Reykjavík University

Department of
Computer Science

University of Camerino

School of Science
and Technology

Master of Science in Computer Science

Meaning Generation in Autonomous Grounded Systems

Candidate
Gregorio Talevi
Student ID: 119985

Supervisors
Prof. Dr. Kristinn R. Thórisson
Prof. Dr. Emanuela Merelli

A.Y. 2022/2023

Abstract

Artificial intelligence systems capable of operating highly autonomously in physical environments must be able to deal effectively with new situations in a variety of domains. Among other things, such systems must be able to relate new information to their existing knowledge to achieve their goals, while managing the necessarily limited resources available, including energy, and especially time and memory. In this context, “meaning” is a practical tool for exploring the relationships between a situated controller and its environment, and which, by highlighting the most relevant pieces of knowledge, supports efficient resource management and goal achievement.

Current research on meaning in artificial intelligence is mainly active in the subfield of Natural Language Processing (NLP). The recent advent of Large Language Models (LLMs) has drawn interest to this area, focusing on patterns and relationships in written texts. While LLMs achieve new state of the art performance in many standardized language-related tasks, they still leave unaddressed fundamental issues like (hallucination) out-of-the-lab autonomous and self-supervised learning, robust revisable knowledge, and especially neglect the fundamental aspects of dynamism and subjectivity characterizing meaning generation.

No theory exists to date providing a complete and practical overview of all the above issues, linking empirical learning paradigms, models of revisable knowledge and meaning generation in autonomous goal-directed systems. This is a nontrivial task, as each topic is very broad, and connecting them all also requires a higher-level view of the phenomenon of intelligence.

The theory outlined in this thesis builds on some results of past research in general machine intelligence, in particular cumulative learning, Task Theory, and the Constructivist AI Methodology, formalizing and contextualizing the phenomenon of meaning and several related concepts in the context of infinite worlds. The hope is that this work will catalyze and propel future research in empirical AI.

Should this theory prove correct, we can expect it to inspire further research on the topic of meaning in AI as a pragmatic tool for achieving goals and tasks, thus contributing to the quest for artificial general intelligence.

Keywords: *Meaning, Information, Knowledge, Task theory, Artificial Intelligence, General Machine Intelligence, Empirical Artificial Intelligence*

Contents

Acronyms

AERA Autocatalytic Endogenous Reflective Architecture

AGI Artificial General Intelligence

AI Artificial Intelligence

AIKR Assumption of Insufficient Knowledge and Resources

GMI General Machine Intelligence

NARS Non-Axiomatic Reasoning System

NLI Natural Language Interpretation

NLP Natural Language Processing

NLU Natural Language Understanding

Glossary

Agent An **agent** is an embodied system consisting of a controller (the mind) and a body. The body is the agent's interface to the **world** which allows the perception of the external **environment**, through the flow of data from the body's sensors to the controller, and the execution of atomic actions, by means of the commands sent from the controller to the body's actuators. The body contains two lists of variables that the controller can read and write to: $B = \langle V_S, V_A \rangle$. Since the body is a physical entity, its sensors and actuators are physical objects in the world as well and are treated as such (Thórisson, Bieger, Thorarensen, et al., 2016).

Environment An **environment** is a view of a **world**. The body of an agent is considered to be part of it. (Thórisson, Bieger, Thorarensen, et al., 2016).

Failure A **failure** state (negative goal) is an undesirable, possibly partial, **state** that the agent should avoid (Thórisson, Bieger, Thorarensen, et al., 2016).

Goal A **goal** state (positive goal) is a desirable, possibly partial, **state** that the agent should reach (Thórisson, Bieger, Thorarensen, et al., 2016).

Phenomenon A **phenomenon** (process, state of affairs, occurrence) Φ, where W is the **world** and $\Phi \subset W$, is composed of a set of elements $\{\phi_1, \phi_2, ..., \phi_n \in \Phi\}$ of various kinds including relations $\mathfrak{R}_\Phi$ that couple elements of Φ with each other and with those of other phenomena. The elements that a phenomenon is made up of can be any subdivision of Φ, including sub-structures, causal relations, whole-part relations and so on. The relations $\mathfrak{R}_\Phi \subseteq 2^W \times 2^W$ that extend to other phenomena identify the phenomenon's *context.* The set of relations can be partitioned in *inward facing* relations $\mathfrak{R}^{in}_\Phi = \mathfrak{R}_\Phi \cap (2^\Phi \times 2^\Phi)$ and *outward facing* relations $\mathfrak{R}^{out}_\Phi = \mathfrak{R}_\Phi \setminus \mathfrak{R}^{in}_\Phi$ (Thórisson, Kremelberg, et al., 2016).

Problem A **problem** can be *atomic* or *compound.* An atomic problem is specified by an initial **state**, **goal** states and **failure** states. A compound problem can be created by composition of atomic problems using operators such as conjunction, disjunction and negation. A problem for which a solution is known to exist is called a closed problem (Thórisson, Bieger, Thorarensen, et al., 2016).

Problem space The problem space is the set of all valid states of the task (Belenchia, 2021).

Solution A **solution** is a sequence of atomic actions that results in a path through the **state** space that reaches all of the **goal** states and none of the **failure** states (Thórisson, Bieger, Thorarensen, et al., 2016).

Solution space The solution space is the subset of the problem space defined by the task's goals and constraints, made up of all the solution states reachable from any initial state of the task (Belenchia, 2021).

State A *concrete* state S is a value assignment to all of the variables in a **task-environment**: $S = \bigcup_{v \in V} \{\langle v, x_v \mid x_v \in d_v \rangle\}$. A **state** can be either *concrete* or *partial*: a *partial* state S^- only assigns values to a subset of the variables. When considering real variables, partial states can be represented using error bounds: $S^- = \bigcup_{v \in V^-} \{\langle v, x_l, x_u \mid x_l < x_u \wedge (x_l, x_u) \subseteq d_v \rangle\}$; this way a partial state covers a set of concrete states. A state is valid if and only if all invariant relations hold: $\text{valid}(S) \iff \forall_{r \in R} r(S)$. In practice the presence of noise and the partial observability of variables makes the use of partial states more practical than concrete states, therefore by state is always meant a partial state unless otherwise noted. (Thórisson, Bieger, Thorarensen, et al., 2016).

Task A **task** is a problem assigned to an **agent**, $T = \langle S_0, \mathcal{G}_{top}, \mathcal{G}_{sub}, G^-, B, t_{go}, t_{stop}, I \rangle$, where S_0 is the set of permissible initial states, $\mathcal{G}_{top}$ is the task's set of top-level goals, $\mathcal{G}_{sub}$ is the set of given sub-goals, G^- is its set of constraints, B is a controller's body, and t refers to the permissible start and stop times of the task. An *assigned* task will have all its variables bound and reference an agency that is to perform it (accepted assignments having their own timestamp t_{assign}). This assignment includes the manner in which the task is communicated to the agent, for example if the agent is given a description of the task a-priori, receives additional hints or if it only gets incremental reinforcement signals as certain **states** are reached. A task is performed successfully when the **world**'s history contains a path of states that solved the problem (Thórisson, Bieger, Thorarensen, et al., 2016).

Task-Environment By **task-environment** is meant the tuple of a **task** and the **environment** in which it is to be performed. The separation of a task from its environment is not always clear and somewhat arbitrary, therefore the term task-environment is used to encompass all the relevant aspects of both (Thórisson, Bieger, Thorarensen, et al., 2016; Bieger and Thórisson, 2017).

World A **world** W is an interactive system consisting of a set of variables V, dynamics functions F, an initial state S_0, domains D of possible clusters of particular constraints on their values, and a set of relations between the variables R: $W = \langle V, F, S_0, D, R \rangle$. The variables $V = \{v_1, v_2, ..., v_{||V||}\}$ represent anything that may change or hold a particular value in the world. The dynamics functions act as the laws of nature in the world and as a whole can be seen as an automatically executed function that periodically or continually evolves the world's current state into the next: $S_{t+\delta} = F(S_t)$. It is useful to the decompose the dynamics into a set of transition functions: $F = \{f_1, f_2, ..., f_n\}$ where $f_i : S^- \rightarrow S^-$ and S^- is a partial state. The domains $d_v \in D$ specify which values each variable v can take, and for physical domains these are usually subsets of real numbers. The relations are Boolean functions over variables that hold true in any state the world will ever find itself in. If the world is a closed system with no outside interference, the domains and relations are implicitly fully determined by the dynamics functions and the initial state. In an open system where changes can be caused externally, instead, the explicit definition of domains and invariant relations can restrict the range of possible interactions (Thórisson, Bieger, Thorarensen, et al., 2016).

CHAPTER 1

Introduction

Artificially intelligent (AI) systems are designed to carry out tasks on behalf of humans. Such tasks, which ultimately take place in the physical world, require an intelligent system to understand the context in which it finds itself at any point in time and relate that context to its own (and its master's) goals, gathering and organizing information, learning and adapting, all while operating with a high degree of autonomy. Indeed, the requirement for autonomy is of primary importance here, in that the more autonomous a system is, the less help and programming is required for it to accomplish tasks, yet autonomy is but one of many requirements that such a system is expected to meet. As outlined by Thórisson (2020e) and Nivel and Thórisson (2013), autonomy must be associated with qualities such as predictable robustness in novel circumstances and graceful degradation in case of failure to prevent undesirable effects from happening. If the system fails unpredictably, the designer's intervention will be needed to figure out what went wrong and how to fix the problem, undercutting its level of autonomy. Regardless of the application domain in which they are used, AI systems must be reliable. This, in turn, necessitates the use of methodologies for building AI systems that follow explainable principles of operation. Let us take as a reference the only example of intelligence unanimously recognized: humans. The human brain naturally manages patterns, connections, and narratives to make sense of the world. *Meaning* provides a framework for organizing and processing information, helping individuals relating complex concepts and situations to their goals (e.g., I am hungry, I see an apple $\xrightarrow{this\ means\ that}$ if I eat the apple I will satisfy my hunger). Humans are also capable of introspection, of asking questions about the context and about the achievement of their objectives. These abilities underlie phenomena such as communication and collaboration, which are fundamental elements of problem-solving. An AI system that does not work by explainable principles of operation cannot, consequently, introspect and explain itself – or, at least, not on its own – and, if it cannot explain the meaning of its own actions, even less can we assume that it can grasp the meaning of any other phenomenon at all.The state-of-the-art AI technologies that to date are most being researched and interested in by both the scientific community and industry are almost entirely based on mechanisms that, by design, do not lend themselves well to interpretation (see neural networks). We cannot evaluate at any given moment what these systems are learning or why they are learning certain things except by defining ad-hoc tests, which, however, cannot possibly cover all possible cases and are therefore to be considered not exhaustive. The lack of a certain degree of meaning management in these AI systems makes them fundamentally unstable and unreliable and, therefore, unsuitable for adoption in complex and dynamic environments where a high degree of autonomy is required. The purpose of this thesis is to resume and neatly organize the foundational concepts as well as requirements necessary for the most comprehensive definition of the phenomenon known as “meaning” and the process behind its generation. This definition will be contextualized within the research framework of Constructivist AI, with reference to the Constructivist definitions of knowledge, information, task, and, in particular, previous work on Constructivist Task Theory.

The manuscript is structured in 5 chapters:

1. In Chapter 1, we introduce the topic of meaning in AI, and the need for a framework addressing meaning in the field of artificial intelligence from a more general perspective. We will introduce the subfield of Artificial General Intelligence and the related Constructivist AI methodology;

2. In Chapter 2, we go through the some related concepts required for the following chapters, including: learning, agents, control, causality, fundamentals of a task theory, extended causal diagrams and a theory of understanding;

3. In Chapter 3, we analyze the concept of meaning starting from the analysis of its common usage, arguing in favour of its pragmatic nature, and introduce core meaning-related concepts, like models suitable for knowledge representation, causal chains, reasoning and the concept of relevance;

4. In Chapter 4, we provide our definition of meaning and meaning generation, complimented with a characterization of goals, and formalization of the concepts of implication and relevant implications; considerations on the limitedness of energy and time influence all of the definitions given;

5. Finally, in Chapter 5 we briefly summarize what has been presented in this thesis and raise some thoughts on connections with other work and future developments.

Our contribution to achieving the set objective includes:

- Research, consultation and organization of corpus of sources and related work;

- Introducing the intuition of the concept of meaning, identifying the aspects necessary for its formalization;

- Definition of requirements necessary for meaning generation and features affecting the way it is handled;

- Review, reformulate, and combine the fundamental aspects of implication and relevance characterizing the generation of meaning, already described in previous work on Anytime Bounded Rationality (Nivel, Thórisson, B. Steunebrink, and Schmidhuber, 2015) and Understanding (Thórisson, Kremelberg, et al., 2016);

- Formalization of the meaning generation process;

- Reformulation of all formulas produced in light of the principle of limited time and resources;

- Considerations and comparisons between meaning and other fundamental concepts in order to contribute to a more complete narrative of the phenomenon of intelligence through meaning, understanding, task theory, and possible influences of meaning on issues of teaching, symbols, and resource management.

1.1 Artificial General Intelligence

Artificial Intelligence (AI) is the field dedicated to the design and development of systems that show behaviours or possess qualities typically associated to what we call ‘intelligence’. Intelligence is a natural phenomenon most commonly associated with human minds, but is also a characteristic of many animals (Thórisson, 2020b). Philosophers, psychologists, biologists, artificial intelligence researchers, all have attempted over the years to give a definition of intelligence. Despite numerous efforts following the growing interest in defining intelligence in the last century (White and Hall, 1980; Buxton, 1985), a well-defined and widely accepted definition of intelligence still eludes us to this day. Noteworthy is the collection of 70-odd definitions of intelligence made by Legg and Hutter (2007).

Although it might seem minor, the problem of having a well-established definition of intelligence is actually of primary importance for any scientific field, like AI, that puts intelligence as its central subject of study. The risk is, otherwise, to unknowingly misdirect research efforts to the point of leaving the field’s area of interest altogether. Instead, we can resort to a working definition, possibly incomplete and subject to future revision, of the phenomenon of interest so as to guide research.

Of the many definitions of intelligence that have been given, we intend, in this work, to refer to two in particular. The first one is the one given by Wang (1995) and Wang (2019):

> “Intelligence is the capacity of an information-processing system to adapt to its environment while operating with insufficient knowledge and resources.” (Wang, 1995)

In its works, Wang introduces the “Assumption of Insufficient Knowledge and Resources” (AIKR), which identifies the normal working environment of an intelligent system and states that there is not (and cannot be) an infinite amount of time, knowledge, or other resources to carry out any task, so an agent must do its best with what it has. The second definition we will refer to in this work is the reformulation of Wang’s definition proposed by Thórisson (2020a):

> “Intelligence is discretionarily constrained adaptation under insufficient knowledge and resources.” (Thórisson, 2020a)

Thórisson further summarizes this definition by simply stating it as “figuring out how to get new stuff done”. Thórisson’s definition is more specific, clearly separating this use of ‘adaptation’ from its sense in the context of natural evolution, whose course is determined by physical laws. Thórisson claims that, to be called intelligent, in contrast to evolution, the adaptation in question needs to have a capacity to handle arbitrary constrains of many forms, as well as the capacity of inventing such constraints in light of multiple and often conflicting goals (Thórisson, 2020a). Both of these definitions are considered *working* definitions, that is, they have the sole purpose of guiding research in the right direction, while remaining open to revisions and improvements in the immediate future.

What emerges from both of these definitions is that there is some association between the intelligence of a system and its *generality*: an agent that can adapt to more, novel, environments and can achieve more goals with limited time and resources is intuitively more intelligent than an agent with a more limited spectrum of use. The goal of Artificial General Intelligence (AGI), also called “Strong AI” or “General Machine Intelligence” (GMI), is therefore to develop systems that can “learn to perform multiple a-priory unknown tasks in multiple unknown environment” (*Bounded Recursive Self-Improvement* 2013, p. 3), similarly to what humans do. In contrast, the most researched “intelligent” systems today (i.e. machine learning and deep learning systems) are mostly limited to a single, pre-defined task in an unchanging, pre-defined fixed environment (*Bounded Recursive Self-Improvement* 2013). Therefore, given that even room temperature controllers “achieve goals in a wide range of situations”, Wang and Thórisson’s definitions more robustly differentiate general intelligence – the kind we normally associate with the concept – from other controllers and processes that might also qualify (Belenchia, 2021).

1.2 AI methodologies

A methodology is a systematic and structured approach or set of principles and practices used to conduct research, solve problems, or achieve specific goals in various fields, such as science, engineering, business, social sciences, and more. Methodologies provide a framework for organizing and executing tasks or processes in a consistent and efficient manner. While there are several methodologies inherited from the field of computer science that are used to design, develop and deploy specific AI applications – such as Agile methods and MLOps – there are a few approaches to AI that can be properly called methodologies (Thórisson, 2022a). Some of the most influential AI methodologies include the Belief-desire-intention software model, the subsumption control architecture and decision theory. More recently, and more influential for this work, two other AI methodologies have emerged: Constructi*on*ist and Constructi*v*ist. **Constructionist AI** (Thórisson, Benko, et al., 2004) is an approach in the field of artificial intelligence that focuses on creating intelligent systems by simulating or emulating aspects of human cognition and learning. This approach employs the divide-and-conquer method inherited from computer science as the main way towards the complete understanding of a phenomenon: the problem is recursively fragmented into subproblems, each small enough to be solved by a team of researchers within a few years (Thórisson, 2009). The fundamental assumption underlying both the Constructivist approach and the bulk of today’s research in AI is that intelligence can also be described and recreated using a series of modules, each of which performs a different function. Therefore, we could consider current major research efforts on artificial intelligence “constructionist” to an extent. However, the manual work of breaking down problems requires constant effort by teams of researchers and, as Thórisson (2009) points out, for any subject of study there is no guarantee that theories developed to effectively address individual subproblems can be later combined in a straightforward manner to form a complete theory. Moreover, when dealing with complex dynamic systems, any subdivision will necessarily ignore important interconnections between the various parts, compromising the possibility to understand how the whole system works (Thórisson, 2009). The human mind is known to be one such complex and dynamic system, exhibiting, as argued by Thórisson (2008), the properties of a heterogeneous, large, densely-coupled system (HeLD). The number of modules required to recreate the more complex functions of intelligence would be very large and would require a highly efficient mechanism for orchestrating the interactions among the various components. The “cognitive” development would be due not only to the dynamic modules, but also to the ability to reorganize and evolve an ever-growing architecture (Thórisson, 2009). For all these reasons, the Constructionist approach is fundamentally unsuitable for the study of the broad, complex and dynamic phenomenon that is intelligence, showing, on the other hand, to be very effective in the development of targeted AI systems for well-defined industrial applications.

Constructivist AI is the name of an innovative and relatively recent approach to AI designing and building that calls for a fundamental shift, from Constructionism's hand-crafting to self-organizing architectures and self-generated code (Thórisson, 2012). This new artificial intelligence methodology, partially inspired by Piaget's theories on cognitive development, was proposed by Thórisson, 2012 to address the numerous significant challenges involved in building artificial general intelligence AGI systems, by replacing the top-down architectural design approaches that are ubiquitous today with methods that allow a system to autonomously manage its own cognitive growth. The topics originally listed as major players in the transition from Constructivist AI (Thórisson, 2009) are: *temporal grounding*, *feedback loops*, *pan-architectural pattern matching*, *small white-box components* and *architecture meta-programming and integration.* The Constructivist AI approach has been successfully demonstrated in the HUMANOBS project (*Bounded Recursive Self-Improvement* 2013), where a domain-independent AI system autonomously learned real-time socio-communicative behavior through observation (Thórisson, Nivel, et al., 2014).

1.3 Meaning in AI

The study of "meaning" has long been primarily the interest of philosophers, linguists and psychologists, who have studied it within their respective fields of study (Robert A. Wilson, 1999; Ignelzi, 2000) . In 1956, a conference was held at Dartmouth College in the United States, attended by many prominent figures in computational intelligence (John McCarthy, Marvin Minsky, Claude Shannon to name a few). During that conference the term "artificial intelligence" was coined and many research topics within the field were decided. Since then, many research topics related to intelligence have been further developed in artificial intelligence, including the concept of "meaning".

The period of the first 20-30 years of research in the field of AI is today known as "Good Old-Fashioned AI" (GOFAI) (Haugeland, 1985) and was characterized by the prevalent study of what is called Symbolic AI. During the period between the 50s and the 60s, the notion that if a machine can manipulate numbers, then it can also manipulate symbols was gradually established, and it was theorized that symbol manipulation might be the essence of human thought. The hypothesis, called the "Physical Symbol System Hypothesis" was formulated by scientists Allen Newell and Herbert A. Simon in the mid-1970s and reads:

> "A physical symbol system [such as a digital computer, for example] has the necessary and sufficient means for general intelligent action." (Newell and Simon, 1976)

In Newell and Simon's own definition, a physical system of symbols consists of a set of entities, called symbols, which are physical structures that can appear as components of another type of entity called expressions (or symbolic structures). Thus a symbolic structure consists of a number of exemplars (or tokens) of symbols that are physically related to each other in a certain way (e.g., by the fact that one token is next to another token). At each instant the system will contain a collection of these symbolic structures. In addition to these structures, the system will also contain a collection of processes that operate on expressions to produce other expressions. A physical system of symbols is therefore a machine that produces a changing set of symbolic structures over time. According to this approach, *meaning* is associated with symbols and represented through symbols and rules to manipulate symbols. Symbolic AI used tools such as logic programming, production rules, semantic nets and frames, and it developed applications such as knowledge-based systems, of which expert systems are the best known example. To limit complexity and organize data into information and knowledge, ontologies were used[1]. Symbolic systems were based on explicit rules and declarative knowledge, with an emphasis on manual programming. Discourse representation theory and first-order logic have been used to represent *sentence meanings*. However, this approach had obvious limits: it was difficult to capture the complex and shady meaning of words and concepts.

Meaning appears therefore to be deeply tied to problems like knowledge representation and reasoning. Semantic networks, conceptual graphs, frames, and logic are all approaches to modeling knowledge such as domain knowledge, problem-solving knowledge, and the semantic meaning of language.

After a period of disillusionment and subsequent disinterest in the potential offered by artificial intelligence, known as "AI winter"[2], new interest has been fueled by new applications of artificial neural networks, the origins of which can be traced back to the work of McCulloch and Pitts (1943). By focusing on specific sub-problems, "narrow" AI systems achieved commercial and academic success in the 1990s and early 21st century Russell and Norvig (2003). These kinds of systems are now used and studied extensively.

1.3.1 Natural language

Of the areas of study in artificial intelligence, the ones that to date are most commonly associated with the search for meaning are Natural Language Processing (NLP) and Natural Language Understanding (NLU), the latter also known as Natural Language Interpretation (NLI)[3]. Natural language *processing* is the title used to refer to the theory that focuses on treating language as data to perform tasks such as identifying topics without necessarily understanding the intended meaning. Natural language *understanding*, in contrast, constructs a meaning representation and uses that for further processing, such as answering questions.

The "classical" process of meaning extraction starts with text transformation using techniques like parsing, tokenization, lemmatization, part-of-speech tagging, stemming, stop-words removal, all of which were already used by symbolic systems. Up to the 1980s, most NLP systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in NLP with the introduction of machine learning algorithms for language processing. Gradually, statistical and machine learning techniques for NLP became popular, as well as vectorized representations of textual data for information retrieval (Russell and Norvig, 2003). Today's state-of-the-art approaches are transformers, deep learning architectures that rely on attention mechanisms (Vaswani et al., 2023), machine learning techniques trying to mimic cognitive attention. These new systems open the way to new possibilities; however, they are opaque and do not yet produce semantic representations that can be interpreted by humans.

[1] An ontology is a formal approach to representing, naming, and defining the categories, properties, and relations between concepts, data, and entities in general.

[2] More than one period of disinterest in AI has been recorded, interspersed with what are called AI summers (Kautz, 2020). In addition, different sources use different dates for the AI winters. Consider having a look at Howe (1994) and Russell and Norvig (2003).

[3] It is usually customary to speak simply of Natural Language Processing, bringing language comprehension under its umbrella as a further specialization. Therefore we, too, will refer to NLP in this more general sense throughout this work.

1.3.2 The symbol grounding problem

The Symbol Grounding Problem represents a central and intricate challenge in the field of artificial intelligence and cognitive science research, raising the fundamental question of the connection between abstract symbols, sensory perception and the physical world. The Symbol Grounding Problem, as postulated by Harnad (1990), refers to the problem of how a system can assign meaning to symbols or representations that it processes, and is thus related to the problem of what "meaning" itself really is.

In the Symbol Grounding Problem, perception serves as the gateway through which symbols acquire meaning. AI systems, equipped with sensory apparatus, must interpret and map sensory data to symbolic representations. The interaction with the physical world plays an equally central role, establishing grounding as a dynamic process. Contextual understanding and disambiguation are fundamental to symbol grounding, since the same symbol can have different meanings in distinct contexts.

The Symbol Grounding Problem is cited here because it raises interesting arguments about the causal connection between symbols and their real-word reference.

1.3.3 Peirce's Pragmatism

Charles Sanders Peirce, a pioneering figure in American philosophy, laid the groundwork for a school of thought that would come to be known as Pragmatism. Born in 1839, Peirce's intellectual contributions spanned a broad range of fields, from philosophy and logic to semiotics and scientific inquiry. At the heart of Peirce's philosophy is the pragmatic maxim (Peirce, 1878):

> It appears, then, that the rule for attaining the third grade of clearness of apprehension is as follows: Consider what effects, that might conceivably have practical bearings, we conceive the object of our conception to have. Then, our conception of these effects is the whole of our conception of the object.

The maxim asserts that the meaning of a concept resides in its conceivable practical effects or consequences. In other words, the significance of an idea is revealed through its potential impact on experience and behavior. This emphasis on practical consequences as the touchstone of meaning distinguishes Pragmatism from more abstract and speculative philosophies.

Peirce's intellectual legacy extends beyond Pragmatism to include relevant work in semiotics. He proposed 3 parts to a sign: a sign/symbol, and object and an interpretant. Peirce's innovation resides in the idea of detaching the symbol/sign from the object it signified, and introducing the interpretation process as a key entity (Thórisson, 2020c). This leaves room for a possible explanation of how symbols and meaning can change depending on culture, and how people can misunderstand each other.

CHAPTER 2

Related Work

The foundation of any significant research endeavor lies in a thorough exploration and understanding of the key concepts and theories that underpin the investigation. In this chapter, we introduce and elucidate a few necessary notions upon which the research presented in this thesis rests. By delving into the theoretical framework, foundational principles, and seminal works in the field, we aim to provide the reader with a comprehensive understanding of the intellectual works that form the basis of our work. Each concept introduced is a building block that sets the stage for the novel contributions and insights that will unfold in the chapters to come.

We begin by discussing the concepts of agents and control as necessary elements for the development of a theory of intelligence aimed at the execution of practical tasks in a variety of worlds. We then introduce the need to make use of models for effective control of complex systems and the role of cognitive architectures. Next, we discuss the need for learning as a method for handling the high complexity of the worlds in which agents must operate, foremost among them the physical world. Particularly relevant is the learning of causal relationships, to which an in-depth section will be devoted. It will then be shown how to apply the notion of causality to the creation of structural models, usable as a basis for the development of intelligent agents. Finally, we summarize the most recent results on Task Theory and Pragmatic Understanding Theory, which are foundational for discussing our theory of meaning.

The concepts introduced in this chapter cover the contextual aspects of our work, namely the autonomous, situated, and experience-based agents capable of cumulative learning, the qualities this type of agents possess and related frameworks for task achievement and understanding in task execution. These notions further support our view of meaning as a pragmatic phenomenon, in a coherent narrative that finds meaning as its central pivot.

2.1 Autonomous grounded systems

In the field of artificial intelligence, the pursuit of computational agents with the ability to generate meaning has become a paramount challenge. As we delve into the dynamics of meaning and meaning generation, a critical consideration emerges – the role of autonomous agents and the nuanced mechanisms of control that govern their interactions with the environment. At the core of our theory lies the concept of control, a fundamental engineering principle that enables the manipulation of system dynamics to achieve desired outcomes.

Along with the need to deal with control and interaction, the importance of learning for intelligent agents becomes evident. Learning is not merely a feature but a necessity for these agents, as they operate within constraints of limited time and resources. Unlike static systems, intelligent agents encounter dynamic and ever-changing environments, requiring the ability to adapt and optimize their behaviors over time. Learning enables agents to gather insights from experience, allowing them to make informed decisions and improve their performance in novel and uncertain situations.

2.1.1 Control

Control mechanisms define the rules and strategies that govern how agents perceive data, make decisions, and adapt their behavior over time. At its most basic level, control can be achieved through the simple, yet powerful, concept of "feedback loop". In the simplest form of control, feedback is employed to continuously monitor the system's output and adjust the input based on the disparity between the desired and actual states. This mechanism forms the basis of a control loop. A controller can be abstracted as a set of **processes** P in a **state** S that can receive an **input** i, produced by and selected from an **environment**, having at least one **goal** G and returning an **output** o that pushes it toward its goals (Thórisson, 2020d). These components enable a cyclical process where the system iteratively approaches the desired state, minimizing the difference between the (perceived) actual state and desired goal states. Over time, different forms of control have been developed to manage the dynamics of systems in specific ways. We will begin by exploring classical control and then advance to more 'intelligent' approaches.

PID Control The **Proportional-Integral-Derivative (PID)** controller is a cornerstone in control theory, offering a versatile framework for regulating systems by considering the proportional, integral, and derivative components of the error signal. The controller receives an input value from a process and compares it with the desired setpoint (SP), calculating their difference – the error value e(t). The error value is used to calculate a correction by adjusting the output of the PID based on the value of the error signal (proportional action), the past values of the error signal (integral action) and how fast the error signal varies (derivative action), hence the name. Mathematically, the PID control law is expressed as:

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$

where:

- $u(t)$ is the control variable (e.g., the opening of a control valve),
- $e(t)$ is the error signal,
- K_P, K_I, and K_D are the proportional, integral, and derivative gains, respectively.

While PID controllers are versatile and often perform satisfactorily with only roughly tuning, they can perform poorly in some applications and do not, in general, provide optimal control (they do not optimize the objective function). The fundamental difficulty with PID control is that it is a control system based on feedback, with constant parameters, and no explicit knowledge of the process. The PID controllers' reactive nature makes them sensitive to changes in the system dynamics. Finally, the tuning process of a PID controller is a mostly manual and possibly time-consuming process (Ogata, 2010).

Model Predictive Control **Model Predictive Control (MPC)** embraces a predictive approach, optimizing control actions by considering a dynamic model of the system and predicting its future behavior. The main advantage of MPC lies in its approach to optimization of the current timeslot, while keeping future timeslots in account. At each time t the controller solves an optimal control problem over a finite future horizon of N time steps, but applies only the first optimal move and, at time $t+1$, gets new measurements and repeats the optimization again. Differently from PID control, MPC has the ability to anticipate future events and can take control actions accordingly. Model predictive control is a multivariable control algorithm that uses an internal dynamic model of the process, a cost function J over the receding horizon, and an optimization algorithm minimizing the cost function J using the control input u. An example of a quadratic cost function for optimization is given by:

$$J = \sum_{i=1}^{N} w_{x_i}(r_i - x_i)^2 + \sum_{i=1}^{M} w_{u_i} \Delta u_i^2$$

without violating constraints (low/high limits) with:

x_i: i^{th} controlled variable (e.g. measured temperature)
r_i: i^{th} reference variable (e.g. measured temperature)
u_i: i^{th} manipulated variable (e.g. measured temperature)
w_{x_i}: weighting coefficient reflecting the relative importance of x_i
w_{u_i}: weighting coefficient penalizing relative big changes in u_i

etc.

The MPC is easier to tune (compared to PID) and can handle structural changes, but it often requires a large number of model coefficients to describe a response and, if the prediction horizon is not formulated correctly, control performance will be poor even if the model is correct (Woolf et al., 2023).

Reinforcement Learning In the realm of AI and optimal control, **reinforcement learning (RL)** is a machine learning technique that aims to realize autonomous agents capable of achieving tasks. Reinforcement learning provides a paradigm where agents learn optimal (or nearly-optimal) control policies through interaction with their environments. Agents learn to take the actions that will maximise a cumulative reward, the *reward function.* Basic reinforcement learning is modeled as a Markov decision process, where:

S is a set of environment and agent states
A is a set of actions of the agent)
$P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability of transition at time t from state s to state s' under action a
$R_a(s, s')$ is the immediate reward after transition from s to s' with action a

Reinforcement Q-learning, a fundamental algorithm in RL, updates a Q-value table iteratively based on the observed rewards. The Q-value update is expressed as:

$$NewQ(s,a) = Q(s,a) + \alpha \left[R(s,a) + \gamma \max Q'(s',a') - Q(s,a)\right]$$

where:

- $NewQ(s,a)$ is the new Q-value for state s and action a,
- $Q(s,a)$ is the current Q-value,
- α is the learning rate,
- $R(s,a)$ is the reward for taking that action at that state,
- γ is the discount rate, and
- $maxQ(s,a)$ is the maximum expected future reward given the new s' and all possible actions at that state.

Reinforcement learning is overtly devoted to the execution of tasks, which makes it different from other machine learning models. The cons of reinforcement learning include the need for a large amount of data for learning, the intensive use of computational resources and high maintenance cost.

2.1.2 Agents

An agent is an *embodied* controller, a specific type of controller consisting of at least one sensor (a transducer that changes one type of energy to another type), at least one effector (a transduction mechanism that implements an action that a controller has committed to), and a controller (Thórisson, 2020d).

Definition 2.1.1 (**Agent** (Woolridge, 1997))**.** An agent is a computer system situated in some environment and capable of autonomous, flexible action in that environment in order to meet its design objectives.

Agents employ control techniques to act on the environment in which they are situated in order to pursue their goals. When operating in highly-complex, physical worlds, agents are dealing with open, unpredictable and often multi-agent environments: the inherent unpredictability and elevated complexity of such environments pose significant challenges to control activities, making the task of designing autonomous agents that can consistently accomplish tasks across a diverse array of situations particularly challenging.

In their paper of 1970, Roger C. Conant and W. Ross Ashby provide insights into the requirements for effective regulation in complex environments, proving that *"Every good regulator of that system must be a model of a system"* (Conant and Ashby, 1970). In light of the "good regulator" theorem, we understand that there is a fundamental need for models in control, suggesting that any theory of intelligence guiding the design of "intelligent" agents must be based on the concepts of *model* and *representation*, ultimately addressed by cognitive architectures. By cognitive architecture, we intend the internals of a controller for the complex, adaptive control of a situated agent. The architecture of an intelligent system dictates the nature of information processing achievable by an agent controller and defines the overall capabilities of the system within a specific environment (Thórisson, 2020d).

Intelligent agent architectures are typically classified according to the following scheme (from simplest to most complex): **reactive**, **predictive**, and **reflective**. We present each of these architectures below, making use of the descriptions given in Belenchia (2021).

Reactive agents Reactive agents only respond to the perceived sensory information from the environment, obtained through their sensors. Their architecture is mostly fixed through their lifetime, and while learning is possible the agent only ever reacts to stimuli and is incapable of proactive behavior. A system of this kind would be unable to hit a fastball in baseball: human brains typically employ a prediction mechanism to do that and excellent players still only hit a ball about 30% of the time. Most AI architectures are reactive, and examples of this type of systems span the very simple thermostats to the complex control systems of power plants. These types of systems are limited in the sense that they are built with an embedded model of the task they carry out which is unchangeable, bar the customization of a few parameters during run-time (Thórisson, 2020d).

Predictive agents Predictive agents are able to anticipate environment states and act in anticipation of sensory information. Their architecture is mostly fixed as in the case of reactive agents, but by means of predictive models they are able to act in a proactive, goal-oriented mode. Predictive agents also incorporate reactive control to achieve a more robust behavior. In particular, predictive agents are able to perform tasks which involve phenomena happening faster than the action-perception loop of the system. This type of agents, endowed with the capabilities of creating, selecting and evaluating models has the potential to be a truly general learner and also carries the potential to improve its own learning mechanism by modelling the learning itself (Thórisson, 2020b).

Reflective agents Reflective agents go a step beyond predictive architectures by enabling the agent to modify its own architecture (thus exhibiting cognitive growth) through introspection and meta-reasoning (Thórisson, 2020b). Two prominent examples of reflective agents are the Non-Axiomatic Reasoning System (Wang, 2004) and the Auto-catalytic Endogenous Reflective Architecture (*Bounded Recursive Self-Improvement* 2013), both of which aspire to be generally intelligent systems.

By *autonomous* and *grounded* systems, the kind mentioned in the title of this thesis, we mean precisely the intelligent agents described in this section, situated in some environment to fulfill some purpose. The autonomy requirements that such systems must possess are typically variable depending on their purpose, but, in general, greater degrees of autonomy correspond to a lower need for human intervention, so designing systems capable of operating highly autonomously is typically desirable. Autonomy is a key aspect of intelligent agents; we will cover it in Section 2.1.4.

2.1.3 Learning

As expressed in the introduction to this section, learning is a key feature of intelligence and an important tool for intelligent agents operating in dynamic and complex environments, so when we refer to the concept of agent we really mean *learning agents*. Learning is the process that leads to the acquisition of knowledge, or *actionable information* (Thórisson, 2022b). The fundamental need for learning stems from two main assumptions:

- Any complex and dynamic environment is challenging to model and will always present performers with novel situations;
- Learners are expected to work under the assumption of insufficient knowledge and resources (AIKR) (Wang, 2012).

The second assumption is particularly interesting, as it argues the substantial impossibility for an agent performing a task to take advantage of an infinite amount of resources or time. Agents must face the limits of our physical world: given that there is no such thing as an infinite amount of time, energy or resources, agents cannot record and store all their experiences in an immense lookup table to be queried on demand, but must implement increasingly efficient compression methods for information storage and retrieval (Thórisson, 2020c). Learning is thus a way of synthesizing a large amount of information while retaining aspects of it that are necessary for carrying out tasks.

But AIKR tells us even more: most of the time, for an agent in a complex world, information available to help the agent reach any goal or do any task is incomplete, incorrect, or absent and the agent should make effective use of whatever is available to it (Thórisson, 2022c). This brings out additional features of the learning process: knowledge acquisition does not happen all-at-once, so learning is rooted in *experience*, and agents must be capable of updating their knowledge with increasingly refined representations – i.e. *cumulative* learning (Thórisson, 2022b). The following are some of the main properties of the learning process, as identified and reported by Thórisson (2022b):

- **Purpose**[1]: the purpose of learning is to adapt and respond in rational ways to problems, to achieve foreseen goals;
- **Speed**: how quickly an agent learns;
- **Data**: learning should concern data in its various forms (e.g. continuous, discrete, symbolic, big, small, etc.);
- **Quality**: quality can be assessed on multiple dimensions. Noteworthy are *reliability* (consistent performance under repeated application) and applicability (correct application in relevant circumstances);
- **Retention**: can be measured using a battery of test administered multiple times;
- **Transfer**: the learner's ability to use something they have learned in similar or completely different situations;
- **Meta-Learning**: hard to measure, but might involve observing changes in knowledge acquisition on the above dimensions (Speed, Data, Quality, Retention, and Transfer);
- **Progress signals**: for artificial learners these are known and thus do not have to be measured.

The learning process involves several steps: the acquisition of models (through pattern extraction), evaluating the performance of existing models (identifying and eliminating unreliable models and updating the reliability score of the models), and monitoring the learning activity itself (learning about the learning process) (Nivel, Thórisson, B. Steunebrink, Dindo, et al., 2014). The agents in the focus of our attention are *grounded*, that is, they are situated

[1] This factor determines how the rest of the features in this list are measured

in an environment (physical or virtual) in which they have to perform the tasks assigned to them. As we already argued when we introduced AIKR, two features of learning are particularly relevant for grounded agents: **experience** and **cumulativeness**. We discuss them in more detail in the following paragraphs.

Experience-based learning Situated agents are typically put in a complex environment (such as the physical world) that cannot be known or specified entirely a priori, thus requiring agents to learn through experimentation and interaction with their surroundings – in other words *learning from experience* (Thórisson, 2022b).

Cumulative learning Cumulative learning is a learning mechanisms introduced to unify several separate research tracks in a coherent form that can be easily related to AGI requirements: multitask learning, lifelong learning, transfer learning and few-shot learning (Thórisson, 2022b; Thórisson, Bieger, Li, et al., 2019). Cumulative learning *subsumes* all of these aspects of learning in an organized framework. In short, cumulative learning provides a way of contextualize every tiny bit of information acquired into coherent knowledge (thus enabling multitask learning) in a continuous learning process (online lifelong learning); it makes use of a robust knowledge acquisition mechanism (which means that the acquisition of new knowledge does not mess up previously acquired knowledge) that is in turn used to facilitate learning new things (transfer learning). Transfer learning capability is, finally, used to acquire new knowledge with less data, making few-shot learning theoretically possible.

These two features combined form the idea of learning commonly associated with human beings (the only universally recognized example of an intelligent being). We therefore say that AGI agents of our interest must possess experience-based cumulative learning.

2.1.4 Autonomy

Autonomy is the capacity to govern one's own actions, to "act on its own" (Thórisson, 2022d). Autonomy has been commonly associated with artificial intelligence, robotics and multi-agent systems, and it has been analyzed both with respect to tasks (Chandrasekaran, Josephson, and Benjamins., 1999) and cognitive capacity and architecture (Thórisson and Helgason, 2012; Wooldridge and Jennings, 1995). Implemented systems show different autonomy capabilities: while systems that can move autonomously in closed spaced of limited extension are already pervasive to date (think about autonomous home vacuum cleaners), it is not yet common to encounter unsupervised robots on the street that perform some tasks entirely on their own. This difference is related, for example, to the difficulty of acting in spaces that cannot be modeled entirely a priori. Since not all systems are equally autonomous, identifying different levels of autonomy helps us understand how to make systems *more* autonomous. Thórisson and Helgason (2012) proposed a framework for the comparison of autonomy of systems, based on four dimensions: **Learning**, **Meta-Learning**, **Resource Management**, and **Realtime**. An effective visualization of this framework is given by the diagram shown in the Figure 2.1.

We can distinguish at least three main levels of autonomy: **mechanical** (automation), **cognitive**, and **biological** (Thórisson, 2020e). The lowest level of autonomy is where we put systems that perform some function, defined in advance, that remains fixed once they are deployed, such as thermostats, deep neural networks, and the like. Their architecture is fixed and their goals are clearly defined. This is typical of mechanical systems, but it does not mean that these systems are bound to performing simple tasks (think of neural networks for image

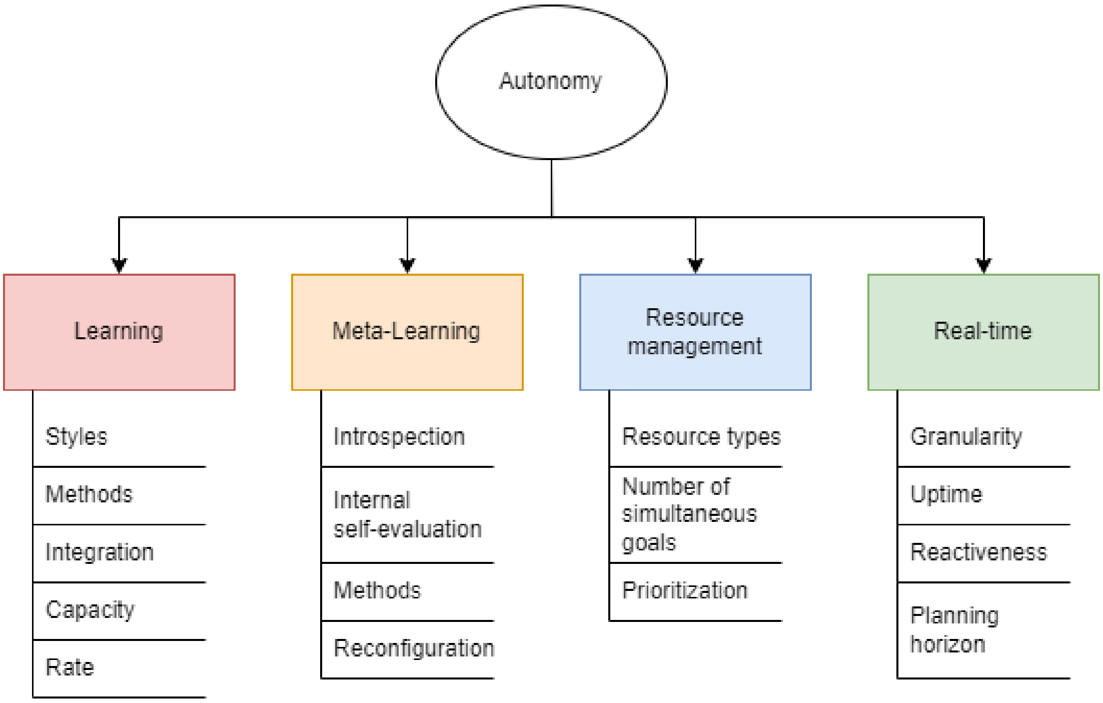


Figure 2.1: "Autonomy comparison framework focusing on mental capabilities." (Thórisson and Helgason, 2012)

recognition). Far these reasons, instead of calling these systems "autonomous", they are often referred to as "automatic" (Thórisson, 2020e). A small set of elements that exhibit slightly more autonomy than mechanical systems are simple reinforcement-based learning systems. These systems can change their function at runtime, but they cannot change their goals (let alone create subgoals) or handle unspecified variables. We could then place this small group at a level of autonomy somewhere between the mechanical and cognitive levels of autonomy. Moving up a whole level, we encounter systems possessing some cognitive autonomy. This is the ability to adapt and handle novelty ("figure things out"), as well as create new concepts. We can put at this level humans and a strict set of animals that show higher-level cognitive abilities, like dogs, crows, and parrots. The highest level is the biological one, where life resides. We consider it the "most autonomus" level because it is a prerequisite for the others (Thórisson, 2020e). However, it is also a distraction to those focusing on artificial intelligence because intelligence calls for discretionarily constrained adaptation, that is, the ability of the system to constrain its own behavior from knowledge by choice, through selecting and generating goals, sub-goals, new knowledge, and other factors, at its own discretion, through reasoning and logic (Thórisson, 2020a).

We therefore focus on the design of systems equipped with cognitive autonomy. Cognitive autonomy comes with its own requirements, in particular over the four processes of autonomous **selection**, **goal-generation**, **control of resources**, and **novelty handling** (Thórisson, 2022d).

Selection By 'selection' we mean the autonomous selection of **variables** and **processes**. Autonomous variable selection involves figuring out, from a set of variables, (a) *which ones* are relevant, (b) *how much*, and (c) *in what way*. Autonomous process selection, on the other hand, is deciding, for example, what types of learning algorithms to use (learning to learn)

(Thórisson, 2022d).

Goal-generation Starting with primary goals, goal generation consists in identifying aspects relevant to the achievement of the primary goal, defining intermediate steps – or *subgoals* – that, once achieved, lead to the achievement of the final goal (Thórisson, 2022d). For example, in carrying out a task, this process starts with breaking down the problem into many subproblems that can be solved individually.

Control of resources The resources available to an agent are of at least three types: computing power, time, and energy. Resource control is the set of activities designed to: (a) control the resource use, (b) plan for it, (c) assess it, and (d) explain it (Thórisson, 2022d).

Novelty handling The ability to handle novelty autonomously requires *autonomous hypotheses creation* related to variables, relation, and transfer functions (Thórisson, 2022d).

We are interested in autonomy in the execution of tasks, and, because learning is necessary to perform tasks, we want agents, among other things, to be capable of learning autonomously. Autonomous grounded systems emerge as agents that are capable of autonomous application of control mechanisms and experience-based cumulative learning to fulfill their duties.

2.2 Causality

The concept of causality refers to the relationship between two phenomena, in which the first phenomenon is in some way the *cause* of the second, that is, it determines their existence. When it comes to performing tasks, causality plays a central role. Any intelligent system that has to perform a task has as its goal the attainment of some target state, that is, a set of variables that take values within particular ranges. In order to reach this state, it is necessary to understand the functions that determine the evolution of these variables and how to act to modify them at will, that is, to *cause* controlled changes in the variables. Thus, any agent that is performing a task needs to learn, among various types of relationships, the *causal* relationships that relate its perceptions and actions to the task's goals (Belenchia, 2021). Perhaps one of the most influential and prolific authors on the topic of causality is Judea Pearl (Pearl, 1988; Pearl, 2009; Pearl and Bareinboim, 2014). In this section we will introduce Pearl's approach to causality, causal relationships and causal diagrams. Causal diagrams will also be discussed as a basis for the representation of tasks, topic that we will take up later when we introduce the basics of Task Theory introduced by Thórisson and Belenchia.

Pearl and Mackenzie's introduce causality using the metaphor of a three step ladder, where each step corresponds to a level of causal reasoning (Belenchia, 2021): **association**, **intervention** and **counterfactuals** (Pearl and Mackenzie, 2018, pp. 27-43). These levels describe a progression in cognitive abilities from the lowest level of reasoning by association to the highest level of counterfactual reasoning (Belenchia, 2021).

The first step of the ladder, association, is the lowest level of causal reasoning, and is concerned with finding associations between variables. This requires making predictions based on passive observations of the world. A possible example of reasoning at this level comes from the observation that a rooster crows at dawn. Simply having the observation of the simultaneity of the two events, we cannot say that the rooster crowing causes the sunrise or, vice versa, that the sunrise causes the rooster crowing. We can simply note that given one of the two events, there is a certain probability that the other will also occur (based on our experience). Any type of association, in general, cannot discriminate causes from effect and neither can tell if any causal relationship is present at all (Belenchia, 2021), but associations can still serve as good predictors even without causal knowledge. Pearl and Mackenzie claim that most learning systems are also stuck at this level of cognition, taking machine learning and deep learning algorithms as examples. However well designed, these algorithms can only receive data unbundled from causal relationships; therefore, they will be limited to identifying associations of observations (Pearl and Mackenzie, 2018; Belenchia, 2021).

Moving up a bit, on the second rung of our ladder, we find "intervention". The focus shifts from observing the world to actively intervening in it. This level of reasoning responds to questions such as 'if I do X, how will X affect the probability of Y?'. This kind of reasoning can be used to answer more questions, specifically the ones involving causality. An agent that can reason about interventions can predict the consequences of its and other agents' actions, as well as generate plans to reach desired states (Belenchia, 2021). In order to do this, data alone is not enough; a causal model of the world is also required. Causal models allow backward chaining from the desired goal states to the agent's range of possible actions, which then acts in a goal-driven manner. Pearl and Mackenzie cite children as examples of goal-driven reasoning organisms (Pearl and Mackenzie, 2018).

Climbing further up our metaphorical ladder, we reach the highest level of reasoning, the one labeled 'counterfactuals', where it is possible to answer questions about past states that never were but could have been, had circumstances been different. Practical examples made possible by this kind of reasoning are uchronic literary products, such as the novel "The Man in the High Castle" written by Philip K. Dick, representing an alternate universe in which the Axis Powers won World War II. An agent implementing this type of reasoning can reflect on its past actions and figure out how to do better in future similar situations, but also learn from the experience of others (Belenchia, 2021). Pearl and Mackenzie argue that only modern humans are capable of using this kind of reasoning, and that it is precisely the ability to understand why things have unfolded one way and not another that has made possible the technological breakthroughs that form the basis of modern civilization (Pearl and Mackenzie, 2018; Belenchia, 2021).

Cause types A cause can be **necessary**, **sufficient**, **contributory**, or a combination of the three (Epp, 2004, pp. 25-26).

- x is a *necessary* cause of y if the presence of y must imply the prior occurrence of x, but the presence of x does not imply that y will occur (Pearl and Mackenzie, 2018);
- x is a *sufficient* cause of y if the presence of x must imply the subsequent occurrence of y. However, the presence of y does not require the prior occurrence of x, as another cause z may independently cause y;
- x is a *contributory* cause of y if the presence of x must increase the likelihood of y (if the likelihood is 100%, then x is *sufficient*).

2.2.1 The manipulative approach to causation

In this work, we refer to the approach to causality known as "manipulative approach" (Pearl, 2009, pp. 223-228). This approach is based on a precise view of the physical world as a set of independent, invariant mechanisms, each described by a set of variables. Interaction among these mechanisms occurs through shared variables (i.e., variables that are part of multiple mechanisms), so in order to predict the (causal) consequences of an action (the effect of which is understood as a change in the values of certain variables) it is necessary to reconstruct the interactions that occur between the various mechanisms until a new steady state is reached (Belenchia, 2021). The manipulative approach assumes that each action performed has a limited and local effect, altering only a subset of mechanisms while other remain untouched (Belenchia, 2021). In contrast, using Pearl's own example, tipping a single tile of a domino array is not a limited and local action at all, as it will knock down all the other tiles that have been aligned to that domino piece. But this action is *local* in the sense that it only affects the physical mechanism keeping that single tile erect and still. Because of this locality, which does not influence other factors in the system, it is possible for any other agent capable of reconstructing the causal model of the example to understand the consequences of the action (Belenchia, 2021). In this way, causal diagrams prove optimal in the analysis of the effects of actions, as they can provide answers to this kind of questions without the need to compute all possible outcomes of the action (Pearl, 2009).

2.2.2 Causal models

Causal models are models describing the causal mechanisms of a system. They can allow some questions to be answered from existing observational data without the need for an interventional study. Causal models have found applications in signal processing, epidemiology and machine learning (Pearl, 2009). We are primarily interested in the implications that causal models have relative to the last of the points mentioned, namely, the application of causal models to learning processes and, more generally, as tools for knowledge representation in intelligent systems. A causal model is formally defined as:

Definition 2.2.1 (**Causal Model**, **Structural Causal Model** (Pearl, 2009, p. 203))**.** A causal model is a triple:

$$M = \langle U, V, E \rangle$$

where:

i) U is a set of background variables that are determined by factors outside the model, also called *exogenous* variables;

ii) V is a set $\{v_1, v_2, ..., v_n\}$ of variables, called endogenous, that are determined by variables in the model, i.e. $U \cup V$;

iii) E is a set of structural equations $\{e_1, e_2, ..., e_n\}$ such that each e_i is a mapping from (the respective domains of) $U_i \cup PA_i$ to V_i, where $U_i \subseteq U$ and $PA_i \subseteq V \setminus V_i$ and the entire set E forms a mapping from U to V. Or equivalently, each e_i in:

$$v_i = e_i(PA_i, U_i), \qquad i = 1, ..., n \tag{2.1}$$

assigns a value to v_i that depends on the values of a select set of variables in $U \cup V$, and the entire set E has a unique solution $V(u)$.

It is important to note that these structural equations are not necessarily reversible and each of them represents an *autonomous mechanism* (Pearl, 2009, p. 27). By autonomous mechanism is meant that each equation is not affected by changes in other equations, and therefore an intervention that targets one variable v_i leaves all other equations in place for any v_j with $j \neq i$. Instead, we say that these equations are irreversible because, when dealing with *interventions* they cannot be reversed to determine any other variable than v_i. Using the example of Belenchia (2021), if the structural causal model includes the equation $v_i = 5 \cdot v_j + u$, then the equation $v_j = \frac{v_i - u}{5}$ does not necessarily hold. Intuitively, if the value of v_j is on the right hand side of the structural equation of v_i, it means v_i is determined by v_j, that is, a change in v_j influences the value of v_i. The opposite does not necessarily hold unless, of course, v_i is also present on the right hand side of the structural equation of v_j too. Therefore the equality sign in structural equations behaves as in standard algebra only when dealing with *observations*, that is, when the equations represent only the *observed* relationships between variables in the model. This different interpretation of equalities is further clarified by the operational definition of structural equations given by (Pearl, 2009, p. 160) (Belenchia, 2021).

Definition 2.2.2 (**Structural Equations** (Pearl, 2009, p. 160))**.** An equation $y = f(x) + \epsilon$ is said to be structural if it is to be interpreted as follows: in an ideal experiment where the value of X is set to x and any other set Z of variables (not containing either X or Y) is set to some value z, the value y of Y is given by $f(x) + \epsilon$, where ϵ is not influenced by either x or z.

The consequence of this definition is that it is only concerned with the value of y: nowhere it states anything about what values x or ϵ can take when controlling for Y.

2.2.3 Causal diagrams

From the definition of causal patterns given above, it is possible to define an additional concept, that of causal *diagrams*. A causal diagram is a representation of the relationships between the variables of a causal model. More precisely, any structural causal model has an associated graph where each vertex corresponds to a variable v_i and a directed edge is drawn from each $pa \in PA_i$ to v_i (Peters, Janzing, and Schölkopf, 2017). In other words, a directed edge is drawn from any variables occurring on the right hand side of each equation (2.1) to the vertex occurring on the left hand side (Belenchia, 2021). The resulting graph, which we call "causal diagram", is finally assumed to be a directed acyclic graph. The concept of causal diagrams is especially useful in the context of intelligent agents performing tasks. Such agents, once they have generated or otherwise obtained a causal model of the system in which they are situated, can understand how to reach a target state from the causal diagrams identifiable in the model they possess[2]. Let us break down and analyze the above definition of causal diagrams by drawing on some concepts from graph theory, resuming the descriptions given by Belenchia (2021).

A **graph** is a pair $G = (V, E)$ that consists in a set of vertices, also called nodes, V and a set of unordered pairs of vertices, called edges, $E \subseteq V \times V$. A graph is **directed** if the set of edges E consists in ordered pairs of vertices $(i, j) \in E$, where i represents the **source node** and j represents the **destination node**, and each edge is rendered as $i \rightarrow j$. Edges of this type can be properly called directed edges or arrows. Two nodes i and j are considered **adjacent** if either $(i, j) \in E$ or $(j, i) \in E$, and a graph G is **fully connected** if all nodes are adjacent with each other. The **in-degree** of a node is the number of incoming directed edges, while the **out-degree** of a node is the number of outgoing directed edges. A node i is a **parent** of a node j if $i, j \in E$ but $j, i \notin E$; in such case j is also called a **child** of node i. The set of parents of a node j and the set of children of a node i are denoted $\mathbf{PA}_j$ and $\mathbf{CH}_i$ respectively. A **path** is a sequence of edges joining a sequence of adjacent nodes, with all

[2] In order for such agents to actually reach the target state other considerations are necessary, we will discuss these later along with Task Theory

edges and nodes distinct. A **directed path** is a path such that the destination node of each edge in the path is the source node of the following edge in the path. If there exists a directed path from any two nodes i and j, i is called an **ancestor** of j and j is a **descendant** of i. A **directed acyclic graph** (DAG) is a directed graph with no directed cycles, that is, for any two nodes i and j, either there is a directed path from i to j, from j to i, or neither of the two. Consequently, if $(i, j) \in E$ then $(j, i) \notin E$.

2.2.4 The common cause principle

As Belenchia (2021) reminds us, every student of statistics knows well that "correlation does not imply causation," although "some correlations do imply causation"(Pearl and Mackenzie, 2018, p. 77). The concept of correlation can be traced back to the works of Sir Francis Galton in 1888, when he noted that the forearm and height measurements could be represented as points on a line, which he later called a *regression* line, which proved reliable in predicting one of these two values knowing the other (Galton, 1888). The formula for correlation was later introduced by Karl Pearson, the father of statistics, and the slope of the regression line was called the correlation coefficient. Correlation is a type of association, the first "step" of Pearl and Mackenzie's ladder.

Not all associations are meaningful, however. Take as an example the correlation between the number of movies Nicholas Cage releases in a year and the number of drownings in swimming pools in the same year (Geraghty, 2018). Does this mean that, whenever Nicholas Cage releases a new movie, people get excited and decide to go jump in the pool? Or perhaps Nicholas Cage draws inspiration for his films in years when there are many drownings? This association might appear moderately strong (having a coefficient of 0.66!), but it just seems spurious. What may have happened instead is the advent of a hidden common cause, a *confounder*, that somehow caused both events. This is exactly what is stated by Reichenbach's Common Cause Principle, asserting that it is not possible to discriminate which associations are meaningful and which are not without referring to the concept of causation (Pearl and Mackenzie, 2018, p. 72). In short, what this principle says is that for any association found in the data, there must be a causal explanation for it. While this statement might appear intuitively true (and in fact, it is in most cases), there are some known exceptions (Belenchia, 2021). Association between variables might arise as, e.g., a consequence of selection bias, or spurious associations might arise when looking at time-series data of phenomena that are both developing over time. Finally, some associations appear solely due to random chance. When none of these considerations apply, the Common Cause Principle provides the only possible explanation for the observed dependence (Peters, Janzing, and Schölkopf, 2017; Belenchia, 2021, p. 7).

2.3 LTE

According to the definitions of intelligence introduced in Chapter 1, the main purpose of intelligence is to figure out how to get new stuff done given limited resources and knowledge, i.e. achieving things on a budget. AI systems must necessarily be developed taking into account the limitations imposed by the physical world. If for disquisitions of a purely theoretical nature we can assume that we have, for example, an infinite amount of time to solve a problem, or an infinite amount of space to store any piece of information encountered, the same assumption cannot be made for system that are to operate in the physical world. There is not an infinite amount of any physical resource or time that can be made available to a system performing a task. This is exactly what is stated in the **Assumption of Limited Time and**

Energy (LTE). This assumption is especially relevant since no task, no matter how small, takes zero time or zero energy to be carried out (Thórisson).

No task takes zero time or zero energy (Thórisson) If te is a function that returns time and energy, an act of perception (reliable measurement) $te(p \in P) > 0$ a commitment to a measurement m, $com(m \in M)$, that is, the measurement is deemed reliable, in $\{te(\{com(m \in M)\}) > 0\}$, and an action $te(a \in A) > 0$ to record it (to memory) or announce it (to the external world, e.g. by writing). It follows deductively that any Task requires at least one of these, possibly all. For a task T whose Goals have already been achieved at the time T is assigned to an agent A, A still needs to do the measurement of the Goal State s, $s \in S$, or at least* commit to the measurement of this fact, and then record it. Even in this case we have $te(T) > 0$. Anything that takes zero time or zero energy is by definition not a Task.

*This could be the case if, for instance, task T_1 assigned to agent A at time t_1 comes with Instructions telling A that T_1 has already been done at the time of its assignment t_1. Sort of like you were to get a shopping list for use on your upcoming shopping trip where at least one item was crossed out.

Limited Time & Energy (Thórisson) For all $\{T, T_e\} : te(T) > 0$ where $T_e \subset W$ is a task-environment in a world, T is a task, and te is a function that returns time and energy.

2.4 Extended structural causal diagrams

Before we can introduce the main results of our research on meaning, we need to review the concept of structural causal models described by Pearl as a useful basis for the development of generally intelligent systems (Belenchia, 2021). Pearl's idea is to use such diagrams, together with a causal inference engine (engine) (Pearl and Bareinboim, 2014, Figure 1), for the creation of agents capable of causal reasoning (Pearl and Mackenzie, 2018). Peters, Janzing, and Schölkopf (2017) illustrate possible applications of structural causal models applied to machine learning and deep learning technologies (Belenchia, 2021). As expressed in Belenchia (2021), causal structural models can be effectively used in task modeling in a way that is less related to the characteristics of specific learning systems.

However, the causal calculus used to derive the causal effects of a causal diagram presents several problems. First, it requires very large, if not infinite, amounts of data in order to be effectively applied. However, any agent operating in the physical world is expected to work under the Assumption of Insufficient Knowledge and Resources (AIKR) (Wang, 2012), according to which, in complex, physical environments, information on how to achieve any goal or task is partial, incorrect or absent most of the time, so an agent should be able to make use of whatever is available to it. Causal calculus also does not deal with the observability and manipulability aspects of the variables. In fact, causal calculus always admits operators that act on all the variables represented. However, for any given task related to the physical world, there are variables that are not directly manipulable or observable. The observability and manipulability of such variables might even vary over time, even as a function of the agent's actions. It would then need to be allowed to clearly specify which variables are goals so that these goals can be verified by the performer at any time. Finally, time and resource aspects in general are completely overlooked. As mentioned above, this is not permissible according to AIKR. Time is a fundamental resource for any task that must be performed in the physical world, along with other types of resources, energy for example, that are inevitably depleted during the task's execution. For all these reasons, the approach of causal calculus seems ill-suited to be used as an effective reasoning tool in any real time physical world scenario.

To overcome the identified limitations of causal calculus, Belenchia (2021) proposes an extension of structural causal diagrams that includes attributes to represent the three types of variables identified by Thórisson and Talbot (2018b) within them: *manipulatable* variables, *observable* variables and *goal* variables. Each of the variables in a structural causal model can be assigned multiple types or none at all (in which case we refer to the variable as *factor*).

Manipulatable variables

The first type of attribute we are going to illustrate are manipulatable variables. A manipulatable variable ('manipulatable' for short) is a variable that a controller can affect.

In the case of tasks in physical environments, an example of a variable that can be manipulated directly by the agent is typically the physical interface between the controller and one of the actuators it controls. Belenchia (2021) presents as an example a simple robotic arm, whose actions can be modeled by a manipulatable variable that can also take the values $\{Move\ up, Move\ down, Move\ left, Move\ right\}$. The domain assumed by the variable is, in this case, very simple, but it can be made more complex (consistent with what the physical actuator allows), and a single actuator more can also be controlled by multiple manipulable variables.

The difference between the variable and the actuator lies in the fact that the variable is an abstract specification (at best understood as a signal carrying information), while the actuator is physically constrained in terms of space, time, and energy. An actuator consumes a certain amount of resources, both energy and time, in performing a task, and is subject to the limits imposed by physical laws, whereas a manipulatable variable could, theoretically, take on arbitrary values (Belenchia, 2021).

Manipulatables are not necessarily referred to a controller's actuators, they can be used to represent entire sub-tasks and sub-goals, namely, aspects of a task the controller already knows how to perform at some level of precision. Taking another example from Belenchia (2021), a controller which knows how to open and close doors, may have a manipulatable variable whose domain consists of the two actions $\{Open\ \ door, Close\ \ door\}$, which, in turn, directly affect a set of variables describing a door. Any part of the task already known to the controller should be modeled with an appropriate manipulable and should not be part of the actual task (Belenchia, 2021).

Manipulatable variables are represented in causal diagrams as a variable with at least one outgoing arrow to another (non-manipulatable) variable representing the affected physical entity. More manipulatables might influence – even constraining – the same non-manipulatable, preventing the execution of certain actions. This is why it is appropriate to speak of *partial manipulability* of variables: the manipulability of variables depends on time and the current state of the task (Belenchia, 2021).

According to the extension proposed by Belenchia, manipulatable variables have no input arrows in a structured causal model, so the controller is the only source of change in these variables and it is assumed never to be part of the task.

In a broader sense, manipulatable variables could also be considered those that can be influenced by the controller indirectly, through other manipulatable variables. We distinguish, therefore, between *direct* and *indirect* manipulatable variables. This difference may come in handy in distinguishing variables that are manipulatable in general and those that are not manipulatable at all.

Observable variables

An observable variable, or 'observable' for short, is defined as a variable whose value can be accessed by the controller. In the case of agents operating in the physical world, controllers receive values of observable variables from their sensors at any given time. Again, it is good to distinguish between the observable variables and the phenomena to which they refer: the observable variables are the result of an *subjective* measurement process[3], while the phenomenon is the objective entity that is perceived. The existence of this difference is due to the very nature of the physical world, where a sensor will never collect objective values, but in its readings (limited in quantity and quality by the agent's experience) noise will always be present. Migliore è il sensore, migliore sarà la fedeltà della rappresentazione che produrrà, ma esso sarà sempre soggetto a fattori interni ed esterni alla task che ne modificheranno la precisione (Belenchia, 2021).

In a causal diagram, observable variables have at least one incoming arrow from the (non-observable) variable representing the physical entity under observation. As in the case with manipulable variables, further incoming arrows can be drawn from other non-observable variables representing other factors that may affect the resulting observation by the sensor. These additional factors allow the *partial observability* of variables: they can improve or compromise the quality of observation, or even prevent it altogether. In this sense, the observability of a variable is time-dependent and might change over time during the execution of the task.

Goal variables

A goal variable is a variable that, possibly together with other variables, contributes to describing the desirable state toward which an agent tends in its execution of a task. In order for a goal to be completed, the values of all the variables that constitute the goal must be within specified ranges. This formulation applies as much to positive goals as it does to negative ones: in the former case, the goal will be considered *successfully* completed, while in the latter case it will be considered *failed*. There can be multiple goals, positive and negative, in a task; in this case, the task will be successfully completed if and only if all the positive goals are successfully completed, while it will be considered failed if even one of the negative goals is reached.

2.4.1 Time, energy and other resources

Since this work focuses on the development of meaning in the physical world, consideration must be given to the use of resources by the various functions performed by the agent. In particular, the previously introduced AIKR raises the need to consider the resources available for the performance of any task as insufficient. The kind of resources we refer to, considering the physical world, are at the very least time and energy.

Causal diagrams can be extended to model time series data, thus proving particularly useful as a tool for studying the temporal aspects typical of any task that is to be completed in the physical world. Reasoning about causation on variables that refer to different moments in time might be considered easier than timeless data, given that causation can occur only forward in time (Peters, Janzing, and Schölkopf, 2017). We can imagine causal diagrams rooted in time as extending indefinitely into the future, thus including an infinite number of nodes. The nodes of a causal diagram are now denoted by X_t^j, where $j \in \{1, \ldots, d\}$ is the index denoting a

[3] Pattee (2001) is recommended reading for a more in-depth discussion of the concept of "measurement"

variable in the d-dimensional vector $\mathbf{X_t}$ and $t \in \mathbb{Z}$ is the temporal index of the variable. We can also distinguish between two types of causal relations that can be represented by temporal causal diagrams: *concurrent* relations, occurring during the same time step, and *consecutive* relations, occurring from a past time step to a future one (a causal relationship cannot occur from the present to the past, so it is not considered in this list) (Peters, Janzing, and Schölkopf, 2017). According to this classification, we say that a causal diagram has *instantaneous effects* if it contains only connections between variables that share the same temporal mark. A causal diagram that, on the other hand, contains only connections between variables with different temporal marks is said to contain no instantaneous effects (Peters, Janzing, and Schölkopf, 2017).

Although instantaneous effects are considered impossible from a physical point of view (in fact, any causal relationship necessarily takes a certain amount of time to occur), this distinction is necessary because there can be effects that take place over a shorter period of time than is detectable by an agent's sensors (Belenchia, 2021). Therefore, to the agent the event will be instantaneous (think of how the pressing of a switch and the subsequent turning on of a light bulb appears instantaneous to a human observer).

If time can be represented as a sequence of nodes with associated time indexes, energy and other resources, on the other hand, are more like fuel that can be consumed a little at a time to perform actions. Taking energy as an example, it is used to power sensors that sense the environment and to keep the basic functions of the controller's body active, but also to act on the environment itself through the manipulation of manipulable variables. In this sense, we can associate the action of modifying a variable with a certain amount of energy to be deducted from the available energy reserve (the same applies to all other kinds of resources) (Belenchia, 2021).

2.5 Task Theory

As both Thórisson, Bieger, Thorarensen, et al. (2016) and Belenchia (2021) clearly express, tasks are of primary importance for Artificial Intelligence research. AI systems are built to carry out tasks, whether they are performed in partially or completely known or even unknown environments. Tasks are not only a generic pivot around which AI systems are built, but are of decisive relevance to the careful design, training and evaluation phases of any AI system being implemented. The lack of a task theory in AI has led to the use of extensive domain knowledge to guide the design of task-specific systems, targeting a limited variety of environments, and the use of psychological theories of human intelligence for evaluation, with very poor results. Furthermore, for the development, training and evaluation of generally intelligent systems (the ones we ultimately care about) domain knowledge and tests like e.g. the Turing test or IQ tests don't nearly cover the wide variety of situations these systems would be facing and a task theory that can model a broad range of tasks and environment becomes absolutely necessary (Thórisson, Bieger, Thorarensen, et al., 2016). Despite its relevance to the field, there is still no comprehensive general framework that encompasses all aspects of tasks, but significant work has been done in recent years by Thórisson, Bieger, Thorarensen, et al. (2016), Bieger, Thórisson, et al. (2016), Thórisson, Bieger, Schiffel, et al. (2015), Eberding et al. (2021), and Belenchia (2021) that has provided additional insights, allowing us to move even closer, bit by bit, to defining such a theory. Below I introduce the main insights and results set forth in the aforementioned works, which will also be the starting point for the work laid out in this paper. With reference to the AI Constructivist doctrine (already mentioned in Section 1.2) the ideas expressed in the referenced works converge into what the constructivists refer to as "Task Theory" (with capital 't's – not to be confused with a generic "task theory" – with lower-case 't's).

2.5.1 Uses and requirements of a task theory

The three main aspects where a task theory would be most useful are evaluation, training and design (Thórisson, Bieger, Thorarensen, et al., 2016). The **evaluation** of AI systems is a way of measuring the progress of a system during its development by making comparisons with earlier versions and other systems. The difficulty in evaluating AGI systems (the ones we are ultimately interested in) lies, among other reasons, in the need to measure not performance on a specific task, but to provide a somewhat general measure of the cognitive abilities of such systems. A task theory would enable the evaluation of different systems, at different stages of development and on different tasks, by relating the tasks' parameters to attributes like determinism, ergodicity, continuity, asynchronicity, dynamism, observability, controllability, periodicity and repeatability (Thórisson, Bieger, Thorarensen, et al., 2016). In addition, a task theory would make it easier to construct both new task-environments and their variations and scaling them up or down in complexity. The creation and tuning of task-environments is also relevant to the **training** phase of AI systems. Finally, as we mentioned earlier, the **design** of AI systems today is mostly a matter of trial-and-error, intuition and domain knowledge. A task theory would help and speed up the design of narrow-AI systems by allowing the prediction of task requirements, such as time, energy or other resources. A task theory would also come with the ability to compare and describe properties of tasks (Thórisson, Bieger, Thorarensen, et al., 2016).

Summing up the above discussion, in (Thórisson, Bieger, Thorarensen, et al., 2016) the authors lay out the requirements for a task theory:

1. *Comparison* of similar and dissimilar tasks;
2. *Abstraction* and *concretization* of (composite) tasks and task elements;
3. Estimation of time, energy, cost of errors, and other resource requirements (and yields) for *task completion*;
4. Characterization of task complexity in terms of (emergent) quantitative measures like *observability*, *feedback latency*, form and nature of *information/instruction* provided to a performer, etc.;
5. Decomposition of tasks into subtasks and their atomic elements;
6. Construction of new tasks based on combination, variation and specifications.

A task theory fulfilling these requirements is expected to allow the development of frameworks that can construct task models, which would be suitable for, e.g., produce variants of tasks and execute tests in batch mode while providing huge amounts of data for the AI system being tested. The theory should be also grounded in physical reality by including and addressing the aspects of energy, time and other resources in tasks (Thórisson, Bieger, Thorarensen, et al., 2016).

In the following sections we are going to outline the principles and developments of constructivist Task Theory, which to date represents perhaps the only attempt to define and organize in a scientific and formal manner the basic requirements and concepts for a theory of tasks.

2.5.2 Foundational concepts

At the core of a task theory reside the fundamental concepts of environment, state, agent, goal, problem, and task. For these concepts to effectively support (modular) construction and analysis of tasks, they must be defined in a way that does not stray too far from their intuitive notions (Thórisson, Bieger, Thorarensen, et al., 2016). We already introduced the notion of *agent* and at least mentioned the other concepts, therefore we will provide a more comprehensive discussion of these other concepts here.

At the highest level of our conceptual framework for task-environments lies a **World**, denoted as W. This world is an interactive system comprising a set of variables V, dynamics functions F, an initial state S_0, domains D representing potential values for those variables, and a potentially empty set of invariant relations between the variables R. In a concise notation, we express this as $W = \langle V, F, S_0, D, R \rangle$. The variables of V (taking values in their respective domains) represent particular aspects of the world that may change or hold a particular value. The dynamics functions describe how the world transitions from a state to another. Invariant relations are Boolean functions on variables that always remain true, regardless of the possible states in which the system might ever find itself. **Environments** are subsets of the variables, domains, functions, etc. of the world, like 'views' on the world itself. A concrete **State** S is a value assignment to all variables of a system. A state is said to be *partial* if it only defines assignments to a subset of the variables of a system. Partial states are more *practical*, in the sense that any agent will almost always deal with partial states: noise and partial observability make it impossible, in most cases, to know most values with absolute precision (Thórisson, Bieger, Thorarensen, et al., 2016). A **Goal** state is a (partial) state that an agent should reach or avoid (*failure* state). A **problem** is specified by (at least) an initial state, desirable goal states and failure states. The *solution* to a problem is a sequence of actions resulting in a path through the state space reaching all of the desirable goal states and none of the failure states. If a solution to a problem P is known, P is said to be a *closed* problem. Finally, we define a **Task** as an *assigned problem*, that is, a problem assigned to an agent to perform. A task is performed successfully once a path that solves the problem can be found in the history of the world's states. Since tasks are ultimately performed in environments and environments are such an integral part of the task that they change the nature of the task itself, we talk of the **Task-Environment** pair.

In addition to the previous concepts is the notion of causality, which we have already extensively presented and discussed in the previous sections. Causality, as already expressed, is fundamental to being able to perform a task, as the simple correlation identified between two or more variables may not be of practical use. Picking up on the example from earlier about the correlation between the number of movies starring Nicholas Cage and drowning incidents in swimming pools, if one were to rely solely on the correlation between these two events, one might think that removing Nicholas Cage from the Hollywood scene and preventing him from starring in movies would also succeed in reducing the number of pool accidents. Or, again, that preventing a rooster from crowing would result in a few more hours of sleep each night. The concepts of causal models and diagrams are also good candidates for the representation of task-environments themselves, as they allow for the representation of aspects like the task-environment's variables, goals, constraints, dynamics functions, etc. We will introduce later, in Section 2.4, an extension of causal structural diagrams suitable for this purpose.

2.5.3 Intricacy and Difficulty of a task

In his work, Belenchia (2021) lays out and discusses some key principles of tasks. We will briefly report these principles, providing limited discussion when deemed necessary, as the rationale and implications of each of the proposed statements are very thorough and do not belong in this paper (for an in-depth discussion we recommend reading Belenchia (2021)).

In short, Belenchia states that the environment, including the body of the controller, is part of the task. He also argues that the level of detail is part of the task, meaning that any task is limited to its level of detail and the same task, presented at another level of detail, is a different task. He further claims that a controller's sensors and actuators define the limits of relevant spatio-temporal task detail, so the finest possible level of detail for a task depends on what the body allows the controller to observe and manipulate. Therefore, tasks described at more fine-grained levels of detail than what the controller's body allow would be experienced by the controller at coarser level of detail. Finally, Belenchia asserts that a task is unaffected by variables which do not constrain its solution space.

Based on these assumptions, Belenchia defines the concept of task **Intricacy**. Intricacy is a measure of a task's "complexity" based based on objective, purely physical and measurable parameters. By "complexity" is meant the "complicatedness" of a task, rather than the classical computer science measure of algorithms.

Definition 2.5.1 (**Intricacy** of a task T)**.** The intricacy of a task T is defined as the measure of a task's "complexity" based purely on physical, measurable parameters. It can be measured in either of the following ways:

1. The minimal number of relational models required to capture the subset of $\mathfrak{R}_T^{in}$ which includes only relations on the causal path to some goal.[4]
2. The number, length and type of mechanisms of causal chains that affect observable variables on a causal path to at least one goal.
3. The size of the smallest solution tree that can be constructed where all nodes are manipulatable variables.
4. Size of the solution space relative to the number of possible action sequences.

Intricacy is defined in relation to the *physics* of the task. it can be intuitively seen as a measure of what physical mechanisms need to be known by any intelligent controller to perform the task in the given environment (which includes the controller's body). The task's intricacy is *invariant* on the initial values of the task's variables (Belenchia, 2021).

Given this objective measure, it is possible to talk about the subjective *difficulty* of a task for the specific agents performing it. From the fundamental principle concerning the effect of superfluous variables on the task, it follows that the difficulty of performing a particular task is not uniquely determined by the task itself (i.e. its intricacy), but also depends on the performing agent (Belenchia, 2021). Some controllers may be better or worse at performing the task than others for a variety of reasons, from having performed similar tasks in the past, to being faster (or slower) at learning cause-and-effect relationships, and even sensor accuracy. Difficulty is therefore dependent on both the task and a controller.

Definition 2.5.2 (**Difficulty** of a task T for a controller C)**.** The difficulty of a task T assigned to a controller C is defined as the cross product of the task's intricacy and the level of understanding of the performing controller: $\{T \times C\}$.

[4] The models referred by this definition are of the type described in Section 3.4 and by $\mathfrak{R}_T^{in}$ is meant the set of inward facing (causal) relations of the task (see Section 2.6).

2.6 Thórisson's Theory of Understanding

A concept closely related to meaning is *understanding*. We consider the ability to understand, e.g., the meaning of the actions we take or the situations we find ourselves in to be important, and , when we assign a task, we tend to assign to someone who understands what they have to do. An artificial agent capable of performing a task repeatedly and reliably often makes us wonder whether the agent "understands" what it is doing or not (Thórisson, 2022e). In the context of this thesis the concept of understanding that is used comes from the theory of *pragmatic understanding* laid out in Thórisson, Kremelberg, et al. (2016) and Bieger and Thórisson (2017). This theory is "pragmatic" in that it focuses on the practical utility of having a certain level of understanding to complete tasks (Thórisson, Kremelberg, et al., 2016).

Understanding is not a skill understood in a general sense, but is always considered in reference to a *phenomenon*. A phenomenon is defined as $\Phi \subset W$ where W is the world is composed of a set of elements $\{\varphi_1, \varphi_2, ..., \varphi_n \in \Phi\}$ of various types, including relations $\mathfrak{R}_\Phi$ that bind elements of Φ with each other and with elements of other phenomena (Belenchia, 2021). These relations are of several types (*causal* relations are especially important, but we consider also, for example, mereological relations) and can be partitioned in two sets: the set of *inward facing* relations $\mathfrak{R}_\Phi^{in} = \mathfrak{R}_\Phi \cap (2^\Phi \times 2^\Phi)$ and the set of *outward facing* relations $\mathfrak{R}_\Phi^{out} = \mathfrak{R}_\Phi \setminus \mathfrak{R}_\Phi^{in}$. An agent understanding only $\mathfrak{R}_\Phi^{in}$ can be said to understand the phenomenon Φ but not its relation to other phenomena, while an agent understanding only $\mathfrak{R}_\Phi^{out}$ understands the phenomenon's relations to other phenomena but is unable to understand its inner workings (Belenchia, 2021; Thórisson, Kremelberg, et al., 2016).

An intelligent agent's understanding of a phenomenon is related to the models of the phenomenon that the agent possesses and creates. These are a particular type of models, in that they support certain operations with respect to the phenomenon in question. A set of models M_Φ for a phenomenon Φ consists in information structures that can be used to (1) predict Φ, (2) produce effective plans to achieve goals with respect to Φ, (3) explain Φ and (4) re-create Φ. The better these models represent elements $\varphi \in \Phi$ including their relationships $\mathfrak{R}_\Phi$, the greater is the accuracy of M_Φ with respect to Φ (Thórisson, Kremelberg, et al., 2016). Thus, considering an agent A's knowledge to be a set of models M, Thórisson, Kremelberg, et al. (2016) define understanding as:

Definition 2.6.1 (Understanding)**.** An agent's A understanding of phenomenon Φ depends on the accuracy of M with respect to Φ, M_Φ. Understanding is a (multidimensional) gradient from low to high levels, determined by the quality (correctness) of representation of two main factors in M_Φ:

U1 The completeness of the set of elements $\varphi \in \Phi$ represented by M_Φ.

U2 The accuracy of the relevant elements φ represented by M_Φ.

The understanding of a phenomenon Φ is then evaluated over the following four dimensions, ordered by the increasing level of understanding required to master each:

1. To *predict* ϕ,
2. To *achieve goals* with respect to ϕ,
3. to *explain* ϕ,
4. To *(re)create* ϕ.

Each of these capabilities can assume a value in the range $[0, 1]$ as a function on **U1** and **U2** (Thórisson, Kremelberg, et al., 2016), where 0 represents the absolute lack of ability and 1 represents achieved perfection. A good level of understanding can only be achieved by leveraging all four of these skills.

Prediction To predict a phenomenon is, equivalently, to *infer* the values of some variables given the values of other variables. Prediction does not require representation of causal relationships, but it can be done using just correlations. In this sense, Thórisson, Kremelberg, et al. (2016) refers to prediction as the "crudest form of evidence for understanding". Predictions can occur forward, backwards or parallel in time, or even all at once (Bieger and Thórisson, 2017). Prediction based on correlation involves using the relationship between two or more variables to make informed guesses about the future values of one variable based on the known values of another. Correlation measures the strength and direction of a linear relationship between variables, providing insights into how changes in one variable might be associated with changes in another. If a correlation is found between a manipulatable variable (independent variable) and another variable of interest (dependent variable), it opens up the possibility of influencing the dependent variable by intentionally manipulating the independent variable. Remember that correlation does not imply causation: establishing correlation between two variables simply means that they tend to vary together, but it does not necessarily mean that one variable is the cause of the other. As Bieger and Thórisson (2017) proposes, the predictive ability of a system can be tested by asking questions of various kinds. Specifically, the tests involve the subject under test possessing or receiving as input information I of the form $I \subset (V, t, S)$, where V is a variable and S is a state at a specific point in time t. A series of questions are then asked, whose formulation consists in presenting subject under test with a second set Q consisting of tuples of the same form as the input, in which, however, some values may be missing. For example, if the states S are omitted in the tuples of Q, the request that is made to the subject is to identify "possible and likely joint state-value assignments" (Bieger and Thórisson, 2017, p. 3) to be associated with the variables given at time t. Another type of question that can be asked by appropriately modifying the set Q is to ask whether, given other values for the same variables and a time t, those values can be assumed by the variables at time t or not. In a similar way, omitting the time-related information would be equivalent to asking at what point in time the variables obtain (if possible) these values. Other combinations are possible, even omitting a combination of variables, values, and time (Bieger and Thórisson, 2017).

Goal achievement Mere observation of correlations, sufficient for prediction, proves inadequate for achieving task goals or accurately predicting the effects of actions on variables (Thórisson, Kremelberg, et al., 2016; Belenchia, 2021). Goal achievement is made possible by knowledge about the causal relations on the variables that directly affect the goal variables (variables representing the goal state(s)) of a task. In other words, achieving goals requires knowledge of how certain variables can be controlled by the agent. The type of variables we mainly refer to is, of course, *manipulatable* variables. The variables are part of a model of interaction with the world that can be adopted by the agent to produce plans on how to achieve a goal by employing that model. In order to measure goal achievement ability, it is therefore necessary to make use of tasks, situating the learner in a relative environment. Since we are talking about goal achievement with respect to a phenomenon ϕ, the task in question will have to be related to the phenomenon in some way. Assessment of the ability to achieve goals starts with constructing task-environments from different sets of variables so that there are causal connections between the variables and the goals. This activity is supported by the concepts and methodologies from the work on Task Theory. The task-environment specification obtained in this way is defined in terms of variables, some of which can be observed and others controlled by the agent. This description allows us to abstract from details related to specific implementations (Thórisson, Kremelberg, et al., 2016). The ability to achieve goals will ultimately be related to the ability to produce *effective* plans to achieve a goal related to the phenomenon ϕ, where an effective plan is one that can be proven, through implementation, useful, efficient, effective, and correct (Thórisson, Kremelberg, et al., 2016).

Explanation Moving on in our "ladder" of understanding, explanation is an even more incisive element in determining an agent's understanding of a phenomenon. Indeed, it is possible for the predictive model of a phenomenon, while not containing causal representation information, to capture certain aspects of it accurately enough to enable the achievement of goals. The further addition of explanation to the dimensions of understanding allows for a more precise assessment of an agent's ability to capture causal relationships (Thórisson, Kremelberg, et al., 2016). The explanation of a phenomenon requires, even more than goal achievement, the ability to understand the causal relationships that enable a phenomenon, in that the precise explanation of a phenomenon requires the identification of a set of *necessary* and *sufficient* elements that characterize it (Bieger and Thórisson, 2017). In particular, this process of explaining a phenomenon can occur at different levels. As we introduced in the section on Task Theory, the level of detail considered changes the nature of a task. Assessing a controller's explanations of a phenomenon on multiple levels of detail is one way to identify the controller's actual understanding of that phenomenon in relation to the maximum extent of the explanations provided(Bieger and Thórisson, 2017).

(Re)creation The ability to create or recreate a phenomenon is the last of the four dimensions on which meaning is based and probably the most important (Thórisson, Kremelberg, et al., 2016). The ability to create a phenomenon is understood here as the ability to *produce models* containing the necessary and sufficient features of the phenomenon, which have already been considered when evaluating the ability to provide explanations. We talk in this case of re-creating phenomena from a theoretical, rather than a practical point of view: an example of this is the models developed to date to explain the laws of the universe (Bieger and Thórisson, 2017).

CHAPTER 3

On Defining Meaning

To properly analyze and propose a theory of meaning generation, some fundamental concepts must be dissected. Starting from the analysis of the concept of meaning in common usage, the following chapter will:

- Argue for the fundamentally pragmatic nature of meaning;
- Discuss some typical features meaning, which will be later formalized, and
- Explain how causality, causal chains, models, and reasoning are linked to meaning.

This somewhat "empirical" inquiry on the nature of meaning, supported by the rigorous theoretical framework introduced in the previous chapters, will progressively clarify the conceptual basis on which a formalization of meaning generation shall be grounded.

3.1 The pragmatic nature of meaning

The importance of considering the concepts of causal relations and task theory set forth earlier lies in the essentially pragmatic, causal, and dynamic nature of meaning. Fundamentally, the assumptions to keep in mind in our research are twofold, namely that 1) we are addressing intelligent, autonomous systems situated in a task-environment, and 2) we intend to equip such systems with mechanisms that support their work of performing tasks. The first assumption is related to the scope of our research. With the idea of making a contribution to research in the field of Artificial Intelligence, we refer to a specific type of intelligent systems at have already been the focus of previous studies. Thus, exploring the concept of meaning in the context of previous studies on these intelligent systems allows us to lean on an established theoretical framework and build on its results using a proven methodology. The second assumption is a consequence of the first. The intelligent systems under consideration, which are also the focus of this research, are intended to perform tasks. The same constructivist conception of 'intelligence' understands it as a tool for dealing with practical needs. The ability to understand and generate meaning is a typical feature of intelligence, and, therefore, it should retain the same pragmatic nature. To have systems capable of receiving and understanding instructions to perform tasks requires the ability to translate instructions into goal states and to reason about achieving those states. Systems that cannot process instructions this way are hardly identifiable as “intelligent”. Therefore, analyzing meaning from a pragmatic point of view is essential for developing autonomous, situated, intelligent agents that can operate effectively in complex, dynamic environments.

3.2 Approaching meaning

Having previously introduced the fundamentals of Task Theory, learning by reasoning, and understanding, it is now possible to discuss the concept of *meaning*. In order to give a more rigorous definition of meaning, we must first get an idea of what meaning actually is. Because it is such an ingrained concept in common usage and used almost unconsciously, the reader is unlikely ever to have asked “what does meaning actually mean?", i.e. “what is the meaning of meaning?". This is a question that might appear to be of exclusively philosophical interest, equal to questions such as “what is the meaning of life" and the like. However, the search for meaning is a central activity in performing tasks and achieving goals. For example, the reader will be familiar with crossword puzzles, word games where solvers enter words or phrases into a grid according to a set of clues. In solving the crossword puzzle, it is not uncommon to encounter clues such as “The center of Rome – 2 letters". At first glance it might seem that a word such as Colosseum” would fit the part of the clue that concerns the center of Rome, but the 2-letter limit contradicts this intuition. In fact, the clue is written in a deliberately ambiguous manner to confuse the solver and refers to the center of the *word* ‘Rome’, corresponding to the letters “om”. Grasping the meaning of the clue allows the player to solve both this particular crossword puzzle and future ones that may contain similar clues.

Let us consider a different example. Suppose you are reading the latest news and learn about a forest that recently caught fire: knowing that the forest has gone up in flames due to weather-related causes is different from knowing that a forest is on fire due to arson. The causes of the event have different implications: in the former case, the fire could have been caused by lightning strike, volcanic eruption, spontaneous combustion, or the action of other weather agents, and would require the intervention of firefighters and civil defense to limit the damage; in the latter case, the cause of the fire is attributable to the action of one or more people, committed for a wide variety of reasons (from political dissent to profit-seeking from the destruction of forest areas, etc.[1]), and, in addition to the intervention already mentioned, would cause an investigation to be opened to identify and arrest the culprits. Even different situation would be to be in the forest at the time of the fire outbreak and having to leave in a hurry so as not to get hurt. In other words, the *meaning* of the forest fire appears different depending on the person’s experience of it, because the event has different repercussions on each person’s daily life. From these examples we can already derive some aspects that characterize meaning.

3.3 Dissecting the Concept of ‘Meaning’

Based on how the concept of ‘meaning’ is used in everyday language, a number of its features can be discerned and isolated, in preparation for proposing a theory of ‘meaning generation’ that is detailed enough to be implementable in an artificial intelligence system.

Based on the general use of the concept, and in light of the related work (see Chapter 2), the following features are considered necessary (but not necessarily sufficient) to capture the

[1] For an overview of the various causes (natural and otherwise) of forest fires, see Civile (2008) from the Italian Department for Civil Defense

phenomenon of 'meaning'.

First of all, we recall that the *meaning* associated to a concept or an event is typically a description or a relationship that connects the concept or event to another subject. Meaning is not an intrinsic property of a given "datum" (where a datum could be an event, a perception from a sensor, or even the result of an internal reasoning process), but emerges from the perspective of a subject who generates or associates meaning with that datum. Let us consider the example event of rocks rolling down a hill. The event does not carry meaning on its own, but it has *some* meaning to those who witness it, and it varies depending, for example, on where that observer is in: a person in the valley who sees the rocks rolling towards him will have a different reaction from someone at the top of the hill. The subjectivity of the meaning associated with the rolling rocks depends both on contextual aspects, such as location in space, but also on the perception of the dangerousness of the event based on both experience and the ability to predict how the system will evolve: anyone who has had experience with rolling objects and heavy objects will easily understand that a heavy rolling object could cause serious damage. Take as another example a golf game in which two players are competing for a major title and the score is 6 to 14. A spectator watching the game but unfamiliar with the sport might think that, as is the case in many other sports, the player with the advantage is the one with the higher numerical score. Therefore, when the player with the lowest score is proclaimed the winner at the end of the game, our spectator will realize they have associated a different meaning with the same numbers. Numerical values read on a game board take on meaning based on associations to other concepts, such as similarity to the score of a soccer game. In this sense, the focus of a theory of meaning shifts from the *datum* to the *processing subject* who possesses a representation of the datum that it uses to derive derive connections with other known facts.

1. ▷ *The meaning of a datum is dependent on subjective representations.*

An excellent example that helps us highlight another characteristic of meaning is the one reported by Thórisson, Kremelberg, et al. (2016):

> "If I hear an announcement that the gate to the flight to my vacation destination has closed, this will mean something very different depending on which side of the gate I am on at that point in time; in one case I may start crying and the other not. And if I have a drink in either contingency it will likely be for very different reasons."

The meaning associated with the event of gate closure has *temporal* relevance, as the same event, shifted in time, results in a different effect. If, for example, the gate closure was moved earlier by two hours, it would cause additional organizational inconvenience to passengers. But the same meaning in this example also has a *spatial* connotation, in that the position in space of the passenger (in this case, inside or outside the gate) determines his or her discouragement. The same can also be said about the previous example of the rolling rocks: being in the valley before or after the rocks fall does not constitute a risk, just as one is equivalently safe by being completely away from the valley in question at the exact moment the event occurs.

2. ▷ *Meaning is dependent on the spatio-temporal context.*

Meaning is related to a more general set of situational elements other than space and time. Irony and sarcasm are complex forms of communication that involve conveying meanings in a way that is not literal or contrary to what is actually said. Both require additional effort to be understood, as they involve a deeper level of understanding and an awareness of the duality between what is being expressed and what is actually meant. Communication in general, not just verbal communication, is dependent on the broader context in which one is embedded. To a strange hand gesture that I have never seen before I can associate different meanings if I am meeting with a friend of mine or if I am, for example, stuck in traffic and have just performed a risky maneuver. In all these cases, meaning is not simply identifiable in a well-defined and circumscribed set of stand-alone data, but must be looked for and *grounded* in a broader context. As the context changes, so does the meaning associated with data and events. Hearing 'great job' after breaking a cup or after solving a complicated problem leads to the reconstruction of different meanings. The situational context in which data and events occur is therefore inseparable from meaning.

3. ▷ *Meaning is related to the situation*[2].

The previous claim about the situation can be further expanded with the concept of goals. Agents located in a task-environment are involved in performing tasks whose outcome (success/failure) is defined by the goals associated with those tasks. If positive goals are achieved, then the agent has succeeded; if, on the other hand, negative goal states are reached, or if no goals are achieved in the time period relevant to the performance of the task, then the agent has failed. In any case, given a space-time context an agent will have assigned tasks and, consequently, goals to achieve. Depending on the goals, the behavior of agents in environments changes. Taking the example from earlier, if my goal is not to catch a flight because I just got off the plane and am on my way home, the announcement of the gate closing will have no impact on my mood. It appears, then, that another aspect on which meaning depends is *goals*. The gate closure event has some relevance[3] only when it affects one of the goals of the subject toward which the meaning is directed. Of course, it is possible for one event to affect more than one goal. Suppose then that we were able to catch our flight to go on a relaxing vacation that we had been planning for some time. Just before we left, we also received a promotion at work, getting a position of greater responsibility. As soon as we arrive at our destination, we learn that our boss will also be in the same destination as us to attend a business conference. The news affects both the vacation and our work experience, as the trip is not only an opportunity for personal relaxation, but can also become an opportunity to establish professional relationships with the boss, discuss the promotion, and show commitment. In this way, the news influenced the meaning attached to both the trip and the promotion, connecting them in an unexpected way.

In this sense, a datum has more than one meaning if it influences multiple goals, or influences the same goal in different ways.

4. ▷ *Meaning is related to goals.*

At the beginning of this section, we restated a generic definition of meaning, as it is understood in common sense, which identifies it as a description or implication. What emerges from our discussion so far is that meaning is mainly related to the connections between datum, situation and goal via the subjective representation that the subject from which meaning

[2] Situation and context are used here generically, but with different connotations. The notion of context will be formally defined later in relation to the environment, while the concept of 'situation' is used here in the common sense of the term to refer to a *broader* context consisting not only of spatio-temporal references, but also, for example, cultural aspects and the like, and thus it will not be formalized

[3] We use "relevance" in this context in the common sense of the term; later we will give a more formal definition of "relevance"

is generated has of all these aspects. In fact, leaving all these aspects unchanged, it is the *relationships* that connect them that make up the meaning. Two different aspects influence these relationships: (a) knowledge and (b) the process of generating relationships.

In the case of agents endowed with cumulative learning (a notion introduced in Section 2.1.3), i.e., the type of agents we refer to in this thesis, we know that every piece of information the agent receives through its sensors is immediately passed to the knowledge synthesis process. Therefore, all the information that the agent receives from any source ultimately flows into its knowledge. Lack of knowledge concerning a phenomenon obviously prevents us from being able to relate back to that phenomenon, or, more generally, to the aspects of that phenomenon that have no representation in our knowledge. This is why, for example, not knowing certain aspects that characterize jazz music or contemporary art makes it impossible to grasp their meaning (here understood as the reason for the existence of such things and the practical utility they should have)[4]. However, the fact that some knowledge is not in the mind of the subject generating meaning does not mean that certain connections cannot exist in an absolute sense; the subject's view of the world will always necessarily be incomplete of some aspects.

In the second case, on the other hand, even assuming that we possess all the necessary knowledge, there is no guarantee that the meaning of something is understood by the generating agent. This leads us to have to explicitly introduce a second claim, namely that meaning is the result of a *process* that generates it. So far we have only hinted at the idea of meaning as the result of a meaning-generating process, and, in presenting the characteristics of meaning, we have mentioned again and again that it is recreated by the subject in some way. Instead, we now claim that in order for meaning to be explicitly accessible to an agent, it must be generated as the result of a process. The fact that connections between two elements (e.g., a datum and a goal) exist and are identifiable from the models present in the agent's knowledge does not imply that this agent will be able to identify these connections, since the meaning generation process implemented by the agent may not isolate these connections among the set of all possible connections present in the agent's knowledge. Recall that the agent is subject to AIKR, so the resources at its disposal (computational, but not only) are to be considered insufficient[5]: a running process with limited resources available will produce limited results. The result of the meaning generation process is some explicit representation of meaning, that is, an additional element of knowledge that enables the agent with the ability to achieve its goals given given representations of a datum and context along with its own knowledge.

5. ▷ *Meaning is lies in the relationships between the elements of knowledge.*

6. ▷ *Meaning is the result of a meaning generation process.*

Two aspects that the meaning generation process should cover are the production of predictions and the reconstruction of the causes of a given phenomenon. Prediction-making occurs when the influence of a datum on goals is reflected not in the immediate, but at a later moment in time: in this case, the connection between datum and goals is not directly identifiable in the agent's knowledge (e.g., as a connection between two different models), but must be produced from those models that, by being executed multiple times over time, allow the goals to be achieved.

[4] In this case the understanding of the meaning of art in its various forms is simply used as an example to illustrate the reconstruction of meaning and has nothing to do with a person's judgment of these things, nor is it meant to be a comment against of those who criticize jazz music or contemporary art for the most diverse reasons

[5] This is because, otherwise, an agent might assume that it has an infinite amount of resources at its disposal and come up with impractical solutions to problems

Cause reconstruction of a datum, on the other hand, is useful when the datum is linked to the goals not through its direct manipulation, but by a common cause that influences both the datum and the goals 2.2.4. These two types of connection-seeking movements can thus imply deductive (forward) and abductive (backward) reasoning starting from both datum and goals.

7. ▷ *Meaning involves prediction and identification of causes.*

The nature of meaning is such that it does not remain unchanged, but varies over time due to various factors, including personal experiences (this is especially the case with situated systems operating in the physical world), learning from teaching, reflection, and changes in circumstances. Positive or negative events can influence our perception of words and concepts, altering the meaning associated with certain situations. Over time, societies go through cultural and social changes that affect the meaning of social conventions and traditions. Reflecting on past events or reasoning about events that might have happened can lead to a reworking of the meaning attributed to those events. Finally, newly acquired information can influence our perspective of past events. These are but a few examples of situations where the meaning of something, whether be it an action or a concept, changes. Because it is related to both the subjective representation (a set of models) of a datum, context, the goals of the system, and the set of relationships that connect them, meaning changes whenever at least one of these three elements changes completely or in part. Therefore, meaning is *defeasible* and *revisable* knowledge.

8. ▷ *Meaning is a type of actionable information that is subject to change.*

From past work on Task Theory, we know that the *level of detail* is part of a task (Belenchia, 2021). In fact, any phenomenon or task of the world can be described at different levels of detail, from highly detailed descriptions (down to the atomic or even subatomic level) to very abstract ones (Belenchia, 2021). Speaking of tasks, Belenchia (2021) argues that the same task, proposed at a different level of detail, is not the same task, and cites the example of a task related to an electronic circuit. An electronic circuit can be described at the level of its electronic components, at the even lower level of the chemical reactions in its circuits or at the higher level of the implemented logic circuit. The task of obtaining some output in such circuit is very different according to the level of description being used, because effectively the variables and the mechanisms changed together with the level of detail. Therefore variations in the level of detail effectively result in *different* tasks (Belenchia, 2021). Having thus tied the meaning of an event to its representation and the goals of the meaning-processing agent, it is obvious that at a different representation of the event, or at changing goals pursued by the agent, the calculated meaning will be different. In this sense, there is a limit imposed by the agent's sensors (the *granularity* of the sensors) that defines the level of detail perceived and, consequently, the perceived level of detail of both a given task and the meaning computed by the agent. Thus, there is always a limit to the level of detail an agent can bring back in the meaning it produces, and this limit is imposed by the agent's own embodiment.

9. ▷ *Meaning is related to the level of detail.*

What emerges from the above discussion is intended to capture the general intuition of the concept of meaning which enables us to go on into more detail and define more precisely the meaning generation process.

3.4 Models for knowledge representation

A model is a **representation** of something – the thing being modeled. More specifically, in computational systems, a model is structured information that, behaving similarly to the modeled subject – although of a “simpler” form – can be manipulated in order to obtain useful information (e.g., answers to a question) (Belenchia, 2021). As mentioned earlier, the limited nature of energy, time and resources available to any system (intelligent or not) prevents such systems from recording, with absolute accuracy and without loss of information, their every experience, so that they can later draw on a huge database the knowledge required to perform their tasks. A model will therefore be as good as it is useful to the agent when manipulating the model – especially when performing tasks (Thórisson, 2020c). For example, reducing the concept of *car* to “a tool for moving things or people” does not contribute significantly to solving the problem of going, for example, to your office for work or to a supermarket to pick up groceries. In fact, it could be said that, according to such a model, a car is fully equivalent to a bicycle or a bus, as all of these are means of ‘moving around’. Here, by introducing the concept of *fatigue* as related to the use of important life resources that need to be optimized, it could be inferred that moving by car requires less effort than moving by, say, bicycle, exposing additional decision factors. Introducing the further concept of *spending* and related connections generically linking money (here understood as another resource to be optimized) to the survival of the individual, one could further infer that moving by car is much more ‘wasteful’ than moving by bus, further guiding the choice of transportation – a necessary choice to achieve some other purpose – toward one path rather than others.

But the accuracy of a model is not the only aspect one should focus on when evaluating its goodness: in fact, by itself, a model is useless without an appropriate **process** that employs it. Therefore, aspects of the design of such a process also become relevant (Thórisson, 2020c). Models encode *actionable* information, in the sense that they can be used to ‘get stuff done’, e.g., predicting future states, derive the causes of observed events, explain phenomena, etc. The process that enables all of these things can be abstracted as a sequence of steps somewhat similar to the control loop we saw earlier: retrieve relevant models, apply them to situations to derive predictions, perform some action based on the model’s predictions, and monitor the results (Belenchia, 2021). Performance monitoring is a crucial element of this mechanism: models that consistently enable goal achievement will then be deemed important, while models that perform poorly (i.e., more unlikely to lead to goal achievement) will be discarded, so that over time, only the most useful models are kept in memory. Considering such a mechanism, it can be said that these models can constitute a non-axiomatic and defeasible knowledge base (Thórisson, 2020c).

So far we have clarified the role of models in the representation of knowledge and as a starting point for the execution of actions to achieve goals. Let us now introduce some relevant qualities of models suitable for meaning.

Bi-directionality With reference to the pragmatic nature of meaning, we intend to refer in this thesis to a particular type of models – *bi-directional* models. By “bi-directional” we mean that these models, as constituted, can be ‘read’ and interpreted both for predictive purposes and in a restorative manner. More precisely, we refer to the *forward-chaining execution* of such models when they produce predictions and *backward-chaining execution* when, starting from goals, a path of states is traced back to an executable action (Thórisson and Talbot, 2018a; Thórisson and Talbot, 2018b). This ability to interpret (and produce actions accordingly) models in two ‘directions’ is fundamentally related to our pragmatic and goal-oriented intuition of meaning. Meaning, in a practical sense, is related to the causes and consequences of certain phenomena, so it is important to allow exploration in this sense of causes and consequences in order to determine the meaning of that phenomenon.

Causality Another quality of models that makes them particularly suitable for managing meaning is their ability to represent causal relationships. In this sense, causal models can be seen as structured, actionable information that encodes procedural processes (Belenchia, 2021). The causality of such models allows us to deal with the causes and consequences that link the variables describing a task-environment and the actions that can be performed, enabling action-oriented behavior[6]. We identify two main parts of a causal model: a left-hand side (LHS), representing the cause, and a right-hand side (RHS), representing the effect. The LHS is the 'input' of the model, and represents a *pre-conditional pattern* composed of variables, values and so on. The RHS then represents the *post-conditions* of the LHS pattern. When applying forward-chaining, whenever the LHS pattern is observed, a prediction based on the RHS is generated by a process of *deduction.* The converse happens in backward-chaining: when a RHS pattern representing a goal is observed, a sub-goal based on the LHS is generated, meaning that, in order to reach the goal specified in the RHS, the LHS must be reached. Sub-goals can be further backward-chained until a command for some actuator is produced, and in this way models can be used to produce effective plans to achieve goals. If the RHS is not a goal, backward-chaining can be used to derive *potential causes* for the observed state. Further information can be incorporated into the model; for example, we could include the *accuracy* of the model, that is, the number of successful applications of the model (how many times the model was correct out of all the times it was applied). In order to make it possible to navigate the models in these two directions, there must also be a set of (learnable) functions that indicate how the transformation from LHS to RHS and vice versa actually occurs. Ultimately, these functions will need to take into account the mechanisms that come into play in the physical world (Thórisson and Talbot, 2018a; Thórisson and Talbot, 2018b).

Pseudo-axiomatic and resources The models should constitute a *defeasible* and *revisable* knowledge base – in a word, pseudo-axiomatic. In the context of agents based on direct experience of their environment, such agents will not be able to experience every aspect of their environment at the same time, but will have to *accumulate* experience over time. In this sense, the knowledge acquired by the agent at any given moment in time t will necessarily be partial and, most likely, incomplete. An agent who assumes its knowledge to be axiomatic will have no reason to update it. Similarly, if every aspect of knowledge is questioned, as being considered "flawed", no action will ever be performed. For this reason, it is necessary for the models that constitute knowledge to be accompanied by information regarding their usefulness in guiding the agent's actions: in this way, even if incomplete, the most useful models (which are assumed to correctly capture at least some aspects of the environment) will be retained until they are updated or discarded when incorrect. Similar discussion can be made for the inclusion of information on time, energy, and resource consumption in general. Models that abstract from resource and time consumption do not give certainty about, for example, the time required for their execution, and could lead the agent using them to generate impractical solutions to the problems they must solve (such as, for example, using an infinite amount of time or memory).

[6] Recall the crucial difference between causation and correlation set forth in the previous chapter, which requires causation as a fundamental element to perform task

Compositionality Finally, we consider *compositionality* as one of the most important qualities that models for that support meaning must possess. Compositionality of models is a very important feature, as it allows them to represent and handle compound phenomena. Causal models, which are composed of an LHS and an RHS, can be put together in hierarchical fashion to specify pre or post conditions for the execution of other models. For example, a model M_j might be featured in the LHS of another model M_i, specifying in that case that upon successful execution of model M_j model M_i will be observed, predicting that occurrence. Viceversa, if M_j is featured on the RHS of M_i it specifies that, upon successful execution of model M_i model M_j will be successfully observed. Models can be built in *conjunctive form*, in which case they constitute a causal chain whose effect is the result of multiple, temporally correlated, required transformations. Models built in *disjunctive* form, on the other hand, specify a causal chain whose effect is determined by the occurrence of the most likely pre-condition (Nivel, Thórisson, B. Steunebrink, and Schmidhuber, 2015). The simplest terms featured as inputs in the LHS or RHS of a model are called *facts*, and the observed event, a likelihood value indicating the reliability of the observation and a time interval specifying the period within which the fact is believed to be true. Therefore facts are *grounded in time* and valid only within a certain time period. The execution of models is models is time-dependant and the truth of facts is valid only with respect to the specified likelihood value and within a specific time-interval (Nivel, Thórisson, B. Steunebrink, Dindo, et al., 2014).

Bi-directional compositional causal models are the fundamental unit ('bit') of knowledge in the kind of intelligent systems we are studying.

3.5 Causality & causal chains

In Section 2.2 we introduced Judea Pearl's approach to causality and the need to manage causal relationships in order to perform tasks effectively. In the previous section we also introduced the notions of bidirectionality and compositionality, pointing to them as important qualities that models should possess to support meaning handling. It is now useful for us to delve into how causal relationships create links between multiple models until they form outright *chains*. Such chains make it possible to reconstruct relationships between even seemingly disconnected models through the identification of the intermediate models that connect them. This ability to link different models together also underlies the concept of meaning, since meaning resides in implications and relationships between phenomena.

Causal models are *modular*, in the sense that a model M_a can be used as precondition for a model M_b and vice-versa. Putting multiple models together in this way forms a **chain** in which every model except the first and last is present at the same time as LHS and RHS exactly once.

Definition 3.5.1 (Causal chain). Given a set of causal models $M = \{m_1, \ldots, m_n\}$, we denote C_M the causal chain over $m_1, \ldots, m_n$ and c_i its elements, where c is a bi-directional relationship linking two causal models, respectively a left-hand term LT and a right-hand term RT, and i represents the position of c in the chain. The right-hand term of c_i is also the left-hand term of c_{i+1}. The only elements that appear once in C_M, as LT and RT, respectively, are called $head$ and $tail$.

A chain between models M_a and M_b appears of the form $M_a \rightarrow m \rightarrow M_b$, where the effect of M_a on M_b comes from the influence of the intermediate model m. This type of relationship naturally occurs quite often between phenomena. For example, we can represent the relationship between depressing a car's brake pedal and the resulting reduction in car speed as

push brake pedal → activate braking mechanism → reduce car speed

The intermediate model m separates the models M_a and M_b, acting as a 'mediator'; without m, M_a and M_b would be independent of each other. Judea Pearl captured this idea in its **d-separation** criterion (Pearl, 1988).

Definition 3.5.2 (d-Separation (Pearl, 2009, pp. 16-17)**).** A path p is said to be d-separated (or blocked) by a set of nodes Z if and only if at least one of the following is true:

1. p contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ and $m \in Z$;
2. p contains a collider $i \rightarrow m \leftarrow j$ such that $m \notin Z$ and neither does any of his descendants.

A set of nodes Z d-separates X from Y if and only if Z blocks all paths from any node in X to any node in Y.

The d-separation criterion applied to any two disjoint sets of variables X and Y allows the identification of another disjoint set of variables Z which makes X and Y independent of each other when Z is controlled for. For the application of the d-Separation criterion the variables need to be represented as nodes of a directed acyclic graph (DAG) whose arrows, correctly represent their causal relationships(Pearl, 2009, p. 16).

As we have seen, chaining is one of three situations where the d-separation criterion can be applied. A **fork** is a situation in which two models share a common cause $M_a \leftarrow m \rightarrow M_b$ where m is also called a 'confounder' of M_a and M_b, because M_a and M_b would appear correlated even though there's no causal relationship between the two. An example of a fork is observed when, during a snowfall, people turn on the heating. It would look like snow causes the heating to be turned on, or the converse, but, in fact, it is the low temperatures that cause both events

snow ← low temperatures → heating ON

Finally, a **collider** is a relationship of the form $M_a \rightarrow m \leftarrow M_b$ where m is influenced by two different causes M_a and M_b. When we observe m alone, we cannot be sure that it was M_a rather than M_b that caused it. An example of a collider is a videogame where the GAME OVER screen appears whenever the player touches an enemy or when the player's time to complete the level runs out

touching enemy → GAME OVER ← time out

The concept of d-separation comes in handy when, in the meaning-generation phase, hypotheses are to be generated about the possible causes or consequences of a model. Considering situations in which a confounder or collider might be present particularly supports the *backward* process of hypothesis-making by allowing the process to correctly recreate causal chains culminating in the goals.

3.6 Learning by reasoning

Most of machine learning study nowadays is focused on algorithm-based learning. A machine learning system is usually described as a "learning algorithm" taking raw data and background knowledge as input and producing some output (Wang, 2009). By definition, a (deterministic) "algorithm" is a well-defined, step-by-step procedure, consisting in an unambiguous set of instructions that can be executed by a computer or followed by a human to achieve a desired outcome. For the same input, the algorithm always produces the same output using a constant amount of computational resources, namely time and space. Therefore, by repeatedly providing the same input, a typical machine learning system will use the same amount of resources to produce identical output every time. However, not all machine learning systems fit the above description. That is the case for the Non-Axiomatic Reasoning System (NARS) and the Autocatalytic Endogenous Reflective Architecture (AERA), two AGI-aspiring systems that rely on reasoning and experience, as well as user-provided knowledge, to answer questions and carry out tasks. This basically means that these systems continuously collect new knowledge and provide answers based on the available knowledge and resources. AERA and NARS systems based on reasoning have already achieved promising results (see Section). Abstracting from the architectural design principles and implementation aspects behind these systems, which are beyond the scope of this paper, we introduce in this subsection the topic of learning by reasoning.

Reasoning is the establishment of pseudo-axioms for the world and the process of applying logic to information according to these rules[7]. Reasoning can be effectively used as a method for learning when applied to acquired information. Deduction, induction and abduction can be used to simulate, generalize and infer new information from acquired information, respectively. Reasoning is most effectively used in combination with experience-based learning: an intelligent agent making use of some reasoning is capable of working in situations where nothing is certain (only some things are more probable than others) with uncertain assumptions (Thórisson). In this kind of situation, agents equipped with reasoning can produce hypothesis to explain the workings of the surrounding environment and, through direct experience, select and retain only those that are most useful (this process is also called *ampliative reasoning*). Logic is also an most effective way to compress information. Because of the high ratio of possible states in the physical world to the storage capacity of any type of mind/memory, it is not conceivable that understanding (i.e. useful, reliable knowledge) of a large amount of phenomena in the physical world can be achieved without the use of reasoning (Thórisson).

We report below the main types of implemented inference used in NARS and AERA (Thórisson, 2022e).

Deduction Results of two statements that logically are necessarily true. For example:

$$\text{Premise 1:} \quad \text{all beans from bag } A \text{ are white beans.} \tag{3.1}$$
$$\text{Premise 2:} \quad B \text{ are beans from bag } A. \tag{3.2}$$
$$\text{Result:} \quad B \text{ are white beans.} \tag{3.3}$$

Induction Generalization from observation. Induced knowledge can always be refuted by new evidence. For example:

$$\text{Premise 1:} \quad B \text{ are beans from bag } A. \tag{3.4}$$
$$\text{Premise 2:} \quad B \text{ are white beans.} \tag{3.5}$$
$$\text{Result:} \quad \text{all beans from bag } A \text{ are white beans.} \tag{3.6}$$

[7] We resort to the use of *pseudo*-axioms because our world is non-axiomatic, or at least that is what we are forced to assume until we discover the ultimate laws that describe the workings of the universe

Abduction Reasoning from conclusions to (likely) causes. For example:

$$\text{Premise 1:}\quad B \text{ are white beans.} \tag{3.7}$$

$$\text{Premise 2:}\quad \text{all beans from bag } A \text{ are white beans.} \tag{3.8}$$

$$\text{Result:}\quad B \text{ are beans from bag } A. \tag{3.9}$$

Analogy The ability to find similarities between even very different phenomena.

With reference to the models and causal diagrams previously introduced and duly extended (Section 2.4) and characterized (Section 3.4), we can define reasoning over these models and diagrams. As defined in Section 2.2.3, a causal model is a representation of a phenomenon by means of a set of variables and their relationships with variables external to the phenomenon. The relationships between variables are understood as cause-and-effect relationships, where the value of one or more variables, the *premises*, affect in a more or less direct way that of another, the *conclusion*. This view of cause-and-effect relationships between variables thus lends itself to analysis by logical reasoning. In particular, causal diagrams associated with models can be read deductively (forward chaining) and abductively (backward chaining). Moreover, since models are hierarchically composable (see Section 3.4), a model can be included in the causal diagram of a "higher-level" model. In this sense, we can apply the reasoning not only to variables but also to the models themselves. A model M_a could be linked to a model M_b through a series of causal equations and, thus, M_a is a cause of change for M_b.

Other types of reasoning can be applied to causal diagrams. For example, induction and analogy can be used effectively in hypothesis generation processes. For example, the application of analogy on a set of models $M = m_0, m_1, \cdots, m_n$ could identify a subset S of models similar to each other. If another model n_j shared the same relationship with multiple models of S, inductive reasoning could generate the hypothesis that the same relationships are present between n_j and the other models $s_i \in S$.

With particular reference to the extension of diagrams presented in Section 2.4, reasoning can also effectively support task achievement. Indeed, once the goal states of a given task have been identified, through abductive reasoning it is possible to trace the goal variables back to the variables that can be manipulated by the agent at a given time, and thus figure out how to control the goal variables and make them assume the values required to successfully complete the task.

3.7 Relevance

The practical essence of our theory requires us to take into account the differences present in the models that constitute the knowledge base, since some models will be more useful than others to an agent in carrying out tasks. In fact, models that describe the performance of a country's economy are unlikely to come in handy in the task of baking a chocolate cake. For this reason, the reconstruction of implications should not treat pieces of knowledge, i.e., models, indiscriminately, but prioritize those most likely to lead to the desired outcomes. Recall that meaning is a tool to support the achievement of goals, so any 'utility' metric will necessarily have to reference goals of some kind. We call this feature of models as **relevance**. Relevance is the metric used to filter knowledge in order to guide the meaning generation process toward achieving goal states by narrowing the scope of the search.

As we have already mentioned, relevance possesses certain characteristics such as **situation** dependence and the need to connect with **goals**. The goals we are interested in for the purpose of calculating relevance are, once again, explicit goals. Implicit goals, having no explicit representation within the agent's knowledge, cannot be the subject of the meaning generation process. We also distinguished active and inactive goals, say that the former are currently pursued by the system and the latter are not. But when does a goal becomes active or inactive? This strictly depends on the system's implementation of the whole goal mechanism. In a sense, goals that are to be pursued are the most relevant to the system's top *drives* – the set of directives by the system's designer defining the purpose of the system and that change only when the system is re-purposed. Another feature of relevance is **temporal grounding**. A set of models that are relevant at a given time are not guaranteed to maintain the same relevance over time, especially over long periods of time. Models that often prove useful are retained, otherwise they are deleted to free up memory space.

3.7.1 Model relevance

A proper definition of relevance comes from Nivel, Thórisson, B. Steunebrink, and Schmidhuber (2015). In this work the authors present a value-driven computational model of *anytime bounded rationality* robust to variations of both resources and knowledge that leverages continually learned knowledge to anticipate, revise and maintain concurrent courses of action spanning over arbitrary time scales for execution anytime necessary. This model of anytime bounded rationality adopts an architecture for the execution of programs assigned to jobs. A system's experience constitutes *defeasible* knowledge, and is represented using non-axiomatic temporal term logic, where the truth value of knowledge is neither eternal nor absolute. A term exposes three components: (a) arbitrary data, (b) a *time interval* of the form $[\text{early deadline, late deadline}]$ expressed in microseconds, world time, and (c) a *likelihood* in $[0,1]$, the degree of data ascertainment.

In order to address relevance, Nivel, Thórisson, B. Steunebrink, and Schmidhuber (2015) focus on the behaviour of the system *tending* to an input x, which can be a sensory/reflective input, an interference or a drive. The value of tending to x at time t is made dependent on both its *urgency* (for situational awareness) and *likelihood*:

$$Urgency(x,t) = 1 - \frac{THZ(x,t)}{\text{Max}_i(THZ(x_i,t)) + U}$$

$$TrendingValue(x,t) = Urgency(x,t) \times \mathit{Likelihood}(x,t)$$

where $THZ = Max(LD(x) - t, 0)$ stands for "time horizon", LD "late deadline", x_i being all the inputs in the system and U a system parameter meant to keep urgencies positive. Because a goal can be achieved by means other than expending energy to derive subgoals, the authors define the value of pursuing a goal as decreasing as the probability of its achievement (i.e., the most likely prediction of its goal state) increases:

$$P(x,t) = \text{Max}_i(\mathit{Likelihood}(p_i,t))$$

$$Effort(x,t) = \begin{cases} \mathit{Likelihood}(x,t) & \mathit{Likelihood}(x,t) \geq P(x,t) \\ 1 - P(x,t) & \text{otherwise} \end{cases}$$

$$TendingValue(x,t) = Urgency(x,t) \times Effort(x,t)$$

where p_i are the predictions of x's target state. The global relevance of a model m is then defined as the normalized maximum of the tending values of all its inferences $x_i(T,m)$ of type T (*predictions* or *goals*) that are still 'alive' (meaning that they have not been removed from memory) at time t:

$$UR(m,T,t) = \text{Max}_i(TendingValue(x_i(T,m),t))$$
$$Relevance(m,T,t) = \frac{UR(m,T,t)}{\text{Max}_i(UR(m_i,T,t))}$$

where m_i are the models in the system. If none of the models of the $x_i(T,m)$ are alive, then m's relevance is computed as

$$Relevance(m,T,t) = \frac{\text{Min}_i(UR(m_i,T,t))}{\text{Max}_i(UR(m_i,T,t))}$$

3.7.2 Model reliability

Models are *variable defeasible knowledge*, therefore their construction, deletion, and revision are triggered by experimental evidences of their predictive performance. An inference results from the processing of evidences by chains of models and is defeated or confirmed upon further (counter-)evidences. Its likelihood is continually revised depending on the context and reliability of said models and, notably, decreases with the length of the chains (Nivel, Thórisson, B. Steunebrink, and Schmidhuber, 2015):

$$Reliability(m,t) = \frac{e^{+}(m,t)}{e(m,t)+1}$$

where $e^{+}(m,t)$ is the number of successful predictions produced until any time t by a model m, and $e(m,t)$ is the total number of predictions. The likelihood, at any time t, of an inference y produced by a model m from an input x is then defined as:

$$Likelihood(y,t) = Likelihood(x,t) \times Reliability(m,t)$$

CHAPTER 4

Autonomous Meaning Generation

This chapter proposes a theory of pragmatic meaning generation in the context of artificial intelligence systems. The main contributions towards this include:

- Proposing a characterization of goals, an essential element of the generation of meaning;
- Presenting a definition of meaning based on the formalization of previously introduced concepts;
- Review and new formulation of the concept of implication, definition of a relevance metric on implications, and discussion of reliability on implications;
- Formalization of a meaning generation process detailed enough to be implemented in an artificial intelligence system;
- Review of the formulas introduced in this chapter in light of the LTE assumption.

At the end of this chapter we will also discuss the relationship between meaning and understanding and the qualities of intelligent controllers that enable them to handle meaning in varying degrees.

4.1 A characterization of goals

We previously introduced goals as an extension to causal diagrams and, then, as a characteristic element of meaning. In this subsection, we further explore the concept of goals and propose a classification system for the different types of goals.

A goal is a desirable and possibly partial state that an agent should achieve. Goals are central to the definition of a problem: a problem is in fact defined as a goal to which are associated constraints arising from the particular task-environment in which the problem is contextualized; the task-environment assumes particular ranges of values for variables, as well as possible element groupings (Thórisson, 2022c). This classification system clarifies the forms in which goals can be identified and how they affect the meaning generation process.

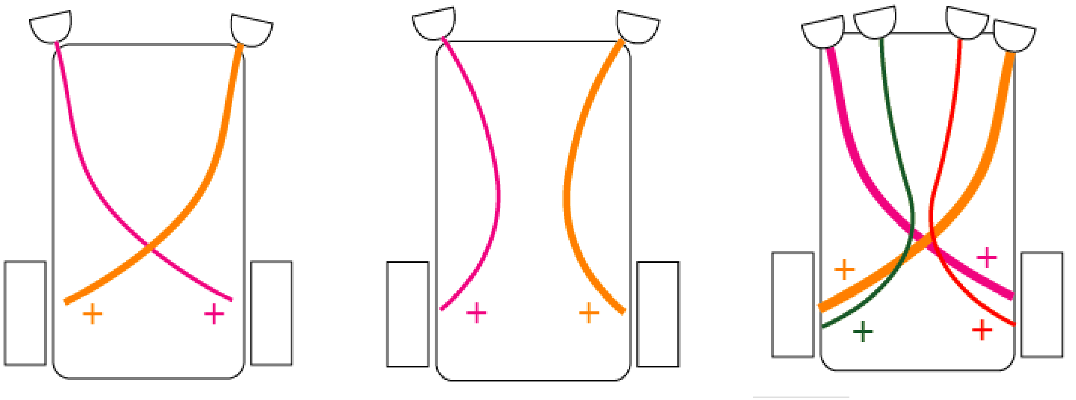
Let us first introduce the distinction between **implicit** and **explicit goals**, taking as an example Braitenberg's vehicles. A Braitenberg vehicle is an example of a reactive agent that can autonomously move around based on sensory inputs. It has primitive sensors to perceive the surrounding environment directly connected to its motors, each of which drives a wheel. In its simplest configuration a perception from one of its sensors immediately produces a corresponding wheel movement, as shown in Figure 4.1. In its basic configurations, the behavior apparently exhibited by this reactive agent is either actively chasing that which its sensors sense (Figure 4.1a), or avoiding it (Figure 4.1b). In other words, the agent could be said to have the "goal" of chasing/avoiding something. In this case, our system is actually designed to chase some goal, but it does not have a data structure that represents the goal explicitly. In order for an AI system to examine and reflect on its goals, for, e.g., generating meaning, it must have a representation of its goals in some sense "explicit", i.e., a discrete representation that can be manipulated, compressed / decompressed, and related to other data structures for various purposes (Thórisson). *Explicit* goals assume that the system's control mechanisms can operate on the system's internal knowledge, analyze patterns, and use these to generate new control mechanisms and goals. For this reason, in order to define the concept of *implication* (a fundamental aspect of meaning, as argued before) as a process that relates perceptions to goals, we must specify that these goals must have a clear representation for the AI system, that is, the system must be aware of them in some clear way. Let us therefore denote the goals that belong to this type as "explicit" goals, and the others as "implicit" goals.

Different, however, is the differentiation between *active* and *inactive* goals. An **active goal** is to be understood as one of the goals that the system is currently pursuing. Take as an example an autonomous agent powered by an internal battery. Such an agent might have the goal of maintaining the charge level of its internal battery above a set threshold (similar to what is usually suggested as to date is suggested to be done with lithium batteries to prolong their useful life). Such an agent would then have to resort, periodically, to a power source to recharge itself. If the agent were aware of appropriate "charging stations" (similar to those for today's electric cars) it could generate the intermediate goal of "finding a charging station". However, the latter goal might be activated only when certain conditions are met. Similarly, a human being who is not hungry will not search for food. Whether or not a goal is active is thus another relevant factor in computing meaning. If a goal is inactive, the events that relate back to that goal may not be significant in the immediate term (if I am not hungry, I am not going to eat that apple I just saw in my fridge, but in a few hours I might give it a thought).

Finally, let us distinguish between *positive* and *negative* goals. A **positive goal** (see Goal) is a desirable, possibly partial, state that the agent should reach. Conversely, a **negative goal** (see Failure), is an undesirable, possibly partial, state that the agent should avoid. To successfully carry out a task, positive goals must be met while, at the same time, avoiding negative goals: any process for meaning generation should produce results that satisfy this condition. In this sense, **goals are attractors in the state space of the agent's knowledge**[1], in that they catalyze the agent's relevant knowledge in such a way as to enable it to reach or avoid certain states. If the agent knows how to achieve a goal, it means that it can make a plan for getting there.

Goals differ from general laws of the universe. Goals are localized, while universal laws are not. Goals exist under the constraints of universal laws, but extend behavior within what these laws dictate. An entity like a rock has no local rules. An active goal further constrains the behavior of a controller; the form that these extended constraints take are determined by the controller's knowledge. In a world of physics, only global rules exist that apply equally

[1] K. R. Thórisson, personal communication



(a) Braitenberg vehicle example control scheme: "love". (b) Braitenberg vehicle example control scheme: "hate". (c) Braitenberg vehicle example control scheme: "curious".

Figure 4.1: An example where (a) the vehicle follows (and ultimately crashes into) what its sensors perceive, (b) the vehicle avoids what its sensors perceive, and (c) the vehicle approaches what its perceived by its sensors without crashing into it (thinner wires means weaker signals). *(From K. R. Thórisson's Advanced Topics in A.I. course at Reykjavik University; reproduced by permission.)*

to everything. A rock does not have a goal to roll down a hill, because it does not have local rules in the form of goals that determine its "nature" at a particular place and time.[2]

According to this view on goals, we might even go as far as to say that the most primitive function of meaning is the computation of whether something will have a positive or negative impact on one's active goals.

4.2 On defining meaning

In our pursuit of a more robust understanding of the concept of meaning, we intend to give a more rigorous description of the concept of meaning by building on the intuition set forth in the previous section. We contend that meaning is tied to and grounded in a physical world denoted as W. This world is described by a complex interplay of variables, where each variable's domain is constrained by the laws established by the world itself. The values of these variables change over time according to a set of dynamics functions associated with them.

Definition 4.2.1 (World (Thórisson, Bieger, Thorarensen, et al., 2016)**).** A **world** W is an interactive system consisting of a set of variables V, dynamics functions F, an initial state S_0, domains D of possible clusters of particular constraints on their values, and a set of relations between the variables R: $W = \langle V, F, S_0, D, R \rangle$. The variables $V = \{v_1, v_2, \ldots, v_{||V||}\}$ represent anything that may change or hold a particular value in the world. The dynamics functions act as the laws of nature in the world and as a whole can be seen as an automatically executed function that periodically or continually evolves the world's current state into the next: $S_{t+\delta} = F(S_t)$. It is useful to the decompose the dynamics into a set of transition functions: $F = \{f_1, f_2, \ldots, f_n\}$ where $f_i : S^- \to S^-$ and S^- is a partial state. The domains $d_v \in D$ specify which values each variable v can take, and for physical domains these are usually subsets of real numbers. The relations are Boolean functions over variables that hold true in any state the world will ever find itself in. If the world is a closed system with no out-

[2] K. R. Thórisson, personal communication

side interference, the domains and relations are implicitly fully determined by the dynamics functions and the initial state. In an open system where changes can be caused externally, instead, the explicit definition of domains and invariant relations can restrict the range of possible interactions.

A world is, by definition, a highly complex and diffuse system. Typically, any situated system experiences only certain parts of the world at any given time. Therefore, we narrow our scope of interest from the world to a subset of it, the *environment.* A subset of the variables of W is called *environment* E, and represents a limited view of the world.

Definition 4.2.2 (Environment (Thórisson, 2022c; Thórisson, Bieger, Thorarensen, et al., 2016)**).** An **environment** is a view of a world. An environment E of a world W is a pair $\langle V, F \rangle$, where $V \in V_W$ is a subset of the variables of W and $F \in F_W$ is the subset of dynamics functions of W describing how the environment's current state evolves into the next. The body of an agent is considered to be part of the environment. The constraints placed by W are valid in every environment of W.

The same environment observed at different points in time may appear slightly different based on the values assumed by its state variables. Thus, an environment with a specific instantiation of its variables is here called *context.* Equivalently, a context is an environment observed at a specific point in time.

Definition 4.2.3 (Context, Perceived Context). Given an environment E and a set of variables $v_1, \cdots, v_n$ representing the state of E, we define the **context** of the environment at a given point in timeC_{Et} as the elements making up the environment (variables and dynamics functions) plus a state S which defines the exact values of every variable associated to E at time t. An agent situated in an instantiated task-environment will typically not be able to perceive the entire context. Therefore, we refer to the **perceived context** as the set of observable variables of the context (the set of variables perceived by the agent).

This set of definitions now allows us to introduce with greater clarity and precision the concept of meaning. Meaning is not a concept that exists in itself, but is always referred to some datum[3] d perceived at time t by some entity – which we call *agent* – A. An agent A computes the meaning of a datum d perceived at time t in a context C by relating d to its own knowledge K and identifying relationships with its own explicit and possibly active goals $G = \{g_1, g_2, ..., g_n\}$. The process of linking d to G involves the use of some form of reasoning over K to generate the relationships linking d to G. Agent-environment interactions provide the agent with a way to test hypotheses and predictions and to refine its models through experience. The information collected by the agent at any point in time is possibly partial and subject to noise, necessitating the use of pseudo-axiomatic logic that produces defeasible and revisable knowledge.

Definition 4.2.4 (Meaning). Meaning is the whole set of actionable information enabling an agent situated in a context to go toward a goal attractor. Meaning is associated with a datum (a phenomenon, event, etc.). The set of associations that, at a given time and in a given context, reconnect the datum to one of the agent's explicit goals through identifiable links in the agent's knowledge places the agent in a position to successfully reach and achieve its goal. We therefore talk about the meaning of datum d perceived at time t in a context C by an agent A with knowledge K and a set of explicit goals $G = \{g_1, g_2, ..., g_n\}$. Meaning

[3] By "datum" we mean an event, perception possibly subject to noise, or any pattern recalled for some reason, something that can be represented by the agent's mind (Thórisson, Kremelberg, et al., 2016)

is computed with the help of some form of reasoning over K to generate the relationships linking d to G. The knowledge of the agent is ultimately linked to a subset of the actions the agent can perform to act on its surroundings.

The first important consideration that emerges from this definition is that meaning is a special kind of actionable information (i.e., knowledge). Not all knowledge represents, per se, meaning. Knowledge organized in such a way as to link a datum to a goal through the execution of actions that influence the surrounding environment. The kind of knowledge that enables this is *causal* knowledge. Since not all types of knowledge constitute meaning, there must be a way of *generating* meaning. It is precisely this process of meaning generation that we are going to explore in the following sections, trying to provide as coherent and detailed an explanation of this process as possible so that it can be implemented in an intelligent system in the future.

A second element to focus on is the reference to a specific type of goal – *explicit* goals. Explicit goals, as already expressed in Section 4.1, imply the system's ability to manipulate internal knowledge and reason about what are the goals it is expected to achieve. Implicit goals cannot, by definition, be linked back to anything, as they lack a representation in the controller's mind.

Finally, since the central pivot of meaning generation is the agent, the set of characteristics that agent possesses influence its ability to handle meaning. We will therefore delve deeper into the relationship between an agent's qualities and the degree of meaning in Section 4.6.

Bit of meaning The tiniest "bit" of meaning might be just a single value associated to an action that directly controls a single goal variable. The meaning of the action is computed in relation to the goal variable that does or does not acquire a 'goal' value in response to the agent's action.

4.3 Implication

We defined meaning as the entirety of information that defines the state of an implemented controller going towards a goal attractor. We also argued that meaning has to be *generated* through a process by some entity. While meaning is a set of *static* information, in that it requires a process that appropriately interprets that information and translates it into actions, meaning generation is a *dynamic* process. We present in this section the fundamental building blocks of a meaning generation process – implications.

Since meaning is a series of connections within the knowledge base available to a controller, the meaning generation process must first proceed to create these connections. Taking up the concept of a causal chain already introduced in Section 3.5 and with reference to the previous formulation provided by Thórisson, Kremelberg, et al. (2016), we then define the concept of implication.

Definition 4.3.1 (Implication). Given C_t a context at a specific moment in time, the implications I of a perceived datum d computed by an agent A are the elements of a set of reasoning chains, namely deductions, inductions, abductions, and analogies $R_t = De_t \cup In_t \cup Ab_t \cup An_t$, over the knowledge K_t of A and the PC_t perceived context by the agent. A single implication is a chain of causal relations $cr \in K_t$ linked together by deduction, induction, and analogy starting from d_t and ending with an explicit goal $g_i \in G_t$, and by abduction starting from an explicit goal g_i and ending with d_t, represented

$$\begin{aligned} R_t(d_t, PC_t, K_A, G_A) = De_t(d_t, PC_t, K_A, G_A) \cup In_t(d_t, PC_t, K_A, G_A) \\ \cup Ab_t(d_t, PC_t, K_A, G_A) \cup An_t(d_t, PC_t, K_A, G_A) \end{aligned}$$

$$I_t(d_t, C_t, A(K_t, G_t)) = R_t(d_t, PC_t, K_A, G_A)$$

It might also be possible for the set of all implications to be empty $R = \emptyset$, if no implication can be derived from a given combination of input values. The datum can therefore be considered "meaningless". An implication is a triple $\langle Head, Body, Tail \rangle$, where $Head$ is the origin of the implication, i.e., the model of the datum, $Tail$ is a goal state, and $Body$ is the ordered set of causal relationships forming a causal chain going from $Head$ to $Tail$.

According to this definition, starting from the agent's set of models describing the datum and the agent's perception of the datum and context, deduction, induction, and analogy are concerned with producing direct implications toward the agent's explicit goals, while abduction acts from those goals and tries to trace back to the datum. These two movements of agent knowledge research, *forward* and *backward*, continue until a complete causal chain is reconstructed, possibly resulting from the union of chains created in forward and backward fashion that meet halfway. In other words, given a large set of variables and dynamics functions defined on a subset of them, the causal diagrams linking these variables to the agent's goals, if any, are generated. Figure 4.2 provides an insight into the structure of the implications.

Given a model m, the process of implication generation will pose m as the LHS of a causal relationship and will search, through a pattern matching mechanism, among the other models in the agent's knowledge base for other models that are influenced (even if only presumably, through hypothesis generation and application of analogy) by m, placing them as candidates for the RHS of the causal relationship and continuing the chain of implications that has as its goal reaching a goal state. "Reaching a goal state" means that the last model in the causal chain, i.e., the "tail" of the implication, directly controls one or more variables that characterize the goal state, allowing the agent, through implementation of the actions associated with the model, to move closer toward the goal state. This process of generating implications also occurs through the production of **predictions** and **hypotheses** by application of reasoning to the knowledge to which the agent has access. For example, imagine that an agent possesses a model that predicts the upward movement of one of its mechanical arms. Such a model might indicate that given a generic (x, y, z) position of the arm in space and applying the $move\_arm\_up$ action the position of the arm in space will become (x, y, z'), where $z' = z+i$. Knowing this, the agent is able to predict, by inference, that by applying the $move\_arm\_up$ action several times, it will be able to raise the arm high enough to pick up an object from a shelf.

When an implications ends in a negative goal is called a **negative implication**. In addition to what has already been said about implications, this type of implication comes in the form of actionable information that makes it possible to avoid failure states.

Definition 4.3.2 (**Negative Implication**)**.** An implication that ends in a negative goal is called a **negative implication**. In contrast to normal implications, a negative implication puts the system that generated it in a position to *avoid* failure by knowing what actions (identified by the implication generation process or by analogy) can cause that goal state to be reached.

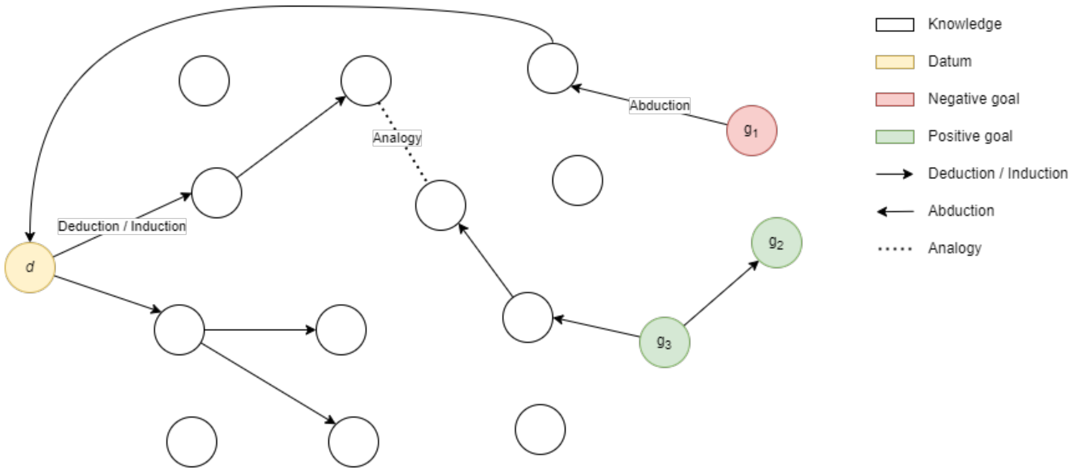


Figure 4.2: Visualization of the meaning generation process. Connections are identified between the datum and goals through the application of reasoning to the agent's knowledge

4.3.1 Partial implication

Given the previous definition of implication, we note immediately that the meaning generation process may not produce any implications at all. This could be due, for example, to particular operational constraints on the agent, such as a very small time window for meaning computation. However, in light of the previously introduced assumption about an agent's operational conditions, i.e., that it is assumed to be constrained to work with insufficient resources and knowledge (see AIKR in Section 2.4), it is necessary to make our definition of implications more flexible. If an agent performs its computation in a limited time frame and fails to produce implications due to lack of resources, instead of returning an empty set of implications and throwing away the work done, we would prefer instead that it returns the *partial result* of its computation so that it can resume its work at a later time. Given this assumption and our definition of implication, let us go on to define the notion of *partial implication*. A partial implication is a chain of causal relationships that originates in the datum and ends in a knowledge element other than a goal. In any relevant case-scenario, the production of implications is subject to limitations in resources of energy, space and time (LEST), so it is plausible that any process for generating implications may fail to return any results. The concept of "partial implication" supports an *anytime* process whose execution that can be paused (returning the results of its partial computation) and later resumed (starting from a previously generated set of partial implications).

Definition 4.3.3 (Partial implications). Given C_t a context at a specific moment in time, the partial implications PI_t of a perceived datum d_t computed by an agent A are the elements of a set of reasoning chains, namely deductions, inductions, abductions, and analogies $R_t = De_t \cup In_t \cup Ab_t \cup An_t$, over the knowledge K_t of A and the P_t perceived context by the agent. A single partial implication is defined as an implication originating either from d_t and ending with any knowledge element other than one of the agent's explicit goals $g_i \in G_t$, or from an explicit goal g_i and ending with any knowledge element other than d_t. A partial implication is a triple $\langle Head, Body, Tail \rangle$, where $Head$ is the origin of the partial implication, i.e., either the datum or a goal, $Tail$ is a piece of the agent's knowledge $k \in K$ other than a goal, and $Body$ is the ordered set of causal relationships forming a causal chain going from $Head$ to $Tail$.

Partial implications can thus be either deductive, inductive and analogy implications that start from a datum and tend toward a goal in a forward manner, or abductive implications that originate from a goal and tend toward the datum in a backward manner. A partial implication can be taken up and extended at a later time just by starting from its tail and progressing toward the target, keeping the head and body in memory to eventually recreate the full implication. This then leads us to supplement the previous definition of implication with the possibility of developing computation from a set of partial implications.

Definition 4.3.4 (**Implication (supplement)**). The function computing implications can accept as input a set previously computed partial implications PI. Coupled with a context at a specific moment in time C_t and the knowledge K possessed by the agent A, the computation resumes from the $Tails$ of each partial implication as they were the $datum$ of a process generating implications.

$$\begin{aligned} R_t(PI_\tau, PC_t, K_A, G_A) = De_t(PI_\tau, PC_t, K_A, G_A) &\cup In_t(PI_\tau, PC_t, K_A, G_A) \\ \cup Ab_t(PI_\tau, PC_t, K_A, G_A) &\cup An_t(PI_\tau, PC_t, K_A, G_A) \end{aligned}$$

$$I_t(PI_\tau, C_t, A(K_t, G_t)) = R_t(PI_\tau, PC_t, K_A, G_A)$$

where $\tau < t$

There is another fundamental aspect of meaning generation that should be included in the definition of a meaning generation process that is intended to be used in practice. So far, our definition of implication generation is comparable to the simple application of breadth-first search (BFS) to a knowledge graph. However, the number of possible causal diagrams that can be defined on a given knowledge base can grow exponentially and quickly become computationally intractable. Let us take a simplified model of a country's economy as an example. The three main variables we consider are employment rate, inflation, and interest rate. Each of these variables influences others: for example, the employment rate could increase the production of goods and services, but also lead to higher wage demands; likewise, higher inflation could increase the cost of living and reduce people's real income. Each of these variables influences, in turn, a set of other variables, greatly expanding the number of aspects to be taken into account in just a few steps. Figuring out, in such a model, how to act on a specific factor thus becomes a nontrivial task. The treatment of the problem must therefore make use of some informed search technique or similar methodology to reduce the number of states to be evaluated. In our case, we resort to the concept of *relevance*.

4.3.2 Relevant implications

In the context of the paper in which it is defined, relevance is used as a metric to assign the model to a chaining job that will eventually execute it according to a scheduling algorithm. Instead, the importance of the definition of relevance of a model for this work is related to the possibility of extending that notion of relevance and making it applicable to implications, since *the process of making implications must proceed guided by relevance*. In the process of meaning generation, the search for connections (known or conjectured) between models is aimed at achieving goal states. As we mentioned in Section 4.3, a simple BFS on the graph defined on the agent's knowledge is not a viable approach in real-case scenarios. Proceeding by choosing *relevant* models each time allows us to reduce the scope of the search by targeting the paths that seem most promising. In constructing causal chains by application of non-axiomatic reasoning, the objective is to reconstruct a chain of models such that, knowing how to control an initial model, it is possible to influence the variables related to a goal state. Each time an intermediate model (i.e., one that is neither at the beginning nor at the end) is added to this chain, this new model is chosen from the set of models M causally related to the previous $tail$ of the partial implication, sorting these models using a metric based on *relevance* computed in the *current situation*. Relevance is an anytime algorithm[4] executed *on the fly* to figure out what is relevant at any given moment in a given time frame.

Definition 4.3.5 (Relevance in implication-making). Given the tail of a partial implication $Tail_{PI}$ and a set of candidate models M causally related to $Tail_{PI}$, the new tail of the partial implication $Tail'_{PI}$ will be the model $m_i \in M$ that achieves the highest relevance score calculated given the current context C_t and a time period τ:

$$Tail'_{PI} = \mathrm{Max}_i(Rel(m_i, T, \tau))) \qquad \text{for} \quad m_i \in CM(Tail_{PI})$$

where $CM(Tail_{PI})$ is a function that returns M, the set of causal models related to $Tail_{PI}$ and Rel is the function computing the relevance.

Repeated application of the relevance metric in the process of constructing an implication allows us to discard paths that seem less useful for achieving the goal, effectively *pruning* the search tree.

Having defined the role of relevance in the construction of an implication, we are now interested in understanding the relevance of a complete implication so that we can compare it with other implications. The way we define the relevance of an implication differs from the previous definitions of relevance introduced. A complete implication may be treated just like any other knowledge element, so considering an entire implication as a single, large model, one could calculate its relevance with the formulas introduced in Section 3.7.1. However, two implications connecting the same datum to the same goal, in the same situation, created at the same time may share the same relevance value, since it is computed by abstracting from the details of the composition of individual chains, in which case it would not be possible to assess whether the two chains are actually equivalent or not. We therefore propose a more capillary system to compute the relevance of an implication that gets into the merits of the composition of individual causal chains. The relevance value of an implication should emerge from the relevance of its constituents, specifically:

- The **relevance of the models** that constitute the implication: the more relevant the models that make up an implication are, the more relevance the implication is expected to have[5]. In general, the relevance of individual models should result in a value directly proportional to the relevance of the implication;

- The **type of goal** to which the implication refers. The more important the goal where the implication ends, the more relevant the implication should be, so the "importance" of a goal should result in a value directly proportional to the relevance of the implication. Since it is theoretically possible to compose and organize goals hierarchically, it would be appropriate to assign an importance value to each level of the hierarchy (if such a hierarchy system were implemented), where goals higher in the hierarchy are assumed to be important (they are more "general" than others in a sense) and thus take on higher values. The value that matters for the purpose of estimating the relevance of the implication is the value associated with the highest goal reached by the implication; in the case where the implication ends in a subgoal, we proceed to identify the most

[4] An anytime algorithm is an algorithm that can return a valid results even if the computation is stopped before its termination (albeit the *quality* of the results may vary).

[5] Note that the relevance of models changes over time, thus an implication's relevance is not guaranteed to remain unchanged over time

important goal to which the subgoal is connected. Similarly, higher values could be assigned to 'active' goals and lower values to 'inactive' goals, as active goals could be said to be more important than inactive goals, since they are the focus of the controller's *attention*[6];

- The **length** of the implication: the longer an implication is, the more models it contains, each of which may become less relevant over time. Therefore, one or more models are more likely to be discarded or become less relevant over time. Furthermore, in longer implications the goal attainment mechanism is be more convoluted and time-consuming to apply (running a large number of models requires more computational resources than running a small set of models). The length of implication should translate to a value that is inversely related to the relevance of the implication;
- The **number of variables** contained in the models that make up the implication. Since models can also be organized in hierarchies, a single high-level model (which would count as 'one' in determining the number of models present in the implication) might contain a large number of variables and become *heavier* to execute, taking up more computational resources to deal with (in terms of required resources). Therefore, the number of variables contained in the models should also translate into a factor inversely proportional to the meaning of the implication.

Building on this discussion, we provide a definition of relevance of an implication.

Definition 4.3.6 (Relevance of an implication). Given an implication Im linking a datum to an explicit goal, we define the **relevance** RI of the implication as its raw relevance $RR(Im)$ weighted by the hierarchical importance G_{Im} of the goal it references, divided by its length $len(Im)$ times the total number of variables contained in the models $m_i \in M_{Im} = Head_{Im} \cup Body_{Im} \cup Tail_{Im}$:

$$len(Im) = |Head_{Im}| + |Body_{Im}| + |Tail_{Im}|$$

$$num\_var(m) = |V_m|$$

$$RR(Im) = \sum_i Rel(m_i, T_i, \tau_i) \qquad \text{for} \quad m_i \in M_{Im}$$

$$RI(Im) = \frac{RR(Im) \cdot G_{Im}}{len(Im) \cdot \sum_i num\_var(m_i)} \qquad \text{for} \quad m_i \in M_{Im}$$

where V_m is the set of variables appearing in a model m and $len(Im)$ is the number of models $m_0, \cdots, m_n$ making up a single implication. $len(Im)$ can be calculated as $len(Im) = |Body_{Im}| + 2$ if $Head$ and $Tail$ consist of simple (not compound) models.

4.3.3 Reliable implications

Similar to what has already been done for the concept of relevance, we could define the reliability of an implication based on the reliability of individual models that constitute it. This kind of work is less obvious and is more difficult, however, since we would have to take into account the reliability of the individual models weighted by the length of the chain, but also the reliability of the chain as a whole (i.e., treated as a single piece of knowledge with which

[6] Here attention is understood as a mechanism for managing the controller's resources; attention decides which aspects are worth focusing on and which are not. For a more precise definition, see Section 5.2.1

to associate a reliability value), since it is more difficult to handle the wide variety of situations that might arise. Thus we defer a more in-depth discussion of the concept of reliable implications to future work on the subject, and here we focus mainly on the conditions that *invalidate* the chain.

In particular, we can assume that there is an **reliability threshold**, that is, a *minimum* level of reliability deemed acceptable for the use of the model in question. However, this value may not directly contribute to the elimination of the model in question, as the process used for model elimination may be based on a more nuanced set of controls. That said, if the reliability of any of the models in the chain falls below the reliability threshold, or if the model is discarded, the chain as a whole is **invalidated** and can no longer be used to direct the agent's actions[7]. Nevertheless, such a chain need not be thrown away, as the "failing" model breaks the implication into two partial implications, one originating from the datum and the other from the goal, each of which, taken individually, is still valid. The two partial chains thus created can then be fed back to the implication generation process to be extended.

Different, on the other hand, is the case of partial implications: this time the chain is "anchored" at its origin, i.e., its $Head$, therefore, in the case of "failure models", it will be necessary to discard the whole part of the chain from the failing model (included) to its $Tail$.

4.4 Meaning Generation

After introducing a set of notions on which to base our theory, we proposed a definition of meaning as the whole set of actionable information enabling an agent situated in a context to go toward a goal *attractor.* This definition gives us an insight into how meaning is essentially related to the execution of tasks, but it does not provide any practical tools for the agent to use to facilitate its activity. Indeed, that role is fulfilled by a process that *generates* meaning and appropriately places it in the agent's knowledge context. Building on the notions introduced in this chapter, we will provide in this section a list of design requirements for a meaning generation process and a formulation of such a process.

4.4.1 Design requirements

Before we can introduce our process of meaning generation (from now on abbreviated PMG), we must define the characteristics that such a process must possess. Many of these aspects are consequences of what has already been expressed regarding the constituent elements of meaning that have emerged from the intuitions and definitions set forth in the previous chapters. However, the design requirements laid out in this section are not necessarily sufficient, as many other assumptions could be added later, or even removed or revised from the following list.

1. **Context awareness**: the PMG must take into account the context in which the agent is placed, since any task assigned to the agent is inseparable from the environment, which imposes a set of constraints on what is possible and not possible. The context provides information about the current state;

2. **Knowledge as a Single Source of Truth**: the PMG must operate over the knowledge of the agent, which, as we argued in Section 3.3, represents the single source of truth

[7] This behavior is entirely analogous to an audio jack, which, by breaking at some intermediate point, prevents signal passage altogether

of the agent, that is, it is the only organized and explicit information base to which the agent is presumed to have access;

3. **Generate actionable information**: the result of the PMG must be a type of information that can direct the agent's actions toward its goals. To do this, it must make use of causal implications;

4. **Causal implications**: the PMG must use implications, defined as connections between a datum and goals identified in the agent's knowledge, since they constitute the essence of meaning. Specifically, such implications must be generated by applying logical reasoning to causal knowledge;

5. **Comply with the principles of LTE and AIKR**: a PMG is subject, like any process, to work with limited and often insufficient resources. In this sense, the PMG must deal with the possibility of disruptions and support the efficient management of resources by employing mechanisms such as partial implications and relevance;

6. **Explainability**: the PMG must be open and allow the rationale behind the results produced to be traced, both for reasons of review and identification of problems within the process itself and for ethical considerations for its practical applicability.

4.4.2 Definition

The design requirements set forth in the previous section provide the starting point for the following formulation of the meaning generation process. Meaning generation is a dynamic process that creates knowledge of a particular functional kind, namely the kind that can be used to achieve goals in a particular situation given a datum. We define the meaning generation process as the recursive application of a function that computes relevant implications from a datum, a context, and the agent's goals by identifying them in the agent's knowledge base. With each invocation of the function, the process produces a set of partial implications that are the input to the next application of the function. The construction of the partial implications is guided by a relevance mechanism that reduces the scope of the search by discriminating the most relevant models in the agent's knowledge base and brings the agent closer to the goal states. By providing an explicit representation of the computed meaning, the process lends itself to interventions for validation and control and revision of its correctness.

Definition 4.4.1 (Meaning generation). Given a datum d and an agent A with knowledge K, a set of goals $G = \{g_0, g_1, \cdots, g_n\}$, and the perceived context PC, the process of meaning generation (PMG) is defined as the recursive application of a function that computes the set of relevant implications RI linking d to one or more elements of G isolating the most relevant causal connections that can be found in the pair (K, PC), of the form $d \xrightarrow{RI(K,PC)} G$, optionally starting from or integrating a set of partial implications, represented

$$MG_0(d, A_0(K, G, PC), RI_0 = \emptyset) = Rel(I(d, PC, A(K, G)))$$

$$MG_t(d, A_t(K, G, PC), RI_{t-1}) = Rel(I_t(d, PC, A(K, G)) \cup I_t(RI_\tau, PC, A(K, G)))$$

where $\tau < t$ and Rel is the function that, for each implication Im produced by I, computes its relevance $RI(Im)$.

The generation of meaning is therefore the recursive application of a function that computes implications starting from available knowledge or previously generated implications.

This meaning generation process can be executed with respect to any perceived datum or element present in the agent's knowledge. The process is executed according to the ways defined by the agent's internal mechanisms and, once started, is assumed to continue running and producing partial implications until one of the following conditions occurs: (a) the resources available to the PMG are exhausted, or the process is stopped for resource-saving reasons, (b) the PMG identifies a sufficient number of relevant implications such that the agent temporarily suspends the process to avoid wasting resources, or (c) the PMG identifies the optimal implications (those that are *minimal* in some sense, presumably based on an internal metric that optimizes for e.g., the number of models traversed, variables controlled, or resources/time spent) present in the agent's knowledge and is paused until models are added or discarded. In general, it is assumed that the meaning generation process is always active, in the perspective of online lifelong learning, except in cases of resource limitation where the agent can discretionally terminate one or more processes altogether (e.g., if it starts a predefined routine to cope with a the specific need). The result of the meaning generation process is a set of relevant implications that represent *the* meaning.

The role of hypotheses What if there are no models in the agent's knowledge that connect back to the goals? This is not an obvious question, as an agent involved in the performance of a task is expected to spend almost the entirety of the time it takes to complete that task in a state in which no causal connections to the task goals are present in its knowledge, assuming that, once in possession of a solution, the task is completed instantaneously. For this reason, a meaning generation process should be coupled with a hypothesis generation process, which, starting from the agent's interactions with the task-environment, identifies possible variables of interest and generates a representation of them. For optimal performance, the two processes should work in parallel. Hypothesis generation mechanisms are outside the scope of this thesis and, therefore, will not be discussed.

4.5 Meaning generation under LTE

Having introduced the LTE assumption and having clarified that all tasks (at least those worth doing) require a certain amount of energy and time to accomplish, we reformulate our previous definitions of *implications*, *implication relevance*, and *meaning generation* to include precise temporal references to make clear the sequence of computation steps.

Definition 4.5.1 (Implication under LTE). Given a context C at a specific moment in time t, the implications I of a perceived datum d_t computed by an agent A are the elements of a subset of reasoning chains, namely deductions, inductions, abductions, and analogies $R_{t+z} = De_t \cup In_t \cup Ab_t \cup An_t$, over the knowledge K_t of A and the PC_t perceived context by the agent. A single implication is a chain of causal relations $cr \in K_{t+x}$ linked together by deduction, induction, and analogy starting from d_t and ending with an explicit goal $g_i \in G_{t+x}$, and by abduction starting from an explicit goal g_i and ending with d_t, represented

$$\begin{aligned} R_\tau(d_\tau, PC_\tau, K_A, G_A) = De_\tau(d_\tau, PC_\tau, K_A, G_A) \cup In_\tau(d_\tau, PC_\tau, K_A, G_A) \\ \cup Ab_\tau(d_\tau, PC_\tau, K_A, G_A) \cup An_\tau(d_\tau, PC_\tau, K_A, G_A) \end{aligned}$$

$$I_{t+z}(d_t, C_t, A(K_{t+x}, G_{t+x})) = R_{t+x}(d_t, P_t, K_{A,t+x}, G_{A,t+x})$$

$t + z$ because computation in the physical world takes time. The computation of implications begins y time steps after recalling or perceiving d and requires additional x time steps to complete, therefore $z = x + y$. We explicitly refer to a subset and not the complete set of reasoning chains because of time and energy constraints (not enough time to compute nor enough memory to store all possible implications). It might also be possible for the set to be empty $I = \emptyset$, if the agent does not have enough time or resources to compute implications or if no implication can be derived from a given combination of input values.

Similarly, we can rewrite the definitions of relevant implications and of the process of meaning generation[8].

Definition 4.5.2 (Relevance of an implication under LTE). Given an implication Im linking a datum to an explicit goal, we define the **relevance** RI_{t+z} of the implication as its raw relevance $RR_{t+y}(Im)$ weighted by the hierarchical importance G_{Im} of the goal it references, divided by the length $len(Im)$ times the total number of variables contained in the models $m_i \in M_{Im} = Head_{Im} \cup Body_{Im} \cup Tail_{Im}$:

$$len(Im) = |Body_{Im}| + 2$$

$$num\_var(m) = |V_m|$$

$$RR_{t+y}(Im) = \sum_i Rel_{t+x_i}(m_i, T_i, \tau_i) \qquad \text{for} \quad m_i \in M_{t,Im}$$

$$RI_{t+z}(Im) = \frac{RR(Im)_{t+y} \times G_{Im}}{len(Im) \times \sum_i num\_var(m_i)} \qquad \text{for} \quad m_i \in M_{Im}$$

where V_m is the set of variables appearing in a model m.

x_i is the amount of time required to compute the relevance of a model m_i, $y = \sum_i(x_i)$, $z = y + f$ where f is the amount of time required to execute the computation of RI only.

Definition 4.5.3 (Meaning generation process under LTE). Given a datum d and an agent A with knowledge K, a set of goals $G = \{g_0, g_1, \cdots, g_n\}$, and the perceived context PC, the process of meaning generation (PMG) is defined as the recursive application of a function that computes the set of relevant implications RI linking d to one or more elements of G isolating the most relevant causal connections that can be found in the pair (K, PC), of the form $d \xrightarrow{RI(K,PC)} G$, optionally starting from or integrating a set of partial implications, represented

$$MG_0(d, A_0(K, G, PC), RI_0 = \emptyset) = Rel(I(d, PC, A(K, G)))$$

$$MG_{t+z}(d, A_t(K, G, PC), RI_{t-1}) = Rel_{t+y}(I_{t+x}(d, PC, A(K, G)) \cup I_{t+w}(RI_{t-1}))$$

where $t + z > t + y > max(t + x, t + w)$ because computation requires time in the real world. After computing the two sets of (possibly partial) implications, the function used to compute the relevant implications requires additional time to complete and return its results. In this case we are not interested in defining passage of time in the base case, as it follows logically from the recursive application of the function MG.

We conclude this chapter by introducing some results of our definitions of meaning and meaning generation and hinting at future work. In particular, we explore the consequences of our theory on the degree meaning an agent may possess and the relationship between meaning and understanding.

[8] We also do not include partial implication formulas in this dissection because the definition of partial implication is equivalent to the definition of implication. The definition of relevance on partial implications is omitted for brevity

4.6 Degrees of meaning

In light of the definitions of meaning and meaning generation formalized here, we can define more precisely under what conditions meaning is being generated. A consequence of this clarification is the possibility of indicating which types of agents are capable of generating meaning and which are not – and which mechanisms would enable them to generate meaning.

Building on the definitions and insights already discussed at length, we can isolate four pillars on which meaning rests: a datum, one or more goals, a situation (context), and knowledge. When even just one of these elements is missing, because, for example, it is not handled by the controller or not taken into account by the designer, we can expect limitations to emerge in the ability to handle meaning. Let us analyze each of these four aspects in more detail.

Datum By "datum" we denote one of the subjects of meaning, which we might call *active*, since it is the one to which the meaning is presumed to refer. In the absence of a datum, the origin of any meaning cannot be found. The datum is, yes, understood as a representation in the agent's knowledge, but it is also the reference to the object closest to the agent that can be leveraged to achieve the goals. That is, the difference between datum and any other piece of knowledge held by the agent is that the datum is, at the time the meaning generation process is initiated, the item that is intended to be used in some way to achieve goals. This could be because, for example, the datum represents a physical object that can be materially used to perform actions, but this is not necessarily always the case. Since the datum is missing, the first step for initiating the goal-setting process is also missing, and, consequently, meaning cannot be generated.

Goals We have previously introduced a characterization of goals in order to refer to them more precisely. We have already talked about desirable and avoidable states, and mentioned how active and inactive goals can influence the process of generating relevant implications. But now we are particularly interested in the distinction between implicit and explicit goals. Implicit goals are so defined because they are not explicitly represented in the knowledge of the agent, who, for this reason, cannot inspect and reason about them to improve his or her performance in carrying out the task. Lacking the explicit representation of goals, the meaning generation process (as defined in this paper) simply cannot be applied[9]. Therefore, systems such as Braitenberg vehicles, whose behavior is hardwired, do not compute meaning. This category also includes reactive systems, which do not make decisions according to the input they receive, but, whatever the input, a standard action programmed into the system is executed. But there is more: explicit goal representation allows for **choices** to be made with respect to the goal itself. Conversely, it could be said that there is no reason to endow an agent with an explicit goal representation if it cannot *choose to (re)act* in different ways. A rolling rock does not generate meaning, since it has neither an explicit goal representation nor the ability to *act*, as a result of a "reasoned choice", to change its behavior, but is subject simply to the physical laws of the world in which it is embedded. In conclusion, another possible discriminator for meaning production is the possibility for agents to **choose** to **act** with respect to a **goal**.

Context Context is a set of information derived from the task-environment in which the agent is situated, including the constraints defined by the environment and the current state

[9] When we talk about representation, we do not necessarily refer to *symbolic* type representation. Therefore, a subsymbolic-type representation could equally enable the generation of meaning

of the environment. Without context, meaning cannot exist. This idea is further supported by the fact that other views of meaning (such as, for example, the semantics of languages – natural or programming languages) equally situate meaning in a context. A datum, if abstracted from context – whether physical or virtual, cannot be related to anything, not even goals. This is because the goals themselves are grounded in the task-environment, and, in the absence of manipulable variables to manipulate and observable variables to observe, the variables that would define the goal state are also missing. Talking about the meaning of something requires therefore the explication of context (this could have further repercussions on the role of meaning in communication).

Knowledge Finally, we come to knowledge. Knowledge is the set of actionable information, i.e., bi-directional causal models, to which the agent has access. Knowledge is pseudo-axiomatic, that is, it constitutes a basis of "truth" on which the agent bases its actions, in the awareness that it may be incorrect and subject to revision. The complete absence of knowledge implies an inability to perform actions. Lack of *causal* knowledge leaves the agent with only the information about the correlation of two events, which, as expressed in Section 2.2.4, may not be sufficient to complete a task. In the case of meaning, the observable correlation would be that between the datum and one of the goals: observing the presence of the datum and the goal at the same time would mean that the datum has some meaning, otherwise nothing could be said. Therefore, since our definition of meaning refers to causal knowledge, agents without causal knowledge will be somewhat limited in the production of meaning.

4.7 Meaning and understanding

We introduced earlier the definition of understanding as a multidimensional gradient that depends on completeness and accuracy of the set of elements related to a phenomenon ϕ represented by a set of models M that constitute knowledge K of an agent A. Specifically, we said that the understanding of ϕ increases as agent A's ability to predict, reach goals, explain and recreate the phenomenon in question increases.

Meaning and understanding are intuitively closely related. In the common usage of these concepts, it is not strange to speak of "understanding the meaning" of something. Here we intend to focus on the relationship between meaning and understanding from both Thórisson's definition of understanding and the definitions of meaning and meaning generation given in this thesis.

4.7.1 Meaning to understanding

We can draw initial parallels between these two concepts from the definition of meaning used in this work. Meaning is described here as the set of information that defines the agent's (the subject possessing that information) state of achieving certain goals. It is immediately apparent to us from this definition that meaning requires a set of *descriptions* for the possible achievement of goals. In this sense, the models related to meaning, which we might even refer to as "models of the connections between models" (possible alternative definition of implications), links meaning to the goal achievement dimension of understanding.

The process of constructing implications also employs reasoning to generate new hypotheses through deductive, inductive, abductive, and analogy reasoning. These hypotheses are based on the knowledge possessed by the agent, and their quality depends on the agent's mechanisms that implement the reasoning. In this sense, the creation of new causal models that could link to the agent's goals is a form of relevance-driven prediction making (generation of models that are assumed to be likely to be relevant to the achievement of the goals). Meaning then also appears to be potentially related to the predictive dimension of understanding. Similarly, the use of causal models supports the aspect of explanation present in understanding, as the meaning generation process applies abduction mechanisms on the agent's knowledge. In addition, because meaning is related to the level of detail, these reconstructions can cover both compound models taken as a whole and broken down into their individual component models.

Finally, the re-creation of a phenomenon is related, according to the definition of understanding used, to the production of models that exhibit the necessary and sufficient features of the phenomenon. The meaning-generating process defined here does not directly address this dimension of understanding, however, the production of multiple implications linking a datum to the same goal enables another process that, taking as input that set of implications and their relevance values, can attempt to discern the necessary and sufficient causes for the achievement of that goal.

In summary, the causal models produced as a result of the meaning generation mechanism can almost always also be directly used by an understanding evaluation process due to their properties of causality and connection through abduction with both goal and other models. We can thus see meaning generation as a process that guides the development of understanding relative to a phenomenon (such as the "datum" at the center of meaning generation or the agent's goals). In this sense, the better the meaning generation process, the better the models produced ("better" in the sense of reliability and relevance) and the greater the connections identified, and, therefore, the greater the understanding shown by the agent of a given phenomenon. Conversely, in the absence of a good meaning production process the understanding may be constrained.

4.7.2 Understanding to meaning

Conversely, we can treat meaning as a phenomenon subject to understanding. Viewed from this perspective, the understanding of meaning spans the four dimensions already illustrated: prediction, goal achievement, explanation, and re-creation. From our definition of meaning we know that meaning does not exist 'in a vacuum' (to use Thórisson's words), but is always defined for a datum and in relation to a context and to an agent's knowledge and goals. We also defined a process for meaning generation that brings all these concepts together. Since the phenomenon of meaning rests on many aspects, to "understand the meaning" (of something) necessarily requires understanding datum, knowledge, goals, context, and process of generation. The analysis of this verse is more convoluted than the previous one and is, therefore, deferred to future work.

CHAPTER 5

Conclusion

In this chapter we conclude the thesis by summarizing the main results of our work, outlining some ideas for future work and pointing out open issues.

The work in this thesis is motivated by the need to dissect the concept of meaning in systems operating in complex environments such as the physical world. With a view to making an appreciable contribution to constructivist research, a careful review of major previous related works was conducted, attempting to reconnect them neatly with the results of this research. An "empirical" method was then applied for the elicitation of features of *meaning* from the common usage of the concept. Once the characterization of the concept of meaning was defined, a theory for the process of meaning generation implementable by an agent was discussed. Such a theory takes and expands on the concepts of "implications" and "relevance", discussed in earlier constructivist work, and combines them into a meaning generation process described in sufficient detail to be implemented in an intelligent system. Finally, connections of the concept of meaning with other concepts related to constructivist research were explored in order to provide valuable insights for future work.

5.1 Testing

The particular formulation of the notion of meaning and the meaning generation process described in this thesis can easily find application in the implementation of artificial intelligence systems. This represents a significant opportunity to test the validity of the theories set forth here. In particular, the implementation of the tests can be done through employment of the constructivist Task Theory already introduced. Thanks to the extended causal diagram notation – in which manipulatable, observable and goal variables are present – and the notion of intricacy, it is possible to produce the precised description of task-environments including a measure of their "complexity". A task-environment described in this way can then be submitted to an intelligent agent to perform a series of tests. In particular, it is possible to compare the performance of the same agent on that task-environment in cases where the agent has equipped a meaning-generation mechanism and those where it does not. In addition, multiple implementations of the meaning-generation mechanism can be tested to evaluate alternatives and improvements.

Thereafter, one could proceed with the creation of increasingly sophisticated task environments to see whether the application of a meaning-generation mechanism to an intelligent agent significantly affects the agent's performance in the tasks. The goal would be to understand whether, as task complexity increases, meaning generation provides an increasingly significant advantage in performing the task. Potential candidate artificial intelligence systems for implementing this mechanism are the Non-Axiomatic Reasoning System (NARS) and Autocatalytic Endogenous Reflective Architecture (AERA), both AGI-aspiring systems, Should the use of meaning generation mechanisms prove to be significantly useful in the performance of tasks, it would be appropriate to compare systems that implement meaning generation with other control systems such as reinforcement learners on the same tasks.

Finally, it would be particularly interesting to evaluate the transfer learning capability of an agent equipped with meaning generation by having the same agent perform two similar tasks (i.e., sharing a number of variables and the causal relationships between them) and see if, by retaining the models generated by the meaning generation process while performing the first task, it is able to effectively reuse them in the second task. In particular, in the case of NARS and AERA, it would be interesting to evaluate the results against the transfer learning capability already present in their respective implementations.

5.2 Future work and open issues

In this section we outline possible directions for future work on a theory of meaning and some open issues that have yet to be addressed. As hinted at in Section 4.3.3, one of the possible directions of this work could be to explore the concept of reliability of an implication. In addition, it would be interesting to further explore the analysis begun in Section 4.7 on the relationship between meaning and understanding. Below we provide other insights for future work.

5.2.1 Attention and planning

A transversal resource management process is necessary for any system that must operate in physical environments, since the world is much more complex and wide-ranging than what the system's resources allow it to explore at any given time. It is critical to select what to devote time to and invest resources in. We call this mechanism attention (Thórisson, 2022f). In applying the concept of relevance to the search for implications that lead to goals, the similarity with techniques of informed search and pruning on graphs has been invoked. As a technique for narrowing the scope of search, the application of relevance to both individual models and the model search function effectively acts as a mechanism to contain resource use. Further research efforts in methodologies could then be devoted to making this agent knowledge search process even more efficient, so that only patterns that are relevant in a given situation are selected and retained.

A process of meaning generation could also support *planning* activities. Planning is the set of operations involved in examining alternative ways of proceeding, based on predictions about future events and the assessed quality of the solutions provided so far, at any point in time. Attention and planning are related in the sense that good plans are also not using more resources than are strictly necessary. Our process for generating meaning can produce multiple viable alternatives for achieving goals from the knowledge possessed. Associating implications with relevance provides information about the expected utility of each implication. By describing in more detail how cost information (in terms of energy and time) can be incorporated into the models generated by the process, planning that takes into account the reasoned use of resources could be further supported.

5.2.2 Symbols and communication

One of the possible future developments is the comparison of the theories of meaning and meaning generation set forth in this thesis with theories of communication and symbol meaning. In particular, it would be interesting to review the previous work on meaning in communication between two subjects whose purpose is to share the same concept and assess the feasibility of offering new insights. Work in this area could focus on understanding when meaning is the same for two agents and how meaning can be exchanged and negotiated.

One type of datum whose meaning we often want to evaluate are symbols. Symbols are a static entity, so they do not contain meaning per se, but it is generated and assigned to them. Of particular interest are symbols used in natural language representation (text). Understanding the meaning of words would go a long way toward understanding the meaning of concepts expressed using those words. The disciplines of natural language processing (NLP) and natural language understanding (NLU), which nowadays are widely studied, are also interested in this topic. The newer approaches based on Large Language Models that achieve state-of-the-art performance in many language-related tasks operate as “black-boxes”, as they are not designed to support explainability processes. In contrast, a model-based meaning generation mechanism constitutes a “white-box”, human readable approach. Our theory of meaning applied to reflective controllers capable of hypothesis-making could lead to overcoming the limitations of previous model-based approaches to meaning, while retaining the white-box approach that is sought.

Linked to both symbols and communication, it would be very interesting to study how an agent can understand the meaning another agent gives to a given sentence, as this mechanism would enable the development of agents that acquire task goals from natural language.

5.2.3 Education

Our theory of meaning built on the foundations of task theory can also be related back to teaching and training. Teaching is one of the fundamental aspects of education, the methodical activity designed to improve a learner’s performance in carrying out a task. The Pedagogical Pentagon defined by Thórisson, Bieger, and B. R. Steunebrink (2017) is a conceptual framework for addressing the five pillars of education: learning, teaching, training, environments, and testing. To teach is to provide directions for carrying out tasks, so it is necessary for learners to have a process for acquiring meaning, that is, tracing vague instructions back to better defined goal states, and then identifying connections with their own abilities and evaluating their development by testing themselves with the task. The training phase of learning involves performing repeated actions over time with the goal of becoming better at some task, while at the same time trying to avoid learning the wrong skills and to avoid forgetting or unlearning desirable skills (Thórisson, 2022g). Since teaching and training are both devoted to conveying notions or developing skills that are useful goal achievement, it would be helpful to have a definition of utility to apply to the knowledge to be taught or trained on. The concept of relevance paired with the features of a good meaning generation process can provide a basis from which to define a metric of empiric usefulness to be associated with the notions to be taught.

5.2.4 Open issues

The process of meaning generation can consume considerable agent resources, both in terms of time and computational load, and in terms of memory. Although a single implication can be implemented as a list of pointers to existing models, in cases where the goal states are far removed from the actual circumstances (e.g., if they are formed by variables unknown to the agent) and the mechanisms governing the task are particularly complex, the chains of partial implications could extend indefinitely in a continuous trial-and-error path pointing towards the goal states. The relevance mechanism currently considered is based on probabilistic estimates of the relevance of acquired models as a function of the utility of individual models assessed according to frequentist metrics. This mechanism, coupled with abductive reasoning for the definition of subgoals, should theoretically allow the least useful models to be progressively eliminated and gradually move closer to the solution, but further investigation would need to be conducted on deadlock and starvation situations of the system, in which the resources available to the agent are consumed in a vain attempt to reach the goal. For example, further investigation would need to be conducted into what would happen if, after eliminating a model from the system because it was deemed ineffective, it was re-learned due to conditions that are repeated cyclically over time at unidentified intervals. If this were to happen, the system's resources could be consumed trying to use the ineffective model again to achieve the goals.

5.3 Final remarks

New perspectives emerge from the outcome of this work on the production of meaning in autonomous grounded systems. The progressive process of defining the concept of meaning has established a solid foundation on which to base a future theory of meaning. New insights into the concepts of implication and relevance, and the proposal of their use in a meaning generation process, will hopefully lead to a more complete definition of the meaning generation process. The hope is that this work will contribute to the establishment of the notion of meaning as a practical phenomenon, for if a theory of pragmatic meaning applied to communication were derived, the implications would be of great relevance to the construction of more general AI systems.

Bibliography

Belenchia, Matteo (2021). “Towards a Theory of Causally Grounded Tasks”. University of Camerino.

Bieger, Jordi and Kristinn R. Thórisson (2017). “Evaluating Understanding”. In.

Bieger, Jordi, Kristinn R. Thórisson, et al. (2016). “Evaluation of General-Purpose Artificial Intelligence : Why , What & How”. In.

Bounded Recursive Self-Improvement (2013). arXiv: 1312.6764 [cs.AI].

Buxton, Claude E. (1985). *Influences in Psychology: Points of View in the Modern History of Psychology*. Academic Press.

Chandrasekaran, Balakrishnan, John R. Josephson, and V. Richard Benjamins. (1999). “Ontology of tasks and methods”. In.

Civile, Sistema Protezione (2008). *Le cause degli incendi boschivi*. URL: https://www.sistemaprotezionecivile.it/allegati/140_Cause_degli_incendi_boschivi.pdf.

Conant, Roger C. and W. Ross Ashby (1970). “Every good regulator of a system must be a model of that system”. In: *Intl. J. Systems Science*, pp. 89–97.

Eberding, Leonard M. et al. (2021). “International Conference on Artificial General Intelligence, AGI-21”. In: pp. 65-7–4.

Epp, Susanna S. (2004). *Discrete Mathematics with Applications*. Thomson-Brooks/Cole. ISBN: 9780534359454.

Galton, Francis (Jan. 1888). “Co-Relations and Their Measurement, Chiefly from Anthropometric Data”. In: *Proceedings of the Royal Society of London Series I* 45, pp. 135–145.

Geraghty, Maurice A. (2018). *Inferential Statistics and Probability - A Holistic Approach*. De Hanza College, Department of Mathematics.

Harnad, Stevan (1990). “The symbol grounding problem”. In: *Physica D: Nonlinear Phenomena* 42.1, pp. 335–346. ISSN: 0167-2789. DOI: https://doi.org/10.1016/0167-2789(90)90087-6. URL: https://www.sciencedirect.com/science/article/pii/0167278990900876.

Haugeland, John (1985). *Artificial intelligence: The very idea*. ISBN: 9780262081535.

Howe, Jim (1994). *Artificial Intelligence at Edinburgh University : a Perspective*. Retrieved 2023-09-10. URL: https://www.inf.ed.ac.uk/about/AIhistory.html.

Ignelzi, Michael (2000). “Meaning-making in the learning and teaching process”. In: pp. 5–14. DOI: 10.1002/tl.8201.

Kautz, Henry (2020). *The Third AI Summer*. Video online. Retrieved 2023-09-10. URL: https://www.youtube.com/watch?v=_cQITY0SPiw&ab_channel=HenryKautz.

Legg, Shane and Marcus Hutter (2007). *A Collection of Definitions of Intelligence.* arXiv: 0706.3639 [cs.AI].

McCulloch, Warren and Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5.5, pp. 115–133.

Newell, Allen and Herbert A. Simon (1976). *Computer Science as Empirical Inquiry: Symbols and Search.* Communications of the ACM. vol. 19, No. 3, pp. 113-126.

Nivel, Eric and Kristinn R. Thórisson (July 2013). "Towards a Programming Paradigm for Control Systems With High Levels of Existential Autonomy". In: pp. 78–87.

Nivel, Eric, Kristinn R. Thórisson, Bas Steunebrink, Haris Dindo, et al. (2014). "Bounded Seed-AGI". In: *Artificial General Intelligence.* Ed. by Ben Goertzel, Laurent Orseau, and Javier Snaider. Cham: Springer International Publishing, pp. 85–96. ISBN: 978-3-319-09274-4.

Nivel, Eric, Kristinn R. Thórisson, Bas Steunebrink, and Jürgen Schmidhuber (2015). "Anytime Bounded Rationality". In: *Artificial General Intelligence.* Ed. by Jordi Bieger, Ben Goertzel, and Alexey Potapov. Cham: Springer International Publishing, pp. 121–130. ISBN: 978-3-319-21365-1.

Ogata, Katsuhiko (2010). *Modern Control Engineering.* Pearson.

Pattee, Howard H. (2001). "The Physics of Symbols: Bridging the Epistemic Cut". In: *Biosystems* 60, pp. 5–21.

Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1558604790.

— (2009). *Causality. Models, Reasoning, and Inference.* 2nd ed. Cambridge, UK: Cambridge University Press. ISBN: 978-0-521-89560-6. DOI: 10.1017/CBO9780511803161.

Pearl, Judea and Elias Bareinboim (Nov. 2014). "External Validity: From Do-Calculus to Transportability Across Populations". In: *Statistical Science* 29.4. ISSN: 0883-4237. DOI: 10.1214/14-sts486. URL: http://dx.doi.org/10.1214/14-STS486.

Pearl, Judea and Dana Mackenzie (2018). *The Book of Why: The New Science of Cause and Effect.* 1st. USA: Basic Books, Inc. ISBN: 046509760X.

Peirce, C.S. (1878). "How to Make Our Ideas Clear". In: *Popular Science Monthly* 12, pp. 286–302.

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf (2017). *Elements of Causal Inference: Foundations and Learning Algorithms.* The MIT Press. ISBN: 0262037319.

Robert A. Wilson, Frank C. Keil. et al. (1999). "The MIT Encyclopedia of the Cognitive Sciences". In: pp. 92–110. ISBN: 9780262338165. DOI: 10.1002/tl.8201.

Russell, Stuart J. and Peter Norvig (2003). *Artificial Intelligence: A Modern Approach (2nd ed.)* New Jersey: Prentice Hall. ISBN: 0-13-790395-2.

Thórisson, Kristinn R. (2008). "Modeling Multimodal Communication as a Complex System". In: Springer Lecture Series in Computer Science: Modeling Communication with Robots and Virtual Humans. New York: Springer, pp. 143–168.

— (Nov. 2009). "From Constructionist to Constructivist A.I." In: AAAI Fall Symposium Series: Biologically Inspired Cognitive Architectures. Menlo Park, CA: AAAI press, pp. 175–183.

— (2012). "A New Constructivist AI: From Manual Methods to Self-Constructive Systems". In: pp. 147–173.

— (2020a). "Discretionarily constrained adaptation under insufficient knowledge & resources". In: *Journal of Artificial General Intelligence* 11.2, pp. 7–12.

— (2020b). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t720-atai-2012:what_is_agi. [Online; accessed 7-July-2021].

Thórisson, Kristinn R. (2020c). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t_720_atai:atai-20:knowledge_representation. [Online; accessed 29-June-2021].

— (2020d). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t-720-atai:atai-20:agents_and_control. [Online; accessed 7-July-2021].

— (2020e). "Seed-Programmed Autonomous General Learning". In: 131, pp. 32–70.

— (2022a). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t-720-atai:atai-22:methodologies. [Online; accessed 4-November-2023].

— (2022b). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t-720-atai:atai-22:learning. [Online; accessed 19-November-2023].

— (2022c). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t-720-atai:atai-22:task-environment. [Online; accessed 19-September-2023].

— (2022d). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t_720_atai:atai-22:self-x. [Online; accessed 02-December-2023].

— (2022e). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t-720-atai:atai-22:understanding. [Online; accessed 15-September-2023].

— (2022f). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t720-atai:atai-22:generality. [Online; accessed 2-December-2023].

— (2022g). *Lecture notes in Advanced Topics in Artificial Intelligence.* http://cadia.ru.is/wiki/public:t_720_atai:atai-22:teaching. [Online; accessed 22-September-2023].

Thórisson, Kristinn R., Hrvoje Benko, et al. (Nov. 2004). "Constructionist Design Methodology for Interactive Intelligence". In: American Association for Artificial Intelligence. Menlo Park, CA: A.I. Magazine, pp. 77–90.

Thórisson, Kristinn R., Jordi Bieger, Xiang Li, et al. (2019). "Cumulative Learning". In: *Proc. 12th International Conference on Artificial General Intelligence*, pp. 198–209.

Thórisson, Kristinn R., Jordi Bieger, Stephan Schiffel, et al. (July 2015). "Towards Flexible Task Environments for Comprehensive Evaluation of Artificial Intelligent Systems & Automatic Learners". In: Proc. 8th International Conference on Artificial General Intelligence (AGI-15). Berlin, Germany, pp. 187–196.

Thórisson, Kristinn R., Jordi Bieger, and Bas R. Steunebrink (Aug. 2017). "The Pedagogical Pentagon: A Conceptual Framework for Artificial Pedagogy". In: Proc. 10th International Conference on Artificial General Intelligence (AGI-17). Melbourne, Australia, pp. 212–222.

Thórisson, Kristinn R., Jordi Bieger, Thröstur Thorarensen, et al. (July 2016). "Why Artificial Intelligence Needs a Task Theory – And What It Might Look Like". In: vol. 9782, pp. 118–128. ISBN: 978-3-319-41648-9. DOI: 10.1007/978-3-319-41649-6_12.

Thórisson, Kristinn R. and Helgi P. Helgason (2012). "Cognitive architectures & autonomy: A comparative review". In: *Journal of Artificial General Intelligence* 3.2, pp. 1–30.

Thórisson, Kristinn R., David Kremelberg, et al. (July 2016). "About Understanding". In: ISBN: 978-3-319-41648-9. DOI: 10.1007/978-3-319-41649-6_11.

Thórisson, Kristinn R., Eric Nivel, et al. (2014). "Autonomous Acquisition of Natural Situated Communication". In: *IADIS International Journal on Computer Science and Information Systems* 9.2, pp. 115–131. ISSN: 1646-3692.

Thórisson, Kristinn R. and Arthur Talbot (July 2018a). "Abduction, Deduction & Causal-Relational Models". In.

— (2018b). "Cumulative Learning with Causal-Relational Models". In: *Artificial General Intelligence.* Ed. by Matthew Iklé et al. Cham: Springer International Publishing, pp. 227–237. ISBN: 978-3-319-97676-1.

Vaswani, Ashish et al. (2023). *Attention Is All You Need.* arXiv: 1706.03762 [cs.CL].

Wang, Pei (1995). "Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence". PhD thesis. Indiana University.

— (2004). "Toward a Unified Artificial Intelligence". In: *AAAI Technical Report.*

— (2009). *The Logic of Learning.* https://cis.temple.edu/~pwang/Publication/learning.pdf.

— (Apr. 2012). "The assumptions on knowledge and resources in models of rationality". In: *International Journal of Machine Consciousness* 03. DOI: 10.1142/S1793843011000686.

— (2019). "On Defining Artificial Intelligence". In: *Journal of Artificial General Intelligence* 10.2, pp. 1–37. DOI: doi:10.2478/jagi-2019-0002. URL: https://doi.org/10.2478/jagi-2019-0002.

White, Margaret B. and Alfred E. Hall (1980). "An overview of intelligence testing". In: *Phi Delta Kappa International* 58.4, pp. 210–216.

Wooldridge, Michael and Nicholas R. Jennings (1995). "Agent theories, architectures, and languages: A survey". In: DOI: 10.1007/3-540-58855-8.

Woolf, Peter et al. (2023). "Chemical Process Dynamics and Controls". University of Michigan.

Woolridge, M. (1997). "Agent-based software engineering". In: *IEEE Proceedings of Software Engineering.*