested a *semantic*, *syntactic*, and *physical* level of systems description; an additional level of *functional architecture* acts as a bridge between Marr's algorithmic and implementation levels. Finally, a third hierarchy, referring to levels of explanation, has been proposed by Dennet [133], who distinguishes three *stances*: the *intentional* stance, which sees the system under analysis as a rational agent performing a task; the *design* stance, concerning the principles that guide the design of a system successfully performing that task; and the *physical* stance, which considers the physical construction of a system according to these principles.

In a recent paper, Weslake makes an interesting connection between explanatory depth and abstraction [559]. By *explanatory depth* he means "*a measure in terms of which explanations can be assessed according to their explanatory value*". Even agreeing with previous accounts associating explanatory depth with the generality of the laws invoked in the explanation, Weslake claims that an important dimension has nevertheless been overlooked, i.e., abstraction, which "*provides a theoretically important dimension of explanatory depth.*" For him *abstraction* is the "*degree to which a whole explanation applies to a range of possible situations.*" However, Weslake does not commit himself to any measure of the degree of abstraction of an explanation, a thorny notion to be defined. In order to illustrate his approach, he takes into consideration the relationship between the macroscopic law of ideal gases, $PV = nRT$, and its microscopic counterpart. The microscopic explanation is more detailed than the macroscopic law, but the latter applies to a wider range of systems, and, then, it is more *abstract*. Finally, Weslake notices that a gain in abstraction is often obtained by omitting representational details, and that "*deep explanations are provided precisely by abstracting away from causal details*."

Before concluding, we mention, as a curiosity, that, in the Greek mythology there existed some minor gods who were called "Abstractions", and who were personifications of some abstract concepts such as the "vengeance" (Nemesis), or the "death" (Thanatos).

2.2 Natural Language

As we have already mentioned, in natural languages there is a distinction between *abstract* and *concrete* words. Abstract words denote ideas and concepts that cannot be experienced through our senses, such as "freedom" or "beauty"; on the contrary, concrete words denote objects that are part of the sensory reality, such as "chair" or "car". As in the philosophical context, also in languages it is not always easy to classify a term as abstract or concrete. Moreover, this classification is dependent on the cultural background where the language originates. For instance, Benner explains that whereas the ancient Greeks privileged abstract thought (they viewed the world through the mind), ancient Hebrews had a preference for concrete thought (they viewed the world through the senses) [47]. As an example, he mentions that the abstract word "anger" corresponds, in the ancient Hebrew, to "nose", because a Hebrew sees anger as "*the flaring of the nose*".

In a sense, the whole language is an abstraction, because it substitutes a "name" to the real thing. And this is another way of considering abstraction in language. By naming an entity, we associate to the name a bundle of attributes and functions characterizing the specific instances of the entity. For example, when we say *car*, we think to a closed vehicle with 4 wheels and a steering wheel, even though many details may be left unspecified, such as the color, the actual shape, and so on. The ontological status of the "universal" names has been debated, especially in the late Medieval time, with positions ranging from the one of Roscelin,[7] who claimed that universals are nothing more than verbal expressions, to that of Guillaume de Champeaux,[8] who, on the contrary, sustained that the universals are the real thing.

Independently from their ontological status, words stand for common features of perceived entities, and they are considered abstractions derived from extracting the characterizing properties of classes of objects. The word *tree*, for instance, represents all the concrete trees that can exist. This view is based on a *referential* view of the meaning of the words. Kayser [283] challenges this view, proposing an *inferential* view of the words' semantic: words are premises of inference rules, and they end up denoting classes of objects only as a side-effect of the role they play. Barsalou sees the process of naming an object as a way to simplify its representation, by endowing it with invisible properties that constitute its very nature [38]. An interesting aspect of naming is the interaction between vision and linguistic [144, 165]. Assigning a name to a seen object implies recognizing its shape, identifying the object itself and retrieving a suitable word for it. The name can then act as the semantic of an image.

The role of the name as an abstraction of the concrete thing also plays a relevant role in magics. According to Cavendish [89], "*the conviction that the name of a thing contains the essence of its being is one of the oldest and most fundamental of magical beliefs.... For the magical thinker the name sums up all the characteristics which make an animal what it is, and so the name is the animal's identity.*" For instance, burying a piece of lead, with the name of an enemy written on top together with a curse, was, supposedly, a way of killing the enemy. View from this perspective, the name is quite dangerous to a person, because he/she can be easily harmed through his/her name. For this reason, in many primitive societies a man had two names: one to be used in the everyday life, and another, the *real* one, is kept secret. For similar reasons also the names of gods and angels were often considered secret. An Egyptian myth tells that the goddess Isis, in order to take over the power of the sun-god Ra, had to discover his name. Magical power or not, a name is, after all, a shortcut allowing a complex set of properties to be synthesized into a word.

[7] French philosopher, who lived in France in the second half of XII century. His work is lost, but references to it can be found in the works by Saint Anselm and Peter Abelard.

[8] French philosopher, who lived in the late XII century in Paris. He also has been a teacher of Peter Abelard, who, later, convinced him to change his opinion about universals.

An approach relevant to both abstraction and language, even though not explicitly stated so, is described by Gärdenfors [193]. He discusses the representations needed for language to evolve, and he identifies two main types: *cued* and *detached.* A *cued* representation "*stands for something that is present in the current external situation of the representing organism*". On the contrary, a *detached* representation may stand for objects or events that are neither present in the current situation nor triggered by some recent situation. Strictly connected with these representations are the notions of *symbol*, which refers to a detached representation, and *signal*, which refers to a cued one. Languages use mostly symbols. Animals may show even complex patterns of communication, but these are patterns of signals, not symbols. Gärdenfors' distinction closely resembles the distinction between abstract and concrete communication; in this context an abstract communication may involve things that have been, that could be, or that are not localized in time and space. A signal system, instead, can only communicate what is here and now.

In natural language abstraction enters also as a *figure of style*. In fact, abstraction is a particular form of metonymy, which replaces a qualifying adjective by an abstract name. For example, in La Fontaine's fable *Les deux coqs* (VII, 13) the sentence "*tout cet orgueil périt* … (all this pride dies)", refers, actually to the dying cock.

2.3 Mathematics

Mathematics is similar to languages with respect to its relation to abstraction; in fact, in some generic way everything in it is abstract, because Mathematics only manipulates objects that are far from the sensory world. As Staub and Stern put it [506], "*most people would agree that Mathematics is more abstract than Geography. The concepts in the domain of Geography refer to real things, such as rivers and volcanoes, and to concrete and perceptible events, such as floods. In contrast, the "objects" dealt with in Mathematics are symbols that do not refer to specific objects or events in the real world.*" This intrinsic abstractness actually proves to be an obstacle for students learning how to make mathematical proofs [151].

Inside the generic abstractness of Mathematics, specific theories of abstraction have been proposed, which have also generated hot debates. One which deserves to be mentioned saw Frege [181] launching a nasty attack to Husserl's book *Philosophy of Arithmetic* [268], which proposed a *theory of number abstraction*, and included, in turn, a critic to Aristotle's and Locke's views on the subject. The idea behind Husserl's theory is that number abstraction is a counting process that forgets about any property or relation involving objects, leaving them as just unities to be counted. This idea was similar to Cantor's description of the counting process [88]. Cantor was a mathematician whose ideas Frege strongly opposed. Actually, there is the suspect that Frege's attack to Husserl was covertly targeted to Cantor himself [408]. In fact, Frege accused Cantor of using the verb *to abstract* with a psychological connotation, which is to be avoided in Mathematics.

Husserl's theory was criticized by Frege because, according to Frege's view, it would change the objects, by emptying them of all their content. This observation appears to be unfair, because Husserl clearly states that the abstraction process does not actually *change* the objects, but simply "*diverts the attention from their peculiarities*". On the other hand, Frege himself appeared to change his position ten years later, by asserting that the process of abstraction can indeed change the objects, or even create new ones.

In the Word iQ dictionary[9] abstraction is defined as "*the process of extracting the underlying essence of a mathematical concept, removing any dependence on real world objects with which it might originally have been connected, and generalizing it so that it has wider applications*." A good illustrative example of this abstraction process is geometry, which started from the observation and measurement of physical spaces and forms, moving then to the abstract axioms of the Euclidean geometry, and, later on, to non-Euclidean geometries, farther and farther removed from the perceived physical world. An interesting aspect of abstraction is that an increase in the level of abstraction is paralleled by a deepening in the understanding of the connections among mathematical concepts. For instance, by abstracting the Non-Euclidean geometry to "*the study of properties invariant under a given group of symmetries*" has revealed deep connections between geometry and abstract algebra. Moreover, abstraction can suggest direction of knowledge transfer among different domains.

Abstraction is also defined as a *process* by Lewis [329], who claims that abstraction can be better characterized by looking at the way an abstract entity is generated from a concrete one by "*subtracting specificity, so that an incomplete description of the original concrete entity would be a complete description of the abstraction.*" Thus, abstraction is about ignoring irrelevant features of an entity.

Even acknowledging some relations between generalization and abstraction in Mathematics, also Staub and Stern claim that the essence of the mathematical abstractness does not reside in generality, but in the principles underlying the use of mathematical constructs [506]. More precisely, these authors link abstraction with the way mathematical concepts are formed from simpler ones. For instance, the notion of a rational number is more abstract than the notion of a natural number, based on the idea of counting and therefore of integers. As counting is an operation that can be directly experienced, the idea of natural numbers appears to be closer to reality, and hence more concrete. The definition of concepts in terms of more "concrete" ones might also parallel the order in which they are acquired.

In Mathematics abstraction does not only play a role in foundational issues, as the ones mentioned before, but it also provides a key to specific approaches. For example, Roşu describes *behavioral abstraction* as an extension of algebraic specification [455]. More precisely, in his approach "*sorts are split into visible (or observational) for data and hidden for states, and the equality is behavioral, in the sense that two states are behaviorally equivalent if and only if they appear to be the same under any*

[9] See http://www.wordiq.com/definition/Abstraction_(mathematics).

visible experiment." Then, Roşu shows that the notion of behavioral abstraction is a special case of a more general abstraction technique, namely *information hiding*.

Another technical notion of abstraction is presented by Antonelli in a recent paper [20]. Starting from the abstraction principle (2.1), introduced by Wright [569] and Hale [230] and reported in Sect. 2.1, he defines an *abstraction operator*, which assigns an object—a "number"—to the equivalence classes generated by the equinumerosity relation, in such a way that each class has associated a different object. According to Antonelli, this principle is what is needed to formalize arithmetic following the "*traditional Frege-Russel strategy of characterizing the natural numbers as abstracta of the equinumerosity relation.*"

More precisely, numbers, as abstract objects, are obtained by applying an abstraction operator to a *concept* (in Frege's sense). However in order to be an abstraction, such mapping from concepts to objects must respect a given equivalence relation [19]. In the case of numbers, the principle of *numerical abstraction*, or *Hume's Principle*, postulates an operator **Num** assigning objects to concepts in such a way that concepts P and Q are mapped to the same object exactly when as many objects falls under P as they fall under Q. The object $\mathbf{Num}(P)$ can be regarded as "the number of P".

His view of abstraction is called by Antonelli *deflationary* [19], because it denies that objects obtained via abstraction enjoy a special status: they are "*just ordinary objects, recruited for the purpose of serving as proxies for the equivalence classes of concepts generated by the given equivalence relation.*" Abstraction principles are linguistically represented by introducing a "term-forming" operator $\Phi(P)$, which stands for the possibly complex predicate expression P.

An interesting overview of the notion of abstraction in Mathematics is given by Ferrari, who tries to establish connections with other fields, such as Cognitive Science, Psychology, and mathematical education practice [166]; the reason is that "*abstraction has been early recognized as one of the most relevant features of Mathematics from a cognitive viewpoint as well as one of the main reasons for failure in Mathematics learning.*"

By looking at the history of Mathematics, Ferrari acknowledges that abstract objects have been characterized by a certain degree of both generalization and decontextualization. However, he points out that maybe their primary role is in *creating new concepts*, when, specifically, a (possibly complex) process or relation are reinterpreted as (possibly simpler) objects, as in Antonelli's approach [19, 20]. An example is provided by the arithmetic operations, which, at the beginning, are learned as procedures, but then become objects whose properties (for instance, associativity) can be investigated. This transition is called *encapsulation* [142] or *reification* [482].

Ferrari argues that generalization, decontextualization and reification are all basic components of abstraction in Mathematics, but that abstraction cannot be identified with any single one of them. For instance, generalization, defined as an extensional inclusion relation, cannot exhaust the abstraction process, which also includes recognition of common properties, adoption of a compact axiom set, and definition of a notation systems to deal with newly defined concepts. Even though generalization and decontextualization do not coincide, generalization implies a certain degree of decontextualization, intended as privileging syntactic rules and disregarding meaning and interpretation related to some given context. For instance, it is possible to work in abstract group theory without any reference to properties of any specific group. As Hilbert suggests, mathematical practice requires the development of the ability of focusing on what is important, without completely getting away from the context. However, reification is the most interesting aspect of abstraction, capturing the dynamics of the formation of new objects. An example has been reported previously, when discussing Antonelli's work on natural numbers as abstractions of the process of counting. Finally, Ferrari stresses the role that *language* plays in mathematical thinking, because mathematical objects cannot be reached but through a suitably defined language.

2.4 Computer Science

According to Guttag [227], "*the central problem in designing and implementing large software projects is therefore to reduce the complexity. One way to do this is through the process of abstraction.*" There is in fact a widely shared agreement that abstraction is a fundamental process in Computer Science, at the point that Kramer wonders whether it is indeed the core of the discipline, and a mandatory cognitive prerequisite for computer scientists and students to develop good software [301]. In his paper Kramer explores two aspects of abstraction: the ability to *remove details* for simplification, and the formulation of *general concepts* by abstracting common properties from specific examples. While mentioning the utility of abstraction in other domains, such as art or map drawing, he cautions the user that abstraction is a strongly purpose-oriented process, which can be misleading if used for other goals than those for which it was created.

Kramer is not alone in stressing the importance of abstraction in Computer Science; Devlin [134] says that "*once you realize that computing is all about constructing, manipulating, and reasoning about abstractions, it becomes clear that an important prerequisite for writing (good) computer programs is the ability to handle abstractions in a precise manner.*" Finally, Ghezzi et al. [201] identify abstraction as a fundamental principle to master complexity in Software Engineering. A specific example is abstract interpretation for program analysis, where a concrete program domain is mapped to an abstract one, in order to capture its semantics.

Actually, as most computers can only manipulate two-state physical devices, the whole software development can be considered abstract. Technically, there are two main types of abstraction in Computer Science: procedural abstraction and data abstraction [62]. *Procedural abstraction* consists in defining what a (sub-)program does, ignoring how it does it: different implementations of the program can differ over details, but the relation input-output is the same for all of them. Sub-routines, functions and procedures are all examples of procedural abstraction. For instance, we can define a function `prod(x, y)`, which outputs the product of two numbers `x` and `y`, without actually specifying how the product is computed.

pname = **proc** (...) **returns** (...)
  requires % states any constraint on use
  modifies % identifies all modified input
  effects % defines the behavior
end pname

dname = **data type is** % list of operations
  Overview % An overview of the data abstraction
  Operations % A specification of each operation
end dname

Fig. 2.1 Specification templates for procedural and data abstraction. When assigning a name to the procedure, its inputs and outputs are defined. For data, their structure and the applicable operations are defined

For Liskov and Guttag [333], "*abstraction is a many-to-one map.*" It ignores irrelevant details; all its *realizations* must agree on the relevant details, but may differ on the irrelevant ones. Abstraction is defined by Liskov and Guttag by means of *specifications*. They introduce templates for procedural and data abstraction, examples of which are reported in Fig. 2.1.
As we will see in Chap. 7, abstraction operators can be represented with Abstract Procedural Types. Let us now introduce examples of procedural and data abstraction in order to clarify these notions.

Example 2.1 Suppose that we want to write a procedure for searching whether an element y appears in a vector X without specifying the actual program to do it. We can define the following abstract procedure:

pname = **Search**(X, y) **returns**({`true, false`})
  requires X is a vector, y is of the same type as the elements of X
  modifies $\emptyset$
  effects Searches through X, and returns `true` if y occurs in X, else returns `false`
end pname □

Data abstraction, on the other hand, consists in defining a type of data and the operations that manipulate it. Data abstraction makes a clear separation between the abstract properties of a data type and its concrete implementation.

Example 2.2 Let us define the data type *complex number* `z` as a pair (`x`, `y`) of real numbers, with associated some operations, such as, for example, `Real(z)`, `Imaginary(z)`, `Modulus(z)` and `Phase(z)`.

dname = **complex** is pair of reals (`x`, `y`)
  Overview A complex number has a *real* part, `x`, and an *imaginary* one, `y`, such that $\mathtt{z} = \mathtt{x} + \mathtt{i}\,\mathtt{y}$, where $\mathtt{i} = \sqrt{-1}$. In polar coordinates `z` has a *modulus* and a *phase*.
  Operations *Real* (`z`) = `x`
  Imaginary (`z`) = `y`

$$Modulus\,(\mathtt{z}) = \sqrt{\mathtt{x}^2 + \mathtt{y}^2}$$
$$Phase\,(\mathtt{z}) = arctg\left(\frac{\mathtt{y}}{\mathtt{x}}\right)$$

end dname □

Data and procedural abstractions have been reunited in the concept of *Abstract Data Type* (ADT), which is at the core of object-oriented programming languages. An ADT defines a *data structure*, with associated *methods*, i.e., procedures for manipulating the data. An ADT offers to the programmer an *interface*, used to trigger methods, which is separated from the actual implementation, which the programmer does not need to see. Thus, abstraction, in this context, realizes *information hiding*. Even though the notion of ADT has been around for a while, a modern description of it is provided by Gabbrielli and Martini [186]. ADTs are only one step in the evolution of object-oriented programming, because they are passive entities, which can only be acted upon by a controlling program; on the contrary, the notion of *object* goes further, by introducing interaction possibilities via message passing, and a sort of autonomy in letting an object invoke operations on other objects. The relationship between classes, objects and data abstraction has been investigated by Fisher and Mitchell [170], who compare three approaches to class-based programming, namely, one called "*premethods*", and two others called "*prototype*". The authors claim that object-based methods are superior to class-based ones.

Introducing an ADT leads spontaneously to the idea of several nested layers of abstraction [170]. A data type may be part of an **is-a** hierarchy, organized as a tree, where each node has one father, but may have several children. The interest in defining such a hierarchy is that it is not necessary to define from scratch every node; on the contrary, a child node automatically inherits the properties of the father (unless specified otherwise) through downward *inheritance* rules, but, at the same time, it may have some more specific properties added. For instance, if an `animal` is defined as a being that *eats*, *moves*, and *reproduces*, a `bird` can inherit all the above properties, with the addition of *has-wings*. The **is-a** relation between a type and a sub-type is called by Goldstein and Storey an *inclusion abstraction* [217]. These authors define other types of abstraction as well, which will be described in Sect. 4.4.2.

Colburn and Shute [111] make a point is differentiating Computer Science from empirical sciences, because the latter ones have concrete models in the form of experimental apparata as well as abstract mathematical models, whereas the former has only software models, which are not physically concrete. Going further along this line, the authors claim that the fundamental nature of abstraction in Computer Science is quite different also from the one in Mathematics with respect to both the primary product (i.e., the use of formalism), and the objectives.

The main reason of abstraction in Mathematics are *inference structures* (theorems and their proofs), while in Computer Science it is *interaction patterns* (pieces of software). Interactions can be considered at many levels, starting from the basic ones between instruction and data in memory, up to the complex interactions occurring in multi-agent systems, or even those between human users and computers. For what concerns formalism, the one of Mathematics is rather "monolithic", based on set theory and predicate calculus, whereas formalism in Computer Science is "pluralistic" and "multilayered", involving programming languages, operating systems [481], and networks [100]. Looking at the objectives of abstraction, Colburn and Shute make an interesting distinction: in Mathematics the construction of models involves getting rid of inessential details, which they call an act of *information neglect*, whereas in Computer Science writing programs involves *information hiding*, because the details that are invisible at a given level of abstraction cannot be really eliminated, because they are essential at some lower level. This is true for programming languages, but also for operating systems, and network architectures.

A teaching perspective in considering abstraction in Mathematics and Computer Science is taken by both Leron [326] and Hill et al. [249]. Leron claims that in Mathematics "*abstraction is closely related to generalization, but each can also occur without the other.*" In order to support his claim, he offers two examples; the first is the formula $(a+b)^2 = a^2+2ab+b^2$, which is *generalized* (but not abstracted) when its validity is extended from natural numbers (a and b) to rational ones. On the other hand, the same formula is abstracted when it is considered to hold for any two commuting elements in a ring. The second example consists in the description "*all prime numbers less than 20*", which is more abstract (but not more general) than "*the numbers 2, 3, 5, 7, 11, 13, 17, 19*". In Computer Science the separation between the high level concepts, used to solve a problem, and the implementation details constitutes what Leron calls an *abstraction barrier*. Above the barrier the problem is solved using suitably selected abstraction primitives, whereas, below the barrier, one is concerned with the implementation of those primitives. Looking at the mathematical examples we may see that Leron attributes to generalization an extensional nature. Moreover, he notices that proofs of abstractly formulated theorems gain in simplicity and insights. Finally, he makes a distinction between descriptions of objects in terms of their structure and in terms of their functionalities, and claims that abstraction is more often linked to the functional aspects.

For their part, Hill et al. [249] claim that "*abstraction is a context-dependent, yet widely accepted aspect of human cognition that is vitally important for success in the study of Computer Science, computer programming and software development.*" They distinguish three types of abstraction: conceptual, formal, and descriptive. *Conceptual abstraction* is the ability to move forward and backward between a big picture and small details. *Formal abstraction* allows details to be removed and attention to be focalized in order to obtain simplifications. Finally, *descriptive abstraction* is the ability to perceive the essence of things, focalizing on their most important characteristics; this type of abstraction also allows "*salient unification and/or differentiation*", namely it is related to generalization.

Abstraction not only plays a fundamental role in Computer Science in general (namely, in discussing programming philosophy), but it also offers powerful tools to specific fields. One is software testing, where abstraction has been proposed as a useful mechanism for model-based software testing [345, 428]. Another one is Database technology. In databases three levels of abstraction are usually considered: the *conceptual level*, where the entities that will appear in the database are defined, as well as their inter-relationships; the *logical level*, where the attributes of the entities and the keys are introduced, and the *physical level*, which contains the actual details of the implementation. Abstraction increases from the physical to the conceptual level.

Beyond this generic stratification, in a database it is often crucial to select an appropriate level of abstraction concerning the very data to be memorized. With a too fine-grained memorization the database may reach excessive size, whereas with a too coarse-grained memorization important distinctions might be masked. The issue is discussed, among others, by Calders et al. [87], who say that "*a major problem … is that of finding those abstraction levels in databases that allow significant data aggregation without hiding important variations*." For instance, if a department store has recorded every day the number and type of sold items, memorizing these raw data over a period of three years may mask some trends that could have been apparent if the data were aggregated, say, by weeks or months. In order to select the appropriate level, database designers exploit *hierarchies* over the values of variables. For instance, for a time variable, `hour`, `day`, `week`, `month`, and `year` constitute a hierarchy of values of increasing coarseness. In an analogous way, `city`, `region`, `country` constitute a hierarchy for a location variable.

In relational algebra, several among the operators can be interpreted, in an intuitive sense, as abstraction operators. For instance, given a relational table R with attributes $(A_1, \ldots, A_n)$ on the columns, the *projection* operator $\pi_{A_{i_1},\ldots,A_{i_r}}(R)$ hides R's columns that are not mentioned in the operator. In an analogous way the *selection* operator $\sigma_\varphi(R)$ selects only those tuples for which the logical formula φ is true, hiding the remaining one. These operators clearly obey the principles of information hiding, because the omitted columns or rows in R are not deleted, but only hidden: they may be visualized again at any time.

Miles Smith and Smith [372] address the issue of abstraction in databases directly. They say that "*an abstraction of some system is a model of that system in which certain details are deliberately omitted. The choice of the details to omit is made by considering both the intended application of the abstraction and also its users. The objective is to allow users to heed details of the system which are relevant to the application and to ignore other details.*" As in some systems there may be too many relevant details for a single abstraction, a *hierarchy* can be built up, in which some details are temporarily ignored at any given level. In Codd's model of a relational database [109] abstraction requires two steps: first, a relational representation compatible with the intended abstraction's semantics must be found. Second, the meaning of this representation must be explicitly described in terms of data dictionary entries and procedures. As we will see in Sect. 4.7.1, a similar approach is adopted by Nayak and Levy [395] for their semantic model of abstraction.

In Computer Science another important aspect is software verification. According to Yang et al. [572] formal program verification must cope with complex computations by means of approximations. *Abstract interpretation* [117] is a theory for defining sound approximations, and also a unifying framework for different approximate methods of program verification tools. Therefore, abstract interpretation is widely exploited in several fields, such as static analysis, program transform, debugging,

Fig. 2.2 "La trahison des images" (1928-9). Magritte's painting is an "image" of a pipe, not the "real thing"

and program watermarking. In their paper the authors describe the foundations of abstracting program's fixpoint semantics, and present a state of the art on the subject.

2.5 Art (Mostly Peinture)

Whatever "art" might be, according to Gortais [219] "*as a symbolic device, art, whether figurative or not, is an abstraction*". This statement is well illustrated by Magritte's picture of a pipe (see Fig. 2.2), where the sentence "*Ceci n'est pas une pipe*"[10] refers to the fact that the painting "represents" a pipe but it is not the "real thing". Certainly, if we look at a person, an event, a landscape in the world, any attempt of reproducing it, be it through a painting, a sculpture, a novel or a music, forgets something existing in the original. Under this respect, art tries to get at the essence of its subject, and hence it is indeed an abstraction of the reality, if abstraction is intended as a process of getting rid of irrelevancies. On the other hand, a work of art is exactly such because it makes present something that was not present before, and may reveal what was not visible before. Moreover, art's true value is in the emotional relation with the public. "*Each work of art will resonate in its own way over the whole range of human emotions and each person will be touched in a different way*" [219].

Art involves an abstract process, exploiting a communication "language" using a set of symbols. In visual arts, this language is based on colors, forms, lines, and so on. The art language of Western cultures had, in the past, a strict link with the reality that was to be communicated: arts were *figurative*. Later on, the language acquired more and more autonomy, and (at least parts of) the arts became *abstract* [219].

[10] "This is not a pipe".

Fig. 2.3 *Nocturne in Black and Gold* by J. McNeill Whistler (1875). It is considered as a first step toward abstraction in painting [*A color version of this figure is reported in Fig. H2 of Appendix H*]

Thus, *abstract art* does not aim at representing the world as it appears, but rather at composing works that are purposefully non-representational and subjective. The use of non-figurative patterns is not new, as many of them appear on pottery and textiles from pre-historical times. However, these patterns were elements of decoration, and did not have necessarily the ambition to be called "art". A great impulse to the abandon of faithfulness to reality, especially in painting, was given by the advent of photography. In fact, paintings were also intended to transmit to posterity the faces of important persons, or memories of historical events. Actually, a complex interplay exists among figurative works, abstract works, and photography. All three may show different degrees of abstraction, and all three may or may not be classified as art at all: history, context, culture, and social constraints, all play a role in this evaluation.

Even before photography, some painter, such as James McNeill Whistler, stressed the importance of transmitting visual sensations rather than precise representations of objects. His work *Nocturne in Black and Gold*, reported in Fig. 2.3, is often considered as a first step toward abstract art.

A scientific approach to abstract art was proposed by Kandinsky [279], who defined some *primitives* (points, lines, surfaces) of a work of art, and associated to them an emotional content. In this way it was possible to define a syntax and a language for art, which were free from any figurative meaning. However, the primitives were fuzzy (when a point starts to be perceived as a surface?), and the proposed language found difficulties in being applied. Kandinsky, with Malevich, is considered a father of the abstract pictorial art. An examples of Malevich' work is reported in Fig. 2.4.

Fig. 2.4 K. Malevich's *Portrait of Ivan Klioune* (1911). The State Russian Museum, St. Petersburg [*A color version of this figure is reported in Fig. H3 of Appendix H*]

In Fig. 2.5 an even more abstract painting, by the contemporary French painter Pierre Soulages, is reported. He says: "*J'aime l'autorité du noir. C'est une couleur qui ne transige pas. Une couleur violente mais qui incite pourtant à l'intériorisation. A la fois couleur et non-couleur. Quand la lumière s'y reflète, il la transforme, la transmute. Il ouvre un champ mental qui lui est propre*."[11]

Since the eighteenth century it was thought that an artist would use abstraction for uncovering the essence of a thing [377, 588]. The essence was reached by throwing away peculiarities of instances, and keeping universal and essential aspects. This idea of abstraction did not necessarily imply, at the beginning, moving away from the figurative. But, once accepted that the goal of art was to attain the essence and not to faithfully represent reality, the door to non-figurative art was open.

An example of this process is reported in Fig. 2.6, due to Theo van Doesbourg, an early abstraction painter, who, together with Piet Mondrian, founded the journal *De Stijl*. In 1930 he published a *Concrete Art Manifesto*, in which he explicitly denied that art should take inspiration from nature or feelings. In Appendix A the text of the Manifesto is reported. Actually, it sounds rather surprising that the type of totally abstract art delineated in the Manifesto be called "concrete art".

[11] "*I love the authority of black. It is a color that does not make compromises. A violent color, but one that stimulates interiorization. At the same time a color and a non-color. When the light is reflected on it, it is transformed, transmuted. It opens a mental field which is its own.*"

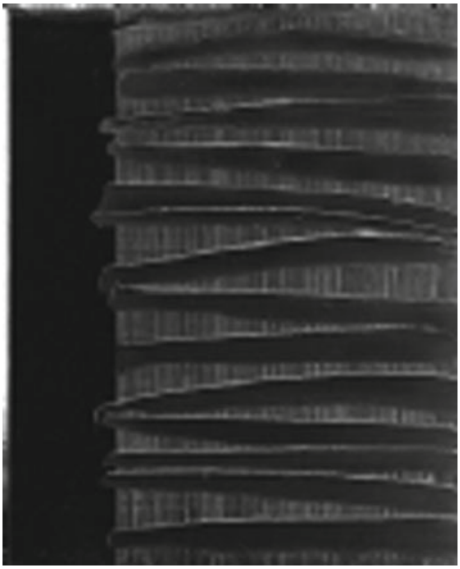


Fig. 2.5 Painting by Pierre Soulages (2008). Bernard Jacobson Gallery (*Printed with the author's permission*)

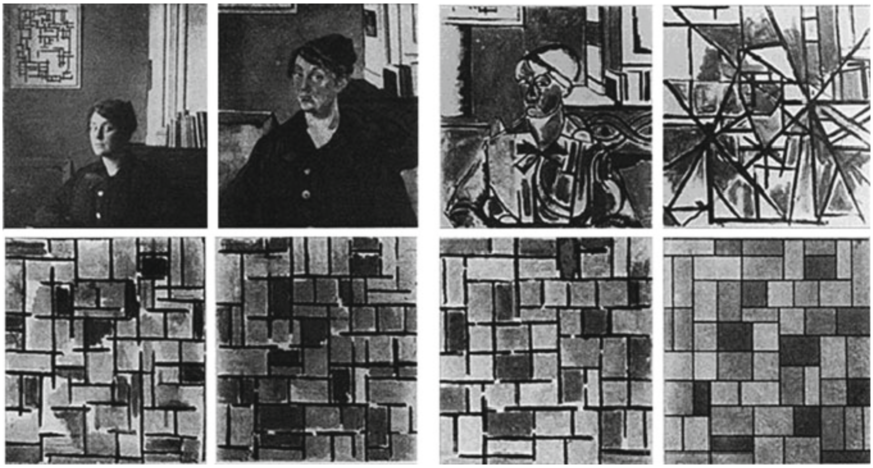


Fig. 2.6 *Studies* by Theo van Doesbourg (1919). From nature to composition

2.6 Cognition

Abstraction is a fundamental dimension of cognition. It is safe to say that without abstraction no high level thinking would be possible. According to Brooks [81], "*Cognitive Psychology has a huge interest in the whole range of issues to do with* *the abstract*". However, the name stands for a large variety of different cognitive phenomena, so that it is difficult to come up with a unifying view.

In Cognitive Science the term "abstraction" occurs frequently; even though with different meanings and in different contexts, it is mostly associated with two other notions, namely, *category formation* and/or *generalization*. Barsalou and co-workers have handled the subjects in several papers (see, for instance, [34]). In particular, a direct investigation of the concept of abstraction led Barsalou to identify six different meaning of the word [35]:

- Abstraction as *categorical knowledge*, meaning that knowledge of a specific category has been abstracted out of experience (e.g., "Ice cream tastes good").
- Abstraction as the *behavioral ability to generalize across instances*, namely the ability to summarize behaviorally the properties of a category's members (e.g., "Bats live in caves").
- Abstraction as *summary representation* of category instances in long-term memory (for instance, the generation of a template for a category).
- Abstraction as *schematic representation*, i.e., keeping critical properties of a category's members and discarding irrelevant ones, or distorting some others to obtain an idealized or caricaturized description (e.g., generating a "line drawing" caricature starting from a person's picture).
- Abstraction as *flexible representation*, i.e., making a representation suitable to a large variety of tasks (categorization, inference, …)
- Abstraction as an *abstract concept*, referring to the distance of a concept from the tangible world ("chair" is less abstract that "truth").

In connection with the above classification of abstraction types, Barsalou introduces three properties of abstraction: *Interpretation*, *Structured Representation*, and *Dynamic Realization*. Regarding interpretation, Barsalou agrees with Pylyshyn [435] on the fact that cognitive representations are not recordings, but interpretations of experience, a process based on abstraction: "*Once a concept has been abstracted from experience, its summary representation enables the subsequent interpretation of later experiences.*" Moreover, concepts are usually not interpreted in isolation, but they are connected via relationships; then, abstractions assemble components of experience into compound representations that interpret complex structures in the world. Finally, abstraction offers dynamic realization, in the sense that it manifests itself in a variety of ways that makes it difficult to define it univocally.

Similar to the notion of category is the one of *concept*. And, in fact, abstraction is also viewed as the process of concept formation, i.e., the process aimed at identifying the "essence" in the sensorial input [522].

An interesting discussion concerns the comparison between abstraction theories in classical Artificial Intelligence (where Barsalou sees them based on predicate calculus), and in connectionism. Barsalou identifies an abstraction as an *attractor* for a statistical combination of properties; here the abstraction is represented by the active units that characterize the attractor. The connectionist view of abstraction suffers from the problem of concept complexity, as neural nets have difficulties in representing structured scenarios.

Abstraction as a representation of a category is contrasted by Barsalou with the *exemplar* representation. In this context, abstraction is intended as a synthesis of common properties associated to a category, as opposed to the memorization of a set of concrete exemplars of the category itself. The two representations are compared in terms of *information storage*, *revision*, and *loss*, and in terms of types of processing that they support. The interesting conclusion of the study is that the two representations are not distinguishable on the basis of empirical findings [34].

The view of abstraction offered by Barsalou consists in an *embodied theory* [35], which is based on *simulation* [36]. According to his view, people have simulators of objects' properties and relations, which are acquired by experience and which they run for interpreting sensory inputs. The set of simulators applied to an instance can be considered as an abstraction.

A more sophisticated view of abstraction is provided later on by Goldstone and Barsalou [216], where conceptual knowledge, however abstract, is strongly grounded on perception; in fact, "*abstract conceptual knowledge is indeed central to human cognition, but it depends on perceptual representations and processes, both in its development and in its active use. Completely modality-free concepts are rarely, if ever, used, even when representing abstract contents.*" Even if trying to link abstract knowledge to perception may seem a counterintuitive approach, we will see in Chap. 6 that this view can provide the basis for a model of abstraction well suited to capture relevant aspects of concept representation. Actually, Goldstone and Barsalou convincingly argue that there are mechanisms shared between cognition and perception that allow abstraction to enter the picture; for instance, *selectivity* lets the attention concentrate on particular aspects of a perception, *blurring* (involuntary or purposeful) removes details from further processing, and *object-to-variable binding* allows perception to have an internal structure as concepts do. Finally, *productivity* generates new objects by aggregating parts.

A set of papers dealing with abstraction in Cognitive Sciences was collected in a special issue of the *Int. Journal of Educational Research* in 1997.[12] In these papers several among the most fundamental questions in abstraction were dealt with, for instance the nature of the notion itself as a *state* or as a *process*, the way in which abstractions are acquired, the possible organization into several levels (introducing thus a gradualness into the notion), and the relationships with generalization and category formation.

In the debate about the relations between generalization and abstraction, Colunga and Smith [112] take the extreme view of identifying the two processes, claiming that "*the processes that create abstract concepts are no different from the processes that create concrete ones*" and then abstraction is nothing else than the "*natural and very ordinary process of generalization by similarity*". According to the authors, the evidence supporting this claim comes from early word learning. Outcomes from experiments with the *Novel Noun Generalization task* [497] show that there are two stages in learning words by children: a slow one, in which learning apparently proceeds through direct association between many single pairs (word, object), and

[12] Vol. 27, Issue 1 (1997).

a fast one, where children seem to use general rules about the nature of words and lexical categories, and they become able to perform *second-order generalization*, namely distinctions not between categories but between features allowing category formation.[13]

The idea of an increasing children's ability to handle abstraction agrees with Piaget's genetic epistemology [417], where he distinguishes *empirical* abstraction, focusing on objects, and *reflective* abstraction, in which the mental concepts and actions are the focus of abstraction. Young children primarily use empirical abstraction to organize the world, and then they increasingly use reflective abstraction to organize mental concepts. The basis for Piaget's notion of abstraction is the ability to find structures, patterns or regularities in the world.

An interesting point is made by Halford, Wilson, and Phillips [231], who draw attention to the role relational knowledge plays in the process of abstraction and in analogy. In their view, the ability of dealing with relations is the core of abstract thinking, and this ability increases with the phylogenetic level, and also with age in childhood. The reason is that the cognitive load imposed by processing relational knowledge depends on the complexity of the relations themselves; actually, the number of arguments of a relation makes a good metric for conceptual complexity. In fact, the cost of instantiating a relation is exponential in the number of arguments. These observations, corroborated by experimental findings, led the authors to conclude that associative processing is not noticeably capacity limited, but that there are, on the contrary, severe capacity limitations on relational processing.

According to Welling, abstraction is also a critical aspect of creativity [556]. He claims that the "*abstraction operation, which has often been neglected in the literature, constitutes a core operation for many instances of higher creativity*". On a very basic level, abstraction can be uncovered in the principles of perceptual organization, such as grouping and closure. In fact "*it is a challenging hypothesis that these perceptual organizations may have formed the neurological matrix for abstraction in higher cognitive functions*". Abstract representation is a prerequisite for several cognitive operations such as symbolization, classification, generalization and pattern recognition.

An intriguing process, in which abstraction is likely to play a fundamental role, is fast categorization of animals in natural scenes [132, 158, 211]. It has been observed that humans and non-human primates are able to classify a picture as containing a living being (or some similar task) after an exposure to the picture of only 30 ms, and with a time constraint of at most 1 s (the median is actually 400 ms) for manifesting recognition. The speed at which humans and monkeys can perform the task (answers may be reached within 250 ms, with a minimum of 100 ms [211]) is puzzling, because it suggests that the visual analysis of the pictures must occur in a single feed-forward wave. One explanation is that recognition happens on the basis of a dictionary of generic features, but how these features are represented and combined in the visual system is not clear. We have here a typical case of abstraction, where the important

[13] For instance they learn that solid things are named by their *shapes* (e.g., a glass "cube"), and non-solid things are named by their *material* (e.g., "water").

discriminant features are selected and used to achieve quick decisions. The specific features involved may have been learned during the evolution of the species, as recognizing a living being (typically, a predator or a pray) may be crucial for survival.

It is interesting to note that the color (which requires a rather long analysis) does not play a significant role in the recognition, as the same recognition accuracy is reached with gray-scale images. The fact that color does not play an essential part suggests that the sensory computations necessary to perform the task rely on the first visual information available for processing. In fact, color information travels along a relatively slow visual pathway (the parvocellular system), and the decision might be taken even before it gains access to mental representations.

According to recent findings [132], recognition might exploit both global aspects of the target and some intermediate diagnostic features. An important one is the size of the animal's body in the picture; in fact, humans are quite familiar with the processing of natural photographs, so that they may have an implicit bias about the scale of an animal target within a natural scene. However this does not seem to be true for monkeys.

A hypothesis about the nature of the processing was investigated very recently by Girard and Koening-Robert [211]. They argue that fast categorization could rely on the quantity of relevant information contained in the low spatial frequencies, because these last could allow a quick hypothesis about the content of the image to be built up. It would be very interesting to come up with a theory of abstraction capable of explaining (or, at least, describing) such a challenging phenomenon.

Another curious cognitive phenomenon, in which abstraction plays a crucial role, is "change blindness" [327, 452, 491, 492], firstly mentioned by the psychologist W. James in his book *The Principles of Psychology* [274]. This phenomenon arises when some distracting element hinders an observer from noticing even big changes occurring in a scene which he/she is looking at. Change blindness occurs both in the laboratory and in real-world situations, when changes are unexpected. It is a symptom of a large abstraction, performed on a scene, which has the effect of discarding a large portion of the perceptual visual input, deemed to be inessential to one's current goal. For example, in an experiment a video shows some kids playing with a ball; asked to count how many times the ball bounces, all observers failed to see a man who traverses the scene holding an open umbrella.[14] Clearly, abstraction is strongly connected to attention, on the one hand, and to the goal, on the other.

Recent studies on the phenomenon include neurophysiological approaches [11, 85], investigation of social effects (changes between images are easier noticed when individuals work in teams as opposed to individually) [530], and level of expertise of the observer (experts are less prone to change blindness, because they can reach a deeper level in analyzing a problem than a novice) [161].

A field where the development of computational models of abstraction could be very beneficial is *spatial cognition*. According to Hartley and Burgess, "the term *spatial cognition* covers processes controlling behaviors that must be directed at

[14] Examples can be seen in the two sites http://nivea.psycho.univ-paris5.fr/#CB and http://www2.psych.ubc.ca/~rensink/flicker/download/.

particular locations, or responses that depend on the spatial arrangement of stimuli" [235]. In spatial reasoning one should be able to abstract time- and space-independent relations from contingent locations, to change among different reference systems, to reason with landmarks and maps, and orient him/herself in unknown environments. All these activities would be impossible without abstraction. Actually there is experimental evidence that not only humans but also animals build up abstract representations of spatial configurations, sharing common spatial features [528]. In this context, Thinus-Blanc states that "*abstraction does not necessarily refer to the highest level of abstraction, but it applies as soon as there is a generalization process taking place. It refers to any cognitive processing, the result of which is not bound to one unique feature or set of features of a given environment, but which can be generalized to various other situations*".

When a subject scans with the eyes the environment, he/she obtains a set of *local views*, because these views depend upon the position and orientation of the subject's eyes, head and body; for this reason local views correspond to a concrete level of spatial description. For spatial representations to be flexible, the time/space dependency should be dropped, and place, angular, and distance relations must be processed in an abstract way. The place occupied by the subject can be defined as the *federating core* of panoramic local views, because it is the point of view of all local views that can be obtained by a 360° rotation around the subject.

Another aspect of spatial cognition where abstraction comes into play is place and spatial relationship *invariance*. When an invisible target place has to be reached, invariant relations among visible landmarks can be exploited. Knowledge of this invariance is abstract, as it does not depend anymore from the concrete descriptions. Abstraction intervenes also in setting up rules for encoding spatial relations and for computing accurate trajectories.

Regarding abstract spatial reasoning, Yip and Zhao [575] have identified a particular style of visual thinking, namely *imagistic reasoning*. Imagistic reasoning "*organizes computations around image-like, analogue representations, so that perceptual and symbolic operations can be brought to bear to infer structure and behavior*". This idea is implemented in a computational paradigm, called *spatial aggregation*, which allows intermediate representations, the *spatial aggregates*, to be formed from equivalence classes and adjacency relations. The authors introduce a set of generic operators, transforming the information-rich input field into more and more abstract aggregates.

Finally, defining the granularity of a spatial region is a classical form of abstraction. According to Hobbs [252], granularity is a means to retrieve a simplified representation of a domain from more complex, richer representations. Spatial and temporal granularities are closely related to the concept of *grain-size* in a local spatial context, defined by Schmidtke [478]. Objects that are smaller than the grain-size can be disregarded as unimportant details. If objects smaller than the grain-size need to be accessed, a change of context is necessary: *zooming out* of a scene, a larger area is covered, but small details are lost, whereas *zooming into* a scene, smaller details are magnified, and objects further away become irrelevant.

The notion of granularity has been addressed also by Euzenat [154–156] in the context of object representation in relational systems. He defined some operators for changing granularity, subject to suitable conditions, and used this concept to define approximate representations, particularly in the time and space domains.

A very interesting link between abstraction and the brain's functioning is provided by Zeki [580–582], who gives to the first part of his book, *Splendors and Miseries of the Brain*, the title "*Abstraction and the Brain*". Zeki suggests that behind the large variety of functions performed by the cells in the brain on inputs of different modalities there is a unifying functionality, which is the *ability to abstract*. By abstraction Zeki means "*the emphasis on the general property at the expense of the particular*". As an example, a cell endowed with *orientation selectivity* responds to a visual stimulus along a given direction, for instance the vertical one. Then, the cell will respond to any object vertically oriented, disregarding what the object actually is. The cell has abstracted the property of *verticality*, without being concerned with the particulars. The ability to abstract is not limited to the cells in the visual system, but extends to all sensory areas of the brain, as well as to higher cognitive properties and judgmental levels.

According to Zeki [582], the brain performs another type of abstraction, which is the basis for the *perceptual constancy*. Perceptual constancy allows an object to be recognized under various points of view, luminance levels, distance, and so on. Without this constancy, recognition of objects would be an almost impossible task. An excellent example is color constancy: even though the amount of red, green and blue of a given surface changes with different illuminations, our brain attributes to the surface the same color. Then, abstraction, in this context, is the capability of the brain to capture the essence of an object, independently from the contextual conditions of the observation. As a conclusion, Zeki claims that "*a ubiquitous function of the cerebral cortex, one in which many if not all of its areas are involved, is that of abstraction*" [582].

2.7 Vision

Vision is perhaps the field where abstraction is most fundamental and ubiquitous, both in human perception and in artificial image processing. Without the ability to abstract, we could not make sense of the enormous number of pixels continuously arriving at our retina. It is abstraction that allows us to group pixels into objects, to discard irrelevant details, to visually organize in a meaningful way the world around us. Then, abstraction necessarily enters into any account of vision, either explicitly or implicitly. In the following we will just mention those works that make more or less explicit reference to some kind of abstraction.

One of the fundamental approach to vision, strictly related to abstraction, is the *Gestalt theory* [558]. "Gestalt" is a German word that roughly means "form", and the Gestalt Psychology investigates how visual perception is organized, particularly concerning the part-whole relationship. Gestalt theorists state that the "whole" is

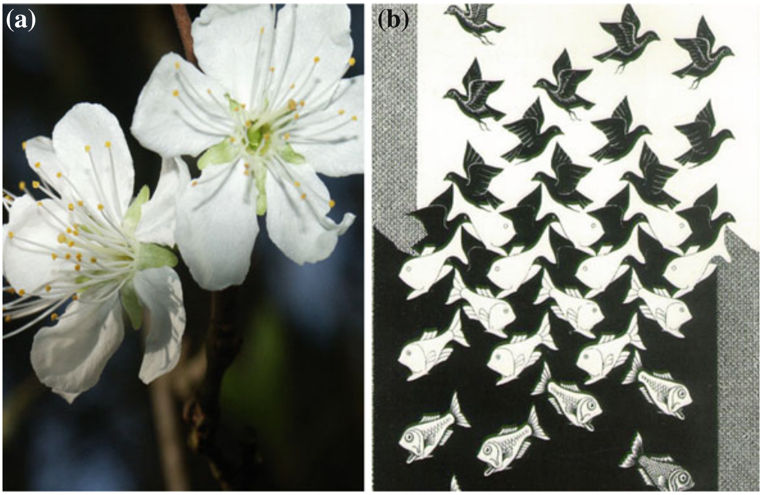


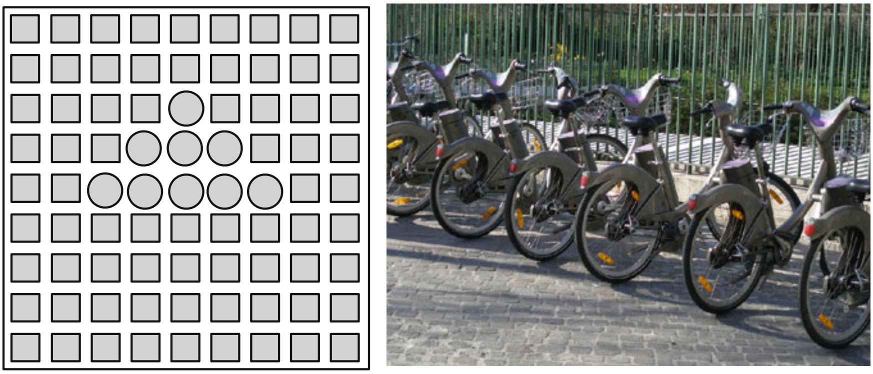
Fig. 2.7 **a** A case of clear separation between foreground and background. **b** A case of ambiguous background in *Sky and Water II*, Escher, 1938 (*Permission to publish granted by The M.C. Escher Company, Baarn, The Netherlands*)

greater than the sum of its parts, i.e., the "whole" carries a greater meaning than its individual components. In viewing the "whole", a cognitive process takes place which consists of a leap from comprehending the parts to realizing the "whole".

Abstraction is exactly the process by which elements are grouped together to form meaningful units, reducing thus the complexity of the perceived environment. According to Simmons [489], parts are grouped together according to function as well; in this way the *functional salience* of parts [538] determines the granularity level from the functional point of view, which often, but not always, coincides with the level suggested by the perceptual one (gestalt).

The Gestalt theory proposes six grouping principles, which appear to underly the cognitive organization of the visual input. More precisely:

- *Foreground/Background*—Visual processing has the tendency to separate figures from the background, on the basis on some feature (color, texture, …). In complex images several figures can become foreground in turn. In some cases, the relation fore/background is stable, whereas in others the mind oscillates between alternative states (see Fig. 2.7).
- *Similarity*—Things that share visual characteristics (shape, size, color, texture, …) will be seen as belonging together, as in Fig. 2.8a. The same happens for elements that show a repetition pattern. Repetition is perceived as a rhythm, producing a pleasing effect, as in Fig. 2.8b.
- *Proximity*—Objects that are close to one another appear to form a unit, even if their shapes or sizes radically differ. This principle also concerns the effect generated when a collection of elements becomes more meaningful than their separate presence. Examples can be found in Fig. 2.9.



(a) (b)

Fig. 2.8 **a** The set of circles in the middle of the array is perceived as a unit even though the surrounding squares have the same color and size. **b** A pleasant repeated arrangement of bicycles in Paris

(a) (b)

Fig. 2.9 **a** The set of *squares* is perceived as two separate entities (*left* and *right*), even though the *squares* are all identical. **b** A ground covered by leaves, where the individual leaves do not matter singularly, but only their ensemble is perceived

- *Closure*—The mind my provide missing parts of an object when there is suggestion of a visual connection or continuity between them, as in the Kanizsa illusion, reported in Fig. 2.10.
- *Continuity*—The eye tends to make lines continuing beyond their ending points, as exemplified in Fig. 2.11.
- *Symmetry*—The eye likes symmetries, and is disturbed by the lacking thereof (see Fig. 2.12).

A good theory of abstraction should be able to explain the computational aspects of the Gestalt theory. This theory has inspired many works on image understanding, whose citation is out of the scope of this book.

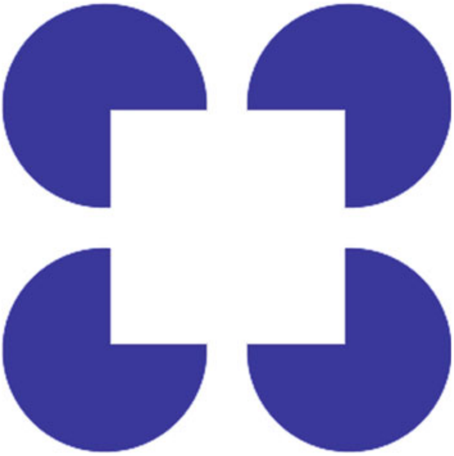


Fig. 2.10 We clearly see a *square* even though the parts of the contour between the circles are not present

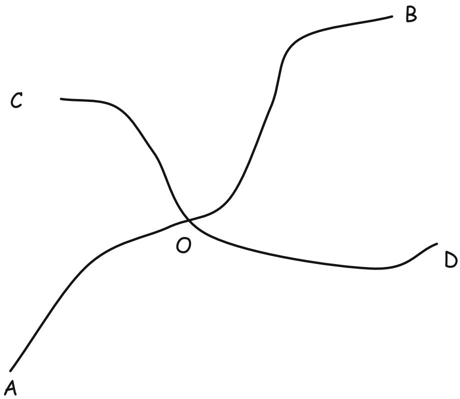


Fig. 2.11 The line AO is automatically continued, by our perception, into line OB, as well as for lines CO and OD

One of the first and most influential work, which has very strict links with abstraction, is Marr's proposal of vision as a process going through a series of representation stages [352, 353]. Particularly relevant for our purposes is the sketchy 3-D representation by means of a series of "generalized cones", as illustrated in Fig. 2.13. The successive stages of a scene representation, from the primal sketch to the 3-D description, can be considered as a series of level of abstraction. Another fundamental contribution to the modeling of human vision was provided by Biederman [59], who introduced the idea that object recognition may occur via segmentation into regions of deep concavity and spatial arrangement of these last. Components can be represented by means of a small set of *geons*, i.e., generalized cones detectable in the image through their curvature, collinearity, symmetry, parallelism, and co-termination. As the geons are free to combine with one another, a large variety of objects can be represented. A *Principle of Componential Recovery* asserts that the identification of two or three geons in an object representation allows the whole object to be recovered, even in presence of occlusion, rotation, and severe degradation.

Fig. 2.12 The symmetry of Notre Dame de Paris appeals to our sense of beauty

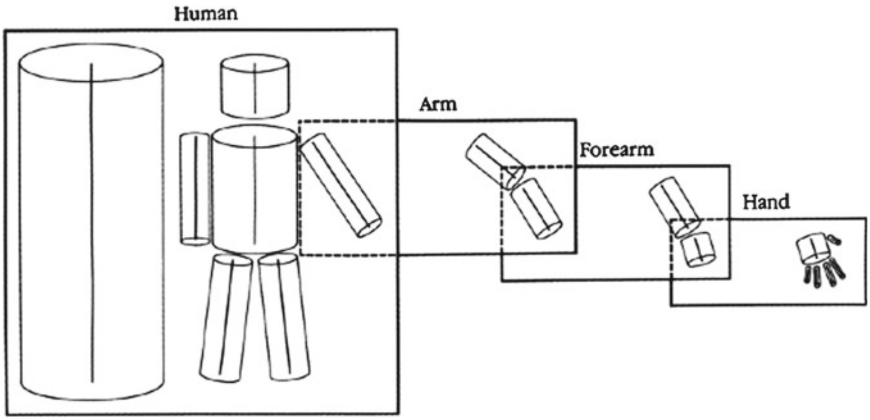


Fig. 2.13 Organization of shape information in a 3-D model description of an object based on generalized cone parts. Each *box* corresponds to a 3-D model, with its model axis on the *left side* of the box and the arrangement of its component axes on the *right*. In addition, some component axes have 3-D models associated with them, as indicated by the way the boxes overlap (*Reprinted from Marr* [*353*])

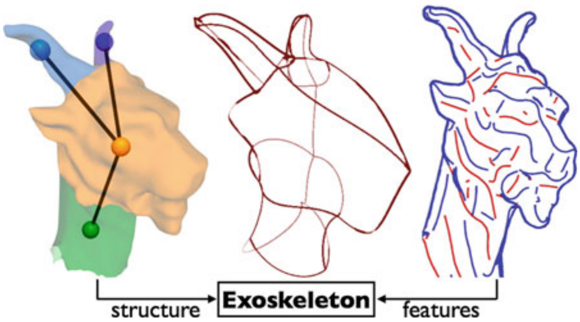


Fig. 2.14 The abstraction technique combines structural information (*left*) with feature information (*right*) (*Reprinted with permission from de Goes et al.* [*125*])

An approach explicitly exploiting abstraction is presented by de Goes et al. [125], who introduce "*the concept of an exoskeleton as a new abstraction of arbitrary shapes that succinctly conveys both the perceptual and the geometric structure of a 3-D model*". The abstraction that the authors propose combines the geometry-driven and the perceptually-driven approaches, generating representations that contain both local and global features of the modeled object, as described in Fig. 2.14.

An approach to vision that typically involves several levels of abstraction is the *multi-scale* image processing [33, 71, 239, 494, 527]. At each level a different resolution allows different details to emerge. As often images are represented via graphs, multi-resolution analysis of graphs and networks is also relevant [456]. Multi-resolution approaches are related to scale-invariance, which is a property that may be required from abstraction of images. Another approach, which combines visual input and functional information to build up concepts, was presented by Hoffmann and Zießler [254]. Their approach is important for abstraction, because it allows concepts to be defined as abstract data types in terms of properties and functions (operations).

Without mentioning abstraction, Chella, Frixione, and Gaglio [94–96] propose an architecture of robot vision that makes large use of it. Their goal is to propose an image processing approach for scene understanding in which there is an interplay among the visual signal (*subconceptual*), the high level linguistic description of the environment, and an intermediate representation based on Gärdenfors' notion of *conceptual space* [194]. This intermediate level is where abstractions, intended as meaningful groupings of pixels from the external world, are formed. In addition, a mechanism of *focus of attention*, allowing only relevant aspects of the input signal to be kept, implements another type of abstraction.

As we mentioned at the beginning of this section, abstraction is the primary mechanism that allows us to make sense of the visual world we perceive, by grouping sets of pixels into meaningful units. Primarily, these units are *objects*. It is then important to define what an object is, what are its characteristic properties, and how can these be extracted in such a way that the presence of the object (and possibly its identity) is detected [15, 509].

The problematics around objects is of the same nature as those concerning knowledge, abstraction, beauty, and so on, i.e., it involves discussions that cannot start form the definition of their subject matter. In fact, the term "object" occurs in a multiplicity of contexts, from Philosophy to Computer Science, from Geometry to Perception. The word "object" comes from the Latin past participle *objectus* of the verb *obicere*, namely "to throw at". In everyday life it is roughly synonym of "thing", and is normally associated to something physical. In Philosophy the word has a much more general meaning, including material things as well as events, ideas, and concepts. Its definition requires that two problems are faced: *change* and *substance*. The first one starts from the consideration that an object may undergo modifications with respect to a given property without loosing its essence. For instance, a house may be restructured, without stopping to be itself for this. On the other hand, a demolished house stops existing. Then, changes have a limit, beyond which the object looses its essence. To locate this limit is not at all obvious. The problem of change is also relevant to abstraction, in the sense that abstraction can increasingly modify objects until (almost) nothing is left of the original ones. An example was provided in Fig. 2.6.

The second problem starts from the observation that the substance that composes an object cannot be experienced directly, but only mediated through its properties. So, it is not possible to conclude for the existence of substance. The way out of this is to say, according to Hume's *bundle theory*, that an object is nothing more than the set of its properties. Things become even more complex when the term object starts to denote also immaterial or conceptual things.

As mentioned in Sect. 2.4, in Computer Science the notion of object is associated to that of abstract data types, and is the basis for *object-oriented programming*. This association is particularly relevant for developing models of abstraction, specifically via the idea of *encapsulation*, which is exactly the information hiding or aggregating process the notion of abstraction is all about. According to Grady Booch [68], "*encapsulation serves to separate the contractual interface of an abstraction and its implementation*". We also recall that encapsulation has been considered by Ferrari (see Sect. 2.3) as one of the main aspects of abstraction in Mathematics.

A field where the notion of an object is not only fundamental but also strictly related with the topics of this book is vision [509], both natural and artificial, including both perceptive issues and computational image processing. For instance, Ballard and Brown [31] say that "*Computer vision is the construction of explicit, meaningful descriptions of physical objects from images*." Or, as Zeki states it [579], "*The brain's task, then, is to extract the constant, invariant features of objects from the perpetually changing flood of information it receives from them.*" Even though acknowledging the fundamental role object recognition has in artificial vision, Stone [509] tries to separate the task of object recognition and identification from several others that vision must attend to, such as vision-guided motion control, determination of depth, tracking changes in lighting, and so on. He maintains that a theory of vision should instead be based on spatio-temporal characteristics, including motion.

Along this line, Amir and Lindenbaum [16] have proposed a quantitative approach to grouping, which consists of a generic grouping method, applicable to many domains [50], and an analysis of the expected grouping quality. In addition, a study of the computational complexity needed for grouping is also presented, as well as a criterion for evaluating the quality of grouping [51, 150].

The notion of abstraction operators bears resemblance with the *visual routines*, introduced by Ullman [541]. These routines are applied to the early representation of an image, and aim at generating visually abstract shape properties and spatial relations. This ability plays a fundamental role in object recognition, visually guided manipulation, and more abstract visual thinking.

A cognitive approach to image classification, which has strong links with abstraction (intended as the process of choosing relevant information), has been proposed by Schyns and co-workers [220]. These authors have designed an algorithm, called *Bubbles*, which allows the parts of an images, which the human attention concentrate on in order to solve a given classification task, to be discovered. In a set of experiments, pictures of human faces were used, and the considered tasks were to decide the gender of the person and whether his/her face was expressive or not. In Fig. 2.15 the regions of the face (relevant features) where the eye of the observers rested the most are reported. The described methodology could provide hints for designing abstraction operators devoted to cognition-based feature extraction.

DeCarlo and Santella [129] describe an approach for stylizing and abstracting photographs, which are translated into line-drawings using bold edges and large regions of constant color, as exemplified in Fig. 2.16. The idea is to facilitate the

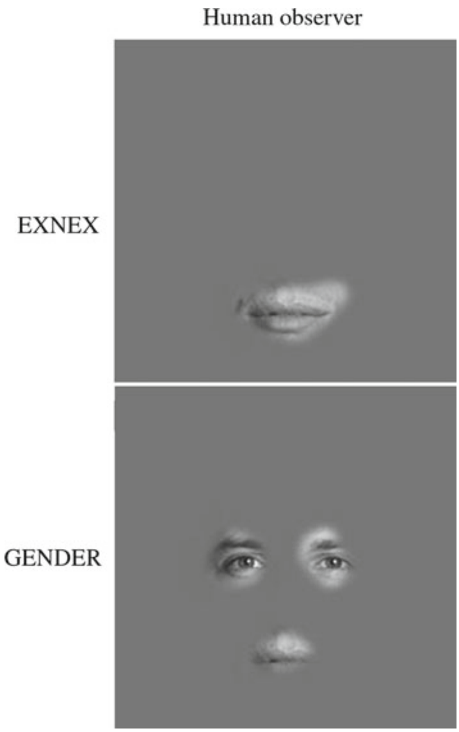


Fig. 2.15 Using the *Bubbles* method, Gosselin and Schyns have identified the information a human observer focuses on when deciding whether a face is/is not expressive (EXNEX), or determining its gender. Expression is sought by extracting information from the mouth, whereas gender classification requires both mouth (but less precise) and eyes (*Reprinted with permission from Gosselin and Schyns* [220])

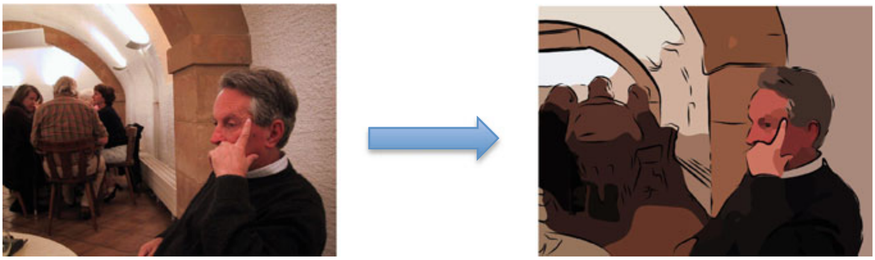


Fig. 2.16 Example of a picture (*left*) and its rendering with lines and color regions (*right*) (*Reprinted with permission from DeCarlo and Santella* [*129*])

observer to easily extract the core meaning of a picture, leaving aside details. A human user interacts with the system, and simply looks at an image for a short period of time, in order to identify meaningful content of the image itself. Then, a perceptual model translates the data gathered from an eye-tracker into predictions about which elements of the image representation carry important information.

In order to cope with the increased resolution power of modern cameras, image processing requires a large amount of memory to store the original pictures. Then, different techniques of *image compression* are routinely used. Image compression can be *lossy* or *lossless*. Lossy compression methods exploit, among other approaches, color reduction, Fourier (or other) transforms, or fractals. They throw away part of the content of an image to accomplish a trade-off between memory requirements and fidelity. For instance, in natural images the loss of some details can go unnoticed, but allows a large economy in memorization space. Lossy compression can be seen as an abstraction process, which (irreversibly) reduces the information content of an image.

When reduction in information is not acceptable, a lossless compression is suitable. There are many methods that can be used, including run-length encoding, chain codes, deflation, predictive coding, or the well-known Lempel-Ziv-Welch algorithm. Lossless compression is a process of image transformation, because the content of the image is preserved, while its representation is made more efficient.

A technique related to abstraction, which is widely used in graphics, is the *Level of Detail* (*LOD*) approach, described by Luebke et al. [348]. In building up graphic systems, there is always a conflict between speed and fluidity of rendering, and realism and richness of representation. The field of *LOD* is an area of interactive computer graphics that tries the bridge the gap between performances and complexity by accurately selecting the precision with which to represent the world. Notwithstanding the great increasing in the power of the machines devoted to computer graphics, the problem is still up-to-date, because the complexity of the needed models has increased even faster.

The idea underlying LOD, illustrated in Fig. 2.17, is extremely simple: in rendering, objects which are far, or small, or less important contain much less details that the more close or important ones. Concretely, several versions of the same object are

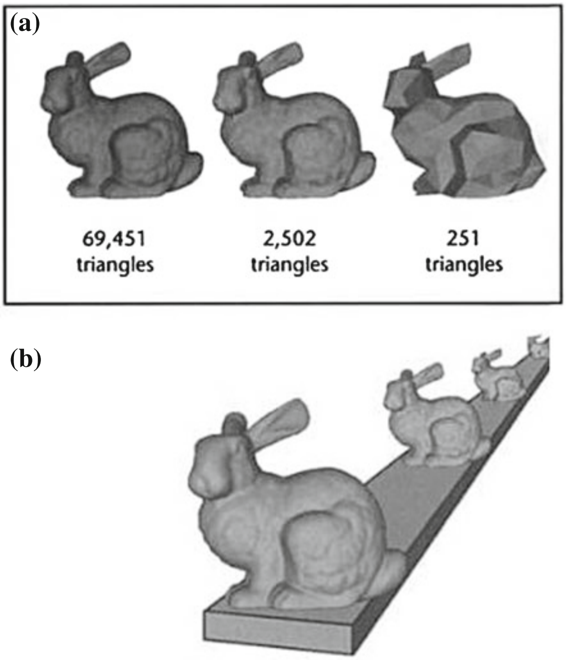


Fig. 2.17 The fundamental concept of LOD. **a** A complex object is simplified. **b** Creation of LOD for rendering small or distant or unimportant objects (*Reprinted with permission from Luebke et al.* [*348*])

created, each one faster and with less details than the preceding one. When composing a scenario, for each object the most suitable LOD is selected.

The creation of the various versions starts from the most detailed representation of an object, the one with the greatest number of polygons. Then, an abstraction mechanism reduces progressively this number, trying to keep as much resemblance as possible with the original one. In recent years several algorithms have been described to automatize this simplification process, which was performed manually in the past. As the generated scenes are to be seen by humans, an important issue is to investigate what principles of visual perception may suggest the most effective simplification strategies.

An approach inspired by the LOD has been described by Navarro et al. [394] to model and simulate very large multi-agent systems. In this case the trade-off is between the amount of details that must be incorporated into each agent's behavior and the computational power available to run the simulation. Instead of *a priori* choosing a given level of detail for the system, the authors propose a dynamic approach, where the level of detail is a parameter that can be adjusted dynamically and automatically during the simulation, taking into account the current focus and/or special events.

2.8 Summary

Abstraction is a notion that plays a fundamental role in a multiplicity of disciplines. By summarizing the basic definitions from various disciplines, five main views of abstraction emerge:

- Abstraction is to take a distance from the concrete world.
- Abstraction coincides with (or is a close variant of) generalization.
- Abstraction is information hiding.
- Abstraction is to keep relevant aspects and to discard irrelevant ones.
- Abstraction is a kind of reformulation or approximation.

In most contexts abstraction has been considered at an informal level, except in Computer Science, Artificial Intelligence, and, in part, Philosophy, where formal or computational models have been proposed.

In later chapters all these notions will be discussed in detail, trying to come up with a computational model of abstraction sufficiently general to unify several among the existing approaches, but concrete enough to be used in practice to help solving non trivial problems.

http://www.springer.com/978-1-4614-7051-9

Abstraction in Artificial Intelligence and Complex Systems
Saitta, L.; Zucker, J.-D.
2013, XVI, 484 p., Hardcover
ISBN: 978-1-4614-7051-9