Towards Modeling Morality Computationally with Logic Programming

Ari Saptawijaya[⋆] and Luís Moniz Pereira

Centro de Inteligência Artificial (CENTRIA), Departamento de Informática
Faculdade de Ciências e Tecnologia, Univ. Nova de Lisboa, 2829-516 Caparica, Portugal
ar.saptawijaya@campus.fct.unl.pt, lmp@fct.unl.pt

Abstract. We investigate the potential of logic programming (LP) to model morality aspects studied in philosophy and psychology. We do so by identifying three morality aspects that appear in our view amenable to computational modeling by appropriately exploiting LP features: dual-process model (reactive and deliberative) in moral judgments; justification of moral judgments by contractualism; and intention in moral permissibility. The research aims at developing an LP-based system with features needed in modeling moral settings, putting emphasis on modeling these above mentioned morality aspects. We have currently co-developed two essential ingredients of the LP system, i.e., abduction and logic program updates, by exploiting the benefits of tabling features in logic programs. They serve as the basis for our whole system, into which other reasoning facets will be integrated, to model the surmised morality aspects. Moreover, we touch upon the potential of our ongoing studies of LP based cognitive features for the emergence of computational morality, in populations of agents enabled with the capacity for intention recognition, commitment and apology.

Keywords: abduction, program updates, argumentation, reactive behavior, deliberative reasoning, morality, emergence.

1 Introduction

The importance of imbuing agents more or less autonomous, with some capacity for moral decision making has recently gained a resurgence of interest from the artificial intelligence community, bringing together perspectives from philosophy and psychology. A new field of enquiry, *computational morality* (also known as machine ethics, machine morality, artificial morality and computational ethics) has emerged from their interaction, as emphasized e.g., in [5, 17, 65]. Research in artificial intelligence particularly focuses on how to employ various techniques, namely from computational logic, machine learning and multi-agent systems, in order to computationally model moral decision making (to some improved extent). The overall result is therefore not only important for equipping agents with the capacity for moral decision making, but also for helping us better understand morality, through the creation and testing of computational models of ethical theories.

[⋆] Affiliated with Faculty of Computer Science at University of Indonesia, Depok, Indonesia.

Recent results in computational morality have mainly focused on equipping agents with particular ethical theories, cf. [6] and [51] for modeling utilitarianism and deontological ethics, respectively. Another line of work attempts to provide a general framework to encode moral rules, in favor of deontological ethics, without resorting to a set of specific moral rules, e.g., [11]. The techniques employed include machine learning techniques, e.g., case-based reasoning [39], artificial neural networks [21], inductive logic programming [3, 7], and logic-based formalisms e.g., deontic logic [11] and non-monotonic logics [51]. The use of these latter formalisms has only been proposed rather abstractly, with no further investigation on its use pursued in detail and implemented.

Apart from the use of inductive logic programming in [3, 7], there has not much been a serious attempt to employ the Logic Programming (LP) paradigm in computational morality. Notwithstanding, we have preliminarily shown in [24, 44–48] that LP, with its currently available ingredients and features, lends itself well to the modeling of moral decision making. In these works, we particularly benefited from abduction [30], stable model [19] and well-founded model [64] semantics, preferences [15], and probability [9], on top of evolving logic programs [1], amenable to both self and external updating. LP-based modeling of morality is addressed at length, e.g., in [33].

Our research further investigates the appropriateness of LP to model morality, emphasizing morality aspects studied in philosophy and psychology, thereby providing an improved LP-based system as a testing ground for understanding and experimentation of such aspects and their applications. We particularly consider only some – rather than tackle all morality aspects – namely those pertinent to moral decision making, and, in our view, those particularly amenable to computational modeling by exploring and exploiting the appropriate LP features. Our research does not aim to propose some new moral theory, the task naturally belonging to philosophers and psychologists, but we simply uptake their known results off-the-shelf. We identify henceforth three morality aspects for the purpose of our work: dual-process model (reactive and deliberative) in moral judgments [13, 38], justification of moral judgments by contractualism [58, 59], and the significance of intention in regard to moral permissibility [60].

The remainder of the paper is organized as follows. Section 2 discusses the state-of-the-art of approaches that have been sought in computational morality. In Section 3 we detail the potential of LP for computational morality in the context of our research goal, and give a direction on how LP can be exploited to model the three chosen morality aspects. Section 4 presents two novel implementation techniques for abduction and knowledge updates, which serve as basic ingredients of the system being developed. Section 5 summarizes an application concerning a princess-saving moral robot. We conclude, in Section 6, by pointing out the importance of cognitive abilities in what regards the emergence of cooperation and morality in populations of individuals, as fostered in our own work, and mention directions for the future in this respect.

2 State of the Art

The field of computational morality, known too as machine ethics [5], has started growing, motivated by various objectives, e.g., to equip machines with the capability of moral decision making in certain domains, to aid (or even train) humans in moral decision making, to provide a general modeling framework for moral decision making, and to understand morality better by experimental model simulation.

The purpose of 'artificial morality' in [14] is somewhat different. The aim is to show that moral agents successfully solve social problems that amoral agents cannot. This work is based on the techniques from game theory and evolutionary game theory, where social problems are abstracted into social dilemmas, such as Prisoner's Dilemma and Chicken, and interactions of agents in games are implemented using Prolog.

The systems TruthTeller and SIROCCO were developed based on case-based reasoning [39]. Both systems implement the ethical approach casuistry [29]. TruthTeller is designed to accept a pair of ethical dilemmas and describe the salient similarities and differences between the cases, from both an ethical and a pragmatic perspective. On the other hand, SIROCCO is constructed to accept an ethical dilemma and to retrieve similar cases and ethical principles relevant to the ethical dilemma presented.

In [21], artificial neural networks, i.e., simple recurrent networks, are used with the main purpose of understanding morality from the philosophy of ethics viewpoint, and in particular to explore the dispute between moral particularism and generalism. The learning mechanism of neural networks is used to classify moral situations by training such networks with a number of cases, involving actions concerning killing and allowing to die, and then using the trained networks to classify test cases.

Besides case-based reasoning and artificial neural networks, another machine learning technique that is also utilised in the field is inductive logic programming, as evidenced by two systems: MedEthEx [7] and EthEl [3]. These are advisor systems in the domain of biomedicine, based on prima facie duty theory [53] from biomedical ethics. MedEthEx is dedicated to give advice for dilemmas in biomedical fields, while EthEl serves as a medication-reminder system for the elderly and as a notifier to an overseer if the patient refuses to take the medication. The latter system has been implemented in a real robot, the Nao robot, being capable to find and walk toward a patient who needs to be reminded of medication, to bring the medication to the patient, to engage in a natural-language exchange, and to notify an overseer by email when necessary [4].

Jeremy is another advisor system [6], which is based upon Jeremy Bentham's act utilitarianism. The moral decision is made in a straightforward manner. For each possible decision d, there are three components to consider with respect to each person p affected: the intensity of pleasure/displeasure (I_p), the duration of the pleasure/displeasure (D_p) and the probability that this pleasure/displeasure will occur (P_p). Total net pleasure for each decision is then computed: $total_d = \Sigma_{p \in Person}(I_p \times D_p \times P_p)$. The right decision is the one giving the highest total net pleasure.

Apart from the adoption of utilitarianism, like in the Jeremy system, in [51] the deontological tradition is considered having modeling potential, where the first formulation of Kant's categorical imperative [32] is concerned. Three views are taken into account in reformulating Kant's categorical imperative for the purpose of machine ethics: mere consistency, common-sense practical reasoning, and coherency. To realize the first view, a form of deontic logic is adopted. The second view benefits from nonmonotonic logic, and the third view presumes ethical deliberation to follow a logic similar to that of belief revision. All of them are considered abstractly and there seems to exist no implementation on top of these formalisms.

Deontic logic is envisaged in [11], as a framework to encode moral rules. The work resorts to Murakami's axiomatized deontic logic, an axiomatized utilitarian formulation of multiagent deontic logic, that is used to decide operative moral rule to attempt to arrive at an expected moral decision. This is achieved by seeking a proof for the expected moral outcome that follows from candidate operative moral rules.

The use of category theory appears in [12], where it is used as the formal framework to reason over logical systems, taking the view that logical systems are being deployed to formalize ethical codes. The work is strongly based on Piaget's position [28]. As argued in [12], this idea of reasoning *over* – instead of reasoning *in* – logical systems, favors post-formal Piaget's stages beyond his well-known fourth stage. In other words, category theory is used as the meta-level of moral reasoning.

Belief-Desire-Intention (BDI) model [10] is adopted in SophoLab [66], a framework for experimental computational philosophy, which is implemented with JACK agent programming language. In this framework, the BDI model is extended with the deontic-epistemic-action logic [63] to make it suitable for modeling moral agents. SophoLab is used, for example, to study negative moral commands and two different utilitarian theories, viz. act and rule utilitarianism.

We have preliminarily shown, in [44, 45], the use of integrated LP features to model the classic trolley problem[1] [18] and the double effect[2] as the basis of moral decisions on these dilemmas. In particular, possible decisions in a moral dilemma are modeled as abducibles, and abductive stable models are computed to capture abduced decisions and their consequences. Models violating integrity constraints, i.e., those that contain actions violating the double effect principle, are ruled out. A posteriori preferences, including the use of utility functions, are eventually applied to prefer models that characterize more preferred moral decisions. The computational models, based on the prospective logic agent architecture (shown in Figure 1) and developed on top of XSB Prolog, successfully deliver moral decisions in accordance with the double effect principle. They conform to the results of empirical experiments conducted in cognitive science [27] and law [40]. In [46–48], the computational models of the trolley problem dilemmas are extended, using the same LP system, by considering another moral principle, viz. the triple effect principle [31]. The work was extended further, in [24], by

[1] The trolley dilemmas, adapted from [27]: "There is a trolley and its conductor has fainted. The trolley is headed toward five people walking on the track. The banks of the track are so steep that they will not be able to get off the track in time." The two main cases of the trolley dilemmas:

Bystander: Hank is standing next to a switch that can turn the trolley onto a side track, thereby preventing it from killing the five people. However, there is a man standing on the side track. Hank can throw the switch, killing him; or he can refrain from doing so, letting the five die. Is it morally permissible for Hank to throw the switch?

Footbridge. Ian is on the bridge over the trolley track, next to a heavy man, which he can shove onto the track in the path of the trolley to stop it, preventing the killing of five people. Ian can shove the man onto the track, resulting in death; or he can refrain from doing so, letting the five die. Is it morally permissible for Ian to shove the man?

[2] The doctrine of double effect states that doing harms to another individual is permissible if it is the foreseen consequence of an action that will lead to a greater good, but is impermissible as an intended means to such greater good [27].

introducing various aspects of uncertainty, achieved using P-log [9], into trolley problem dilemmas, both from the view of oneself and from that of others. The latter by tackling the case of jury trials to proffer rulings beyond reasonable doubt.

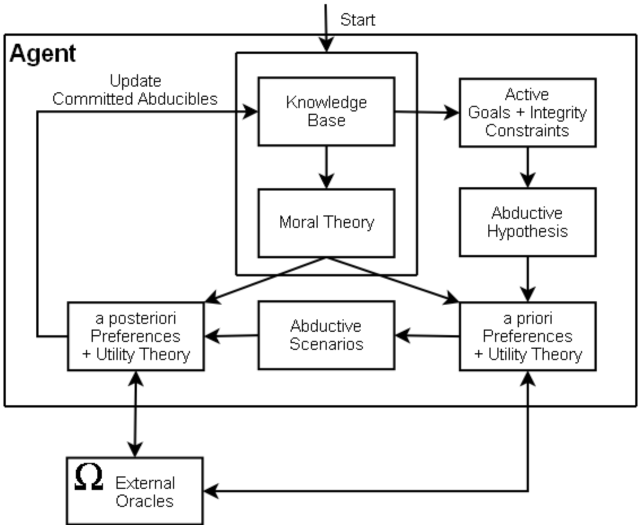

Fig. 1. Prospective logic agent architecture

3 Potential of Logic Programming for Computational Morality

Logic programming (LP) offers a formalism for declarative knowledge representation and reasoning. It thus has been used to solve problems in diverse areas of artificial intelligence (AI), e.g., planning, diagnosis, decision making, hypothetical reasoning, natural language processing, machine learning, etc.

Our research aims at developing an LP-based system with features needed in modeling moral settings, to represent agents' knowledge in those settings, and to allow moral reasoning under morality aspects studied in philosophy and moral psychology.

The choice of the LP paradigm is due to its potentials to model morality. For one thing, it allows moral rules, being employed when modeling some particular aspects, to be specified declaratively. For another, research in LP has provided us with necessary ingredients that are promising enough at being adept to model morality, e.g. default negation is suitable for expressing exception in moral rules, abductive logic programming [30] and (say) stable model semantics [19] can be used to generate possible decisions along with their moral consequences, and preferences [15] are appropriate for enabling to choose among moral decisions or moral rules.

We have identified three important morality aspects, from the fields of philosophy and psychology, that in our view are amenable to computational model by exploiting appropriate LP features: (1) the dual-process of moral judgments [13, 38], (2) justification of moral judgments [58, 59], and (3) the significance of intention in regard to moral permissibility [60]. The choice of these aspects is made due to their conceptual closeness with existing logic-based formalisms under available LP approaches as explained below. The choice is not meant to be exhaustive (as morality is itself a complex subject), in the sense that there may be other aspects that can be modeled computationally, particularly in LP. On the other hand, some aspects are not directly amenable to model in LP (at least for now), e.g., to model the role of emotions in moral decision making.

The development of the system is driven by the above three important morality aspects. The following LP features, being an integral part of the agent's observe-think-decide-act life cycle, serve as basic ingredients for the system to bring about moral reasoning:

1. **Knowledge updates, be they external or internal**. This is important due to constantly changing environment, and also particularly relevant in moral settings where an agent's moral rules are susceptible to updating, including when considering judgments about others, which are often made in spite of incomplete, or even contradictory, information.
2. **Deliberative and reactive decision making**. These two modes of decision making correspond to the dual-process model of moral judgments. Furthermore, reactive behavior can be employed for fast and frugal decision making with pre-compiled moral rules, thereby avoiding costly deliberative reasoning performed every time.

Given these basic ingredients, the whole process of moral decision making are particularly supported with the following capabilities of the system, justified by our need of modeling morality:

- To exclude undesirable actions. This is important when we must rule out actions that are morally impermissible under the moral rules being considered.
- To recognize intentions behind available actions, particularly in cases where intention is considered a significant aspect when addressing permissibility of actions.
- To generate alternatives of actions along with their consequences. In moral dilemmas agents are confronted with more than one course of action. They should be made available, along with their moral consequences, for an agent to ultimately decide about them.
- To prefer amongst alternatives of actions based on some measures. Preferences are relevant in moral settings, e.g. in case of several actions being permissible, preferences can be exercised to prefer one of them on the grounds of some criteria. Moreover, it is realistic to consider uncertainty of intentions, actions or consequences, including to perform counterfactual reasoning, in which cases preferences based on probability measures play a role.
- To inspect consequences of an action without deliberate imposition of the action itself as a goal. This is needed for instance to distinguish moral consequences of actions performed by an agent to satisfy its goals from those of its actions and side-effects performed unwittingly, not being part of the agent's goals.
- To provide an action with reasons for it (not) to be done. Reasons are used to justify permissibility of an action on grounds that one expects others to accept. In other words, morality in this way is viewed as striving towards argumentative consensus.

With respect to the first morality aspect, we look into recent approaches in combining deliberative and reactive logic-based systems [34, 35]. Inspired by these approaches, we have proposed two implementation techniques which are the basis for our system. First, we have improved the abduction system ABDUAL [2], employed for deliberative moral decision making in our previous work [24, 44–48]. We particularly explored the benefit of LP tabling mechanisms in abduction, to table abductive solutions for future reuse, resulting in a tabled abduction system TABDUAL [54]. Second, we have adapted evolving logic programs (EVOLP) [1], a formalism to model evolving agents, i.e., agents whose knowledge may dynamically change due to some (internal or external) updates. In EVOLP, updates are made possible by introducing the reserved predicate $assert/1$ into its language, whether in rule heads or rule bodies, which updates the program by the rule R, appearing in its only argument, whenever the assertion $assert(R)$ is true in a model; or retracts R in case $assert(not\ R)$ obtains in the model under consideration. We simplified EVOLP, in an approach termed EVOLP/R [55, 56], by restricting assertions to fluents only, whether internal or external world ones. We discuss both TABDUAL and EVOLP/R in Section 4.

The second morality aspect views moral judgments as those about the adequacy of justification and reasons for accepting or rejecting the situated employment of broad consensual principles, whilst allowing for exceptions. This view is supported by *contractualism* [58], one of the major schools in moral philosophy. Contractualism provides flexibility on the set of principles to justify moral judgments so long as no one could reasonably reject them, i.e., reasoning becomes an important feature [59]. Thus, morality can be viewed as (possibly defeasible) argumentative consensus, which is why contractualism is interesting from a computational and AI perspective. We are researching the applicability of argumentative frameworks, such as [16, 52, 62], to deal with this aspect.

Finally, we shall employ results on intention recognition, e.g., [23] for the third morality aspect, about intention in regard to moral permissibility. Counterfactuals will also play some role in uncovering possible implicit intentions, and "What if?" questions in order to reason retrospectively about past decisions. With regard to counterfactuals, both causal models [8, 41] and the extension of inspection points [43] to examine contextual side effects of counterfactual abduction may be considered, meaning foreseeable extraneous consequences in future or past hypothetical scenarios.

The lighter conceptual and implementation advantages of EVOLP/R will help in combining with TABDUAL, to model both reactive and deliberative reasoning. Their combination also provides the basis for other reasoning facets needed in modeling other morality aspects, notably: argumentative frameworks and intention recognition to deal with the second and the third aspects, respectively.

4 TABDUAL and EVOLP/R

We recently proposed novel implementation techniques, both in abduction and logic program updates, by employing tabling mechanisms in LP. Tabling mechanisms in LP, known as the tabled logic programming paradigm, is currently supported by a number of Prolog systems, to different extent. Tabling affords solutions reuse, rather than recomputing them, by keeping in tables subgoals and their answers obtained by query evaluation. Our techniques are realized in XSB Prolog [61], one of the most advanced tabled LP systems, with features such as tabling over default negation, incremental tabling, answer subsumption, call subsumption, and threads with shared tables.

4.1 Tabled Abduction (TABDUAL)

The basic idea behind tabled abduction (its prototype is termed TABDUAL) is to employ tabling mechanisms in logic programs in order to reuse priorly obtained abductive solutions, from one abductive context to another. It is realized via a program transformation of abductive normal logic programs. Abduction is subsequently enacted on the transformed program.

The core transformation of TABDUAL consists of an innovative re-uptake of prior abductive solution entries in tabled predicates and relies on the dual transformation [2]. The dual transformation, initially employed in ABDUAL [2], allows to more efficiently handle the problem of abduction under negative goals, by introducing their positive dual counterparts. It does not concern itself with programs having variables. In TABDUAL, the dual transformation is refined, to allow it dealing with such programs. The first refinement helps ground (dualized) negative subgoals. The second one allows to deal with non-ground negative goals.

As TABDUAL is implemented in XSB, it employs XSB's tabling as much as possible to deal with loops. Nevertheless, tabled abduction introduces a complication concerning some varieties of loops. Therefore, the core TABDUAL transformation has been adapted, resorting to a pragmatic approach, to cater to all varieties of loops in normal logic programs, which are now complicated by abduction.

From the implementation viewpoint, several pragmatic aspects have been examined. First, because TABDUAL allows for modular mixes between abductive and non-abductive program parts, one can benefit in the latter part by enacting a simpler translation of predicates in the program comprised just of facts. It particularly helps avoid superfluous transformation of facts, which would hinder the use of large factual data. Second, we address the issue of potentially heavy transformation load due to producing the *complete* dual rules (i.e., all dual rules regardless of their need), if these are constructed in advance by the transformation (which is the case in ABDUAL). Such a heavy dual transformation makes it a bottleneck of the whole abduction process. Two approaches are provided to realizing the dual transformation *by-need*: creating and tabling all dual rules for a predicate only on the first invocation of its negation, or, in contrast, lazily generating and storing its dual rules in a trie (instead of tabling), only as new alternatives are required. The former leads to an eager (albeit by-need) tabling of dual rules construction (under local table scheduling), whereas the latter permits a by-need-driven lazy one (in lieu of batched table scheduling). Third, TABDUAL provides a system predicate that permits accessing ongoing abductive solutions. This is a useful feature and extends TABDUAL's flexibility, as it allows manipulating abductive solutions dynamically, e.g., preferring or filtering ongoing abductive solutions, e.g., checking them explicitly against nogoods at predefined program points.

We conducted evaluations of TABDUAL with various objectives, where we examine five TABDUAL variants of the same underlying implementation by separately factoring out TABDUAL's most important distinguishing features. They include the evaluations of: (1) the benefit of tabling abductive solutions, where we employ an example from declarative debugging, now characterized as abduction [57], to debug incorrect solutions of logic programs; (2) the three dual transformation variants: complete, eager by-need, and lazy by-need, where the other case of declarative debugging, that of debugging missing solutions, is employed; (3) tabling so-called *nogoods* of subproblems in the context of abduction (i.e., abductive solution candidates that violate constraints), where it can be shown that tabling abductive solutions can be appropriate for tabling nogoods of subproblems; (4) programs with loops, where the results are compared with ABDUAL, showing that TABDUAL provides more correct and complete results. Additionally, we show how TABDUAL can be applied in action decision making under hypothetical reasoning, and in a real medical diagnosis case [57].

4.2 Restricted Evolving Logic Programs (EVOLP/R)

We have defined the language of EVOLP/R in [56], adapted from that of Evolving Logic Programs (EVOLP) [1], by restricting updates at first to fluents only. More precisely, every fluent F is accompanied by its fluent complement $\sim F$. Retraction of F is thus achieved by asserting its complement $\sim F$ at the next timestamp, which renders F supervened by $\sim F$ at later time; thereby making F false. Nevertheless, it allows paraconsistency, i.e., both F and $\sim F$ may hold at the same timestamp, to be dealt with by the user as desired, e.g., with integrity constraints or preferences.

In order to update the program with rules, special fluents (termed *rule name fluents*) are introduced to identify rules uniquely. Such a fluent is placed in the body of a rule, allowing to turn the rule on and off, cf. Poole's "naming device" [50]; this being achieved by asserting or retracting the rule name fluent. The restriction thus requires that all rules be known at the start.

EVOLP/R is realized by a program transformation and a library of system predicates. The transformation adds some extra information, e.g., timestamps, for internal processing. Rule name fluents are also system generated and added in the transform. System predicates are defined to operate on the transform by combining the usage of two features of tabling in XSB Prolog: incremental and answer subsumption tabling.

Incremental tabling of fluents allows to automatically maintain the consistency of program states, analogously to assumption based truth-maintenance system in artificial intelligence, due to assertion and retraction of fluents, by relevantly propagating their consequences. Answer subsumption of fluents, on the other hand, allows to address the frame problem by automatically keeping track of their latest assertion or retraction, whether obtained as updated facts or concluded by rules. Despite being pragmatic, employing these tabling features has profound consequences in modeling agents, i.e., it permits separating higher-level declarative representation and reasoning, as a mechanism pertinent to agents, from a world's inbuilt reactive laws of operation. The latter are relegated to engine-level enacted tabling features (in this case, the incremental and answer subsumption tabling); they are of no operational concern to the problem representation level.

Recently, in [55], we refined the implementation technique by fostering further incremental tabling, but leaving out the problematic use of the answer subsumption feature. The main idea is the perspective that knowledge updates (either self or world wrought changes) occur whether or not they are queried, i.e., the former take place independently of the latter. That is, when a fluent is true at a particular time, its truth lingers on independently of when it is queried. Fluent updates are initially kept pending in the database, and on the initiative of top-goal queries, i.e., by need only, incremental assertions make these pending updates become active (if not already so), but only those with timestamps up to an actual query time. Such assertions automatically trigger system-implemented incremental upwards propagation and tabling of fluent updates. In order to delimit answers in the table, which in some cases could lead to iterative non-termination, the propagation is bounded by some given predefined upper global time limit. Though foregoing answer subsumption, recursion through the frame axiom can thus still be avoided, and a direct access to the latest time a fluent is true is made possible by means of existing table inspection predicates. Benefiting from the automatic upwards propagation of fluent updates, the program transformation in the new implementation technique becomes simpler than our previous one, in [56]. Moreover, it demonstrates how the dual program transformation, introduced in the context of abduction and used in TABDUAL, is employed for helping propagate the dual negation complement of a fluent incrementally, in order to establish whether the fluent is still true at some time point or if rather its complement is. In summary, the refinement affords us a form of controlled, though automatic, system level truth-maintenance, up to the actual query time. It reconciles high-level top-down deliberative reasoning about a query, with autonomous low-level bottom-up world reactivity to ongoing updates.

4.3 LP Implementation Remarks: What's Still Left to be Done

Departing from the current state of our research, the integration of TABDUAL and EVOLP/R becomes naturally the next step. We shall define how reactive behavior (described as maintenance goals in [34, 35]) can be achieved in the integrated system. An idea would be to use integrity constraints as sketched below:

$$\begin{gathered} assert(trigger(conclusion)) \leftarrow condition \\ false \leftarrow trigger(conclusion), not\ do(conclusion) \\ do(conclusion) \leftarrow some\_actions \end{gathered}$$

Accordingly, fluents of the form $trigger(conclusion)$ can enact the launch of maintenance goals, in the next program update state, by satisfying any corresponding integrity constraints. Fluents of the form $\sim trigger(conclusion)$, when asserted, will refrain any such launching, in the next program update state. In line with such reactive behavior, is fast and frugal moral decision making, which can be achieved via pre-compiled moral rules (cf. heuristics for decision making in law [20]).

Once TABDUAL and EVOLP/R are integrated, we are ready to model moral dilemmas, focusing on the first morality aspect, starting from easy scenarios (low-conflict) to difficult scenarios (high-conflict). In essence, moral dilemmas will serve as vehicles to model and to test this morality aspect (and also others). The inclusion of other ingredients into the system, notably argumentation and intention recognition (including counterfactuals), is in our research agenda. The choice of their appropriate formalisms still need to be defined, driven by the salient features of the second and the third morality aspects to model.

5 Application: a Princess Saviour Moral Robot

Apart from dealing with incomplete information, knowledge updates (as realized by EVOLP/R) are essential to account for moral updating and evolution. It concerns the adoption of new (possibly overriding) moral rules on top of those an agent currently follows. Such adoption is often necessary when the moral rules one follows have to be revised in the light of situations faced by the agent, e.g. when other moral rules are contextually imposed by an authority.

Moral updating is not only relevant in a real world setting, but also in imaginary ones, e.g., in interactive storytelling; cf. [37], where the robot in the story must save the princess in distress while it should also follow (possibly conflicting) moral rules that may change dynamically as imposed by the princess and may conflict with the robot's survival.

It does so by employing Prospective Logic Programming (PLP), a declarative framework supporting the specification of autonomous agents capable of anticipating and reasoning about hypothetical future scenarios. This capability for prediction is essential for proactive agents working with partial information in dynamically changing environments. The work explores the use of state-of-the-art declarative non-monotonic reasoning in the field of interactive storytelling and emergent narratives and is supported by a concrete graphics application prototype to enact the story of a princess saved by a robot imbued with moral reasoning. Note that ACORDA [36], an ad hoc abduction implementation on top of the updates system EVOLP [1], is used in the previous LP implementation for this application, without exploiting tabling features. From that experience, we now move on to a new single integrated system, as described in Section 4.3, that fully exploits tabling technology.

In order to test the PLP framework and the integration of a virtual environment for interactive storytelling, a simplified scenario was developed. In this fantasy setting, an archetypal princess is held in a castle awaiting rescue. The unlikely hero is an advanced robot, imbued with a set of declarative rules for decision making and moral reasoning. As the robot is asked to save the princess in distress, he is confronted with an ordeal. The path to the castle is blocked by a river, crossed by two bridges. Standing guard at each of the bridges are minions of the wizard which originally imprisoned the princess. In order to rescue the princess, he will have to defeat one of the minions to proceed.[3]

Prospective reasoning is the combination of pre-preference hypothetical scenario generation into the future plus post-preference choices taking into account the imagined consequences of each preferred scenario. By reasoning backwards from this goal, the agent generates three possible hypothetical scenarios for action. Either it crosses one of the bridges, or it does not cross the river at all, thus negating satisfaction of the rescue goal. In order to derive the consequences for each scenario, the agent has to reason forwards from each available hypothesis. As soon as these consequences are known, meta-reasoning techniques can be applied to prefer amongst the partial scenarios.

This simple scenario already illustrates the interplay between different LP techniques and demonstrates the advantages gained by combining their distinct strengths.

[3] More at online demo: `http://centria.di.fct.unl.pt/~lmp/publications/slides/padl10/quick_moral_robot.avi`

Namely, the integration of top-down, bottom-up, hypothetical, moral and utility-based reasoning procedures results in a flexible framework for dynamic agent specification. The open nature of the framework embraces the possibility of expanding its use to yet other useful models of cognition such as counterfactual reasoning and theories of mind.

6 Emergence and Computational Morality

The mechanisms of emergence and evolution of cooperation in populations of abstract individuals with diverse behavioral strategies in co-presence have been undergoing mathematical study via Evolutionary Game Theory, inspired in part on Evolutionary Psychology. Their systematic study resorts as well to implementation and simulation techniques, thus enabling the study of aforesaid mechanisms under a variety of conditions, parameters, and alternative virtual games. The theoretical and experimental results have continually been surprising, rewarding, and promising.

Recently, in our own work we have initiated the introduction, in such groups of individuals, of cognitive abilities inspired on techniques and theories of Artificial Intelligence, namely those pertaining to both Intention Recognition and to Commitment (separately and jointly), encompassing errors in decision-making and communication noise. As a result, both the emergence and stability of cooperation become reinforced comparatively to the absence of such cognitive abilities. This holds separately for Intention Recognition and for Commitment, and even more when they are engaged jointly.

From the viewpoint of population morality, the modeling of morality in individuals using appropriate LP features (like abduction, knowledge updates, argumentation, counterfactual reasoning, and others touched upon our research) within a networked population shall allow them to dynamically choose their behavior rules, rather than to act from a predetermined set. That is, individuals will be able to hypothesize, to look at possible future consequences, to (probabilistically) prefer, to deliberate, to take into account history, to adopt and fine tune game strategies.

Indeed, the study of properties like the emergent cooperative and tolerant collective behavior in populations of complex networks, very much needs further investigation of the cognitive core in each of the social atoms of the individuals in such populations (albeit by appropriate LP features). See our own studies on intention recognition and commitments, such as in e.g. [22, 23, 25, 26, 49]). In particular, the references [42, 49] aim to sensitize the reader to these Evolutionary Game Theory based studies and issues, which are accruing in importance for the modeling of minds with machines, with impact on our understanding of the evolution of mutual tolerance, cooperation and commitment. In doing so, they also provide a coherent bird's-eye view of our own varied recent work, whose more technical details, references and results are spread throughout a number of publishing venues, to which the reader is referred therein for a fuller support of claims where felt necessary.

In those works we model intention recognition within the framework of repeated interactions. In the context of direct reciprocity, intention recognition is performed using the information about past *direct* interactions. We study this issue using the well-known repeated Prisoner's Dilemma (PD), i.e., so that intentions can be inferred from past individual experiences. Naturally, the same principles could be extended to cope with indirect information, as in indirect reciprocity. This eventually introduces moral judgment and concern for individual reputation, which constitutes “per se” an important area where intention recognition may play a pivotal role.

In our work too, agents make commitments towards others, they promise to enact their play moves in a given manner, in order to influence others in a certain way, often by dismissing more profitable options. Most commitments depend on some incentive that is necessary to ensure that the action is in the agent’s interest and thus, may be carried out to avoid eventual penalties. The capacity for using commitment strategies effectively is so important that natural selection may have shaped specialized signaling capacities to make this possible. And it is believed to have an incidence on the emergence of morality. Not only bilaterally wise but also in public goods games, where in both cases we are presently researching into complementing commitment with apology.

Modeling such cognitive capabilities in individuals, and in populations, may well prove useful for the study and understanding of ethical robots and their emergent behavior in groups, so as to make them implementable in future robots and their swarms, and not just in the simulation domain but in the real world engineering one as well.

7 Message in a Bottle

In realm of the individual, Logic Programming is a vehicle for the computational study and teaching of morality, namely in its modeling of the dynamics of knowledge and cognition of agents.
In the collective realm, norms and moral emergence has been studied computationally in populations of rather simple-minded agents.
By bridging these realms, cognition affords improved emerged morals in populations of situated agents.

Acknowledgements AS acknowledges the support of Fundação para a Ciência e a Tecnologia (FCT/MEC) Portugal, grant SFRH/BD/72795/2010.

References

[1] J. J. Alferes, A. Brogi, J. A. Leite, and L. M. Pereira. Evolving logic programs. In *JELIA 2002*, volume 2424 of *LNCS*, pages 50–61. Springer, 2002.

[2] J. J. Alferes, L. M. Pereira, and T. Swift. Abduction in well-founded semantics and generalized stable models via tabled dual programs. *Theory and Practice of Logic Programming*, 4(4):383–428, 2004.

[3] M. Anderson and S. L. Anderson. EthEl: Toward a principled ethical eldercare robot. In *Procs. AAAI Fall 2008 Symposium on AI in Eldercare*, 2008.

[4] M. Anderson and S. L. Anderson. Robot be good: A call for ethical autonomous machines. In *Scientific American*, October 2010.

[5] M. Anderson and S. L. Anderson, editors. *Machine Ethics*. Cambridge U. P., 2011.

[6] M. Anderson, S. L. Anderson, and C. Armen. Towards machine ethics: implementing two action-based ethical theories. In *Procs. AAAI 2005 Fall Symposium on Machine Ethics*, 2005.

[7] M. Anderson, S. Anderson, and C. Armen. MedEthEx: a prototype medical ethics advisor. In *IAAI 2006*, 2006.

[8] C. Baral and M. Hunsaker. Using the probabilistic logic programming language P-log for causal and counterfactual reasoning and non-naive conditioning. In *IJCAI 2007*, 2007.

[9] C. Baral, M. Gelfond, and N. Rushton. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144, 2009.

[10] M. E. Bratman. *Intention, Plans and Practical Reasoning*. Harvard University Press, 1987.

[11] S. Bringsjord, K. Arkoudas, and P. Bello. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, 21(4):38–44, 2006.

[12] S. Bringsjord, J. Taylor, B. van Heuveln, K. Arkoudas, M. Clark, and R. Wojtowicz. Piagetian roboethics via category theory: Moving beyond mere formal operations to engineer robots whose decisions are guaranteed to be ethically correct. In M. Anderson and S. L. Anderson, editors, *Machine Ethics*. Cambridge U. P., 2011.

[13] F. Cushman, L. Young, and J. D. Greene. Multi-system moral psychology. In J. M. Doris, editor, *The Moral Psychology Handbook*. Oxford University Press, 2010.

[14] P. Danielson. *Artificial Morality: Virtuous Robots for Virtual Games*. Routledge, 1992.

[15] P. Dell'Acqua and L. M. Pereira. Preferential theory revision. *J. of Applied Logic*, 5(4):586–601, 2007.

[16] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[17] Economist. Morals and the machine. Main Front Cover and Leaders (page 13), The Economist, June 2nd-8th 2012.

[18] P. Foot. The problem of abortion and the doctrine of double effect. *Oxford Review*, 5:5–15, 1967.

[19] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *5th Intl. Logic Programming Conf.* MIT Press, 1988.

[20] G. Gigerenzer and C. Engel, editors. *Heuristics and the Law*. MIT Press, 2006.

[21] M. Guarini. Computational neural modeling and the philosophy of ethics: Reflections on the particularism-generalism debate. In M. Anderson and S. L. Anderson, editors, *Machine Ethics*. Cambridge U. P., 2011.

[22] T. A. Han. *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation: From Artificial Intelligence Techniques to Evolutionary Game Theory Models*, volume 9 of *SAPERE*. Springer, 2013. ISBN 978-3-642-37511-8.

[23] T. A. Han and L. M. Pereira. State-of-the-art of intention recognition and its use in decision making. *AI Communications*, 26(2):237–246, 2013.

[24] T. A. Han, A. Saptawijaya, and L. M. Pereira. Moral reasoning under uncertainty. In *LPAR-18*, volume 7180 of *LNCS*, pages 212–227. Springer, 2012.

[25] T. A. Han, L. M. Pereira, F. C. Santos, and T. Lenearts. Good agreements make good friends. *Nature Scientific Reports*, 3(2695):DOI: 10.1038/srep02695, 2013.

[26] T. A. Han, L. M. Pereira, F. C. Santos, and T. Lenearts. Why Is It So Hard to Say Sorry: The Evolution of Apology with Commitments in the Iterated Prisoner's Dilemma. In *IJCAI 2013*, pages 177–183. AAAI Press, 2013.

[27] M. D. Hauser. *Moral Minds: How Nature Designed Our Universal Sense of Right and Wrong*. Little Brown, 2007.

[28] B. Inhelder and J. Piaget. *The Growth of Logical Thinking from Childhood to Adolescence*. Basic Books, 1958.

[29] A. R. Jonsen and S. Toulmin. *The Abuse of Casuistry: A History of Moral Reasoning*. University of California Press, 1988.

[30] A. Kakas, R. Kowalski, and F. Toni. The role of abduction in logic programming. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 5. Oxford U. P., 1998.

[31] F. M. Kamm. *Intricate Ethics: Rights, Responsibilities, and Permissible Harm*. Oxford U. P., 2006.

[32] I. Kant. *Grounding for the Metaphysics of Morals, translated by J. Ellington*. Hackett, 1981.

[33] R. Kowalski. *Computational Logic and Human Thinking: How to be Artificially Intelligent*. Cambridge U. P., 2011.

[34] R. Kowalski and F. Sadri. Abductive logic programming agents with destructive databases. *Annals of Mathematics and Artificial Intelligence*, 62(1):129–158, 2011.

[35] R. Kowalski and F. Sadri. A logic-based framework for reactive systems. In *RuleML 2012*, volume 7438 of *LNCS*, 2012.

[36] G. Lopes and L. M. Pereira. Prospective programming with ACORDA. In *ESCoR 2006 Workshop*, IJCAR'06, 2006.

[37] G. Lopes and L. M. Pereira. Prospective storytelling agents. In *PADL 2010*, volume 5937 of *LNCS*. Springer, 2010.

[38] R. Mallon and S. Nichols. Rules. In J. M. Doris, editor, *The Moral Psychology Handbook*. Oxford University Press, 2010.

[39] B. M. McLaren. Computational models of ethical reasoning: Challenges, initial steps, and future directions. *IEEE Intelligent Systems*, pages 29–37, 2006.

[40] J. Mikhail. Universal moral grammar: Theory, evidence, and the future. *Trends in Cognitive Sciences*, 11(4):143–152, 2007.

[41] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge U. P., 2009.

[42] L. M. Pereira. Evolutionary tolerance. In L. Magnani and L. Ping, editors, *PCS 2011*, volume 2 of *SAPERE*, pages 263–287. Springer, 2012.

[43] L. M. Pereira and A. M. Pinto. Inspection points and meta-abduction in logic programs. In *INAP 2009*, 2009.

[44] L. M. Pereira and A. Saptawijaya. Moral Decision Making with ACORDA. In *Local Procs. of LPAR 2007*, 2007.

[45] L. M. Pereira and A. Saptawijaya. Modelling Morality with Prospective Logic. In *EPIA 2007*, 2007.

[46] L. M. Pereira and A. Saptawijaya. Modelling Morality with Prospective Logic. *International Journal of Reasoning-based Intelligent Systems*, 1(3/4):209–221, 2009.

[47] L. M. Pereira and A. Saptawijaya. Computational Modelling of Morality. *The Association for Logic Programming Newsletter*, 22(1), 2009.

[48] L. M. Pereira and A. Saptawijaya. Modelling Morality with Prospective Logic. In M. Anderson and S. L. Anderson, editors, *Machine Ethics*, pages 398–421. Cambridge U. P., 2011.

[49] L. M. Pereira, T. A. Han, and F. C. Santos. Complex systems of mindful entities – on intention recognition and commitment. In L. Magnani, editor, *Model-Based Reasoning in Science and Technology: Theoretical and Cognitive Issues*, volume 8 of *SAPERE*. Springer, 2013.

[50] D. L. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36 (1):27–47, 1988.

[51] T. M. Powers. Prospects for a Kantian machine. *IEEE Intelligent Systems*, 21(4): 46–51, 2006.

[52] I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.

[53] W. D. Ross. *The Right and the Good*. Oxford University Press, 1930.

[54] A. Saptawijaya and L. M. Pereira. Tabled abduction in logic programs (technical communication of ICLP 2013). *Theory and Practice of Logic Programming, Online Supplement*, 13(4-5), 2013.

[55] A. Saptawijaya and L. M. Pereira. Incremental tabling for query-driven propagation of logic program updates. In *LPAR-19*, volume 8312 of *LNCS ARCoSS*. Springer, 2013.

[56] A. Saptawijaya and L. M. Pereira. Program updating by incremental and answer subsumption tabling. In *LPNMR 2013*, volume 8148 of *LNCS*. Springer, 2013.

[57] A. Saptawijaya and L. M. Pereira. Towards practical tabled abduction usable in decision making. In *KES-IDT 2013*, Frontiers of Artificial Intelligence and Applications (FAIA). IOS Press, 2013.

[58] T. M. Scanlon. Contractualism and utilitarianism. In A. Sen and B. Williams, editors, *Utilitarianism and Beyond*. Cambridge U. P., 1982.

[59] T. M. Scanlon. *What We Owe to Each Other*. Harvard University Press, 1998.

[60] T. M. Scanlon. *Moral Dimensions: Permissibility, Meaning, Blame*. Harvard University Press, 2008.

[61] T. Swift and D. S. Warren. XSB: Extending Prolog with tabled logic programming. *Theory and Practice of Logic Programming*, 12(1-2):157–187, 2012.

[62] F. Toni. Argumentative agents. In *Procs. Intl. Multiconference on Computer Science and Information Technology*, volume 5, 2010.

[63] J. van den Hoven and G-J. Lokhorst. Deontic logic and computer-supported computer ethics. *Metaphilosophy*, 33(3):376–386, 2002.

[64] A. van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of ACM*, 38(3):620–650, 1991.

[65] W. Wallach and C. Allen. *Moral Machines: Teaching Robots Right from Wrong*. Oxford U. P., 2009.

[66] V. Wiegel. *SophoLab; Experimental Computational Philosophy*. PhD thesis, Delft University of Technology, 2007.