A formal framework for the symbol grounding problem

Benjamin Johnston and Mary-Anne Williams
University of Technology, Sydney
Broadway, Ultimo 2007, Australia
johnston@it.uts.edu.au

Abstract

A great deal of contention can be found within the published literature on grounding and the symbol grounding problem, much of it motivated by appeals to intuition and unfalsifiable claims. We seek to define a formal framework of representation grounding that is independent of any particular opinion, but that promotes classification and comparison. To this end, we identify a set of fundamental concepts and then formalize a hierarchy of six *representational system classes* that correspond to different perspectives on the representational requirements for intelligence, describing a spectrum of systems built on representations that range from symbolic through iconic to distributed and unconstrained. This framework offers utility not only in enriching our understanding of symbol grounding and the literature, but also in exposing crucial assumptions to be explored by the research community.

Introduction

The symbol grounding problem [1] represents a long standing (and often misunderstood) point of contention within the Artificial Intelligence community (*e.g.*, [2,3,4]) and continues to concern researchers exploring Artificial General Intelligence[1] (AGI). The problem, as it is classically conceived, concerns the nature of the abstract symbols used in computer systems, how they may be seen as having a real-world meaning, and how that meaning can be made intrinsic to a system.

Consider the problem of a knowledge base designed to reason about the possible security threats posed by terrorist organizations. The system may have an internal symbol ***nuclear_weapon*** that we as humans understand as representing the real-world concept of nuclear weapons. However, to a purely symbolic computer system, the symbol ***nuclear_weapon*** is nothing more than an arbitrary token that has no more *intrinsic* meaning than any other symbol in a computer, say ***waffle_blah***. The symbol grounding problem concerns the question of how the meaning of symbols can be embedded into a system, and grounding is said to be the process of ensuring that these abstract symbols have meaning.

While there is the philosophical question of whether a machine really can have intrinsically ground symbols (indeed, this is the motivation for Searle's Chinese Room argument), the symbol grounding problem poses the more practical question of whether a purely symbolic system *could* solve problems that apparently demand deep intelligence and understanding. Is it possible, for example, for a purely symbolic system to understand the nuanced relationship between a nuclear weapon and a dirty bomb, or to explain that a zebra is like a horse with stripes, or even to determine what other letter of the alphabet an upside-down 'M' resembles; without requiring the specific answers to these questions to be explicitly given to the system in advance?

Our objective is not to argue a specific position on the symbol grounding problem, but rather, to provide the first formal framework for the symbol grounding problem. Our approach offers standardized terminology for discussing assumptions and ideas and also raises important new research questions. In particular, we aim to allow an AI researcher to express their assumptions regarding the symbol grounding problem as elegantly as a computer scientist might use computational complexity classes to motivate the need for heuristics in preference to brute force search. Furthermore, we hope that our framework will direct future arguments about grounding away from appeals to intuition and toward a greater emphasis on formalizable and falsifiable claims.

In this paper, we will first define symbol systems and representations, and review the problem of grounding such symbols and representations. We then introduce our formal notation and explore 'semantic interpretability'. These serve as preliminaries for the primary contribution of this paper: the definition of our representational system classes and their correspondences with the published literature. We then conclude with some observations about the representational classes and their relevance, including a brief overview of future research directions.

Symbol Systems and Symbol Grounding

Harnad [1] first proposed the symbol grounding problem as a question concerning semantics: "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads?" While Harnad's original formulation of the problem is largely philosophical, his motivation is clearly of a pragmatic nature: he implicitly assumes that the property of 'intrinsic interpretability' is crucial for intelligence. We therefore prefer to reformulate the symbol grounding problem in more straightforward terms: "Is it possible to use formal symbolic reasoning to create a system that is intelligent?" Of course, this reformulation presupposes a definition of what it means to be or to appear intelligent—but the reformulation is an improvement in the sense that it brings us closer to something objectively measurable.

Harnad saw the mechanisms of an isolated formal symbol system as analogous to attempting to learn Chinese as a second language from a Chinese-Chinese dictionary. Even though characters and words are defined in terms of other characters, reading the dictionary would amount to nothing more than a 'merry-go-round' passing endlessly from one symbol string (a term) to another (its definition); never coming to a 'halt on what anything meant' [1]. He therefore argued that since symbols only refer to other symbols in a symbol system, there is no place where the symbols themselves are given meaning. The consequence of this is that it is impossible for a formal symbol system to distinguish between any two symbols except using knowledge that has been explicitly provided in symbolic form. This, in the view of Harnad, limits the comprehension and capabilities of a symbolic system in the same way that a non-speaker armed with a Chinese-Chinese dictionary may manage to utter random syntactically correct sentences, but would exhibit extremely poor performance in understanding real conversation.

1 For example, consider recent debate on an AGI email list: www.mail-archive.com/agi@v2.listbox.com/msg07857.html

Of course, Harnad's argument is not universally accepted by computer scientists. One objective of this paper is to explore the problem, so we will use our representational system classes to outline and classify the diversity of opinions later in this paper.

The symbol grounding problem concerns symbolic systems, but what *is* a formal symbol system? Harnad [1] provides eight criteria:

> A symbol system is: (1) a set of arbitrary "physical tokens" (scratches on paper, holes on a tape, events in a digital computer, *etc.*) that are (2) manipulated on the basis of "explicit rules" that are (3) likewise physical tokens and strings of tokens. The rule-governed symbol-token manipulation is based (4) purely on the shape of the symbol tokens (not their 'meaning'), *i.e.*, it is purely syntactic, and consists of (5) 'rulefully combining' and recombining symbol tokens. There are (6) primitive atomic symbol tokens and (7) composite symbol-token strings. The entire system and all its parts—the atomic tokens, the composite tokens, the syntactic manipulations both actual and possible and the rules—are all (8) 'semantically interpretable': the syntax can be systematically assigned a meaning (*e.g.*, as standing for objects, as describing states of affairs).

It is interesting to note that criteria 1–7 can be used to describe any universal or Turing-complete language. However, Harnad is not claiming that meaning or intelligence is incomputable—he proposes his own computational framework for solving the symbol grounding problem. It is the 8th criterion of a formal symbol system that defines the essential point of difference between his conception of symbolic systems and representations used in arbitrary computation. The requirement of interpretability in criterion 8 is intended to capture the essence of familiar symbolic systems such as logical theorem proving and rule based systems, and to exclude highly distributed, 'holographic' or connectionist representations that do not behave in a symbolic manner (even though they operate on a digital computer). For example, while connectionist approaches to building intelligent systems can be framed so as to meet criteria 1–7, connectionist methods do not typically allow for a systematic assignment of real-world interpretation to hidden layer neurons (*i.e.*, hidden neurons learn with a bias towards performance rather than any particular 'meaning'), they therefore do not satisfy criterion 8, and are therefore not (directly) subject to Harnad's criticism.

Harnad's 8th criteria for a symbol system is essential to understanding the symbol grounding problem. We will consider how changes to this criterion (whether in its phrasing or in comprehension) influence an understanding of the symbol grounding problem. Specifically, we further generalize the symbol grounding problem as follows: "What kinds of reasoning can be performed by systems constrained by different representational criteria?" In this formulation, we can regard different research groups as working from both different assumptions of what constitutes intelligence (*i.e.*, the kind of reasoning) and different representational constraints.

Notational Preliminaries

Problems We begin by assuming the existence of a set, $\mathcal{P}$, that contains all problems that may be posed to an intelligent system. Each problem is a declarative sentence (in some formal language) about the world and an agent is said to be able to solve a problem if its determination of the truth of that statement matches the 'real world' truth. A problem might be a query posed by a person to a theorem prover or a question-answering system, or it might represent an encoding of the inputs and outputs of a robotic system (*i.e.*, "given certain sensory inputs x, the appropriate behavior at time t, is to perform action a"). While a real life agent may encounter complex situations and exhibit nuanced performance that is neither success nor failure, we assume that these situations can be analyzed as a large set of binary sub-problems, and the agent's performance is a measure of how many of the sub-problems can be successfully solved. If an agent, f, believes statement p, we denote[2] this as $f \vdash p$. If an agent, f, can correctly solve a problem, $p : \mathcal{P}$, then we denote this as $f \sim p$.

Problem Sets We define a *problem-set* as a set of problems: an object of type $\mathbb{P}(\mathcal{P})$. An agent, f, can solve a problem-set, $ps : \mathbb{P}(\mathcal{P})$, if it can solve all problems within that set. This is denoted $f \sim ps$, and we have $f \sim ps \Leftrightarrow \forall\, p : ps \bullet f \sim p$.

Intelligence We use problem-sets to define intelligence. In this paper, we do not choose any particular definition of intelligence: we assume a range of definitions so that we can not only denote the largely subjective and unformalizable 'I'll know it when I see it' attitude of many AI researchers towards intelligence, but also offer scope for formal definitions of intelligence. As such, a given definition, I, of intelligence is a set of problem-sets; *i.e.*, $I : \mathbb{P}(\mathbb{P}(\mathcal{P}))$. An agent, f, is considered intelligent with respect to a definition of intelligence I, if it can solve *all* problems in *some* problem-set. This is denoted $f \sim I$, and we therefore have $f \sim I \Leftrightarrow \exists\, ps : I \mid \forall\, p : ps \bullet f \sim p$.

This approach to representing definitions of intelligence, admits many definitions beyond that of simple IQ tests or fixed checklists of skills. Consider that how one may regard Albert Einstein and Elvis Presley as both possessing exceptional intelligence, even though their genius is expressed in different ways. Their distinct skills correspond to different problem-sets within our common interpretation of 'genius'.

We allow many definitions of intelligence; for example:

- A set $I_{\text{Harnad}} : \mathbb{P}(\mathbb{P}(\mathcal{P}))$ for those systems that Harnad would regard as exhibiting intelligence,
- A set $I_{\text{IQ=100}} : \mathbb{P}(\mathbb{P}(\mathcal{P}))$ to denote sets of problems that a person of average Human intelligence would be able to solve,
- A set $I_{\text{Market}} : \mathbb{P}(\mathbb{P}(\mathcal{P}))$ of buying decisions that a trading agent would need to solve successfully in order to exceed break-even on a market over a particular time interval,
- Given a formal definition of intelligence with a precise threshold, we may have a set $I_{\text{Formal}} : \mathbb{P}(\mathbb{P}(\mathcal{P}))$ denoting those problem-sets that a formally intelligent system could solve.

Formal Systems We define $\mathcal{F}$ as the set of all finite formal systems that satisfy criteria 1–7 of Harnad and are finitely realizable[3]. We define $\mathcal{T}$ as the universal set of symbols, and assume

2 Throughout this paper, we use the Z notation per international standard ISO/IEC 13568:2002. We treat typing as equivalent to membership in a set and denote this with a colon. The power-set operator is denoted as $\mathbb{P}$, the set of natural numbers as $\mathbb{N}$, the set of partial functions from A to B as $A \nrightarrow B$, and the domain and range of a function, f, as dom(f) and ran(f) respectively.

3 By finitely realizable, we mean that the systems's representations and computational processes that can be described in finite space on a Turing machine by a finite agent within the universe, and that the corresponding computations of the system in solv-

that each formal system comprises a set of fixed symbolic transition rules and a dynamic execution state. We assume (without loss of generality) that an execution trace of a formal system, f, on a problem, p, is comprised of a two-dimensional grid of symbols. We denote this as $\boldsymbol{t}(f, p) : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T}$. The two axes of the grid correspond to the state of the system (analogous to a CPU clock) and the position or address of each symbol. The value of each grid-cell is the single symbol stored in the 'address' in that state (be it a byte value stored in RAM, a mark on a tape, a neural network weight, or an 'atom' in a logical programming language).

Representational Units In most non-trivial systems, the individual symbols do not convey meaning alone; the intelligent behavior stems from the manipulation of entire subsequences or subsets of the system's symbolic state. Furthermore, such intelligent behavior stems from the manipulation of only certain possible subsets: those subsets of the system state that correspond to legal guards and arguments of the system's transition rules. For example, if the numbers 1, 25 and 334 are denoted as the fixed-length sequence of digits <001025334> at a given step of the system trace, then a system's transition rules might only accept sequences aligned to three-digit boundaries (*i.e.*, 001, 025 and 334, but neither 00102 nor 253). For a given formal symbolic system $f : \mathcal{F}$, and problem p : P, we define the set of all representational units, $\boldsymbol{a}(f, p) : \mathbb{P}(\mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T})$ as the set of all subsets of the system trace, $\boldsymbol{t}(f, p)$, that can match part or all of a guard or parameter of a transition rule in f.

Semantic Interpretability

'Semantic interpretability,' the cornerstone of Harnard's 8th criteria for a symbol system, presents a challenge to the formalization of the symbol grounding problem. Indeed, we believe that the philosophical difficulties of the symbol grounding problem lie in the elusiveness of 'semantic interpretability'.

The model-theoretic approach to defining semantic interpretability would be to assume some valuation function, m, that maps from symbols to 'real world' counterparts so that the problems that a formal system believes to be true correspond to truths that follow from their real world valuations. That is, if we assume the existence of a universal set, $\mathcal{U}$, containing the complete universe of actual, possible, conjectured and imaginary objects, actions, categories, relations and concepts in the 'real world,' then given a formal system, f, we may attempt to formalize semantic interpretability in a manner such as the following[4]:

$$\exists\, m : \mathcal{P} \rightarrow \mathcal{U} \mid \forall\, p : \mathcal{P} \bullet f \vdash p \Rightarrow m(p) \models p$$

However, such a definition is clearly not what was intended by Harnad; it merely states that the agent has true beliefs for every problem it can solve. Model theoretic methods do not directly apply because problems of intelligence are already assumed to be statements about the 'real world'. Semantic interpretability of a formal system demands inspection of not only its internal symbols, but also the *use* of the symbols. For example, a system that uses both constructive proof and proof-by-contradiction may use the same symbol to denote a concept and its negation: it the use of the symbol in reasoning that reveals the true meaning of the symbol.

ing a problem occur in finite time.

4 We introduce the semantic entailment operator, $u \models p$, to denote that proposition, p, is (or would be) true in every universe consistent with the set u.

Unfortunately, it is impossible to analyze use without defining a particular computational model (and our goal is to retain a level of abstraction from such particulars). In future works, we intend to explore such philosophical challenges of defining semantic interpretability, especially given symbol use. We propose here a working definition.

Let $\mathcal{SI}$ denote the type of semantic interpretations of representational units $\mathcal{SI} = \mathbb{P}(\mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T}) \rightarrow \mathcal{U}$. Then, given a (model-theoretic) semantic interpretation $m : \mathcal{SI}$, that maps from a set of representational units, $r : \mathbb{P}(\mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T})$, to elements of $\mathcal{U}$; we say that a formal system, f, in solving a problem, p, is semantically interpretable if *syntactic entailment* (*i.e.*, computation) corresponds to *semantic entailment* from the model implied by the conjunction of the semantic mapping of the system's entire execution trace. *i.e.*;

$$\boldsymbol{si}(m, f, p)$$
$$\Leftrightarrow$$
$$(f \vdash p \Leftrightarrow \boldsymbol{t}(f, p) \subseteq (\cup \mathrm{dom}(m)) \wedge \{\, u \mapsto e : m \mid u \subseteq \boldsymbol{t}(f, p) \bullet e \,\} \models p\,)$$

While this working definition ignores the internal use of symbols and may be overly stringent for any particular system, we do not believe it limits the generality of our work. Representational units with different purposes may be expanded to include the neighboring section markers, delimiters, type definitions, annotations or positional information that indicates their purpose: thereby embedding the *use* of a representational unit in the formal system into its surface structure.

The nature and existence of 'real world' concepts, and consequently, the membership of the set $\mathcal{U}$ remains an open question that bears upon the symbol grounding problem and the work we describe here. We have assumed that the 'real world' universe includes concepts such as historical, hypothetical and imaginary entities, as well as attributes, verbs and abstract nouns like *up*, *walk*, *happy* and *beauty*. However, one could trivially 'solve' the symbol grounding problem on a technicality by excessively generalizing $\mathcal{U}$, so that the 'real world' concept of any symbol can be taken as "those situations, entities and environments that stimulate the generation of the symbol". Such contrived entities would seem absurd to a human-observer and are also highly context dependent, so therefore do not correspond to our intuitions of meaningful 'real world' entities that belong in $\mathcal{U}$. The relationship between the nature of $\mathcal{U}$ and symbol grounding is an interesting problem, that we plan to explore in future work.

Representational System Classes

We can now use our notions of and notation for semantic interpretability to formalize the differences between attitudes toward the symbol grounding problem. Our goal is to analyze the space of finite formal systems, $\mathcal{F}$, and categorize these into a hierarchy based on the semantic interpretability of their representations. That is, we assume Harnad's criteria 1–7, and build a hierarchy from specializations and generalizations of Harnad's 8th criteria.

For example, $\mathcal{SIR}$ is one such class intended to denote the set of systems with fully Semantically Interpretable Representations. We can use this class to restate Harnad's thesis that a symbolic system cannot be intelligent as follows:

$$\forall f : \mathcal{SIR} \bullet \neg\, (f \sim I_{\mathrm{Harnad}})$$

Or, if we extend the 'solves' operator to representational classes so that $c \sim i \Leftrightarrow \exists f : c \bullet f \sim i$, then we have Harnad's thesis as:

$$\neg\, \mathcal{SIR} \sim I_{\mathrm{Harnad}}$$

By exploring variations of Harnad's definition of symbolic systems, we have identified a range of representational system classes beyond $\mathcal{SIR}$. In particular, we have identified six representational system classes that appear to capture the philosophical position of many AI researchers, and that form (by their definition) a hierarchy of representational expressiveness. Each class represents a set of formal systems and is therefore of the type $\mathbb{P}(\mathcal{F})$.

The following subsections describe the classes ordered from most restrictive to most general. Each class has been defined so that it subsumes (*i.e.*, is a super-set of) those classes that have been presented before it. The subsumption property can be proven syntactically from the formal definitions in this work, and holds irrespective of the foundational assumptions of this work.

3.1 Context-Free Semantically Interpretable Representation

$\mathcal{CFSIR}$: Every symbol must have a unique semantic interpretation.

$$\mathcal{CFSIR} = \{f : \mathcal{F} \mid \exists\, m : \mathcal{SI} \mid \forall\, p : \mathcal{P} \bullet \textbf{\textit{si}}(m, f, p) \wedge \forall\, r : \mathrm{dom}(m) \bullet \#r = 1\}$$

Systems in $\mathcal{CFSIR}$ are those in which every single symbol has some meaning (given by valuation function, *m*): symbols do not acquire meaning from their context in some larger representational unit such as a sentence or structure.

A system that, for example, uses the symbol ***Mouse*** to represent the common house mouse, but also uses that same symbol in the context of a Disney movie to state that Mickey Mouse is a mouse *could not* be regarded as making use of symbols with universal and unambiguous meanings (consider, for example, posing the system the question of whether a mouse can speak English). In a symbolic system with context-free semantic interpretations, such distinctions would first need to be translated into separate symbols: *e.g.*, ***Natural_Mouse*** and ***Cartoon_Mouse***. Whether complete translations are possible for all symbolic systems remains an open question, and is, in fact, the question of whether $\mathcal{SIR} = \mathcal{CFSIR}$.

Systems in $\mathcal{CFSIR}$ include:

1. Semantic web systems based on RDF: every resource is denoted by a globally unique URL that is intended to capture some unique context-free interpretation. RDF provides no facility to contextualize the truth of an RDF triple without complex reification [5].
2. Traditional database systems: in typical database designs, each record is intended to have a unique and context-free interpretation.
3. Internal representations of industrial robotic systems: every variable in the control system of an industrial robot can be assigned a unique meaning (*e.g.*, joint position, current distance sensor reading, x-coordinate of a recognized widget).

3.2 Semantically Interpretable Representation

$\mathcal{SIR}$: Every representational unit must have a semantic interpretation.

$$\mathcal{SIR} = \{f : F \mid \exists\, m : \mathcal{SI} \mid \forall\, p : \mathcal{P} \bullet \textbf{\textit{si}}(m, f, p) \wedge \mathrm{dom}(\mathrm{m}) \subseteq \textbf{\textit{a}}(f, p)\}$$

The set $\mathcal{SIR}$ corresponds to those systems that match Harnad's original definition of formal symbolic systems. Every representational unit in the system must have a semantic interpretation, and every symbol used by the system belongs to a representational unit.

Systems in $\mathcal{SIR}$ (but not $\mathcal{CFSIR}$) include:

1. John McCarthy's early proposal to incorporate context into formal symbolic systems [6], and related efforts that have arisen from this, such as PLC and MCS [7].
2. The CYC project's symbolic engineering wherein symbols have meaning, and that meaning is given within context spaces [8].

3.3 Iconic and Symbolic Representation

$\mathcal{ISR}$: Representational units may have semantic interpretation. Non-interpretable representational units must be composable as sets that in aggregate have semantic interpretation and resemble their meaning.

$$\mathcal{ISR} = \{f : \mathrm{F} \mid \exists\, m : \mathcal{SI} \mid \forall\, p : \mathcal{P} \bullet \textbf{\textit{si}}(m, f, p) \wedge \textbf{\textit{iconic}}(\mathrm{dom}(m) - \textbf{\textit{a}}(f, p))\}$$

In $\mathcal{ISR}$, individual representational units need not have a semantic interpretation, but may be part of an aggregate that is semantically interpretable as a whole. Such aggregations in $\mathcal{ISR}$ must have a structure that somehow resembles the meaning of their referent (*e.g.*, by projection or analogy)—they must be iconic.

For example, the individual pixels of a high-resolution image could not typically be regarded as having a particular meaning when considered individually, but in aggregate may be understood as denoting the object that they depict. A system with hybrid visual/symbolic representations could refer to its symbolic knowledge to answer factual queries, but use high resolution images to compute answers to queries about nuanced physical traits or to compare the appearances of different people. Iconic representations in some way resemble their meaning: be they low-level resemblances such as images, 3D models and perspective invariant features, or more abstract forms such as graphs representing the social networks in an organization or the functional connections of components.

Precisely what, then, does it mean for a symbol to resemble its meaning? If a system resembles its meaning, then a small representational change should correspond to a small semantic change. That is, for a set of iconic representations, *i*, there should exist a computable representational distance function, ***rdist***, and a semantic distance function (with some real world meaning, and therefore a member of $\mathcal{U}$), ***sdist***, and error limit, ε, such that:

$$\textbf{\textit{iconic}}(i) \Leftrightarrow \forall\, i_1, i_2 : i \bullet |\textbf{\textit{rdist}}(\mathrm{i1},\mathrm{i2}) - \textbf{\textit{sdist}}(\mathrm{m(i1)},\mathrm{m(i2)})| \leq \varepsilon$$

Systems in $\mathcal{ISR}$ include:

1. Harnad's [1] proposed 'solution' to the symbol grounding problem via the use of visual icons.
2. The Comirit project that combines 'imaginative' graph-based iconic representation and reasoning with the deductive reasoning of a logical theorem prover [9].
3. Reasoning performed within Gärdenfors' conceptual spaces framework, especially as a mechanism for embedding greater 'semantics' into symbolic systems such as the Semantic Web [10]. The cases or prototypes of a case-based reasoner may also be regarded as a similar form of iconic representation.
4. Setchi, Lagos and Froud's [11] proposed agenda for com-

putational imagination.

3.4 Distributed Representation

$\mathcal{DR}$: *Representational units may have semantic interpretation. Non-interpretable representational units must be composable as sets that in aggregate have semantic interpretation.*

$\mathcal{DR} = \{f : \mathcal{F} \mid \exists\, m : \mathcal{SI} \mid \forall\, p : \mathcal{P} \bullet \boldsymbol{si}(m, f, p)\}$

Every element of the set $\mathcal{DR}$ is a finite system that makes use of two kinds of representations: those that can be systematically assigned meaning, and those that only have meaning in aggregate (and may be of arbitrary form). That is, $\mathcal{DR}$ requires semantic interpretability, but does not require that the units of semantic interpretation correspond to the same representational units that are manipulated by the rules of the formal system.

Consider, for example, a neural network that has been trained to identify the gender of a human face. Some of the network's output nodes may be specifically trained to activate in the presence of masculine features: these output nodes, in addition to the hidden layer neurons that feed into the output nodes, may in aggregate be seen as meaning 'facial masculinity'. Even though it may be impossible to assign a coherent semantic interpretation to the representations and values of the hidden layer neurons that the formal system manipulates, the aggregated network can be seen as capturing specific real-world meaning.

Examples of systems that make representational assumptions or restrictions consistent with $\mathcal{DR}$ include:

1. Hybrid systems wherein neural networks have been trained under supervised learning to recognize symbols of a higher level symbolic reasoning processes. Indeed, all forms of supervised machine learning where the internal structure of the induced representations are regarded as a black box would be consistent with the restrictions of $\mathcal{DR}$.
2. Neural-symbolic systems that, for example, perform symbolic reasoning within connectionist mechanisms (*e.g.*, [12]).

3.5 Unconstrained Representation

$\mathcal{UR}$: *Representational units may or may not have any particular semantic interpretation.*

$\mathcal{UR} = \mathcal{F}$

Every element of the set $\mathcal{UR}$ corresponds to a problem-set that may be solved by a finite formal system (*i.e.*, a Turing-complete machine). The set $\mathcal{UR}$ therefore corresponds to the capabilities of computational systems, irrespective of whether their internal representations can be assigned particular semantic interpretations.

Examples of systems that make representational assumptions or restrictions consistent with $\mathcal{UR}$ include:

1. Neural-network-based systems, in which output or activity is triggered entirely by arbitrary connectionist processes (*e.g.*, [13,14]). In such systems, input nodes correspond to raw sensory data, output nodes are motor commands corresponding to actions, and internal hidden nodes are trained without regard to the development of meaningful cognitive symbols (*i.e.*, black-box intelligence): none of these nodes can be seen as capturing meaningful semantic interpretations.
2. Universal computable models of intelligence such as $\mathrm{AI}\xi^{tl}$ [15]. Such approaches emphasise computation or modelling that maximizes a reward function without regard for the semantic interpretability of the computational processes (though there is an implicit assumption that the most successful representations are likely to be those that best capture the environment and therefore are likely to acquire semantic interpretability in the limit).
3. Reactive systems, such as those concrete implementations of Brooks [16]. Such systems do not attempt to explicitly model the world (or may only partially model the world), and so lack semantic interpretability.

3.6 Non-Formal

$\mathcal{NF}$: *Representation units may or may not have any particular semantic interpretation, and may be manipulated by rules (such as interaction with the environment or hyper-computational systems) that are beyond formal definition.*

$\mathcal{NF} = \mathcal{F} \cup \mathcal{F}^*$

The class $\mathcal{NF}$ extends $\mathcal{UR}$ with a set of 'enhanced' formal symbolic systems, $\mathcal{F}^*$—systems with distinguished symbols that are connected to the environment[5]. While problems associated with action in the physical environment may already be found in the set $\mathcal{P}$, and these may already be solved by systems of other representational system classes (such as $\mathcal{UR}$), the set $\mathcal{NF}$ includes those systems that use embodiment directly as part of its deductive processes: systems where the environment is 'part' of the reasoning, rather than merely the 'object' of a solution. $\mathcal{NF}$ encompasses systems that, for example, need the environment to generate truly random sequences, to perform computations that aren't finitely computable on a Turing machine but may be solved by physical systems, to exploit some as-yet-unknown quantum effects, to build physical prototypes, or more simply, to solve problems about objects and complex systems that simply cannot be described or modelled in sufficient detail on a realizable computer system.

Examples of systems and research that make representational assumptions or restrictions consistent with $\mathcal{NF}$ include:

1. Embodied robotics in the true spirit of Brooks' vision, that treat 'the world [as] its own best model' and that refute the possibility of a disembodied mind [16]. Such work regards direct sensory experience and manipulation of the physical environment throughout problem solving as an essential part of the intelligent thought: that intelligence has co-evolved with the environment and sensory abilities; that it is not sufficient merely to have a reactive system; but that higher order intelligence arises from the complex interactions between reactivity and the environment. Note however, *actual* reactive robots/systems to date would, in fact, be better classified in $\mathcal{UR}$ (as we have done) because they do not yet operate at a level of interaction beyond primitive reactivity.
2. Models of intelligence and consciousness that are not Turing-computable or constrained by Gödel's incompleteness theorem. Examples of these may be found in work such as that of Penrose [17] postulating significant (but currently unspecified) quantum effects on intelligent thought and consciousness.

5 For example, we can allow a Turing machine to interact with the environment by reserving a segment of its tape as 'memory mapped' I/O. Symbols written to this segment of the tape will manipulate actuators and sensory feedback is itself achieved by a direct mapping back onto symbols of the I/O segment of the tape.

Discussion

The representational system classes not only serve to clarify many of the loose and ambiguous concepts that appear in debate on symbol grounding, but offer many other benefits: a language for rapidly communicating assumptions; a tool for analyzing the symbol grounding problem and generating new research assumptions; and a framework for better understanding underlying assumptions and the inter-relationships between assumptions. For example, Harnad's claims may be succinctly summarized as $\neg\, \mathcal{SIR} \sim I_{\text{Harnad}} \wedge \mathcal{ISR} \sim I_{\text{Harnad}}$.

Simple proof techniques can show that the representational classes form a hierarchy (*i.e.*, $\mathcal{CFSIR} \subseteq \mathcal{SIR} \subseteq \mathcal{ISR} \subseteq \mathcal{DR} \subseteq \mathcal{UR} \subseteq \mathcal{NF}$), and it follows that the combined sets of problems that each class may solve also forms a hierarchy (*i.e.*, we have a hierarchy of intelligence). However, it remains an interesting question whether this hierarchy is strict: are there classes of representational systems C_1 and C_2 such that $C_1 \subseteq C_2$ but there exists some definition of intelligence I where $\neg\, C_1 \sim I$, and $C_2 \sim I$ (we denote this, $C_1 < C_2$). *i.e.*, is C_2 strictly more intelligent than C_1? Our intuitions are that this is indeed the case for our hierarchy, and we plan to show this in future work. Here, we briefly outline our reasons for believing so:

- $\mathcal{CFSIR} < \mathcal{SIR}$, because even though the context-sensitive symbols of $\mathcal{SIR}$ could be systematically mapped into sets of context-free symbols in $\mathcal{CFSIR}$ (*e.g.*, ***Mouse*** → ***Cartoon_Mouse***), the potentially unbounded regress of contexts may make it impossible to ensure that this mapping remains finite when problem-sets are unbounded (*i.e.*, it can be done for any *particular* problem, but not *in general*, in advance of knowledge of the problem).
- $\mathcal{SIR} < \mathcal{ISR}$, following the arguments of Harnad.
- $\mathcal{ISR} < \mathcal{DR}$, because we believe that there are pathological concepts that *emerge* from complex chaotic systems so that iconic representations of structure or appearance hinder rather than enhance performance (*i.e.*, systems in which emergence is crucial to understanding the global system behavior, but for which properties of emergence cannot be predicted from local or structural analysis).
- $\mathcal{DR} < \mathcal{UR}$, because we believe that there are pathological situations where an attempt to analyze the situation into concepts diminishes the ability to learn appropriate behaviors (compare this to the manner in which human beings 'discover' false patterns in randomized data, hindering their ability to make optimal decisions using that data).
- $\mathcal{UR} < \mathcal{NF}$, because even though the universe may be formally computable, it may not be possible for any agent *situated within* the universe to describe the universe in sufficient detail such that a Turing machine could compute the solution to all 'intelligent' problems.

Finally, we are careful to emphasise again that we do not claim to have *solved* the problem. Instead, our framework reduces the symbol grounding to two long-standing philosophical challenges: the selection and definition of intelligence, I, and the problem of the nature of 'meaningful' entities in the universe (*i.e.*, the set $\mathcal{U}$, and consequently how to define ***si***(m, f, p)). While our framework does not yet offer precise guidance towards solving these sub-problems, it provides straightforward machinery by which the symbol grounding problem can be understood in such terms. Our contribution lies in formalizing the connections between sub-problems, and thereby narrowing the ambiguity in the problem and closing opportunities for circular reasoning.

Conclusion

By defining a formal framework of representation grounding, we help clarify the contention in important work on symbol grounding as stemming from arguments about different kinds of representational system classes. We have proposed six classes to this end: $\mathcal{CFSIR} \subseteq \mathcal{SIR} \subseteq \mathcal{ISR} \subseteq \mathcal{DR} \subseteq \mathcal{UR} \subseteq \mathcal{NF}$. These capture many perspectives on the symbol grounding problem: the classes have significant power both for explanation and investigation. Not only can future research use these classes to quickly express assumptions, but the abstractions assist in the exploration of the problem, the classification and comparison of existing work, and provide machinery for the development of novel conjectures and research questions.

References

1. Harnad, S. 1990. 'The symbol grounding problem', *Physica D*, 42:335-346.
2. Coradeschi, S. and Saffiotti A. (eds) 2003. *Robotics and Autonomous Systems*, 43(2-3).
3. Williams, M-A., Gärdenfors, P., Karol, A., McCarthy, J. and Stanton, C. 2005. 'A framework for evaluating the groundedness of representations in systems: from brains in vats to mobile robots', *IJCAI 2005 Workshop on Agents in Real Time and Dynamic Environments*.
4. Ziemke, T. 1997. 'Rethinking grounding', *Proceedings of New Trends in Cognitive Science*, 1997, Austrian Society for Computer Science.
5. Bouquet, P., Serafini, L. and Stoermer, H. 2005. 'Introducing context into RDF knowledge bases', *SWAP 2005*.
6. McCarthy, J. 1993. 'Notes on formalizing context', *IJCAI 1993*.
7. Serafini, L. and Bouquet, P. 2004. 'Comparing formal theories of context in AI', *Artificial Intelligence*, 155(1): 41-67.
8. Lenat, D. 1998. 'The dimensions of context space', *Cycorp Technical Report*.
9. Johnston, B. and Williams, M-A. 2008. 'Comirit: Commonsense reasoning by integrating simulation and logic', *AGI 2008*, 200-211.
10. Gärdenfors, P. 2004. 'How to make the semantic web more semantic', In Varzi, A. and Lieu, L. (eds) *Formal Ontology in Information Systems*, 17-34, IOS Press.
11. Setchi, R., Lagos, N. and Froud, D. 2007. 'Computational imagination: research agenda', *Australian AI 2007*, 387-393.
12. Kilkerry Neto, A., Zaverucha, G. and Carvalho, L. 1999. 'An implementation of a theorem prover in symmetric neural networks', IJCNN 1999, 6: 4139-4144.
13. Markram, H. 2006. 'The Blue Brain project', *Nature Reviews: Neuroscience*, 7: 153-160.
14. de Garis, H., Tang, J-Y., Huang, Z-Y., Bai, L., Chen, C., Chen, S., Guo, J-F., Tan, X-J., Tian, H., Tian, X-H., Wu, X-J., Xiong, Y., Yu, X-Q. and Huang, D. 2008. 'The China-Brain project: building China's artificial brain using an evolved neural net module approach', *AGI 2008*, 107-121.
15. Hutter, M. 2000. 'Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions', *EMCL 2001*, 226-238.
16. Brooks, R. 1991 'Intelligence without reason', *IJCAI 1991*, 569-595.
17. Penrose, R. 1999. *The Emperor's New Mind*, Oxford University Press.