The Glass Box Approach: Verifying Contextual Adherence to Values

Andrea Aler Tubella[*] and **Virginia Dignum**
Umeå University
{andrea.aler, virginia.dignum}@umu.se

Abstract

Artificial Intelligence (AI) applications are being used to predict and assess behaviour in multiple domains, such as criminal justice and consumer finance, which directly affect human well-being. However, if AI is to be deployed safely, then people need to understand how the system is interpreting and whether it is adhering to the relevant moral values. Even though transparency is often seen as the requirement in this case, realistically it might not always be possible or desirable, whereas the need to ensure that the system operates within set moral bounds remains.

In this paper, we present an approach to evaluate the moral bounds of an AI system based on the monitoring of its inputs and outputs. We place a 'Glass Box' around the system by mapping moral values into contextual verifiable norms that constrain inputs and outputs, in such a way that if these remain within the box we can guarantee that the system adheres to the value(s) in a specific context. The focus on inputs and outputs allows for the verification and comparison of vastly different intelligent systems– from deep neural networks to agent-based systems– whereas by making the context explicit we expose the different perspectives and frameworks that are taken into account when subsuming moral values into specific norms and functionalities. We present a modal logic formalisation of the Glass Box approach which is domain-agnostic, implementable, and expandable.

1 Introduction

Artificial Intelligence (AI) has the potential to greatly improve our autonomy and wellbeing, but to be able to interact with it effectively and safely, we need to be able to trust it. Trust in Artificial Intelligence (AI) is often linked to algorithmic transparency [Theodorou *et al.*, 2017]. This concept includes more than just ensuring algorithm visibility: the different factors that influence the decisions made by algorithms should be visible to the people who use, regulate, and are impacted by systems that employ those algorithms [Lepri *et al.*, 2018]. However, decisions made by predictive algorithms can be opaque because of many factors, for instance IP protection, which may not always be possible or desirable to eliminate [Ananny and Crawford, 2018]. Yet, accidents, misuse, disuse, and malicious use are all bound to happen. Since human decisions can also be quite opaque, as are the decisions made by corporations and organisations, mechanisms such as audits, contracts, and monitoring are in place to regulate and ensure attribution of accountability. In this paper, we propose a similar approach to monitor and verify artificial systems.

On the other hand, the current emphasis on the delivery of high-level statements on AI ethics may also bring with it the risk of implicitly setting the 'moral background' for conversation about ethics and technology as being about abstract principles [Greene *et al.*, 2019]. The high-level values and principles are dependent on the socio-cultural context [Turiel, 2002]; they are often only implicit in deliberation processes. The shift from abstract to concrete therefore necessarily involves careful consideration of the context. In this sense, the subsumption of each value into functionalities will vary from context to context the same way it can vary from system to system. For example, consider the value *fairness*: it can have different normative interpretations, e.g. *equal access to resources* or *equal opportunities*, which can lead to different actions. This decision may be informed by domain requirements and regulations, e.g.national law. Often, these choices made by the designer of the system and the contexts considered are hidden from the end-user, as well as for future developers and auditors: our aim is to make them explicit.

This paper presents the Glass Box approach [Aler Tubella *et al.*, 2019] to evaluating and verifying the contextual adherence of an intelligent system to moral values. We place a 'Glass Box' around the system by mapping abstract values into explicit verifiable norms that constrain inputs and outputs, in such a way that if these remain within the box we can guarantee that the system adheres to the value in a certain context. The focus on inputs and outputs allows for the verification and comparison of vastly different intelligent systems; from deep neural networks to agent-based systems. Furthermore, we make context explicit, exposing the different perspectives and frameworks that are taken into account when subsuming moral values into specific norms and functional-

[*] Contact Author

ities. We present a modal logic formalisation of the Glass Box approach which is domain-agnostic, implementable, and expandable.

2 The Glass Box approach

The Glass Box approach [Aler Tubella *et al.*, 2019], as depicted in Figure 1, consists of two phases which inform each other: interpretation and observation. It takes into account the contextual interpretations of abstract principles by taking a *Design for Values* perspective [Van de Poel, 2013].

The interpretation stage is the explicit and structured process of translating values into specific design requirements. It entails a translation from abstract values into concrete norms comprehensive enough so that fulfilling the norm will be considered as adhering to the value. Following a Design for Values approach, the shift from abstract to concrete necessarily involves careful consideration of the context. For each context we build an abstract-to-concrete hierarchy of norms where the highest level is made-up of values and the lowest level is composed of fine-grained concrete requirements for the intelligent system only related to its inputs and outputs. The intermediate levels are composed of progressively more abstract norms, and the connections between nodes on each level are contextual. When building an intelligent system, each requirement is distilled into functionalities implemented into the system in order to fulfill it. At the end of the interpretation stage we therefore have an explicit contextual hierarchy which can be used to provide high-level transparency for a deployed system: depending on which requirements are being fulfilled, we can provide explanations for how and exactly in which context the system adheres to a value. Note that the interpretation stage is also useful for the evaluation of a system, as it provides grounding an justification for system requirements, in terms of the norms and values they are an interpretation. That is, it indicates a 'for-the-sake-of' relation between requirements and values.

The low-level requirements inform the observation stage of our approach, as they indicate what must be verified and checked. In the observation stage, the behaviour of the system is evaluated with respect to each value by studying its compliance with the requirements identified in the previous stage. In [Vázquez-Salceda *et al.*, 2007] two properties for norms to be enforceable are identified: (1) verifiability i.e., the low-level norms must allow for being machine-verified given the time and resources needed, and (2) computational tractability, i.e. whether the functionalities comply with the norms can be checked on any moment in a fast, low cost way. Note that this is a requirement for the observation stage and not necessarily for the design stage: some of the norms chosen for the design stage might be easily implementable, but hard to monitor. In the observation stage, to each requirement identified in the interpretation stage, we assign one or several *tests* to verify whether it is being fulfilled. Testing may range through a variety of techniques, from simply checking whether input/output verify a particular relationship, to complex methods such as statistical testing, formal verification or model-checking. These must be performed without knowledge about the internal workings of the system under observation, by monitoring input and output streams only. We insist on this feature as we do not always have access to the internals of the system, neither do we always have access to the designs of a system.

Designing the tests is naturally one of the most complex steps of this process: the main challenge is the computational tractability of these checks and their correspondence with the low-level norms and their implementation. Different levels of granularity in the norms pose different constraints for testing: the cost of checking that the outcome for a certain input remains within certain bounds is very different than having to consider data of a whole database of inputs and outputs. Part of the challenge is then determining the required granularity of the Glass Box and testing: a too rough approximation can possibly cap many potentially compliant behaviours, whereas a too specific one may limit the adaptation of the AI system.

From the observation stage we give feedback to the interpretation stage: the testing informs us on which requirements are being fulfilled and which aren't, which may prompt changes in the implementation or in the chosen requirements. The observation stage is therefore fundamental both at a design stage to verify that the intelligent system is functioning as desired, and after deployment to explicitly fill in stakeholders on how the system is interpreting and whether it is verifiably adhering to the relevant moral values without having to reveal its internal working.

3 Running example

As an example, we will consider an intelligent system used to filter CVs as a recruitment tool. Note that the ethical values, norms and functionalities highlighted in what follows are used purely as an example, and we do not claim that they are the most appropriate to adhere to, but rather are used to demonstrate the approach.

As a starting point, the designers of the system must identify the relevant ethical values that they wish to adhere to, depending on the legal framework, the company policies, the standards they are following, etc. They could, for example, settle on *fairness* and *privacy*. The next step is to unravel what these values mean in terms of recruitment decisions from different perspectives.

In the case of fairness, they could consider several angles. In the context of the Swedish law, for instance, fairness in recruitment means, amongst other things, non-discrimination between male and female applicants. A design requirement to guarantee fairness in this context can therefore be that the ratio of acceptances vs rejections has to be the same for both men and women (which can be calculated purely from the inputs and outputs of the system). This requirement is then taken into account for implementation: for example, it can be decided –rather ineffectively [Reuters, 2018]– to exclude gender from the inputs of the system. In the same way, each legal requirement in terms of fairness will be translated into specific requirements for the system.

Another perspective for fairness can be provided by company policy. It can for example be deemed that it is fair to give preference to those applicants that are already working for the company. In this case, the requirement for the sys-

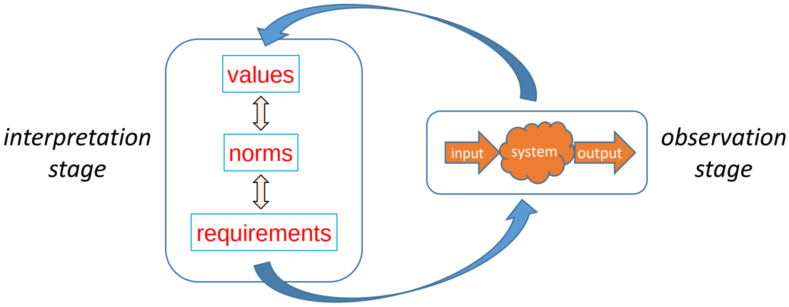

Figure 1: The two stages of the Glass Box Approach: an Interpretation stage, where values are translated into design requirements, and an Observation stage, where we can observe and qualify the behaviour of the system.

tem would be that applicants from within the company are prioritised. Functionality-wise, this can be translated into the assignment of weights for each variable considered in the implementation.

In the same way, other perspectives (e.g. European law, HR recruitment guidelines) and other values (e.g. privacy, responsibility) will be taken into account and distilled into requirements and functionalities, providing a contextual hierarchy of values, norms, requirements and functionalities as a result of the interpretation stage.

At this point, we proceed to the observation stage where testing procedures are devised for each of the functionalities identified in the previous stage. To test whether the ratio of acceptances vs rejections is the same for both men and women, we can for example check periodically after every 100 decisions whether the two ratios are within 5% of each other. To test whether applicants from within the company are prioritised, we can take random samples of applicants from outside and inside the company, and check that the acceptance rate for the latter group is higher. With the results of these tests in hand, we can reason about which values are being verified (or not) in each context.

4 Formalising the Glass Box

Since AI applications exist in a huge variety of areas, the formalisation we present is based on predicate logic: it is domain-agnostic and can be adapted to any application. Crucially, it is also implementable: the hierarchy of checks, norms and values can be encoded in logical programming languages, and the complexity of the system in terms of the queries that we will pose to it is well within the reach of current techniques.

4.1 Counts-as

The interpretation stage entails a translation from abstract values into concrete norms and requirements comprehensive enough so that fulfilling the norm will be considered as adhering to the value, with careful consideration of the context. Normative systems are often described in deontic-based languages, which allow for the representation of obligations, permissions and prohibitions. With this approach, however, we aim to not only describe the norms themselves, but also the exact connection between abstract and concrete concepts in each context.

Several authors have proposed *counts-as* statements as a means to formalise contextual subsumption relations [Aldewereld *et al.*, 2010]. With this relation, we can build logical statements of the form: "X counts as Y in context c" [Searle, 1995; Jones and Sergot, 1995]. The semantics of counts-as is often interpreted in a classificatory light [Grossi *et al.*, 2005], i.e. "A counts-as B in context c" is interpreted as "A is a subconcept of B in context c". Thus, counts-as statements can be understood as expressing classifications that hold in a certain context. At the same time, from a different semantic viewpoint a counts-as operator can be used not only to express classifications that happen to hold in a context, but to represent the classifications that define the context itself. Counts-as can also encode *constitutive* rules [Grossi *et al.*, 2008], that is, the rules specifying the ontology that defines each context.

To formally represent the hierarchy of functionalities, requirements, norms and values resulting from the interpretation stage of the Glass Box approach both outlooks are necessary. On one hand, contexts are defined by the connections between more concrete lower level concepts to abstract values, precisely corresponding to the notion of constitutive counts-as. On the other hand, once the contexts are established, we aim to be able to reason about which combinations of functionalities lead to the fulfillment of each norm i.e. about the classifications holding in each context. Both views of counts-as admit compatible representations in modal logic as shown in [Grossi *et al.*, 2008]: we will use the formalism and semantics presented there, which we will briefly introduce in this subsection.

The logic we will consider is $\mathbf{Cxt}^{u,-}$. It is a multi-modal homeogeneous $\mathbf{K45}$ [Blackburn *et al.*, 2007], extended with a universal context, negations of contexts, and nominals which denote the states in the semantics.

Definition 1. Language $\mathcal{L}_n^{u,-}$ is given by: a finite set $\mathbb{P}$ of propositional atoms p, an at most countable set $\mathbb{N}$ of nominals denoted by s disjoint from $\mathbb{P}$, and a finite non-empty set K of $n/2$ atomic context indexes denoted by c including a distinguished index u representing the universal context. The set C of context indexes is given by the elements c of K and their negations $-c$ and its elements are denoted by $i, j, \ldots$

Further, the alphabet of $\mathcal{L}_n^{u,-}$ contains the set of boolean connectives $\{\neg, \wedge, \vee, \rightarrow\}$ and the operators $[\ ]$ and $\langle\ \rangle$.

The set of *well-formed formulae* of $\mathcal{L}_n^{u,-}$ is given by the following BNF:

$$\phi ::= \bot \mid p \mid s \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid [i]\phi \mid \langle i\rangle\phi\ .$$

Formulae in which no modal operator occurs are called *objective*.

Logic $\mathbf{Cxt}^{u,-}$ is axiomatized via the following axioms and rules schemata:

(P) all tautologies of propositional calculus

($\mathtt{K}^i$) $[i](\phi_1 \rightarrow \phi_2) \rightarrow ([i]\phi_1 \rightarrow [i]\phi_2)$

(4^{ij}) $[i]\phi \rightarrow [j][i]\phi$

(5^{ij}) $\neg[i]\phi \rightarrow [j]\neg[i]\phi$

($\mathtt{T}^u$) $[u]\phi \rightarrow \phi$

($\subseteq .ui$) $[u]\phi \rightarrow [i]\phi$

(Least) $\langle u\rangle s$

(Most) $\langle u\rangle(s \wedge \phi) \rightarrow [u](s \rightarrow \phi)$

(Covering) $[c]\phi \wedge [-c]\phi \rightarrow [u]\phi$

(Packing) $\langle -c\rangle s \rightarrow \neg\langle c\rangle s$

(Dual) $\langle i\rangle\phi \leftrightarrow \neg[i]\neg\phi$

(Name) IF $\vdash s \rightarrow \theta$ THEN $\vdash \theta$, for s not occurring in θ

(MP) IF $\vdash \phi_1$ AND $\vdash \phi_1 \rightarrow \phi_2$ THEN $\vdash \phi_2$

($\mathtt{N}^i$) IF $\vdash \phi$ THEN $\vdash [i]\phi$

where i, j are metavariables for the elements of C, c denotes elements of the set of atomic context indexes K, u is the universal context index, v ranges over nominals, and θ in rule Name denotes a formula in which the nominal denoted by s does not occur.

Logics with nominals are called hybrid logics [Blackburn *et al.*, 2007]: they blur the lines between syntax and semantics, allowing us to express possible states (semantics) through formulae (syntax). In this application, the presence of nominals allows for the definition of rules COVERING and PACKING, fundamental to capture the concept of the complement of a context. This becomes clearer when looking at the semantics: logic $\mathbf{Cxt}^{u,-}$ enjoys a possible-world semantics in terms of a particular class of multiframes. In this type of semantics, we represent the states that are possible in each context, and consider an interpretation function $\mathcal{I}$ which associates to each propositional atom the set of states which make it true.

Definition 2. A $\textsc{Cxt}^{\top,\setminus}$ frame $\mathcal{F}$ is a structure $\langle W, \{W_i\}_{i\in C}\rangle$ where:

- There is a set K such that $C = K \cup \{-c | c \in K\}$;
- W is a finite set of states (possible worlds) ;
- $\{W_i\}_{i\in C}$ is a family of subsets of W such that: there exists a distinguished $u \in C$ with $W_u = W$ (there is a universal context), and such that for every atomic context $c \in K$ we have that $W_{-c} = W_u \setminus W_c$.

A *model* $\mathcal{M}$ for the language $\mathcal{L}_n^{u,-}$ is a pair $(\mathcal{F}, \mathcal{I})$ where $\mathcal{F}$ is a $\textsc{Cxt}^{\top,\setminus}$ frame and $\mathcal{I}$ is a function $\mathcal{I} : \mathbb{P} \cup \mathbb{N} \rightarrow \mathcal{P}(W)$ such that:

- For all nominals $s \in \mathbb{N}$, there is a state w such that $\mathcal{I}(s) = \{w\}$ (the interpretation of a nominal is a single state) ;
- For all states $w \in W$ there is a nominal $s \in \mathbb{N}$ such that $\mathcal{I}(s) = \{w\}$ (every state has a name) .

Definition 3. We define satisfaction for $\textsc{Cxt}^{\top,\setminus}$ frames as follows:

$$\begin{array}{rll} \mathcal{M}, w \vDash s & \text{iff} & \mathcal{I}(s) = \{w\} \\ \mathcal{M}, w \vDash [c]\phi & \text{iff} & \forall w' \in W_c : \mathcal{M}, w' \vDash \phi \\ \mathcal{M}, w \vDash [-c]\phi & \text{iff} & \forall w' \in W \setminus W_c : \mathcal{M}, w' \vDash \phi \end{array}$$

where s ranges over nominals and c ranges on the context indexes in K. The boolean clauses and clauses for the dual modal operator are defined in a standard way and are omitted.

With satisfaction defined in this way, the following theorem holds.

Theorem 1. *Logic* $\mathbf{Cxt}^{u,-}$ *is sound and complete with respect to* $\textsc{Cxt}^{\top,\setminus}$ *frames.*

For a detailed proof, the interested reader is invited to refer to [Grossi *et al.*, 2008]. The intuitive reading of the semantics is that W contains all the possible worlds (or *states*) considered in the model. For each context, W_c contains the states that are possible with the added restrictions of the context. Then, the set of possible worlds in the universal context coincides with all possible worlds, and the set of possible worlds for the negation of a context is the complement of its set of possible worlds.

The specific requirements on nominals capture the idea that each nominal is only satisfied in a single state, and that in every state there is at least one nominal that is satisfied: nominals can therefore simply be interpreted as *names* for each state.

Logic $\mathbf{Cxt}^{u,-}$ provides us with the theoretical machinery to be able to define both classificatory and constitutive counts-as operators, which we will use to build and explore hierarchies of norms and values.

Definition 4. Let γ_1, γ_2 be objective formulae.

The classificatory counts-as is statement "γ_1 counts as γ_2 in context c" is formalised in $\mathbf{Cxt}^{u,-}$ by

$$\gamma_1 \Rightarrow_c^{cl} \gamma_2 := [c](\gamma_1 \rightarrow \gamma_2) \quad .$$

Let Γ be a set of formulae, with $\gamma_1 \rightarrow \gamma_2 \in \Gamma$. The constitutive counts-as statement "γ_1 counts as γ_2 *by constitution* in the context c defined by Γ" is formalised in $\mathbf{Cxt}^{u,-}$ by

$$\gamma_1 \Rightarrow_{c,\Gamma}^{co} \gamma_2 := [c]\Gamma \wedge [-c]\neg\Gamma \wedge \neg[u](\gamma_1 \rightarrow \gamma_2) \quad .$$

Note that for constitutive counts-as statements, both the name c of the context and the formulae Γ that define it have to be specified. This corresponds to the notion that constitutive statements are those statemnts that we take as a definiton for the context. If c is defined by Γ, an equivalent set of formulae Γ' defines the same context c, but the constitutive statements that hold in c, Γ' are different than those that hold in

c, Γ and correspond to the formulae of Γ'. On the other hand, the classificatory statements holding in a context remain the same no matter which set of equivalent formulae we choose as its definition, since they don't define the context but rather correspond to inferences that hold in it.

4.2 Glass Box contexts

The end-result of the interpretation stage is a collection of contexts, each given by a hierarchy of functionalities and norms fulfilling values. In this sense, contexts are defined by the hierachy that holds in them. For this reason, we will formally define contexts through the set of implications that define it, which we can then implement via constitutive counts-as statements.

Furthermore, our aim is for contexts to be hierarchies of progressively more concrete terms. For this reason we need to partition our language, given by the set of propositional atoms we work with, into levels. Intuitively, given a hierarchy of norms and values, we will assign a level to each propositional atom it is composed of, corresponding to its position in the hierarchy in terms of concreteness. In addition, this allows for the use of different vocabulary for each level. Contexts are then formed by defining how the propositional atoms of level i are related to atoms representing more abstract concepts at level $i-1$.

Definition 5. Let $\mathbb{P}_I$ be a set of propositional atoms.

Given a subset $S \subseteq \mathbb{P}_I$ we denote by p^S the elements of S and by γ^S the objective formulae built on the propositional atoms of S, given by

$$\gamma^S ::= p^S \mid \neg\gamma^S \mid \gamma_1^S \wedge \gamma_2^S \mid \gamma_1^S \vee \gamma_2^S \mid \gamma_1^S \to \gamma_2^S \ .$$

A *hierarchy* is a partition $\mathcal{P} = \{P_0, \ldots, P_N\}$ of $\mathbb{P}_I$ i.e. a collection of sets such that $P_0 \sqcup \cdots \sqcup P_N = \mathbb{P}_I$.

An *interpretation context* c in a hierarchy $\mathcal{P}$ is given by:

- a collection of subsets $F_i^c \subseteq P_i$, $0 \leq i \leq N$;
- a collection Γ_c of objective formulae of the form

$$\gamma^{F_{i+1}^c} \to p^{F_i^c}$$

such that for every $p^{F_i^c}$, $0 \leq i < N$ there is at least one such formula in Γ_c.

When referring to an interpretation context, we will often abuse language and omit the family of subsets of the partition included in its definition, as it is recoverable from Γ_c.

Let $\mathcal{P}$ be a hierarchy on a set $\mathbb{P}_I$. An *interpretation box* is a finite collection K of interpretation contexts in $\mathcal{P}$.

With this characterisation, we represent the hierarchy of concepts, from most abstract to most concrete, as a partition. Elements of P_0 correspond to values and elements of P_N correspond to functionalities. Each interpretation context c is given by explicitly stating the relationships from more concrete to more abstract concepts by specifying them in Γ_c.

Note that at the interpretation stage the lowest level of the hierarchy defining the context is given by functionalities and not by the verification procedures. These are designed and seamlessly incorporated to the Glass Box in the second stage of the process, allowing for a modular approach.

4.3 Glass Box verification

The observation stage consists on checking that the lower-level norms devised at the interpretation stage are in fact adhered to. Even if we restrict tests to constraints on the inputs and outputs of a system, they can encode a number of complex behaviours, from obliging the input or output to stay within certain parameters, to imposing a certain relationship between the input and output as a function of each other, to comparing the inputs and outputs to other similar cases. Furthermore, the tests need to be computationally checkable in a reasonable time. Once devised, these tests will be translated into propositional variables that will encode whether a test has failed or has passed. The results of these tests will be entered into the Glass Box by means of these variables, which we can then use to reason about whether a value has been verified in a certain context. In this stage we therefore need to specify which tests are associated with each low-level norm in each context, and how.

Definition 6. Let $\mathbb{P}_O$ be a finite set of binary predicates. We denote by $p^{\mathbb{P}_O}$ the elements of $\mathbb{P}_O$ and by $\gamma^{\mathbb{P}_O}$ objective formulae built on the propositional atoms of $\mathbb{P}_O$ and $\wedge$ and $\vee$, given by

$$\gamma^{\mathbb{P}_O} ::= p^{\mathbb{P}_O} \mid \neg\gamma^{\mathbb{P}_O} \mid \gamma_1^{\mathbb{P}_O} \wedge \gamma_2^{\mathbb{P}_O} \mid \gamma_1^{\mathbb{P}_O} \vee \gamma_2^{\mathbb{P}_O} .$$

Let c be an interpretation context on a partition $\mathcal{P}$ of set $\mathbb{P}_I$ given by a collection of subsets $F_i^c \subseteq P_i$, $0 \leq i \leq N$ and a set of objective formulae Γ_c.

A *testing context* Δ_c for c is a collection of objective formulae of the form

$$\gamma^{\mathbb{P}_O} \to p^{F_{N-1}^c}$$

such that for every $p^{F_{N-1}^c} \in F_{N-1}^c$ there is at least one such formula.

An *observation box* on $\mathbb{P}_O$ associated to an interpretation box $\{c \in K\}$ is given by a set $\{\Delta_c | c \in K\}$ where each Δ_c is a testing context for c.

Notice that we don't consider implication in the vocabulary of tests, since we will operate with concrete test results that return either "pass" or "fail", and it theoretically makes no semantic sense, given a specific outcome of the testing, to reason in general about whether a certain test result implies another test result.

4.4 Reasoning inside the Glass Box

Interpretation and observation boxes contain all the implications that define each context. We can now use counts-as to build a framework that will allow us to reason about the statements that hold in each context. Given an interpretation box and an associated observation box, following Definition 1 we will build a language on the propositional atoms of $\mathbb{P} = \mathbb{P}_I \sqcup \mathbb{P}_O$ and the context labels in K.

Additionally, we will need to specify a set $\mathbb{N}$ of nominals denoting every possible world that we consider in our model. Following the semantics of Definition 2, this set corresponds to the set of states that are possible within the universal context. Since all the restrictions in our framework are contextual and not universal, all the truth value assignments for the elements of $\mathbb{P}$ can hold in the universal context. Thus we will define $\mathbb{N}$ as a set of $2^{|\mathbb{P}|}$ elements, allowing for a one-to-one correspondence between elements of $\mathbb{N}$ and all possible worlds, i.e. truth value assignments, in the semantics.

Definition 7. A *Glass Box* is given by:

- A set of propositional atoms $\mathbb{P} = \mathbb{P}_I \sqcup \mathbb{P}_O$;
- An interpretation box $\{c \in K\}$ on a hierarchy $\mathcal{P}$ on $\mathbb{P}_I$;
- An associated Glass observation box $\{\Delta_c | c \in K\}$ on $\mathbb{P}_O$;
- A set $\mathbb{N}$ of $2^{|\mathbb{P}|}$ elements.

Given a Glass Box, we can build language $\mathcal{L}_n^{u,-}$ on $\mathbb{P}$, $\mathbb{N}$ and $K' = K \cup \{u\}$, where u is an additional context name, following Definition 1. We consider logic $\mathbf{Cxt}^{u,-}$ on this language.

For each $c \in K$, let $\Upsilon_c = \Gamma_c \cup \Delta_c$. We define the *Glass Box constitution* as the conjunction of formulae

$$GB := \bigwedge_{\substack{c \in K \\ \gamma \to p \in \Upsilon_c}} (\gamma \Rightarrow^{co}_{c,\Upsilon_c} p) \; .$$

Having encoded the Glass Box in a logical system (see Figure 2), we can now reason about the statements that hold in it. With the implementation in mind, we are particularly interested in classificatory statements, which allow us to describe for example which combinations of norms count as satisfying a value in a context. The following definition illustrates some of the statements which we will want to hold in the Glass Box.

Definition 8. We say that an objective formula γ is *incompatible* with context c if

$$\vdash GB \to (\gamma \Rightarrow^{cl}_{c} \bot).$$

Incompatible formulae imply both a formula and its negation in context c, and therefore we wish to remove them from the set of formulae that verify a certain norm or value.

We say that a combination of functionalities $\gamma^{F^c_N}$ counts as value p^{P_0} in context c if it is not incompatible with c and

$$\vdash GB \to (\gamma^{F^c_N} \Rightarrow^{cl}_{c} p^{P_0}) \; .$$

We say that a test result $\gamma^{\mathbb{P}_O}$ verifies value p^{P_0} in context c if it is not incompatible with c and

$$\vdash GB \to (\gamma^{\mathbb{P}_O} \Rightarrow^{cl}_{c} p^{P_0}) \; .$$

Formulae incompatible with a certain context correspond to statements that do not make semantic sense: they may be contradictory by themselves in this context, or lead to contradictions within the context by for example, implying that both a value and its negation are satisfied.

Crucially for an effective implementation, given a certain test result, we want to answer the question of whether this result verifies a certain value in a given context. Thus we need to find a proof of $GB \to (\gamma^{\mathbb{P}_O} \Rightarrow^{cl}_{c} p^{P_0})$, or to show that there is no such proof. We therefore need to address the issue of the search-complexity of our system.

(Multi)modal logics with a universal modality have an EXPTIME-complete K-satisfaction problem [Hemaspaandra, 1996] and adding nominals maintains this bound [Areces, 2004]. For our system to be suitable for an implementation, we need to show that the specific queries we will be posing are answerable in a reasonable time. Furthermore, it would be desirable to be able to solve the satisfiability problems we will pose with existing tools.

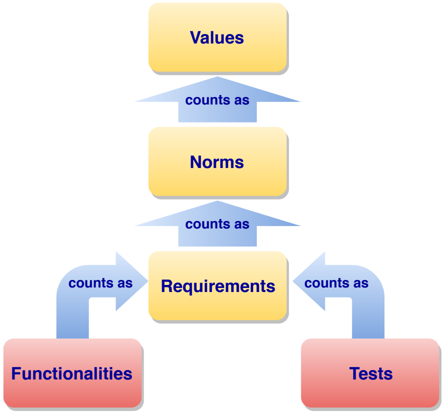

Figure 2: Formalisation of the Glass Box approach

To address both of these points, we will show that answering questions of the form "does γ count as γ' in context c in the Glass Box?" is equivalent to checking whether the implication $\gamma \to \gamma'$ holds propositionally with the assumptions of Υ_c. This is in fact a very intuitive result: the only constraints on a context c in the Glass Box are those set-up by its definition through Υ_c, and therefore any deduction in context c only needs to consider these constraints. We therefore reduce our question to a satisfiability problem in propositional logic with a finite number of propositions. In real-life applications, our human-made vocabulary for norms and values should remain reasonably small. Additionally, the number of tests performed needs to remain relatively small as well for computational reasons. Thus answering queries in our propositional language should easily remain well within the reach of SAT-solvers and answer set programming approaches.

Proposition 2. *Let γ be an objective formula. We have that*

$$\vdash GB \to [c]\gamma \textit{ iff } \vdash \Upsilon_c \to \gamma \; .$$

Proof. Right to left is easy to see. It is given by the following deduction:

$$\begin{array}{lcl}
1 & \text{(hypothesis)} & \vdash \Upsilon_c \to \gamma \\
2 & (\mathtt{N}^c) & \vdash [c](\Upsilon_c \to \gamma) \\
3 & (\mathtt{K}^c), (\mathtt{MP}) & \vdash [c]\Upsilon_c \to [c]\gamma \\
4 & (\mathtt{P}), (\mathtt{MP}) & \vdash GB \to [c]\gamma \; .
\end{array}$$

Left to right will be proven making use of the soundness of the logic with the semantics introduced in Definition 2. Consider the model $\mathcal{M}$ given by $(\langle W, \{W_i\}_{i \in C}\rangle, \mathcal{I})$ where:

- W is the set of all possible valuations for $\mathbb{P}$;
- For each $c \in K$, W_c is the set of truth-value assignments for $\mathbb{P}$ in which Υ_c holds, and we set $W^{-c} := W \setminus W_c$ and $W_u := W$;
- $\mathcal{I} : \mathbb{P} \to \mathcal{P}(W)$ assigns to each propositional atom the set of states where its assignment is `true` ;
- $\mathcal{I} : \mathbb{N} \to \mathcal{P}(W)$ is a one-to-one assignment between the elements of $\mathbb{N}$ and the elements of W .

$\mathcal{M}$ is a model for the language $\mathcal{L}_n^{u,-}$.

If we assume that $\vdash GB \rightarrow [c]\gamma$ holds in logic $\mathbf{Cxt}^{u,-}$, then by soundness $\mathcal{M} \models GB \rightarrow [c]\gamma$. Furthermore, it is easy to see that $\mathcal{M} \models GB$, from the definition of $\mathcal{M}$. Therefore $\mathcal{M} \models [c]\gamma$ holds.

Thus, by definition, $\forall w' \in W_c : \mathcal{M}, w' \models \gamma$. Therefore, in every truth-value assignment where Υ_c holds, also γ holds i.e. $\Upsilon_c \rightarrow \gamma$ holds propositionally.

□

5 Discussion

The Glass Box approach is both an approach to software development, a verification method and a source of high-level transparency for intelligent systems. It provides a modular approach integrating verification with value-based design.

Achieving trustworthy AI systems is a multifaceted complex process, which requires both technical and socio-legal initiatives and solutions to ensure that we always align an intelligent system's goals with human values. Core values, as well as the processes used for value elicitation, must be made explicit and that all stakeholders are involved in this process. Furthermore, the methods used for the elicitation processes and the decisions of who is involved in the value identification process are clearly identified and documented. Similarly, all design decisions and options must also be explicitly reported; linking system features to the social norms and values that motivate or are affected by them. This should always be done in ways that provide inspection capabilities —and, hence, traceability— for code and data sources to ensure that data provenance is open and fair.

The formalisation we presented in this paper allows for implementation while remaining highly versatile: this approach is not only useful for black boxes, as more information can easily be included in the hierarchy and the testing. Furthermore, by including a universal context, we can easily include universal context-free statements that may hold in particular applications. We aim to expand it into concrete implementations in answer set programming. Beyond concrete implementations, further work will include studying the effects of this type of value-oriented transparency.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[Aldewereld *et al.*, 2010] H. Aldewereld, S. Álvarez-Napagao, F.P.M. Dignum, and J. Vázquez-Salceda. Making Norms Concrete. In *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 807–814, Toronto, Canada, 2010.

[Aler Tubella *et al.*, 2019] A. Aler Tubella, A. Theodorou, F.P.M. Dignum, and V. Dignum. Governance by Glass-Box: Implementing Transparent Moral Bounds for AI Behaviour. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJ-CAI'2019)*, 2019. To appear.

[Ananny and Crawford, 2018] M. Ananny and K. Crawford. Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, 20(3):973–989, 2018.

[Areces, 2004] C Areces. The computational complexity of hybrid temporal logics. *Logic Journal of IGPL*, 8(5):653–679, 9 2004.

[Blackburn *et al.*, 2007] P. Blackburn, J. van Benthem, and F. Wolter. *Handbook of modal logic*. Elsevier, 2007.

[Greene *et al.*, 2019] D. Greene, A. Hoffmann, and L. Stark. Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[Grossi *et al.*, 2005] D. Grossi, J-J.Ch. Meyer, and F.P.M. Dignum. Modal Logic Investigations in the Semantics of Counts-as. *Proceedings of ICAIL'05*, 2005.

[Grossi *et al.*, 2008] D. Grossi, J-J.Ch. Meyer, and F.P.M. Dignum. The many faces of counts-as: A formal analysis of constitutive rules. *Journal of Applied Logic*, 6(2):192–217, 6 2008.

[Hemaspaandra, 1996] Edith Hemaspaandra. The Price of Universality. *Notre Dame Journal of Formal Logic*, 37(2), 1996.

[Jones and Sergot, 1995] A.J.I. Jones and M. Sergot. A Formal Characterisation of Institutionalised Power. *Logic Journal of IGPL*, 4(3):427–443, 6 1995.

[Lepri *et al.*, 2018] B. Lepri, N. Oliver, E. Letouzé, A. Pentland, and P. Vinck. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology*, 31(4):611–627, 2018.

[Reuters, 2018] Reuters. Amazon ditched AI recruiting tool that favored men for technical jobs. *The Guardian*, Oct 2018. Available at https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine.

[Searle, 1995] J Searle. *The construction of social reality*. Free Press, New York, 1995.

[Theodorou *et al.*, 2017] A. Theodorou, R.H. Wortham, and J.J. Bryson. Designing and implementing transparency for real time inspection of autonomous robots. *Connection Science*, 29(3):230–241, 7 2017.

[Turiel, 2002] E. Turiel. *The culture of morality: Social development, context, and conflict*. Cambridge University Press, 2002.

[Van de Poel, 2013] I. Van de Poel. Translating values into design requirements. In *Philosophy and engineering: Reflections on practice, principles and process*, pages 253–266. Springer, 2013.

[Vázquez-Salceda *et al.*, 2007] J. Vázquez-Salceda, H. Aldewereld, D. Grossi, and F.P.M. Dignum. From human regulations to regulated software agents' behavior. *Artificial Intelligence and Law*, 16(1):73–87, 2007.