The Grand Challenges and Myths of Neural-Symbolic Computation*

Luis C. Lamb

Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre RS, Brazil
LuisLamb@acm.org

Abstract. The construction of computational cognitive models integrating the connectionist and symbolic paradigms of artificial intelligence is a standing research issue in the field. The combination of logic-based inference and connectionist learning systems may lead to the construction of semantically sound computational cognitive models in artificial intelligence, computer and cognitive sciences. Over the last decades, results regarding the computation and learning of classical reasoning within neural networks have been promising. Nonetheless, there still remains much do be done. Artificial intelligence, cognitive and computer science are strongly based on several non-classical reasoning formalisms, methodologies and logics. In knowledge representation, distributed systems, hardware design, theorem proving, systems specification and verification classical and non-classical logics have had a great impact on theory and real-world applications. Several challenges for neural-symbolic computation are pointed out, in particular for classical and non-classical computation in connectionist systems. We also analyse myths about neural-symbolic computation and shed new light on them considering recent research advances.

Keywords. Connectionist non-classical logics, neural-symbolic computation, non-classical reasoning, computational cognitive models.

1 Introduction

The construction of computational cognitive models integrating the connectionist and symbolic paradigms of artificial intelligence is a standing research issue in the field [1,2,3,4,5]. These models may lead to more effective and richer cognitive computational models, and to a better understanding of the computational processes and techniques of artificial intelligence, with benefits to computer and cognitive sciences. Several efforts have been made in this direction. However, most of them deal with symbolic knowledge expressed as production rules or logic programming [6,7,8]. Recently, several research

* Usually, this research area is referred to as neural-symbolic integration. However, I see *integration* as a methodology of neural-symbolic computation. If our aim is to construct computational models and technologies, I would prefer to name the area as *Neural-Symbolic Computation*.

results have shown how to offer richer knowledge representation, reasoning, and learning by means of non-classical computation in neural-symbolic systems [9,10,11,12]. This is an important development, since non-classical logics and models have offered solid foundations for computer science, including contributions to the basis of model checking, system specification and verification, reasoning in multi-agent and distributed systems and knowledge representation [13,14,15].

In addition, the construction of effective computational cognitive models has recently been pointed out by Valiant as great challenge for computing and cognitive sciences [16]. Further, the UK Computing Research Committee (UKCRC) and the British Computer Society have organised since 2002 an ambitious research enterprise which identified challenges that may lead to long-term positive effects not only for computer science, but also for neural-symbolic computation research [17], as we shall see in the sequel.

At least two of the grand challenges in computing research listed in [17] are particularly relevant to neural-symbolic computation research. It is expected over the next decades that studies on *the architecture of brains and minds* and *journeys in non-classical computation* [17] shall encourage researchers to think about long term prospects for their research fields. We summarise those two challenges.

As regards the first challenge, [17] state that many processes in our brains and minds have not yet been fully understood, including the way humans: (i) see many kinds of things around them; (ii) understand language tasks, such as reading comprehension; (iii) learn new concepts; (iv) decide what to do; (v) control their actions; (vi) remember things; (vii) enjoy or dislike things; (viii) become aware of their thoughts and emotions; (ix) learn about and take account of the mental states of others; (x) appreciate music and jokes; and (xi) sense the passage of time. Further, [17] state that in order to successfully achieve the goals of this project, major breakthroughs are needed, including research efforts in computer science and artificial intelligence. Such efforts include the development of "techniques for specifying and implementing many kinds of abstract mechanisms and processes in present and future physical machines" [17] . The aims of the project are ambitious and we believe that neural-symbolic computation can contribute to its success. The following quote summarises the purpose and desires of the challenge:

> *Inspired by both past and future advances in neuroscience, the project will attempt to build machines that simulate as much as is known about how brain mechanisms work. In parallel with that, the project will attempt to implement many kinds of abstract mental processes and mechanisms in physical machines, initially without requiring biological realism in the implementation mechanisms, but gradually adding more realism as our understanding increases.* [17]

As for the second grand challenge listed above, namely *journeys in non-classical computation* neural-symbolic computation, in particular connectionist non-classical logics [5] can also offer useful methods and computational constructions towards reaching the grand challenge research objectives. Computer science has been a successful human endeavour. Its foundations are based on mathematical abstractions implemented in physical constructions or devices. The *classical* theory of computation is based on logics, algorithms and mathematical abstractions assuming underlying Turing Machine models. However, such abstractions have physical embodiments which are reaching physical limits. As we reach the limits foretold by Gordon Moore in the sixties, alternatives are needed, if computer science is to continue being a successful science with further far-reaching technological implications. The following quote summarises this grand challenge.

> *Today's computing, classical computing, is a remarkable success story. However, there is a growing appreciation that it encompasses only a small subset of all computational possibilities. There are several paradigms that seem to define classical computing, but these may not be valid for all computation. As these paradigms are systematically challenged, the subject area is widened and enriched. The Grand Challenge Journeys in Non-classical Computation is nothing less than a reconceptualization of computation.*[17]

Non-classical connectionist models may shed some light on this challenge as well. Such models are built upon non-classical logics and abstractions *implemented* on a biologically inspired (although not necessarily biologically plausible) computing device. Although incipient, this new approach may well lead to models which will impact in the development of future technologies. They may offer a computational engine inspired in the way the brain works, being noise-tolerant and robust. This may lead to hardware implementations that could overcome the current limitations of the von Neumann architectures, drawing inspiration from on biological models of the brain, with respect to computation, learning and memory models. This would integrate both challenges from a connectionist non-classical model perspective.

All this sets the scene for neural-symbolic computation with respect to the grand challenges for computing research. However, we shall present some domain specific challenges that may contribute for neural-symbolic computation towards reaching more ambitious objectives.

The approach I use in the paper avoids technical details, but tries to offer a bird's eye view of current research in the field. The paper is organised as follows. First, we introduce the basics of non-classical connectionist computation. We then present research challenges and lay some myths of neural-symbolic computation on the line.

2 Preliminaries: A Taste of Connectionist Non-Classical Models

This section introduces the basics of connectionist models and non-classical connectionist models used in this article. The specialist reader can skip this section.

We assume familiarity with neural networks models and only summarise used concepts. A neural network can be seen as a massively parallel distributed processor that stores experiential knowledge [18]. A multilayer perceptron (MLP) is composed of several layers of simple processing units, the artificial neurons. Typically, neural-symbolic systems use some simple network model to compute and learn symbolic knowledge. In particular, there are several methods for representing time and symbolic knowledge in MLPs. In [9] a parallel representation of time is considered, using an ensemble of MLPs, where each network represents a specific timepoint. But before going into further detail, let us briefly describe the distinction between classical and non-classical reasoning.

In order to illustrate the difference between classical and non-classical logics let us consider temporal logic as described in [19]. Let us regard the assignment of truth-values to the propositions as being a description of the world or situation with respect to a particular time t. In temporal logic, the value assigned to a proposition (statement) can vary with the flow of time. This is not the case in classical logics: once a statement (a proposition) is proved its truth-value is definite. However, in artificial intelligence and computer science, as opposed to classical mathematics, time is an extremely relevant dimension as we are frequently working with several states, statements about particular timepoints or intervals, and several interpretations of these states. Under a modal interpretation of time, one could refer to the truth-values of a proposition in a linear timeline (considering both past and future), a branching time interpretation with several futures using modalities such as *always true in future/past*, *sometimes true in the future/past* among several other possibilities. This turns temporal logic into an expressive non-classical logical systems, which perhaps explains the success of this logic in computer science, artificial intelligence and cognitive science [12,13,20,21,22,23].

It is now common knowledge that modal logics have become one of the outstanding logical languages used in computer science from theoretical foundations [13,15] to state-of-the-art hardware [14] and multi-agent technologies [22]. The toolbox of any AI researcher now includes modal logics, as they were found appropriate for researches in several areas of AI. Areas such as knowledge representation, planning and theorem proving also have been making extensive use of modal logics, be they temporal, epistemic, conditional, intuitionistic, doxastic or many-dimensional modal logics, including, for instance, combinations of time and knowledge, time and belief, or space and time, to name a few [13,15,24].

In order to represent rich non-classical, symbolic knowledge in connectionist models, such as modal and temporal knowledge (which have been shown adequate in modelling multi-agent cognition [22]), one typically makes use of a hybrid approach, translating symbolic knowledge into a neural network, e.g. [6,9,12]. For instance, the temporal knowledge representation language either consider fragments of logic programming or extensions of logic programming clauses, including modalities or temporal operators [9,10,11].

To respond to the challenge put forward in [16], we have incorporated in a single model the two fundamental aspects of intelligent behaviour, namely reasoning and learning. We have applied the model in reasoning about distributed knowledge representation benchmarks [9,25], intuitionistic reasoning [10], temporal synchronisation and learning [12]. Thus, the following logic definitions shall be useful, as we shall use temporal reasoning to analyse the effectiveness of neural-symbolic computation.

Definition 1. *An* atom *A is a propositional variable; a* literal *L is an atom A or a negation of an atom ($\sim A$). A* clause *is an implication of the form $A \leftarrow L_1, L_2, ..., L_n$ with $n \geq 0$, where A is an atom and L_i, $1 \leq i \leq n$, are literals. A* program *$\mathcal{P}$ is a set of clauses. An* interpretation *of a program $\mathcal{P}$ is a mapping from each atom of a program to a truth value* true *or* false*. The* Immediate Consequence Operator *$\mathcal{T}_\mathcal{P}$ of a program $\mathcal{P}$ is a mapping from an interpretation $I_\mathcal{P}$ of $\mathcal{P}$ to another interpretation, and is defined as: $\mathcal{T}_\mathcal{P}(I_\mathcal{P})(A)$ is* true *if and only if there is a clause in $\mathcal{P}$ of the form $A \leftarrow L_1, L_2, ..., L_n$ and $\bigwedge_{i=1}^{n} I_\mathcal{P}(L_i)$ is true.*

Definition 2. *A* modal atom *is of the form MA where* $M \in \{\Box, \Diamond\}$ *and A is an atom. A* modal literal *is of the form ML where L is a literal.*

Definition 3. *A* modal program *is a finite set of clauses of the form* $\alpha_{n+1} \leftarrow \alpha_1 \wedge ... \wedge \alpha_n$*, where* α_i $(1 \leq i \leq n)$ *is either an atom or a modal atom, and* α_{n+1} *is an atom.*

Let us start with a simple example. It briefly illustrates how *Connectionist Modal Logics (CML)* can be used for modelling non-classical reasoning.

Example 1. Figure 1 shows an ensemble of three *CILP* [6] neural networks labelled as $(\omega_1, \omega_2, \omega_3)$, which might *communicate* in many different ways. Input and output neurons may represent the modalities $\Box L$, $\Diamond L$, or L, where L is a literal. The idea is to see ω_1, ω_2 and ω_3 as *possible worlds,* and to represent and compute modalities in neural networks. For example, (*i*) "If $\omega_1 : \Box A$ then $\omega_2 : A$"; i.e. if at the possible world ω_1 (here representing a neuron) we have $\Box A$ true, then A could be communicated from ω_1 to ω_2 by connecting $\Box A$ in ω_1 to A in ω_2 such that, whenever $\Box A$ is activated in ω_1, A is activated in ω_2. Similarly, (*ii*) "If $(\omega_2 : A) \vee (\omega_3 : A)$ then $\omega_1 : \Diamond A$" could be implemented by connecting neurons A of ω_2 and ω_3 to neuron $\Diamond A$ of ω_1 through a number of hidden neurons. Examples (*i*) and (*ii*) simulate, in a finite universe, the rules of $\Box$ *Elimination* and $\Diamond$ *Introduction* used in natural deduction style proof systems [13]. The representation of modalities in neural networks is described in detail in [5,11].

2.1 Connectionist Temporal Logics of Knowledge

Connectionist Temporal Logic of Knowledge CTLK [9] uses ensembles of *Connectionist Inductive Learning and Logic Programming (CILP)* neural networks [6,26]. *CILP* networks are single hidden layer networks that can be trained with backpropagation [27]. In *CILP*, a *Translation Algorithm* maps a temporal logic program $\mathcal{P}$ into a single hidden layer neural network $\mathcal{N}$ such that $\mathcal{N}$ computes the least fixed-point of $\mathcal{P}$. Let us illustrate the approach by presenting a simple example.

Example 2. (Next Time Operator) When combining temporal and epistemic logics, one has to define temporal and epistemic operators. Both are modal operators. The K_i modality is known as the *knowledge operator*. $K_i\alpha$ means that *agent i knows* α, where α is some propositional formula. One of the typical axioms of temporal logics of knowledge is $K_i \bigcirc \alpha \rightarrow \bigcirc K_i\alpha$ [28], where $\bigcirc$ denotes the *next time* temporal operator. This means that what an agent i knows today (K_i) about tomorrow ($\bigcirc\alpha$), she still knows tomorrow ($\bigcirc K_i\alpha$). In other words, this axiom states that an agent does not forget what she knew. This can be represented in an ensemble of neural networks with the use of a network that represents the agent's knowledge today, a network that represents the agent's knowledge tomorrow, and the appropriate connections between networks. Clearly, an output neuron $K\bigcirc\alpha$ of a network that represents agent i at time t needs to be connected to an output neuron $K\alpha$ of a network that represents agent i at time $t + 1$ in such a way that, whenever K$\bigcirc\alpha$ is activated, $K\alpha$ is also activated. This is illustrated in Fig. 2, where the black circle denotes a neuron that is always activated, and the activation value of output neuron K$\bigcirc\alpha$ is propagated to output neuron $K\alpha$. Weights must be such that $K\alpha$ is also activated.■

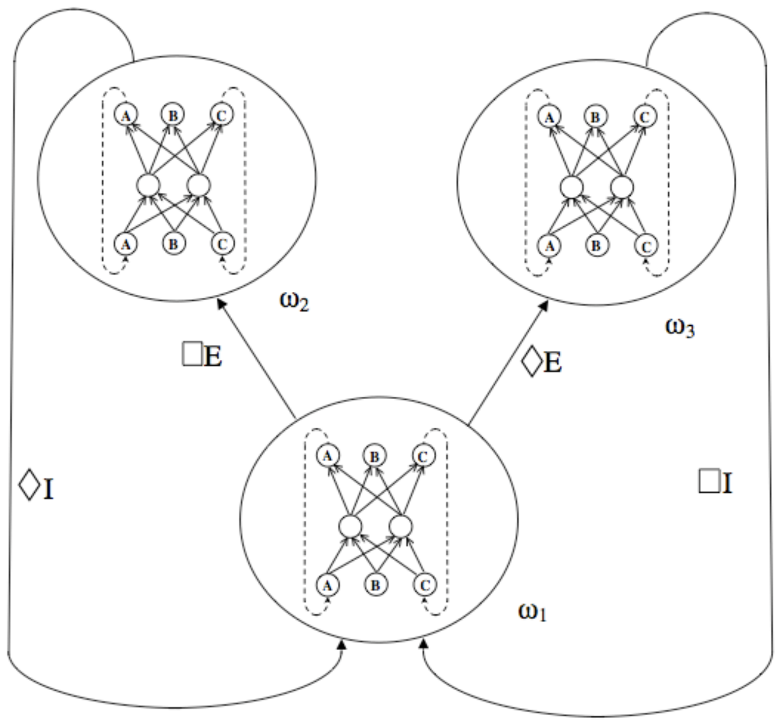

Fig. 1. Modal reasoning in connectionist models

Generally speaking, the idea behind a connectionist temporal logic is to have (instead of a single ensemble) a number n of ensembles, each representing the knowledge held by a number of agents at a given time point t. Figure 2 illustrates how this dynamic feature can be combined with the symbolic features of the knowledge represented in each network, allowing not only the analysis of the current state (possible world or time point), but also the analysis of how knowledge changes through time.

In order to reason over time and represent knowledge evolution, we combine Temporal Logic Programming [29] and the knowledge operator K_i into a Connectionist Temporal Logic of Knowledge (*CTLK*). The implementation of K_i is analogous to that of $\Box$; we treat K_i as a universal modality as done in [15].

Definition 4. (Connectionist Temporal Logic) *The language of* CTLK *contains:*

1. A set $\{p, q, r, ...\}$ *of primitive propositions;*

2. A set of agents $\mathcal{A} = \{1, ..., n\}$*;*

3. A set of connectives K_i $(i \in \mathcal{A})$*, where* $K_i p$ *reads* agent i knows *p;*

4. The temporal operator $\bigcirc$ *(next time); and*

5. A set of extended modal logic clauses of the form $t : ML_{n+1} \leftarrow ML_1, ..., ML_n$*, where* t *is a label representing a discrete time point in which the associated clause holds,* $M \in \{\Box, \Diamond\}$*, and* L_j $(1 \leq j \leq n+1)$ *is a literal.*

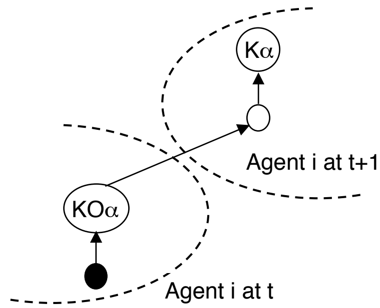

Fig. 2. Temporal reasoning in neural networks

We consider the case of a linear flow of time. As a result, the semantics of *CTLK* requires that we build models in which possible states form a linear temporal relationship. Moreover, to each timepoint, we associate the set of formulas holding at that point by a valuation map. The language used in the example below represents linear temporal knowledge dealing with both past and future as in [12,20]. Therefore, we need to extend the syntax of the above formalisation. The unary past operators ●, ■ and ♦ are respectively defined as *previous time*, *always in the past* and *sometime in the past*. Future time operators ○, □ and ◊ are also defined. The binary $\mathbb{S}$ and $\mathbb{Z}$ operators (*since* and "*zince*") denote that a proposition has been *true* since the occurrence of another, but $\alpha\mathbb{Z}\beta$ also allows the case where α has always occurred. The $\mathbb{U}$ (*until*) and $\mathbb{W}$ (*unless*) operators are defined mirroring $\mathbb{S}$ and $\mathbb{Z}$, in the future time.

Definition 5. *(Extended Temporal Formulas) An atom is inductively defined as follows: (i) If p is a propositional variable, then p is an atom; (ii) If* α *and* β *are atoms, then* ●α, ■α, ♦α, $\alpha\mathbb{S}\beta$ *and* $\alpha\mathbb{Z}\beta$ *are also atoms; (iii) If* α *and* β *are atoms, then* ○α, □α, ◊α, $\alpha\mathbb{U}\beta$ *and* $\alpha\mathbb{W}\beta$ *are also atoms.*

2.2 Temporal Synchronisation and Learning in Neural-Symbolic Systems

In [12] we have shown that temporal synchronisation and learning can be effectively comuted in neural-symbolic systems. A sequential approach to temporal reasoning was used, enabling reasoning and learning about distributed environments in neural networks. Consider the following example [12]. We have applied a neural-symbolic approach to a classical problem of synchronisation in distributed environments, namely, the *Dining Philosophers Problem*, originally from [30]: *n philosophers sit at a table, spending their time thinking and eating. In the centre of the table there is a plate of noodles, and a philosopher needs two forks to eat it. The number of forks on the table is the same as the number of philosophers. One fork is placed between each pair of philosophers and they will only use the forks to their immediate right and left. They never talk to each other, which creates the possibility of deadlock and starvation.*

In [12] we have represented the knowledge of each philosopher (agent) using temporal logic programs, and computed their behaviour in the neural-symbolic model. An agent's policy will model the following behaviour: from the moment that information

$hungry_i$ is known to agent i, she must start trying to get forks (say, from the left) until all forks are in use. When an agent has two forks, she may eat until she is sated (i.e. an external input $sated_i$ is applied). An agent can communicate with the environment through five distinct actions: eat_i, $dropL_i$ and *dropRi*, representing that the agent is returning a fork (left or right) to the table, and $pickL_i$, $pickR_i$, in which the agent tries to allocate the left and the right forks. Since a fork may not be available when an agent tries to pick it, the environment responds to agent i through the information $gotL_i$ and $gotR_i$, denoting that agent i was successfully allocated a fork. The environment randomly sends signals *hungry* and *sated* to the agents, and responds to actions performed by the agents, allowing only one agent to be allocated a particular fork at each time. Agents do not receive any information about their state (being hungry, holding forks, etc); they only receive information about individual events and internally represent their states with respect to these events.

Table 1 illustrates the logic program that represents an agent's behaviour. The upper half of the table describes the original knowledge and the lower half describes knowledge translated by a translation algorithm (given in [12]) into a neural network. In order

$pickL_1 \mathbb{W} gotL_1 \leftarrow hungry_1;\ pickR_1 \mathbb{W} gotR_1 \leftarrow gotL_1$
$eat_1 \mathbb{W} sated_1 \leftarrow gotR_1;\ dropL_1 \leftarrow sated_1$
$dropR_1 \leftarrow sated_1;\ sated_1 \leftarrow sated_1^*$
$GotL_1 \leftarrow GotL_1^*$
$GotR_1 \leftarrow GotR_1^*$
$pickL_1 \mathbb{W} gotL_1 \leftarrow \bullet(pickL_1 \mathbb{W} got_{1,A}), \sim \bullet gotL_1$
$pickL_1 \leftarrow pickL_1 \mathbb{W} gotL_1, \sim gotL_1$
$pickR_1 \mathbb{W} gotR_1 \leftarrow \bullet(pickR_1 \mathbb{W} gotR_1), \sim \bullet gotR_1$
$pickR_1 \leftarrow pickR_1 \mathbb{W} gotR_1, \sim gotR_1$
$eat_1 \mathbb{W} sated_1 \leftarrow \bullet(eat_1 \mathbb{W} sated_1), \sim \bullet sated_1$
$eat_1 \leftarrow eat_1 \mathbb{W} sated_1, \sim sated_1$

Table 1. An agent's temporal knowledge representation

to analyse the learning capacity of the networks representing each agent, we have given each agent the necessary information so that a supervised learning algorithm could be used. Such information is the action the agent executes at each timepoint, according to the default policy, and the agent's state of affairs (such state is stored in the environment). Three different configurations have been used in our experiments. The behaviour of "fully knowledgeable" agents is represented in a network generated by the translation of all rules in Table 1. This generates a network with layers containing, respectively, thirteen, fourteen, and eleven neurons. "Partly knowledgeable" agents are represented by networks generated from the lower part of Table 1, and inserting eight additional hidden neurons to allow for learning of the other rules with the same number of neurons in the hidden layer (fourteen). All the connections to and from these new neurons have been randomly initialized. Finally, agents with no knowledge had all connections randomly set.

Two learning approaches were considered. First, offline learning was implemented, where the agent only receives information from the environment, and her actions do not change the environment. Next, we have carried out online learning, with an agent acting over the environment during the learning experiments. We have used an environment with three agents, where two of them are fully knowledgeable. We have run three different experiments, varying the knowledge level of the remaining agent. These learning experiments have shown that an effective cognitive computational model for integrated temporal learning and reasoning can be built. The experiments have indicated that temporal knowledge can be successfully represented, computed and learned by connectionist models, and they has illustrated the capabilities of the model's learning dimension. In particular, the learning experiments corroborate the benefits of using symbolic background knowledge in the form of temporal logic rules: agents that make better use of background knowledge had more effective learning performance. Detailed results are reported in [12].

The model described above copes with reasoning, learning and synchronisation in a distributed, multi-agent systems with non-trivial behaviour. However, we believe that neural-symbolic computation can contribute to computer science if we are able to achieve several challenges. These challenges are related to mainstream computing methodology and thought and thus would give neural-symbolic computation a high stand in the scientific community. Next, we proceed to the challenges ahead of neural-symbolic computation.

3 Challenges for Neural-Symbolic Systems

In this section we raise several challenges ahead of neural-symbolic computation. They can also be seen as challenges for computer science, since positive responses would lead to a number of implications. Such implications could impact the great challenges of computing research as described above.

3.1 Challenge 1: To Axiomatise Connectionist Non-Classical Logics

Logicians axiomatise logical systems to present subject-matters as formal and coherent theories, allowing for the deduction of all propositions of a system from a well-defined set of initial assumptions [31]. Axiomatisations are used to understand proofs from initial assumptions, to allow the analysis of the complexity and structure of proofs. Computer scientists axiomatise logic-based theories to better understand the computational power of their theories, their limitations and relationships with the limits of computing. Notwithstanding several results on representing fragments of logical systems within neural networks, and computing such fragments, we have not (yet) presented a systematic axiomatisation of classical and connectionist non-classical systems. The proposed axiomatisation would have to take into consideration several characteristics of neural models, including uncertainty, conditional reasoning (i.e. what kind of implication $A \rightarrow B$ between formulae A and B we have in neural networks. Is it classical, conditional [32], causal [33] or probabilistic implication [34]). What kind of negation do we have in neural networks? Intuitionistic as in [10,35], classical, or negation as failure? The forthcoming book [5] promises to shed some light on this, but research is still needed, as pointed out in [36]. This would be of great benefit to the community, as we would be able to understand the logical and computational limitations and expressiveness of logic-based connectionist systems. This is a hard challenge. Axiomatising a theory of *connectionist logics* which in principle integrate elements of symbolic logic (rigid and exact, well-defined over a long history of mathematical rigour) and neural networks (based on statistical, dynamic and probabilistic models) may demand the development of new foundations, which might be themselves hybrid combinations of mathematical techniques. Since we are dealing with the axiomatisation of a form of computational logic, complexity studies regarding the underlying proposed connectionist logical systems are also welcome.

3.2 Challenge 2: To Show that the Benchmark Scenarios are Effectively Tackled by Neural-Symbolic Computation

In order to respond to this challenge, we might firstly have to axiomatise connectionist classical and non-classical logics. This would allow the neural-symbolic community to establish principled comparisons in terms of knowledge representation and (descriptive) complexity to classical and non-classical logics. In particular, in order to show that the standard (multi-agent or probabilistic) knowledge representation scenarios presented in e.g. [15] are effectively solved by neural-symbolic systems, we need to understand how far neural-symbolic systems go as knowledge representation systems. Related to this challenge is ontology learning and representation in the semantic web [37] or in domains where temporal and epistemic dimensions are of relevance, such as cognitive modelling [38].

3.3 Challenge 3: To Provide a Semantic Foundation for Neural-Symbolic Computation.

Although several sound translations have been done from fragments of logic programming to connectionist models, in a principled way, we do not have at the moment a well-defined *denotational* semantics-style of neural-symbolic computation. I believe that the challenge can draw inspiration from the challenge the late Christopher Strachey confronted in the 1960's when trying to use the λ-calculus as a semantic foundation of programming languages and systems. It is well-known in the history of computer science that Dana Scott came to the rescue and warned the programming languages community that the λ-calculus did not have a model at that time. He then developed such model and alongside Strachey founded the area of denotational semantics [39,40]

3.4 Challenge 4: Model Checking Cognitive Systems

Model checking [14] has been probably the most successful non-classical logic-based technology of the last 30 years. It is founded on temporal logics and has been recognised as an *essential* technology in hardware design. Temporal logics (a form of non-classical logics [13]) have had so much success in computer science that Amir Pnueli was awarded the ACM Turing Award in 1996 for laying its computational foundations [41], and Edmund Clarke, E. Allen Emerson and Joseph Sifakis - pioneers of its most prominent application (model checking) - have been awarded the ACM Turing Award in 2007. As non-classical reasoning researchers we strongly believe that model checking connectionist models may lead to the successful development of applied connectionist systems. Recently, model checking has been useful in multi-agent systems applications [42]. We conjecture that model checking connectionist systems may offer us principles to understand the how such models work, leading to a better understanding of the neural computation process and providing sound computational models.

3.5 Challenge 5: To Understand the Relationship Between Biological and Computational Connectionist Models

Most results the neural-symbolic community has published are related to the expressiveness, learnability and computational power of artificial neural networks as computational systems. However, there is no proof of the correspondence between artificial neural network models and biological models. Even though our neural-symbolic models are biologically motivated, we have not produced definite evidence that they correspond to natural systems. However, this is not a requirement. What we aim is to produce effective neural-symbolic models which are useful in real world applications, and offer a sound theory of *computational* cognitive systems. Such systems may correspond to natural models, but this is not required. Computer science is about building effective computational models, that can be of benefit to humankind, be they biologically motivated, inspired or just mathematical abstractions with no relation to biological constructions. [43] offers an interesting analysis about the recent trends on biologically inspired computing, suggesting that one should look not only at biological models, but to unconventional (non-classical) and novel paradigms.

4 Myths and Achievements of Neural-Symbolic Computation

In this section I present some controversial issues about the limitations of neural-symbolic computation. However, I also present evidence that several results have weaken these myths over the last decades. [1]

4.1 Myth 1: The Propositional Fixation of Neural-Symbolic Computation

In 1988, McCarthy raised this issue on the note *Epistemological challenges for connectionism* [45], a published commentary on Smolensky's *On the proper treatment of connectionism* [46]. Propositional fixation assumes that neural networks cannot go beyond propositional logic. Several researchers have now addressed this issue, and have indicated that this is not necessarily the case. Recently, Garcez, Lamb and Gabbay, in several publications have show that modal and temporal logics (which correspond to the two-variable fragment of first-order logic [47]) can be effectively represented in neural

[1] I draw inspiration from a paper by Hall [44] about the seven myths of formal methods.

networks [9,10,11,48,49,50]. Bader, Hitzler, Hölldobler, and Witzel have proved that neural networks can generate models of first-order logic programs [51,52,53]. Several researchers have also shown that neural-symbolic systems can learn or compute relations and fragments of first-order logics, see e.g. [54,55,56,57]. These results provide evidence that the field is maturing in terms of foundational results.

4.2 Myth 2: Neural-Symbolic Systems do Not Work in Practice

This myth has been challenged. Even though neural-symbolic systems are still in its infancy, they have proved effective learning systems in real world applications. Such applications have included power systems fault diagnosis, computational biology and DNA sequence analysis [6,58]. They have also been successfully used as model allowing for integrated learning and computation of arguments, including circular arguments [59]. Moreover, neural-symbolic systems have allowed for full solutions of standard testbed for distributed knowledge representation and reasoning about uncertainty, time and knowledge. For instance, a full solution for the well-known muddy children puzzle [15] has recently been shown. Connectionist Temporal Logic of Knowledge *(CTLK)* [9,25,35] an extension of Connectionist Modal Logics *(CML)* [11,50] has enabled a full solution of the testbed, integrating a temporal and an epistemic dimension to the problem, leading to a full solution of the problem (although one may claim that the full solution is simple such solution had not been published until 2003). To the best of our knowledge such solution has been firstly shown by Garcez and Lamb in [25]. The proposed solution also allows for learning in the system, rendering a computational cognitive model integrating knowledge representation, reasoning and learning within neural networks with possible applications in the domains of multi-agent systems and cognitive modelling [9,5]. It would be interesting, however, to analyse the reasoning and learning capabilities of neural-symbolic systems in economic multi-agent scenarios such as the minority game [60,61].

4.3 Myth 3: Neural-Symbolic Systems are Not Biologically Plausible

This myth is strongly related to the challenge 3.5 above. Some neural-symbolic systems draw inspiration from biological models. However, that does not imply that they correspond to biological, natural models of the brain and mind. What we aim at building are effective computational cognitive models integrating the connectionist and symbolic paradigms of artificial intelligence. Ideally, such systems will show their effectiveness in practical applications. We do not necessarily need to build models that correspond to biological models of a (human) brain. Teuscher [43] raises a relevant issue, providing evidence and questioning the recent trend in computing: the need for biologically inspired methods and abstractions. He reminds us that *"Trying to cope or mimic life or lifelike behaviour in all scientific disciplines has generally produced disillusion after high initial hopes and hype."* Teuscher quotes Conrad [62], one of the pioneers of bio-inspired computational models: *"...no system can be at once highly structurally programmable, evolutionary efficient and computationally efficient"* [62]. In order to justify the work on neural-symbolic computing, one does not need to prove that one's model is biologically plausible; rather, one has to show that one's model is computationally effective, leading to principled scientific results which in turn may lead to successful technologies.

5 Conclusions

The analysis of the recent grand challenges for computing research [17] provide further evidence that neural-symbolic computation may be a promising research endeavour. The challenges are certainly not easy to achieve and demand multi-disciplinary research, including research on non-classical computational models. Neural-symbolic computation demands knowledge of seemingly incompatible areas, but which can benefit from each other.

Recently, non-classical logics have been shown to be prone to integration with neural networks. The work reported in [5,9,10,12,20] has indicated that modal, temporal and (fragments of) non-classical logics knowledge can be successfully represented, computed and learned by connectionist models. [51,52,53] have show how to compute a class of first-order programs in connectionist models. These results are relevant in terms of expressiveness of neural-symbolic computation and may lead to further applications in real-world scenarios.

Computer science has had a successful history. We believe that the road ahead is challenging and exciting. The grand challenges suggest that we should look at several areas to make due progress. Neural-symbolic computation could contribute towards reaching them. Biology, physics, economics, cognitive and nano sciences are providing us with new data (and opportunities) to set forth on new computational models that may lead to ground breaking computational or scientific paradigms.

Acknowledgments

I am grateful to Artur d'Avila Garcez, Dov Gabbay, Jehoshua (Shuki) Bruck, Matthew Cook, Pascal Hitzler, Steffen Hölldobler, Sebastian Bader, Rafael Borges, Gerson Zaverucha, Kristian Kersting, Luc De Raedt and the participants of Dagstuhl Seminar 08041 *Recurrent Neural Networks: Models, Capacities, and Applications* for several illuminating conversations about neural-symbolic computation. This work is partly funded by the Brazilian Research Council CNPq.

References

1. Touretzky, D., Hinton, G.: Symbols among neurons. In: Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-85, Morgan Kaufmann (1985) 238–243
2. Page, M.: Connectionist modelling in psychology: A localist manifesto. Behavioral and Brain Sciences **23** (2000) 443–467
3. Smolensky, P., Legendre, G.: The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. MIT Press, Cambridge, MA (2006)
4. Valiant, L.: A neuroidal architecture for cognitive computation. Journal of the ACM **47** (2000) 854–882

5. d'Avila Garcez, A., Lamb, L.C., Gabbay, D.: Neural-Symbolic Cognitive Reasoning. Springer (2008) forthcoming.
6. d'Avila Garcez, A., Broda, K., Gabbay, D.: Neural-Symbolic Learning Systems: Foundations and Applications. Perspectives in Neural Computing. Springer (2002)
7. Shastri, L.: Advances in SHRUTI: a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. Applied Intelligence Journal, Special Issue on Neural Networks and Structured Knowledge **11** (1999) 79–108
8. Hölldobler, S., Kalinke, Y.: Toward a new massively parallel computational model for logic programming. In: Proceedings of the Workshop on Combining Symbolic and Connectionist Processing, ECAI 1994. (1994) 68–77
9. d'Avila Garcez, A., Lamb, L.C.: A connectionist computational model for epistemic and temporal reasoning. Neural Computation **18** (2006) 1711–1738
10. d'Avila Garcez, A., Lamb, L.C., Gabbay, D.: Connectionist computations of intuitionistic reasoning. Theoretical Computer Science **358** (2006) 34–55
11. d'Avila Garcez, A., Lamb, L.C., Gabbay, D.: Connectionist modal logic: Representing modalities in neural networks. Theoretical Computer Science **371** (2007) 34–53
12. Lamb, L.C., Borges, R., d'Avila Garcez, A.: A connectionist cognitive model for temporal synchronisation and learning. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence AAAI 2007, AAAI Press (2007) 827–832
13. Broda, K., Gabbay, D., Lamb, L.C., Russo, A.: Compiled Labelled Deductive Systems: A Uniform Presentation of Non-classical Logics. Studies in Logic and Computation. Research Studies Press/Institute of Physics Publishing, Baldock, UK, Philadelphia, PA (2004)
14. Clarke, E., Schlingloff, H.: Model checking. In Robinson, J., Voronkov, A., eds.: Handbook of Automated Reasoning. Volume II. Elsevier (2001) 1635–1790
15. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press (1995)
16. Valiant, L.: Three problems in computer science. Journal of the ACM **50** (2003) 96–99
17. Hoare, C., Milner, A., eds.: Grand Challenges in Computing Research. British Computer Society (2004)
18. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1999)
19. Gabbay, D.: Elementary Logics: a Procedural Perspective. Prentice Hall, London (1998)
20. Borges, R., Lamb, L.C., d'Avila Garcez, A.: Reasoning and learning about past temporal knowledge in connectionist models. In: Proceedings of the Twentieth International Joint Conference on Neural Networks IJCNN 2007. (2007) 1488–1493
21. Elman, J.: Finding structure in time. Cognitive Science **14** (1990) 179–211
22. Fischer, M., Gabbay, D., Vila, L., eds.: Handbook of Temporal Reasoning in Artificial Intelligence. Elsevier (2005)
23. Halpern, J., Harper, R., Immerman, N., Kolaitis, P., Vardi, M., Vianu, V.: On the unusual effectiveness of logic in computer science. Bulletin of Symbolic Logic **7** (2001) 213–236
24. Gabbay, D., Kurucz, A., Wolter, F., Zakharyaschev, M.: Many-dimensional Modal Logics: Theory and Applications. Volume 148 of Studies in Logic and the Foundations of Mathematics. Elsevier Science (2003)
25. d'Avila Garcez, A., Lamb, L.C.: Reasoning about time and knowledge in neural-symbolic learning systems. In Thrun, S., Saul, L., Schoelkopf, B., eds.: Advances in Neural Information Processing Systems 16. Proceedings of NIPS 2003, MIT Press (2004) 921–928
26. d'Avila Garcez, A., Zaverucha, G.: The connectionist inductive learning and logic programming system. Applied Intelligence Journal, Special Issue on Neural Networks and Structured Knowledge **11** (1999) 59–77
27. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation. In Rumelhart, D., McClelland, J., eds.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1. MIT Press (1986) 318–362

28. Halpern, J., van der Meyden, R., Vardi, M.: Complete axiomatizations for reasoning about knowledge and time. SIAM Journal on Computing **33** (2004) 674–703
29. Orgun, M., Ma, W.: An overview of temporal and modal logic programming. In: Proceedings of the International Conference on Temporal Logic ICTL'94. Volume 827 of Lecture Notes in Artificial Intelligence., Springer (1994) 445–479
30. Dijkstra, E.W.: Hierarchical ordering of sequential processes. Acta Inf. **1** (1971) 115–138
31. McCall, S.: Axiomatic method. In Honderich, T., ed.: The Oxford Companion to Philosophy. Oxford University Press (1995) 72
32. Broda, K., Gabbay, D., Lamb, L.C., Russo, A.: Labelled natural deduction for conditional logics of normality. Logic Journal of the IGPL **10** (2002) 123–163
33. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press (2000)
34. Halpern, J.: Reasoning About Uncertainty. MIT Press (2003)
35. d'Avila Garcez, A., Lamb, L.C., Gabbay, D.: A connectionist model for constructive modal reasoning. In: Advances in Neural Information Processing Systems 18. Proceedings of NIPS 2005, MIT Press (2006) 403–410
36. d'Avila Garcez, A., Lamb, L.C.: Neural-symbolic systems and the case for non-classical reasoning. In Artëmov, S., Barringer, H., d'Avila Garcez, A., Lamb, L.C., Woods, J., eds.: We Will Show Them! Essays in Honour of Dov Gabbay. College Publications, International Federation for Computational Logic (2005) 469–488
37. Bader, S., Hitzler, P.: Dimensions of neural-symbolic integration - a structured survey. In Artëmov, S., Barringer, H., d'Avila Garcez, A., Lamb, L.C., Woods, J., eds.: We Will Show Them! Essays in Honour of Dov Gabbay. College Publications, International Federation for Computational Logic (2005) 167–194
38. Mastella, L., Abel, M., Lamb, L.C., Ros, L.F.D.: Cognitive modelling of event ordering reasoning in imagistic domains. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05). (2005) 528–533
39. Scott, D.: Data types as lattices. SIAM J. Comput. **5** (1976) 522–587
40. Tennent, R.: Denotational semantics. In Abramsky, S., Gabbay, D., Maibaum, T., eds.: Handbook of Logic in Computer Science. Volume 3: Semantic Structures. Clarendom Press, Oxford (1994) 169–322
41. Pnueli, A.: The temporal logic of programs. In: Proceedings of 18th IEEE Annual Symposium on Foundations of Computer Science. (1977) 46–57
42. Bordini, R.H., Fisher, M., Visser, W., Wooldridge, M.: Verifying multi-agent programs by model checking. Autonomous Agents and Multi-Agent Systems **12** (2006) 239–256
43. Teuscher, C.: Biologically uninspired computer science. Commun. ACM **49** (2006) 27–29
44. Hall, A.: Seven myths of formal methods. IEEE Software **7** (1990) 11–19
45. McCarthy, J.: Epistemological challenges for connectionism. Behavioral and Brain Sciences **11** (1988) 44
46. Smolensky, P.: On the proper treatment of connectionism. Behavioral and Brain Sciences **44** (1988) 1–74
47. Vardi, M.: Why is modal logic so robustly decidable. In Immerman, N., Kolaitis, P., eds.: Descriptive Complexity and Finite Models. Volume 31 of Discrete Mathematics and Theoretical Computer Science. DIMACS (1997) 149–184
48. d'Avila Garcez, A., Lamb, L.C., Gabbay, D.: A connectionist inductive learning system for modal logic programming. In: Proceedings of the 9th International Conference on Neural Information Processing ICONIP'02, Singapore, IEEE Press (2002) 1992–1997
49. d'Avila Garcez, A., Lamb, L.C., Broda, K., Gabbay, D.: Distributed knowledge representation in neural-symbolic learning systems: A case study. In: Proceedings of AAAI International FLAIRS Conference, Florida, USA (2003) 271–275

50. d'Avila Garcez, A., Lamb, L.C., Broda, K., Gabbay, D.: Applying connectionist modal logics to distributed knowledge representation problems. International Journal on Artificial Intelligence Tools **13** (2004) 115–139
51. Bader, S., Hitzler, P., Holldobler, S., Witzel, A.: A fully connectionist model generator for covered first-order logic programs. In: Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-07, Hyderabad, India, AAAI Press (2007) 666–671
52. Bader, S., Hitzler, P., Hölldobler, S., Witzel, A.: The core method: Connectionist model generation for first-order logic programs. In Hammer, B., Hitzler, P., eds.: Perspectives of Neural-Symbolic Integration. Springer (2007)
53. S.Bader, Hitzler, P., Hölldobler, S.: Connectionist model generation: A first-order approach. Neurocomputing (2008)
54. Basilio, R., Zaverucha, G., Barbosa, V.: Learning logic programs with neural networks. In: Inductive Logic Programming. Springer LNAI 2157 (2001) 15–26
55. Browne, A., Sun, R.: Connectionist inference models. Neural Networks **14** (2001) 1331–1355
56. Hitzler, P., Holldobler, S., Seda, A.K.: Logic programs and connectionist networks. Journal of Applied Logic **2** (2004) 245–272 Special Issue on Neural-Symbolic Systems.
57. Komendantskaya, E.: First-order deduction in neural networks. In: Proceedings of the 1st Conference on Language and Automata Theory and Applications LATA'07, Tarragona, Spain (2007) 307–318
58. Towell, G., Shavlik, J.: Knowledge-based artificial neural networks. Artificial Intelligence **70** (1994) 119–165
59. d'Avila Garcez, A., Gabbay, D., Lamb, L.C.: Value-based argumentation frameworks as neural-symbolic learning systems. Journal of Logic and Computation **15** (2005) 1041–1058
60. Araújo, R.M., Lamb, L.C.: Towards understanding the role of learning models in the dynamics of the minority game. In: 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 15-17 November 2004, Boca Raton, FL, USA. (2004) 727–731
61. Araújo, R.M., Lamb, L.C.: An information-theoretic analysis of memory bounds in a distributed resource allocation mechanism. In: IJCAI-07, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, AAAI Press (2007) 212–217
62. Conrad, M.: The brain-machine disanalogy. BioSystems **22** (1989) 197–213