Towards Computational Morality with Logic Programming

ARI SAPTAWIJAYA

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

submitted 12 May 2013; accepted 5 June 2013

Abstract

When autonomous agents are deployed in some field where moral dilemmas may arise, the need for imbuing them with some capacity of moral decision making becomes indispensable. *Computational morality* has emerged to address this issue, bringing together perspectives from different areas, including artificial intelligence. This PhD research investigates further the appropriateness of logic programming (LP) to model morality aspects studied in philosophy and psychology, those amenable to computational modeling, by exploiting appropriate LP features. The goals of the research are twofold: (1) to develop an LP based system with features needed in modeling moral settings, putting emphasis on modeling some morality aspects, and (2) to employ the developed system for modeling several moral situations and reasoning thereon, with respect to the morality aspects under consideration. We have currently co-developed two essential ingredients of the LP system, i.e., abduction and 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 identified morality aspects.

KEYWORDS: computational morality, abduction, program updates, argumentation, reactive behavior, deliberative reasoning.

1 Introduction

The importance of imbuing more or less autonomous agents with some capacity of 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, emphasized e.g., in (Wallach and Allen 2009; Anderson and Anderson 2011; Economist 2012). Research in artificial intelligence particularly focuses on how employing various techniques, e.g., from computational logic, machine learning and multi-agent systems, in order to computationally model, to some extent, moral decision making. 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.

Recent results in computational morality have mainly focused on equipping agents with particular ethical theories, cf. (Anderson et al. 2005) and (Powers 2006) 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., (Bringsjord et al. 2006). The techniques employed include machine learning techniques, e.g., case-based reasoning (McLaren 2006), artificial neural networks (Guarini 2011), inductive logic programming (Anderson et al. 2006; Anderson and Anderson 2008), and logical formalisms e.g., deontic logic (Bringsjord et al. 2006) and nonmonotonic logics (Powers 2006). The use of the latter formalisms has only been proposed abstractly, but no further investigation on its use has been pursued.

Apart from the use of inductive logic programming in (Anderson et al. 2006; Anderson and Anderson 2008), there has not been a serious attempt to employ the Logic Programming (LP) paradigm in computational morality. We have preliminarily shown in (Pereira and Saptawijaya 2007a; Pereira and Saptawijaya 2007b; Pereira and Saptawijaya 2009a; Pereira and Saptawijaya 2009b; Pereira and Saptawijaya 2011; Han et al. 2012) 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 (Kakas et al. 1998), stable model (Gelfond and Lifschitz 1988) and well-founded model (van Gelder et al. 1991), preferences (Dell'Acqua and Pereira 2007), and probability (Baral et al. 2009), on top of evolving logic programs (Alferes et al. 2002). Kowalski, inspired by these inroads, has addressed LP modeling of morality at length in (Kowalski 2011).

This PhD research will further investigate the appropriateness of LP to model morality aspects studied in philosophy and psychology, thereby providing an improved LP-based system as a testing ground for understanding and experimentation of these aspects and their applications. We shall particularly consider only some aspects – rather than tackle all morality aspects – namely those being pertinent to moral decision making, and, in our view, amenable to computational modeling by exploiting 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.

2 Background and overview of the existing literature

The field of computational morality, known too as machine ethics (Anderson and Anderson 2011), 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 systems TruthTeller and SIROCCO were developed by focusing reasoning on cases, viz. case-based reasoning (McLaren 2006). Both systems implement aspects of the ethical approach known as casuistry (Jonsen and Toulmin 1988). 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 (Guarini 2011), 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 (Anderson et al. 2006) and EthEl (Anderson and Anderson 2008). Both systems are advisor systems in the domain of biomedicine, based on prima facie duty theory (Ross 1930) 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 (Anderson and Anderson 2010).

Jeremy is another advisor system (Anderson et al. 2005), 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 (Powers 2006) the deontological tradition is considered having modeling potential, where the first formulation of Kant's categorical imperative (Kant 1981) 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 (Bringsjord et al. 2006), 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.

Belief-Desire-Intention (BDI) model (Bratman 1987) is adopted in SophoLab (Wiegel 2007), 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 (van den Hoven and Lokhorst 2002) 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.

(Pereira and Saptawijaya 2007a; Pereira and Saptawijaya 2007b) have preliminarily shown the use of integrated LP features to model the classic trolley problem dilemmas[1]

[1] The trolley dilemmas, adapted from (Hauser 2007): "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

and the double effect principle[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, 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 (Hauser 2007) and law (Mikhail 2007). In (Pereira and Saptawijaya 2009a; Pereira and Saptawijaya 2009b; Pereira and Saptawijaya 2011), 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 (Kamm 2006). (Han et al. 2012) further extends the work by introducing various aspects of uncertainty, achieved using P-log (Baral et al. 2009), 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.

3 Research Goals

The goals of the research are twofold: (1) to develop an LP based system with features needed in modeling moral settings, with the emphasis on modeling morality aspects – not on particular moral rules – that are relevant to the computational machinery of moral decision making, thereby providing flexibility of the machinery in terms of the plethora of moral rules that can be put in force, (2) to employ the developed system for modeling several moral situations and reasoning thereof, with respect to morality aspects under consideration, thereby showing that our understanding of moral behavior can in part be computationally modeled and implemented.

Whenever empirical results from morality-related fields (notably from moral psychology experiments) are available, they will be compared with those obtained from our computational model. That is, validating whether the computational results are corroborated by the empirical ones.

4 Current status of the research

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 ap-

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 (Hauser 2007).

propriate LP features: (1) the dual-process of moral judgments (Cushman et al. 2010; Mallon and Nichols 2010), (2) justification of moral judgments (Scanlon 1982; Scanlon 1998), and (3) the significance of intention in regard to moral permissibility (Scanlon 2008). 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.

In the first aspect, two systems are considered in moral decision making, i.e., intuitive/affective and rational/cognitive. With regards to this aspect, we shall look into recent approaches in combining deliberative and reactive logic-based systems (Kowalski and Sadri 2011; Kowalski and Sadri 2012). Inspired by these approaches, we have started to work on two features which will be the basis for our system. First, we have improved the abduction system ABDUAL, employed for deliberative moral decision making in our previous work (Pereira and Saptawijaya 2007a; Pereira and Saptawijaya 2007b; Pereira and Saptawijaya 2009a; Pereira and Saptawijaya 2009b; Pereira and Saptawijaya 2011; Han et al. 2012). We particularly explored the benefit of tabling in abduction, to table abductive solutions for future reuse, resulting in a tabled abduction system TABDUAL (Pereira and Saptawijaya 2012; Saptawijaya and Pereira 2013b). Second, we have adapted evolving logic programs (EVOLP) (Alferes et al. 2002), 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, 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 (Saptawijaya and Pereira 2013a), by restricting assertions to fluents only, whether internal or external world ones. The lighter conceptual and implementation advantages of EVOLP/R will help in combining with TABDUAL, to model both reactive and deliberative reasoning. Their combination will also be the basis for other reasoning facets, described in the subsequent paragraphs, in modeling the other two morality aspects. We discuss both TABDUAL and EVOLP/R in Section 5.

The second aspect views moral judgments as those about the adequacy of the justification and reasons for accepting or rejecting the situated employing, with accepted exceptions, of broad consensual principles. This view is supported by *contractualism* (Scanlon 1982), 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 is an important feature here (Scanlon 1998). In this way, morality can be viewed as (possibly defeasible) argumentative consensus, which is why contractualism is interesting from a computational and artificial intelligence perspective. We shall research the applicability of argumentative frameworks, such as (Dung 1995; Rahwan and Simari 2009; Toni 2010; Dung and Thang 2010), to deal with this aspect.

Finally, we shall employ results on intention recognition, e.g., (Han and Pereira 2011) and exploit their use for the third aspect, about intention in regard to moral permissibility. Counterfactuals will also play some role in uncovering possible implicit intentions. With regard to counterfactuals, both causal models (Baral and Hunsaker 2007; Pearl

2009) and the extension of inspection points (Pereira and Pinto 2009) to examine contextual side effects of counterfactual abduction may be considered.

5 Preliminary results

We are currently still at the stage of developing a minimum LP system with basic ingredients, viz. abduction and knowledge updates, to allow an agent modeling moral dilemmas and making moral decisions.

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 (Alferes et al. 2004). The dual transformation, initially employed in ABDUAL (Alferes et al. 2004), 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 (Saptawijaya and Pereira 2013c), 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 (Saptawijaya and Pereira 2013c).

Restricted Evolving Logic Programs (EVOLP/R) The development of EVOLP/R is currently ongoing. We have defined the language of EVOLP/R, adapted from that of Evolving Logic Programs (EVOLP) (Alferes et al. 2002), 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" (Poole 1988); this being achieved by asserting or retracting the rule name fluent. The restriction thus requires that all rules be known at the start.

From the implementation viewpoint, we exploited Prolog tabling mechanisms, notably these two features of 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.

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 incremental and answer subsumption tabling.

6 Open issues and expected achievements

The integration of TABDUAL and EVOLP/R becomes naturally the next step. We shall define how reactive behavior (described as maintenance goals in (Kowalski and Sadri 2011; Kowalski and Sadri 2012)) can be achieved in the integrated system. An idea would be to use integrity constraints as sketched below:

$$assert(trigger(conclusion)) \leftarrow condition$$
$$false \leftarrow trigger(conclusion), not\ do(conclusion)$$
$$do(conclusion) \leftarrow some\_actions$$

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 ~*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 (Gigerenzer and Engel 2006)).

Once TABDUAL and EVOLP/R are integrated, we shall be ready to model some 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 the 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.

Acknowledgements Ari Saptawijaya acknowledges the support of Fundação para a Ciência e a Tecnologia (FCT/MEC) Portugal, grant SFRH/BD/72795/2010, and of his thesis supervisor, Prof. Luís Moniz Pereira.

References

ALFERES, J. J., BROGI, A., LEITE, J. A., AND PEREIRA, L. M. 2002. Evolving logic programs. In *JELIA'02*. LNCS, vol. 2424. Springer, 50–61.

ALFERES, J. J., PEREIRA, L. M., AND SWIFT, T. 2004. Abduction in well-founded semantics and generalized stable models via tabled dual programs. *Theory and Practice of Logic Programming 4,* 4, 383–428.

ANDERSON, M., ANDERSON, S., AND ARMEN, C. 2006. MedEthEx: a prototype medical ethics advisor. In *Procs. 18th Conf. on Innovative Applications of AI (IAAI-06).*

ANDERSON, M. AND ANDERSON, S. L. 2008. EthEl: Toward a principled ethical eldercare robot. In *Procs. AAAI Fall 2008 Symposium on AI in Eldercare.*

ANDERSON, M. AND ANDERSON, S. L. 2010. Robot be good: A call for ethical autonomous machines. In *Scientific American.*

ANDERSON, M. AND ANDERSON, S. L., Eds. 2011. *Machine Ethics.* Cambridge U. P.

ANDERSON, M., ANDERSON, S. L., AND ARMEN, C. 2005. Towards machine ethics: implementing two action-based ethical theories. In *Procs. AAAI 2005 Fall Symposium on Machine Ethics.*

BARAL, C., GELFOND, M., AND RUSHTON, N. 2009. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming 9,* 1, 57–144.

Baral, C. and Hunsaker, M. 2007. Using the probabilistic logic programming language P-log for causal and counterfactual reasoning and non-naive conditioning. In *Procs. of the 20th Intl. Joint Conf. on Artificial Intelligence (IJCAI'07)*.

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

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

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

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

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

Dung, P. M. and Thang, P. M. 2010. Towards probabilistic argumentation for jury-based dispute resolution. In *Procs. 3rd Intl. Conf. on Computational Models of Argument (COMMA'10)*.

Economist. 2012. Morals and the machine. Main Front Cover and Leaders (page 13), The Economist.

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

Gigerenzer, G. and Engel, C., Eds. 2006. *Heuristics and the Law.* MIT Press.

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

Han, T. A. and Pereira, L. M. 2011. Intention-based decision making with evolution prospection. In *Procs. 15th Portuguese Intl. Conf. on Artificial Intelligence (EPIA 2011).* LNAI, vol. 7026. Springer.

Han, T. A., Saptawijaya, A., and Pereira, L. M. 2012. Moral reasoning under uncertainty. In *Procs. of The 18th Intl. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18).* LNCS, vol. 7180. Springer, 212–227.

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

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

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

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

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

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

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

Kowalski, R. and Sadri, F. 2012. A logic-based framework for reactive systems. In *6th Int. Symposium RuleML'12.* LNCS, vol. 7438.

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

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

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

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

Pereira, L. M. and Pinto, A. M. 2009. Inspection points and meta-abduction in logic programs. In *Procs. 18th. Conf. on Applications of Declarative Programming and Knowledge Management (INAP'09).*

Pereira, L. M. and Saptawijaya, A. 2007a. Modelling Morality with Prospective Logic. In *Progress in Artificial Intelligence, Procs. 13th Portuguese Intl.Conf. on Artificial Intelligence (EPIA'07).*

Pereira, L. M. and Saptawijaya, A. 2007b. Moral Decision Making with ACORDA. In *Local Procs. 14th Intl. Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR'07).*

Pereira, L. M. and Saptawijaya, A. 2009a. Computational Modelling of Morality. *The Association for Logic Programming Newsletter 22,* 1.

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

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

Pereira, L. M. and Saptawijaya, A. 2012. Abductive logic programming with tabled abduction. In *Procs. 7th Intl. Conf. on Software Engineering Advances (ICSEA).* ThinkMind, 548–556.

Poole, D. L. 1988. A logical framework for default reasoning. *Artificial Intelligence 36,* 1, 27–47.

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

Rahwan, I. and Simari, G., Eds. 2009. *Argumentation in Artificial Intelligence.* Springer.

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

Saptawijaya, A. and Pereira, L. M. 2013a. Program updating by incremental and answer subsumption tabling. Accepted as Short Paper at LPNMR 2013. Available at `http://centria.di.fct.unl.pt/~lmp/publications/online-papers/evolpR.pdf`.

Saptawijaya, A. and Pereira, L. M. 2013b. Tabled abduction in logic programs. Accepted as Technical Communications at ICLP 2013. Available at `http://centria.di.fct.unl.pt/~lmp/publications/online-papers/tabdual_lp.pdf`.

Saptawijaya, A. and Pereira, L. M. 2013c. Towards practical tabled abduction usable in decision making. In *Procs. 5th. KES Intl. Symposium on Intelligent Decision Technologies (KES-IDT).* Frontiers of Artificial Intelligence and Applications (FAIA). IOS Press.

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

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

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

Toni, F. 2010. Argumentative agents. In *Procs. Intl. Multiconference on Computer Science and Information Technology.* Vol. 5.

van den Hoven, J. and Lokhorst, G.-J. 2002. Deontic logic and computer-supported computer ethics. *Metaphilosophy 33,* 3, 376–386.

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

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

Wiegel, V. 2007. SophoLab; experimental computational philosophy. Ph.D. thesis, Delft University of Technology.