Norm-regulation of Agent Systems

Magnus Hjelmblom

Report Series / Department of Computer and Systems Sciences
No. 15-014

Stockholm
University

Norm-Regulation of Agent Systems

Instrumentalizing an algebraic approach to agent system norms

Magnus Hjelmblom

ISSN 1101-8526
ISBN 978-91-7649-260-4
DSV Report Series No. 15–014

Printed in Sweden by Holmbergs, Malmö 2015
Distributor: Department of Computer and Systems Sciences, Stockholm University

Abstract

An architecture for norm-regulated multi-agent systems based on an algebraic approach to normative systems is instrumentalized and further developed. The core of the instrumentalization is a Prolog module, which together with a Java library can be used for creating client/server-based runtime systems. Norms are represented as conditional sentences, whose normative consequences are formulated by applying normative operators to descriptive conditions. From such general normative conditions follow normative sentences regarding specific states of affairs. These in turn result in permission or prohibition of individual actions in specific situations. Furthermore, an approach to turning runtime systems into instruments for problem-solving by using evolutionary mechanisms for evolving normative systems, is presented. The construction of norm-creating operators on conditions, which forms the basis for the representation of normative systems, is approached from two angles. (i) A logical analysis based on the Kanger-Lindahl theory of normative positions is conducted. This results in two extended sets of types of normative positions, and based on an algebraic version of one of these extended systems, a set of operators for creating agent-specific norms is constructed. (ii) An alternative analysis, which takes as its starting point a systematic exploration of types of state transitions, yields a set of norm-creating operators based on prohibition of transition types. It is furthermore argued that in the context of a class of transition systems, in which transitions are deterministic and associated with a single agent performing an act, operators based on (ii) specify a meaningful semantics of operators based on (i). Theoretical results together with shared code and example applications contribute to make possible theoretically sound, transparently described, and efficiently implemented norm-regulated autonomous agent systems.

Sammanfattning

En arkitektur för normreglerade multiagentsystem baserad på en algebraisk representation av normativa system instrumentaliseras och vidareutvecklas. Kärnan i instrumentaliseringen utgörs av en Prolog-modul som tillsammans med ett Java-bibliotek kan användas för att skapa client/server-baserad körbar kod. Normer representeras som ordnade par av grundvillkor och följdvillkor. De senare konstrueras genom att normativa operatorer appliceras på deskriptiva villkor. Från sådana generella normativa villkor följer normativa satser om specifika sakförhållanden, vilka i sin tur ger upphov till förbud mot eller tillåtelse att utföra enskilda handlingar i olika situationer. Vidare skisseras en metod för att göra körbara multiagentsystem till verktyg för problemlösning genom att använda evolutionära mekanismer för att odla fram normativa system. Konstruktionen av normskapande operatorer på villkor, vilka ligger till grund för representationen av normativa system, betraktas ur två olika synvinklar. (i) En logisk analys, baserad på Kanger-Lindahls teori om normativa positioner. Denna resulterar i två utökade uppsättningar av typer av normativa positioner och utgående från en algebraisk version av ett av dessa utökade system konstrueras sedan en uppsättning operatorer för att skapa agentspecifika normer. (ii) En alternativ analys, som tar sin utgångspunkt i en systematisk undersökning av olika typer av tillståndsövergångar. Denna ger upphov till en uppsättning av normskapande operatorer som är baserade på förbud mot olika typer av övergångar. Argument presenteras vidare för att inom ramen för en klass av övergångssystem, där övergångar är deterministiska och associerade med en agent som utför en handling, så specificerar operatorer baserade på (ii) en meningsfull semantik för operatorer baserade på (i). Teoretiska resultat tillsammans med tillgängliggjord programkod och exempel på tillämpningar bidrar till att underlätta skapandet av teoretiskt sunda, transparent beskrivna och effektivt implementerade normreglerade system av autonoma agenter.

Acknowledgements

I am thankful to so many people for giving me the necessary nudges out through the door and into the road that led to this thesis, for urging me forward in times of weariness or despair, helping me keep my feet, and gently pulling me back when being swept too far off. First and foremost, I would like to thank my two supervisors, Prof. Magnus Boman and Prof. Jan Odelstad, who have shown that the whole is sometimes greater than the sum of its parts. As individuals, they have numerous strengths, to which I will return, but together they have formed the perfect team of supervisors, each reading and commenting on my writing from different angles. Our 'trialogs' in Kista or Gävle, or via Skype, or by e-mail, have always been stimulating and rewarding. One of Magnus's greatest assets as a supervisor, besides vast experience and subject knowledge, is his ability to inspire and, if one part of the road turns out to be impassable, to find new paths ahead. There is a constant flow of positive energy from Magnus that has always made me go from our meetings with renewed energy and confidence. This is a quality he shares with Jan, who has put a lot of effort into my dissertation, probably a great deal more than should be expected of a co-supervisor. Besides arranging many practical matters, Jan has over and over again put my writing under scrutiny. I have more than once been impressed (and, admittedly, sometimes frustrated) by his ability to find mistakes and ambiguities in the text, while at the same time critically challenging explicit or implicit assumptions and conclusions. This has, of course, always been extremely helpful. The discussions with Jan have always stimulated me and pushed me forward, probably thanks to his unique combination of sharp intellect, attention to detail, and good humour.

I am very grateful to my employer, the University of Gävle, for granting me the opportunity to engage in doctoral studies. I would especially like to express my gratitude to Ulla Ahonen-Jonnarth, head of the Decision, Risk and Policy Analysis (DRPA) group, Camilla Niss and Jonas Boustedt, former, resp., current head of the Department of Industrial Development, IT, and Land Management, and Bengt Eriksson, faculty manager of the Faculty of Engineering and Sustainable Development. As my close colleagues in the DRPA group and the Computer Science (CS) group, respectively, Ulla and Jonas have not only made the necessary institutional arrangements, but also given constant support and encouragement. Ulla's work on simulation of forest cleaning, and our discussions in relation to this area, have given invaluable input to the thesis. Jonas, whom I got to know already in Uppsala as a fellow student on the upper secondary teacher study programme, paved the way for me by making a similar PhD journey a few years ago. He thus cleared many organizational obstacles, but first and foremost showed me the way, at the same time inspiring and (without knowing it) putting a slight pressure on me. I also wish to thank the other members of the DRPA group, Anders Hermansson, Goran Milutinovic, Fredrik Bökman, Eva Boo Höglund and Stig Blomskog, for encouragement and stimulating seminars, and all of my colleagues at the CS group for encouragement and, not the least important, numerous coffee (or, in my case, tea) breaks. Special thanks go to Jörgen Holmberg at the Department of Educational Sci-

ences at the University of Gävle, like me a doctoral student at the Department of Computer and Systems Sciences (DSV) at Stockholm University. We have taken many courses together, often collaborating on various assignments, and together reflecting upon the content of the courses. This has led to many fruitful discussions about various aspects of research in general, and the special situation of being a PhD student at one university while being employed by another, in particular. I am very grateful to teachers and staff at DSV for the opportunity to participate in some course seminars via the internet, and to Jörgen and the support staff in Gävle for making the practical arrangements here. I am also thankful to DSV for the opportunity to carry out doctoral studies there, and especially to Love Ekenberg, former head of department, for facilitating the admission process and signing the necessary agreements between DSV and the Faculty of Engineering and Sustainable Development.

Finally, I would like to acknowledge the external funding received for parts of my work. My first publication emanated from work within the project 'Advanced Decision Support System for Choice of Actions', carried out in co-operation with local companies Rivermen/Fiberdata and Gavlegårdarna and funded by The Knowledge Foundation (KK-stiftelsen), and the EU Objective 2 project 'CML Beslutsstöd'. Some work was carried out within DaGIS, later renamed as Globes, a joint project with the business cluster Future Position X, funded by the European Regional Development Fund. I am very grateful to Roland Norgren at the University of Gävle (now process manager at Future Position X) who has played the important role of spider in the web in these joint projects. Without Roland's efforts, the road would have been longer and more difficult.

In all my life I have been blessed with being surrounded by an abundance of love, encouragement and support: from my parents, Hans and Ulla, from my sisters Anna and Helena, from my wife and life companion Kajsa, and from my children Anton, Samuel and Ester. Kajsa, I love you. Without your patience and love, I would never have made it all the way. A star shines on the hour of our meeting! I am so thankful for everything you and the rest of my family have done for me, and I dedicate this book with love to you all.

Contents

About this thesis

Papers

The thesis includes the following papers, referred to in the text by their roman numerals:

I. Hjelmblom, M., & Odelstad, J. (2009). jDALMAS: A Java/Prolog framework for deontic action-logic multi-agent systems. In Håkansson, A., Nguyen, N., Hartung, R., Howlett, R., and Jain, L. (Eds.), *Agent and Multi-Agent Systems: Technologies and Applications. Lecture Notes in Artificial Intelligence* (Vol. 5559, pp. 110-119). Berlin Heidelberg: Springer. DOI: 10.1007/978-3-642-01665-3_12. ISBN: 3-642-01664-2

II. Hjelmblom, M. (2013). Norm-regulated transition system situations. In Filipe, J. and Fred, A. (Eds.), *Proc. of the 5th International Conference on Agents and Artificial Intelligence, ICAART 2013* (Vol. 1, pp. 109-117). Portugal: SciTePress. DOI: 10.5220/0004260801090117. ISBN: 978-989-8565-38-9

III. Hjelmblom, M. (2014b). Instrumentalization of norm-regulated transition system situations. In Filipe, J. and Fred, A. (Eds.), *Agents and Artificial Intelligence. Revised Selected Papers from the 5th International Conference on Agents and Artificial Intelligence, ICAART 2013. Communications in Computer and Information Science* (Vol. 449, pp. 80-94). Berlin Heidelberg: Springer. DOI: 10.1007/978-3-662-44440-5_5. ISBN: 978-3-662-44439-9

IV. Hjelmblom, M. (In press). Normative positions in multi-agent systems. *Extended version of* (Hjelmblom, 2014a). *Web Intelligence: An International Journal.* IOS Press.

V. Hjelmblom, M. (In press). Offline norm evolution. *Extended version of* (Hjelmblom, 2015a). In Duval, B., van den Herik, J., Filipe, J., and Loiseau, S. (Eds.), *Revised Selected Papers from the 7th International Conference on Agents and Artificial Intelligence, ICAART 2015. Lecture Notes of Artificial Intelligence.* Berlin Heidelberg: Springer.

Other publications

1. Hjelmblom, M. (2014a). Normative positions within norm-regulated transition system situations. *Proc. of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).* (Vol. 3, pp. 238-245). IEEE Computer Society Digital Library. DOI: 10.1109/WI-IAT.2014.173. ISBN: 978-1-4799-4143-8

2. Hjelmblom, M. (2015a). Offline evolution of normative systems. In Loiseau, S., Filipe, J., Duval, B., and van den Herik, J. (Eds.), *Proc. of the 7th International Conference on Agents and Artificial Intelligence, ICAART 2015.* (Vol. 2, pp. 213-221). Portugal: SciTePress. DOI: 10.5220/0005284102130221. ISBN: 978-989-758-074-1

Notes on terminology

The notion of *instrumentalization* has been a recurring theme in the development of this thesis. Already in its forerunner, my master's thesis, it is stated that the algebraic model for Deontic Action-Logic-based Multi-Agent Systems (abbreviated DALMAS) is "inspected and instrumentalized through an executable logic program". The Merriam-Webster dictionary defines the transitive verb *instrumentalize* as

(1) *To render instrumental: direct, organize, adapt.*

The Oxford English Dictionary gives two explanations:

(2) *To make or render instrumental to some end; to fashion into an instrument; to organize.*

(3) *To measure or reckon by means of instruments.*

Here, instrumentalize is used in the sense of (1) and (2); i.e., "turning the object of study into an instrument". It seems, in fact, that the thesis deals with different kinds of instrumentalizations. One is the aforementioned instrumentalization of the DALMAS architecture for norm-regulated multi-agent systems into a physical architecture. In this case, the object of study (the 'instrumentalizandum', i.e., 'that which is being instrumentalized') is the abstract architecture, which is made into an instrument for creating specific DALMASes in the form of logic programs to be run on computers. This is mainly the topic of papers I, II, and III, although the notion of a 'norm-regulated transition system situation', developed in II and III, is intended to be applicable to a broader class of systems than the DALMAS. Another kind of instrumentalization, which is the main topic of Paper V, is to turn the physical architecture into an instrument for problem-solving. This involves creating specific runtime systems, which includes making choices on hardware, operating system, and many other things that affect the runtime properties of any software. Any run of this software is an instrumentalization of the physical architecture, since the details of any run are devised from the application case at hand (e.g., the cleaning of a young forest stand) and the input and output to the program is adapted accordingly. Here, then, it is the physical architecture and the runtime system that are the objects of study, which by implementation and execution on a computer are instrumentalized into a problem-solving tool.

It could furthermore be argued that the DALMAS architecture in itself is the result of instrumentalization of its underlying formal theories and theoretical concepts, such as the theory of normative positions and the notion of condition-implication structures. This can be viewed as instrumentalization on yet another level of abstraction. Further development of these theoretical concepts is the topic of Paper IV. Thus, instrumentalizations on four or even five levels of abstraction can be discerned, the instrumentalizanda on these different levels being (a) general theories and theoretical concepts, (b) the abstract DALMAS architecture, (c) the physical DALMAS architecture, and (d) specific runtime systems and their executions. A chain of instrumentalizations on different levels of abstraction is thus obtained, so that what is the 'instrumentalizans' (i.e., 'that which is the result of instrumentalization') of something on one level also constitutes an instrumentalizandum aiming towards another level. The methods used for instrumentalizations on these various levels of abstraction are discussed in Ch. 2. It can be noted that, in for example ethics and social sciences, the verb *instrumentalize* usually has a negative connotation, as in 'to (ab-)use something or someone for one's own agenda', but it is hardly the case that this negative connotation extends into the realm of the artificial instrumentalizanda appearing in this thesis.

Two other frequent terms with many connotations are 'agent' and 'norm'. In everyday use, an *agent* is usually someone who has been authorized by someone else to act on his or her behalf; in other words a *representative* agent, such as a football player agent, an insurance agent, or a travel agent. In philosophy, on the other hand, an agent is someone who acts, but not necessarily on behalf of someone else; i.e., an agent that might have its own agenda. In economics, the term *rational agent* is often used for such an agent. This is usually the way agents are viewed within artificial intelligence (AI), but in this context terms such as 'intelligent agent', 'autonomous agent' and 'autonomous intelligent agent' are more commonly used, each emphasizing different aspects of the agent concept. An (artificial) *multi-agent system*, abbreviated MAS, is a (computerized) system of multiple interacting intelligent agents, situated in a particular environment. A *one-agent system* is here regarded as a special case with a single agent. Although the agent systems discussed here will primarily be systems of artificial software agents, they could in principle also include human agents or embodied artificial agents. At this stage, no special assumptions are made about, for example, autonomy, reasoning capability, or architecture, etc.

The term *norm* is widely used in everyday life, but in a rather loose sense. Furthermore, the concepts of norm and *normative system* are important objects of study within many academic disciplines, including philosophy, sociology, law, and ethics. It is generally acknowledged that there seems to be a significant difference between what is usually referred to as *normative* and *descriptive* sentences. A normative sentence, as opposed to a descriptive one, does not state what *is* the case, but expresses some kind of value judgment, for example what *ought* (not) or *shall* (not) be the case, or what is *good* or *bad*. Descriptive sentences express matters of fact, but not evaluations or prescriptions. In *A Treatise of Human Nature*, first published anonymously in 1739, Hume found that it is common that claims are made about what ought to be on the basis of statements about what is (Hume, 2000); the position that one cannot derive an 'ought' from an 'is' is often referred to as *Hume's law*. Note that a common view is that the opposite of descriptive is *prescriptive*. I shall, however, stick to the term normative, since in some communities so-called 'prescriptive norms', which prescribe a certain behaviour in a certain situation, are regarded as a special case of the general concept of norms.

A consequence of the diverse areas in which norms and normative systems are studied is that there is a plethora of different perspectives on and definitions of these terms, and related terms such as 'policy' and 'law'. Thus, one should not expect to find a standard definition accepted by everyone. In short, the term norm may in this thesis refer to, e.g., a behavioural rule, an unwritten 'social law' or a written legal rule. In focus is the formal representation and semantics of norms, primarily norms that regulate behaviour, and the idea that regardless of whether a norm expresses a social law or a legal rule, or whether it has emerged bottom-up in a social setting or is imposed top-down by a designer or some authority, its representation has some sort of logical structure.

An important part of the representation of norms in this thesis is the Kanger-Lindahl theory of normative positions. The term originally used by Lindahl (1977) in his dissertation was 'legal position'. The term 'normative position', which came into use later and has become the most widely used term today, can perhaps be a bit misleading, since 'normative' may also refer to, e.g., value judgments. Therefore, 'deontic position', which has been suggested as a replacement, is an appealing alternative. However, since 'normative position' is part of the now well-established term 'Kanger-Lindahl theory of normative positions', I will continue to use it in this thesis.

Disposition

The first part of the thesis is structured as follows. In Ch. 1 I outline and delimit the research area and elaborate the purpose and goal of the research. In Ch. 2 I discuss methodological issues, and in Ch. 3 I give a theoretical background. The main contribution of the thesis is presented in Ch. 4, and finally in Ch. 5 I conclude and present ideas for future research. The second part contains five papers.

Part I

Research Area and Contribution

'You cannot pass,' he said. The orcs stood still, and a dead silence fell. 'I am a servant of the Secret Fire, wielder of the flame of Anor. You cannot pass. The dark fire will not avail you, flame of Udûn. Go back to the Shadow! You cannot pass.'

Gandalf the Grey, standing on the bridge of Khazad-dûm, to the Balrog
(from *The Fellowship of the Ring* by J.R.R. Tolkien)

Chapter 1

Introduction

1.1 Preamble

Imagine an area containing nuclear waste, and a team of robots designed to clear the area of waste. Each robot has its own (partial or full) knowledge of the terrain, the location of waste, and the whereabouts of other robots. This knowledge may come from the robot's different sensors, and perhaps be communicated to other robots via actuators, or it may be centrally computed, stored, and shared with all robots. The robots can move around independently of each other in different directions, and pick up waste in their immediate neighbourhoods.

Now consider a young forest stand, in need of pre-commercial thinning (forest cleaning), and a robot designed to perform cleaning of the stand. The area could, for example, be divided into smaller square-shaped cleaning units, and the robot, which from the beginning might have only partial information on the whole forest stand, can move around in a square to remove some of the trees located there. When done with one cleaning unit, it could then proceed to the next, keeping a store or a log, and so building a model of the whole forest stand.

What both examples have in common is that they describe systems of autonomous agents (embodied as robots) designed for the purpose of solving a particular problem. In many settings, it is in the interest of the system designer to get the best solutions possible, executed as efficiently as possible. To meet both criteria, mechanisms for regulating the behaviour of the agents are needed.

1.2 Research problem

There are many reasons for studying the regulation of MAS in general, and regulation by norms in particular. For example, norm-regulation of MAS designed to solve certain (classes of) problems may be an attractive alternative to planning and replanning, operations which are often expensive in terms of execution time (Verhagen & Boman, 1999). The 'NorMAS roadmap' (Andrighetto, Governatori, Noriega, & van der Torre, 2013) is a comprehensive overview of norm-regulated or norm-governed MAS, often referred to as *normative MAS* (abbreviated *NMAS* or *nMAS*). The concept of NMAS exemplifies the relation between agent theory and the social sciences, for example sociology, philosophy, economics, and legal science, and in particular the use of sociological theories in MAS. The need for social science theories and concepts such as norms in the field of MAS is by now well established (Boella, van der Torre, & Verhagen, 2006). A researcher in this area has a variety of theories at his or her disposal, but when it comes to practical application, the available tools (including shared code) based on these theories are not as many. Criado, Argente, and Botti (2011) identify a number of open issues for norm-regulation of open agent systems, systems whose participants are highly autonomous, heterogeneous, not always trustworthy and have conflicting goals (Luck & McBurney, 2008). One basic area that requires further attention is the specification of normative systems. Criado et al. call for more work on formal languages for explicit representation of norms and normative systems, which includes defining a normative model for designing complex and dynamic agent societies, developing languages for representation of normative systems, and creating tools for automated reasoning about normative systems. The related open issue of norm implementation deals with norms from an operational or computational point of view, studying the implementation of norms in a society by means of control mechanisms. This requires procedures for the detection of norm violations, which are dependent on the verifiability of norms, and mechanisms for letting norms effectively influence the behaviours of agents, either by regimentation or by enforcement (see, e.g., Criado et al., 2011, pp. 246ff). Closely related to norm specification and implementation is also the issue of individual normative reasoning, which includes development of architectures and decision-making procedures for norm awareness and reasoning about norm compliance, architectures in which agents can play different normative roles such as monitors, norm enforcers, or legislators, as well as mechanisms for detection of incoherence or redundancy in normative systems. Furthermore, Criado et al. point to the need for new software tools for solving real-world problems. Such tools should assist developers in the design and implementation of open system applications, for example by facilitating the specification of norms that allow for control and coordination of agent activities. When identifying ten challenges for the interactionistic view of norm-regulated MAS, Boella, van der Torre, and Verhagen (2008, p. 6) highlight this issue by pointing out that often neglected in norm-related discussions is the perspective of the agent programmer in need of tools such as programming primitives, infrastructures, protocols, and mechanisms.

Within economic theory, many researchers have become interested in the interplay between the economic agent's preferences and a normative system; see, e.g., (Arthur, 2006). Normative systems, such as laws, may impose restrictions on the agent's behaviour, thereby limiting the agent's room for manoeuvre. The agent may choose the most desired act, where the agent's preferences are represented by a utility function, but only within the scope determined by the norms. It has been suggested by, among others, Boman (1999), that a model of this kind could be used for regulating the behaviour of artificial agents. The role of norms in this 'agent oeconomicus norma' (Odelstad, 2008, Sect. 1.8.3) model is to delimit the autonomy of the agents. The relation between desires and obligations in agent decision-making has been studied by Broersen, Dastani, Hulstijn, and van der Torre (2002), but generally speaking, the idea that norms can define a scope of action or 'room for manoeuvre' for an agent does not seem to have received much attention.

1.3 Purpose and goals

The purpose of this thesis is to further develop and instrumentalize an abstract architecture for certain kinds of norm-regulated agent systems, including (but not limited to) systems based on the *agent oeconomicus norma* model. Contributions on at least four levels of abstraction are made, the main focus being on representation of the underlying normative system and specification of a semantics for norms, in order to make available instrumentalized normative mechanisms that are usable in many classes of applications. These tools consist of shared code together with example applications as well as theoretical results, documented in this thesis. Thus, I aim to contribute to issues of norm representation and implementation, including procedures for the detection of norm compliance. Another aim is to address the issue of developing new software tools for solving problems modelled as norm-regulated MAS, for example systems of norm-regulated utility-based agents. In particular focus is the perspective of the agent programmer in need of tools for specifying and instrumentalizing norms for controlling and coordinating agent activities. An example of an existing application which could benefit from this work is simulation of social interaction and theories from the social sciences, where norm-regulated MAS have been used in research on, e.g., coordination, cooperation, social control, benevolence, reciprocity, trust, and electronic institutions (Hollander & Wu, 2011). Other application areas include software engineering, adaptive environmental control, autonomous configuration of ad-hoc networks, intelligent buildings, human negotiation support, and second opinions on medical diagnoses (Boman, Davidsson, & Younes, 1999; Hollander & Wu, 2011; Jonker, 2007; Takahashi, Kitamura, & Miwa, 2012). Potential future application areas include, e.g., stability in financial exchange, forest cleaning, robot rescue team coordination, autonomous system management on space missions, computer games, and decision support (Ahonen-Jonnarth & Odelstad, 2010; Dorais, Bonasso, Kortenkamp, Pell, & Schreckenghost, 1999; Hjelmblom & Odelstad, 2009; Kitano & Tadokoro, 2001; Lybäck & Boman, 2004). The overarching goal is to make possible theoretically sound norm-regulated systems, enabling transparent descriptions and efficient implementations of autonomous agent systems. It is furthermore hoped that the thesis will contribute to an increased interest in the *agent oeconomicus norma* model for agent systems.

1.4 Delimitations

Beyond the scope of this thesis are, for example, discussions about gender issues, sustainable development aspects, ethical considerations regarding the potential (mis-)use of the findings and artifacts presented, and the legal status of artifacts such as agents or agent systems.[1] Furthermore, I do not seek to give a comprehensive philosophical account of concepts such as norms, agents, and agency. An overview will be given in Ch. 3, mainly focusing on norms in the context of agent systems. Here are just a few introductory remarks. Agent system norms can be classified according to their purpose. One such classification, proposed by Boella and van der Torre (2008), divides norms into *substantive* and *procedural* norms. Substantive norms, which define the legal relationships of members of a society with other members and with society itself, are further classified into two broad categories, *constitutive* norms and *regulative* norms. Constitutive norms specify, for example, the meaning of different kinds of communicative acts within a given institution, or the definitions of what kind of acts are meaningful in a certain context, while regulative norms specify varying degrees of ideal or sub-ideal behaviour through prohibitions, permissions, or obligations, in order to regulate behaviour. As the examples above indicate, this thesis deals primarily with regulative norms for autonomous agents.

There are two different perspectives on norms regarding the origin of normativity. The *social* or *interactionist* perspective is based on the idea of *norm emergence*, i.e., that norms arise spontaneously within a social setting. The *legalistic* perspective, on the other hand, is that normativity is institutionalized and authority-based (Boella et al., 2008). Although not necessarily contrasting the social perspective of norms with the legal perspective, the NorMAS roadmap stresses that norms in human society regulate the interactions between an individual and the society, i.e., other individuals or groups of individuals. As a consequence,

> ... norms would not make any sense if we consider only one individual in an environment (which possibly might contain other individuals, but which are not distinguished from other environmental elements). (Balke et al., 2013, p. 2)

A social setting is, however, not presupposed here: a normative system will be regarded as a collection of norms, and norms may be applicable in environments with only one individual as well as in societies with more than one individual. Thus, according to this view, both of the robot systems introduced in the preamble might well be norm-regulated. It should furthermore be noted that, although the examples in this thesis and in the papers may suggest a legalistic perspective on norms, this perspective is not presupposed. For example, in a setting where norms emerge by social interaction, there is a need for agents to learn and internalize norms, a process known as norm *immergence* (Criado et al., 2011, p. 255). Thus, an agent must be able to maintain a model of the emerging normative system, which requires explicit representation and

[1] For an introduction to some of these issues, see for example (Chopra & White, 2011).

reasoning about the norms that are being internalized. That the first research challenge identified by Boella et al. (2008) is the development of tools which support communities in recognizing, creating, and communicating norms, further underlines the need for supporting explicit representation of norms.

A common view of MAS norms is that they should be *soft constraints* (see, e.g. Müller, 2006, pp. 217f) that, at least in principle, are possible to violate, and that *hard constraints* which are 'physically unbreakable' are not in fact norms. The approach to formal representation of norms presented here is not tied to a specific view on this matter. In the agent oeconomicus norma model for MAS, the norms can be perceived as soft constraints in the sense that they are not thought of as 'physically impossible' to violate, but as hard constraints in the sense that they define the scope of agent behaviour, with which the agents try to comply as much as they can. In general, I make no assumptions regarding particular norm enforcement mechanisms or the possibility to violate norms.

Chapter 2

Methodology

Which research *methods* are computer scientists, logicians, or mathematicians using when they invent an algorithm, perform a logical-conceptual analysis of a phenomenon, or prove a theorem? To perform asymptotic algorithm analysis of the invented algorithm, or to prove theorems, a deductive approach is clearly required. But to come up with the algorithm, or to find suitable basic concepts and axioms, requires something similar to what might be called an 'open-ended experimental inductive' approach, involving much creativity (Bohm, 2004). Is there a 'method' for this, more than 'hard thinking and playing around with pen and paper'? Does this qualify as a research method? Is a research method always necessary, or even possible? In his classic *The Logic of Scientific Discovery*, Popper (1959) argues that the task of the logic of scientific discovery is to analyse the method of the empirical sciences. According to Popper, there is no logical method of having new ideas; every discovery contains an element of creative intuition. This view is criticized by, among others, Simon (1973); an account of this debate is, however, beyond the scope of this thesis. Let us simply point out here that most agree that computer science borrows some of its characteristics from both non-empirical research such as logic and mathematics and empirical research in natural and social sciences, as well as from engineering (see, e.g., Tedre, 2011). Thus it is partly a (natural) science discipline, partly a mathematical discipline, and partly an engineering discipline. In fact, computer science has been called both an 'unnatural' science (Knuth, 2001) and an 'artificial' science (Simon, 1996). To perform instrumentalizations of various concepts on different conceptual levels, tools from mathematics, logic, and computer science are needed. This section will discuss the main approaches, methods, and tools employed in this thesis.

Paper I employs object-oriented imperative programming as well as logic programming, two basic computer science methods, to further develop, and demonstrate possible applications of, the existing Prolog instrumentalization. Thus, two different programming paradigms are combined. There are many views of what kind of activity programming is. A common view is that it is a mathematical activity which involves the use of formal methods (Sterling & Shapiro, 1994). However, as the art of programming has become an integrated part of the wider area of *software engineering*, which includes analysis and design and other activities related to system development, a mixture of methods and techniques besides formal methods have come into use. Thus, programming can be viewed also as an engineering activity that follows a gradually evolving set of design principles. A third view of programming is that it is (at least partly) a natural science activity, where programs act as scientific theories or models. See, for example, (Knuth, 1997; Sebesta, 2013) and the summary by Turner (2014, Sect. 7). These different views in combination suggest that programming has both inductive and deductive components; the latter is especially true for logic programming (Lloyd, 1987).

One of the aims of papers II and III is to show that the ideas underlying the normative mechanisms employed in Paper I can be put to use in a broad class of MAS modelled as transition systems, not necessarily restricted to the architecture in Paper I. A systematic exploration, based on a logical analysis of possible types of state transitions, is performed. This investigation, which employs basic methods from symbolic logic, results in a set of transition type prohibition operators then turned into a mechanism for normative systems based on prohibition of state transition types. This mechanism is further instrumentalized into a Java/Prolog framework, which, as in Paper I, requires the combination of object-oriented and imperative programming as well as logic programming. Thus, papers II and III deal with instrumentalizations on at least two different levels of abstraction.

Paper IV contributes to the further development of a theoretical foundation for normative systems aimed at application to a class of MAS. It is argued that the transition type prohibitions in papers II and III can form a semantics for normative systems formulated within this theoretical framework. Here, we find a clear analogy with programming languages: a programming language, and the constructs that can be expressed using it, is commonly characterized by its syntax (the rules that govern the form, i.e. the sentence structure, of the language) and its semantics (the intended meaning of well-formed sentences in the language). Similarly, we can talk about the rules that govern the structure of norms, i.e., the syntax, and the intended meaning of well-formed norms, i.e., the semantics, of a language for representing normative systems. To attach a meaning to norms through their effects in terms of permissible or prohibited actions in a given situation, is an important part of instrumentalizing the theoretical concepts into a normative framework for MAS. The tools available for the kind of reasoning and logical-conceptual analysis needed here are those offered by mathematics and logic; in this case, two of them in particular: symbolic logic and (Boolean) algebra. To be more precise, the logic needed includes standard deontic logic, Kanger's 'deontic action-logic' and the Kanger-Lindahl theory of one-agent normative positions. Furthermore, the algebraic tools that are used are applications of the Lindahl-Odelstad theory of joining-systems, which incorporates notions such as Boolean quasi-orderings, Boolean joining-systems and so-called condition–implication structure models of Boolean joining-systems. An overview will be given in Ch. 3.

Paper V deals with instrumentalizations on another level of abstraction, near one end of the instrumentalization chain. Here, the instrumentalizandum is a running MAS, similar to a system employed as an example in Paper I, which through the use of evolutionary mechanisms is turned into an instrument for problem-solving. The main tools and methods used in this paper are the combination of two different programming paradigms, put to work in implementing an evolutionary algorithm, a specific computer science tool for problem-solving.

To conclude, the metaphor of a chain of instrumentalizations on different levels suggests a methodology that is chiefly reductionist, but with some systemic elements. These systemic elements result from the transdisciplinary considerations necessary to complete the chain, in which abstract algebra, deontic logic, and computer science methods are merged into a unified and constructive approach. The research can be characterized as exploratory and open-ended, not directed by any specific hypotheses, and without ramifications from a particular research programme or 'school of thought'.

Personal notes

My first encounter with agents and MAS took place in 1998 when I studied artificial intelligence (AI) at Uppsala University. I was finishing my degree of Master of Education and found employment as a lecturer in computer science at the University of Gävle. A few years later, I myself began teaching an AI course, assigning to the students the task of implementing a simple agent in Prolog. One of my students, Jan Odelstad, was quite different from the others. Odelstad already had a doctor's degree in theoretical philosophy, and had recently, as part of a national 'competence switching' programme, become my colleague at the Division of Computer Science. For a young lecturer, it was sometimes challenging, but always very stimulating and instructive, to have a student with his special combination of experience and eagerness to learn. At that time, Odelstad was working together with Magnus Boman on DALMAS, short for Deontic Action-Logic Multi-Agent System, an abstract architecture for norm-regulated MAS. We began discussing the possibility of turning a DALMAS into an executable Prolog program, and Odelstad showed me their seminal paper 'Algebras for agent norm-regulation' (Odelstad & Boman, 2004), in our internal discussions always referred to as 'OdelBom'. This sparked my interest, and I began trying to understand the idea behind the architecture and writing some pieces of Prolog code. This gradually took the shape of a master's thesis project. Fortunately, Odelstad agreed to be my supervisor, and throughout the project Boman took an active role in our discussions. I had already finished my Master of Education, and in 2007 it was complemented with a Master of Science in computer science at Uppsala University. The thesis resulted in the `dnrDALMAS` Prolog module, a general-level instrumentalization of the DALMAS architecture intended as a tool for implementing specific systems. The thesis, entitled *Deontic Action-Logic Multi-Agent Systems in Prolog*, was later published as a technical report (Hjelmblom, 2008) at the University of Gävle.

As it turned out, my relationship with DALMASes would not end there. Odelstad and Boman were happy to see the DALMASes 'come to life' as Prolog programs, but Prolog is not ideally suited for visualization and animation. Therefore, as part of the project *Advanced Decision Support System for Choice of Actions*, I started working on expanding the `dnrDALMAS` implementation into a client/server architecture consisting of a Prolog logic server and a Java front-end. This resulted in a first paper (Paper I), co-authored with Odelstad and presented at the 3rd International KES Symposium on Agents and Multi-agent Systems (KES-AMSTA 2009) in Uppsala. Odelstad and I continued to discuss various refinements and improvements of the architecture and the implementation. One particularly interesting issue was the interpretation of the Kanger-Lindahl one-agent types of *normative positions*, which make up the logical foundation for the algebraic approach to norms in OdelBom, in terms of permissible or prohibited transitions between different system states. I found their analysis, which formed the basis for the implementation in the master's thesis, to be intuitive and clear, and I saw no need to consider other interpretations. But the reviewer of my master's thesis, Roland Bol, observed that the set of combinations of state transition types seemed to be larger than the set of types of normative positions. This puzzled and disturbed me, and thus gave me the necessary energy to formally enrol as a PhD student, this time with Boman as main supervisor and Odelstad as the assisting supervisor. Different ways of interpreting types of normative positions in terms of permissible or prohibited types of state transitions have been one of the key issues that I have returned to over and over again throughout the doctoral studies. As is probably often the case, I expected this issue to be a quite uncomplicated step on the path toward more distant goals, but it turned out to be more complicated than I initially thought. (And, as is probably also often the case, had I known then how confused and frustrated I would sometimes become, I wonder if I would have given it a try at all.)

Initially, I turned my attention toward the 'mismatch' between the set of (fifteen) possible transition type combinations and the set of (seven) types of normative positions. It seemed that, in order to be able to interpret the types of normative positions in terms of prohibition of combinations of transition types, a larger set of types of normative positions was needed. Therefore, a natural first step was to investigate elaborated versions of the Kanger-Lindahl theory of normative positions. I considered two alternatives. One promising foundation seemed to be the finer-grained system of fifteen normative positions by Sergot (2001). Indeed, I found Sergot's system to be a suitable basis for so-called *system norms*, with a one-to-one correspondence between the set of transition type prohibitions and Sergot's elaborated set of 'cumulative fact/act positions'. However, my main interest was in creating a framework for *agent-specific norms*, that it would be possible to instrumentalize for the regulation of agent behaviour in runtime systems. Inspired by the remarks by Roland Bol, I soon reached the conclusion that only nine of the fifteen transition type combinations were meaningful as the basis for agent-specific norms. In the hope of reducing the number of types from fifteen to nine, I tried to find a plausible interpretation of, in particular, the notion of action within Sergot's system. At the same time, I wanted the interpretation of agency to be consistent with the examples given by Lindahl (1977) when developing the theory of normative positions. This turned out to be a tough nut to crack. Instead, I decided to investigate the other alternative, which had been under consideration from the outset. This approach, based on an observation by Odelstad (2008), resulted in another extended system of fifteen types of normative positions, somewhat different from Sergot's. It should be noted that the goal was to make the theory of normative positions applicable to a specific class of MAS, not to create a new theory at the same level of generality as the original theory. By some additional assumptions, this system was reduced to a system of nine types, which could be given an interpretation in terms of the prohibition of combinations of transition types. These results first appeared in a technical report (Hjelmblom, 2011), and were not until later published as part of two conference papers. Since I was struggling with different ways of formulating and interpreting the extended systems of normative positions, and seemed to get stuck over and over again, I decided to put the potential link between the theory of normative positions and transition type prohibition aside for some time. Instead, I continued to develop the idea that normative systems could be based on transition type prohibition, without any reference to normative positions. This resulted in Paper II, a conference paper that was later extended into Paper III. It can be seen that there is substantial overlap between these two papers, but some parts of II were abridged in III in order to make room for a discussion of an example system.

When I finally felt that I had managed to make the pieces fit together, (Hjelmblom, 2014b) was submitted to IAT 2014. In this conference paper, later extended into Paper IV, a normative framework based on an extended system of types of normative positions was presented and given an interpretation in terms of prohibition of transition types.

Already in OdelBom, the idea was outlined of letting a problem-solving Dalmas determine the optimal normative system for the particular task at hand. This idea was kept alive during our discussions in relation to the thesis work, and in (Hjelmblom, 2015b) it was finally realized. In this paper, later to be extended into Paper V, I demonstrate a method for using evolutionary mechanisms, put to work through a genetic algorithm, as part of the process of designing a normative system for a class of problem-solving MAS.

Throughout the doctoral studies, I have been part of a stimulating and encouraging academic environment at the Faculty of Engineering and Sustainable Development at the University of Gävle and the Department of Computer and Systems Sciences at Stockholm University. Besides studying courses at both universities, I have participated in seminars within the Decision, Risk and Policy Analysis group, as well as general faculty seminars, at the University of Gävle. Attending keynote lectures and presentations, and engaging in discussions with fellow researchers at four different conferences (KES-AMSTA 2009, ICAART 2013, IAT 2014, and ICAART 2015), has provided inspiration, new ideas, and academic insight. This, together with co-reviewing papers submitted to CLIMA XII and ICAART 2014, has broadened and deepened my knowledge of the intriguing field of MAS.

Chapter 3

Background

3.1 Agent systems

3.1.1 The Waste Collectors system

Consider an idealized version of the first of the two opening examples in Ch. 1: We have a system of agents situated in an environment consisting of a grid of squares ordered in rows and columns. Each square is assigned a coordinate consisting of an ordered pair of integers. Some squares contain an amount of waste, represented by a number. An agent in the system (a 'waste collector' or 'waste agent') tries, in cooperation with other agents, to collect as much waste as possible. Each waste collector has a utility function, such that the utility of an action depends on the amount of waste present in the squares surrounding the target square. As long as it stays within the boundaries of the grid, an agent can *move* one square at a time in four directions (north, south, west and east) and pick up the waste, if any, found in the new square. The agents may also *pass*, i.e., do nothing. Informally, a state of the system is determined by the position of agents, the amount of waste carried by each agent, and the amount of waste in each square.

This example is suitable for illustrating how a normative system can restrict the movement of agents in the proximity of other agents. To be able to distinguish different agents from each other and to say something about their spatial relationships, let us define two conditions *Diff* and Lap_n:

1. $\mathit{Diff}(x, y)$ iff x and y are different agents.
2. $Lap_n(x, y)$ iff the 'protected spheres' of x and y overlap with n squares.

Fig. 3.1 shows a state in which the protected spheres of two agents overlap by two squares, i.e., that $Lap_2(x_1, x_2)$ holds. Note that both *Diff* and Lap_n define symmetric relations, and that if $n \neq m$, then $Lap_n(x, y)$ implies $\neg Lap_m(x, y)$ for all x, y. Moreover, it is always the case that $Lap_9(x, x)$.

The Waste Collectors system, originally inspired by an example in (Steels, 1989), was used as a running example in (Odelstad & Boman, 2004), as well as

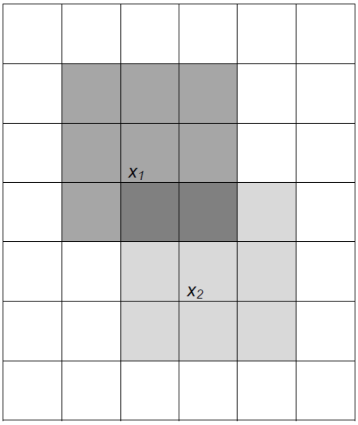


Figure 3.1: The protected spheres of two agents.

in (Hjelmblom, 2008), to illustrate the ideas presented. Instrumentalizations of (different variants of) this example appear in both Paper I and Paper V.

3.1.2 Modelling agent systems

Agent systems can be modelled in many different ways. One of the most common approaches is based on the notion of a transition system. Other formal approaches employ, for example, finite state machines or Petri nets (Wooldridge, 2009).

A *labelled transition system* is usually defined as an ordered triple $\langle S, E, R\rangle$ where S is a non-empty set of *states*; E is a set of transition *labels*, often called *events*; and $R \subseteq S \times E \times S$ is a non-empty set of labelled *transitions*. If (s, ε, s^+) is a transition, s is the initial state and s^+ is the resulting state of ε. An event ε is *executable* in a state s if there is a transition $(s, \varepsilon, s^+) \in R$, and *non-deterministic* if there are transitions $(s, \varepsilon, s^+) \in R$ and $(s, \varepsilon, s') \in R$ with $s^+ \neq s'$. A *path* (or *run*) of length m ($m \geq 0$) of a labelled transition system is a sequence $s_0\varepsilon_0 s_1 \cdots s_{m-1}\varepsilon_{m-1}s_m$ such that $(s_{i-1}, \varepsilon_{i-1}, s_i) \in R$ for $i \in \{1, .., m\}$.

An agent was previously identified as someone or something capable of *action*, but this notion has not been defined or discussed. The meaning or logical structure of action and agency will be further discussed in Sect. 3.2.1. It should be noted here that in an agent system modelled as a transition system, agent actions are usually associated with transitions. Informally, the actions performed by an agent determine its *behaviour*. There are many types of architecture for autonomous agents, of various complexity, ranging from simple reflex agents, through model-based agents, and goal-based agents to complex utility-based agents; an overview is given in the textbook by Russell and Norvig (2010). Similarly, there are many models for regulating the behaviour of agents. The following section discusses the use of norms for agent behaviour regulation.

3.1.3 Multi-agent systems and norms

A number of norm typologies have been proposed within different academic disciplines over the years. Well-known examples, often employed for characterization of norms in MAS contexts, are Morris (1956), Gibbs (1965), Tuomela (1995), and Therborn (2002). Normative MAS is the research field formed by the intersection between normative system theory and the MAS area (Boella et al., 2006). In the first chapter of the NorMAS roadmap, Balke et al. (2013) present three different definitions of normative MAS. According to the 'social definition' (Balke et al., pp. 8f), a normative MAS is governed by restrictions on the patterns of behaviour of the agents in the system. These actively or passively transmitted restrictions, sometimes represented as actions to be performed, have a social function and impact. They govern what actions (or outcomes) are permitted, empowered, prohibited, or obligatory under a given set of conditions, as well as specifying effects of norm compliance or non-compliance. The restrictions can emerge within the system or can be created a priori by the system designer. Norms should be (a) *contextual*, i.e., applicable only in specific contexts, and not in general, (b) *prescriptive*, which means that agents knowledgeable of norms are also aware that they can be held accountable for breaking them, and (c) *followable*, which means that it is possible for agents to perform or refrain from performing prohibited, obligatory or permitted actions. The 'norm-change definition' (Balke et al., p. 9) states that NMAS are MAS together with normative systems in which, on the one hand, agents can decide whether to follow the (explicitly represented) norms, and on the other the normative systems specify how and to what extent the agents can modify the norms. The 'mechanism design definition' (Balke et al., p. 12) characterizes an NMAS as "organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment".

Balke et al. note that it is difficult, if not impossible, to attain a complete consensus on the nature of NMAS. Instead of attempting to give a single definition to be generally accepted, they give a number of guidelines for developing norm-regulated MAS. The first guideline is to justify which definition is used and explain the choices made regarding the representation of norms in the system. It is argued that, when characterizing norms in MAS, one should take into account (i) the rule structure, i.e., conditional structure, of norms, (ii) the fact that there are different types of norms, and (iii) other basic features of normative systems such as norm recognition and hierarchies, norm application, and norm change. We shall return to (i) in Sect. 3.2. As regards (ii), there are many different definitions of norms, and many ways of classifying norms and normative systems. One classification of agent system norms, proposed by Boella and van der Torre (2008), draws on the philosophical and sociological works by Gibbs, Therborn, and others. In this classification, norms are divided according to their purpose into *substantive* and *procedural* norms. The former define the legal relationships of members of a society with other members and with society itself, while the latter define practical connections between regulations and their consequences. Procedural norms are addressed to agents with different roles in the normative system, and connect the ideal behaviour specified by substantive norms to enforcement mechanisms. Substantive norms are further classified into two broad categories, determinative or *constitutive* norms and behavioural or *regulative* norms. The former kind of norms specify, for example, the meaning of different kinds of communicative acts within a given institution, or the definitions of what kind of acts are meaningful in a certain context. A *legislative* norm is a special kind of constitutive (meta-)norm, whose purpose is to define who has the right–and when–to issue, modify or abolish norms. Constitutive norms may, for example, be represented as *counts-as conditionals*. A logical analysis of sentences like 'x counts-as y in $\mathcal{N}$', where $\mathcal{N}$ can be a normative system, was proposed by Jones and Sergot (1996), and contributions have been made by, for example, (Artosi, Rotolo, & Vida, 2004; Boella & van der Torre, 2006; Grossi, Dignum, & Meyer, 2005). See (Grossi & Jones, 2013) for an overview. Norms in the latter class, regulative norms, specify varying degrees of ideal or sub-ideal behaviour through prohibitions, permissions, or obligations, in order to regulate behaviour. As already mentioned, this thesis focuses primarily on regulative norms for autonomous agents. However, as noted in Paper IV, the so-called theory of joining-systems developed by Lindahl and Odelstad, which forms the basis for the algebraic representation of norms employed in the thesis (see Sect. 3.2.4), is well suited for representation of counts-as conditionals. This is further discussed by Lindahl and Odelstad (2013, pp. 627ff).

One perspective on norms is connected to the origin of normativity. The *social* or *interactionist* perspective of norms is based on the idea of *norm emergence*, i.e., that norms arise spontaneously within a social setting. Norms are regarded as conventions that emerge from agent interactions. The *legalistic* perspective, on the other hand, is that normativity is institutionalized and authority-based; norms are explicitly created by a system designer or agents with neglected capabilities (Boella et al., 2008). These two perspectives are related to different views of the norm creation process. The legalistic perspective is related to a top-down approach to norm creation, in which norms are either created offline by the system designer or online by specially appointed leader or legislator agents, the latter potentially regulated by legislative norms. The interactionist perspective, on the other hand, is related to a bottom-up approach in which norms emerge as a society (macro-)level effect of agent interactions carried out at the agent (micro-)level. At the same time, as remarked in Sect. 1.4, the agents learn and internalize the norms they perceive. Thus, there is an interplay between the two notions of norm emergence and norm immergence. In sociology, the relation between individual agent behaviour and characteristics at the level of the social system is known as the micro-macro link. Since MAS research and sociology share the interest in this relation, norms and other social concepts can have an important part to play in multi-agent systems (Boella et al., 2006). An example is the concept of *social commitments*, involving MAS, ethics, and deontic logic (Cohen & Levesque, 2012). The *spheres of commitment* MAS architecture proposed by Singh (1999) synthesizes ideas, particularly that of social context, from the multi-agent systems field with ideas from ethics and legal reasoning. Special emphasis is put on the interplay between commitments and social structure. Singh defines operations on commitments and groups, and MAS provide the context for the different operations. Social *policies*, conditional expressions involving commitments and actions on commitments, are modelled as higher-order commitments. Here the intricate relationship between the notion of norms and commitments becomes visible, and many of the basic conceptual assumptions listed by Singh (p. 98) have a bearing also on norm-regulated MAS. One example is the assumption that commitments not only rely on the social structure of the groups in which they exist, but also help create that structure; in fact, it is argued that commitments are both the ancestors and (possibly, but not exclusively) the descendants of norms.

Gaertner, Clark, and Sergot (2007) discern two categories of regulative norms, system norms and agent-specific norms, based on the 'level' on which they are effective; whether the system is viewed from the point of view of a system designer or an individual agent. As Sergot (2008, p. 16) puts it, *system norms*

> ... express a system designer's point of view of what system states and transitions are legal, permitted, desirable, and so on. There is a separate category of individual *agent-specific* norms that are intended to guide an individual agent's behaviours and are supposed to be taken into account in the agent's implementation, or reasoning processes, in one way or another.

Evidently, the notion of system norms implies a legalistic perspective on normativity. *Agent-specific norms* have a different character from that of system norms. In order for the latter to be meaningful, they need to be formulated in terms of the knowledge (possibly acquired from various perceptions) that an agent has, while the former are formulated in terms of the globally observable state of the world. Paper IV, although discussing both system norms and agent-specific norms, focuses mainly on the representation and semantics of the latter kind of norm.

Fig. 3.2 illustrates a possible architecture for a 'norm-regulated utility-based' agent, i.e., an architecture based on the agent oeconomicus norma model. In the figure, the normative system is distinguished both from the environment (which in a MAS contains other agents) and from the agent itself. It can then be regarded as a common normative framework shared by individual agents which take norms into account when reasoning about actions. The model could, however, be changed in several ways; see the discussion in Paper III, p. 90. For example, the normative system could be considered a part of the environment, in which a regimentation mechanism imposes on the agent system a desired behaviour. Another option is to regard the normative system with its associated mechanisms as an integrated part of the architecture of the individual agent. For example, it can represent the agent's internal normative system (its 'ethics'), or be used in the process of norm immergence to represent the agent's model of the norms effective in the surrounding society.

It has already been pointed out that the algebraic representation of normative systems discussed in this thesis is intended as a generic and useful tool for

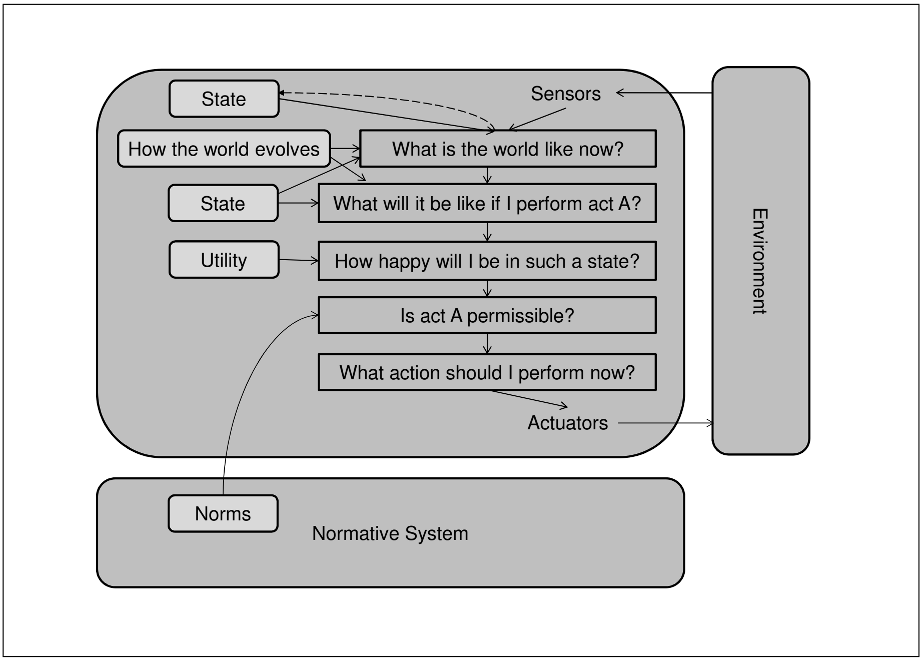

State
Sensors
How the world evolves
What is the world like now?
State
What will it be like if I perform act A?
Utility
How happy will I be in such a state?
Is act A permissible?
What action should I perform now?
Actuators
Environment
Norms
Normative System

Figure 3.2: A norm-regulated utility-based agent. The figure is inspired by Fig. 2.14 in (Russell & Norvig, 2010).

the development of normative MAS. The rule structure of norms, item (i) in the first guideline by Balke et al. (2013), is deeply embedded in this approach, but many of the other design choices mentioned in the guideline, regarding (ii) types of norms and (iii) other basic features of norms, cannot be addressed at the level of the tool itself. Instead, it is up to the developer who uses the tool to address these issues. To make matters more concrete, some of the issues will be discussed in Sect. 3.3 in relation to the Dalmas architecture for norm-regulated utility-based MAS.

3.2 Formal representation of norms

A simple categorization of norms is based on their structure: it is common to divide normative sentences into *categorically normative* and *conditional normative* sentences. A sentence of the former kind is often a descriptive sentence to which a 'norm-creating' operator has been applied. Examples of such operators are 'it shall be (the case) that' or 'it may be (the case) that'. For example, returning to the Waste Collectors example, the norm-creating operator 'it shall be the case that' can be applied to the descriptive sentence 'the protected spheres of agents x and y do not overlap with 9 squares', to form the categorically normative sentence 'it shall be the case that the protected spheres of agents x and y do not overlap with 9 squares' or simply 'the protected spheres of agents

x and y shall not overlap with 9 squares'.

A conditional norm is, in its purest form, an implication where the antecedent (i.e., the 'if-part') is descriptive and the consequent ('then-part') is categorically normative.[1] Hence, a conditional norm connects sentences of two different kinds, *viz.* descriptive sentences and normative sentences. It is a common view that norms have this conditional structure[2], which captures the conditions for their applicability as well as their normative effects. Thus, there is a link between the concepts of *norm* and *rule* (Balke et al., 2013).

As pointed out by Odelstad (2008, p. 19), in the computer science context there are some structural similarities between conditional norms and production rules. A *production rule* (or, simply, *production*) consists of a precondition (the 'if-part' of the production) and an action (the 'then-part'). A production is said to be *triggered* if its precondition matches the state of the world database, and *fired* if its action is executed. The production system's rule interpreter is responsible for determining when a production is to be triggered, and handling situations where more than one production is triggered, i.e., when a *conflict* arises. There are a number of pattern matching algorithms which can be used by the rule interpreter to determine the set of triggered productions, often called the *conflict set*. Most commonly used are forward chaining algorithms, which test the precondition of each rule against the current state of the database. Similarly, there are various conflict resolution strategies that use different algorithms for selecting from the conflict set which productions to execute. In some types of system, all productions in the conflict set are fired in some order, while in others one action must be selected (Luger, 2005, pp. 200ff).

To summarize, a common form of conditional norms is

$$p \to N(q)$$

where p and q are descriptive sentences, and N is a norm-creating operator. A popular approach is to use deontic operators such as Shall ('it is obligatory that', 'it shall be the case that') and May ('it is permissible that', 'it may be the case that') as norm-creating operators. It should, however, be noted that there are other options besides the approach based on (deontic) logic. The basic idea in Paper III is to create a set of norm-creating operators based on the notion of prohibition of different types of state transitions. We shall return to this idea in Sect. 4.2.

3.2.1 Kanger's deontic action-logic

The fundamental ideas of *deontic*[3] *logic* can be traced back to ancient Greece. In 1671, Leibniz pointed out the analogy between deontic concepts and the

[1] In practice, most conditional norms are based on *intermediate* concepts which are neither purely descriptive nor purely normative. Examples of such concepts are ownership, citizenship, and matrimony; cf. (Lindahl & Odelstad, 2013, p. 554).

[2] Note that unconditional norms may be regarded as special cases of conditional norms.

[3] The term is derived from the ancient Greek déon, roughly meaning 'as it should be', 'duly', '(that which is) binding or proper'.

so-called alethic modal concepts 'necessary', 'possible', 'impossible', and 'contingent'. Modern modal logic was founded by Lewis in his 1910 thesis (Garson, 2014). A first attempt at a formal theory for the normative concept 'ought' was made by Mally in 1926, but his system was unsatisfactory since it made possible to prove the absurd. Most modern research within this area has followed the pioneer work 'Deontic Logic', first published in 1951 by von Wright (McNamara, 2014). A comprehensive introdution to deontic logic, including a historical survey, is given by Hilpinen and McNamara (2013).

Deontic logic may be used as a tool for creating normative sentences concerned with obligation, permission, and related concepts.[4] As already mentioned, a normative sentence may be constructed by applying a norm-creating operator (e.g., Shall or May, with the intended interpretation 'it shall be that', and 'it may be that', respectively) to a descriptive sentence, which is a sentence expressing some fact about the state of the world. An example of a purely normative sentence is $\mathrm{Shall}\,p$, which is interpreted as 'it shall be that p', where p is some proposition. A conditional normative sentence consists of a combination of a descriptive sentence and a purely normative sentence. A typical conditional norm has the form $p \to \mathrm{Shall}\,q$, which can be interpreted as 'if p, then it shall be that q'.

An important contribution to deontic logic was made by Kanger, who combined it with a simple characterization of action. The study of the logic of action can be traced back at least to St. Anselm, but the modern study (based on symbolic logic) is attributed to, among others, Anderson, Fitch, Kanger, and von Wright.[5] Today there are two mainstream groups of theories within this area: *stit theory* (where 'stit' is an acronym for 'sees to it that') suggested by Belnap et al., and *dynamic logic*. The former has its roots in the philosophical tradition of (alethic) modal logic, while the latter grew out of the analysis of computer action within computer science.[6] An introduction to the logic of action is given by Segerberg, Meyer, and Kracht (2013).

Kanger's idea was to combine deontic logic with a logical system (in the following referred to as 'action-logic') which includes the rules of propositional logic and elementary set theory, together with rules for the binary action operator Do (Kanger, 1972). Common ways of reading $\mathrm{Do}(x, F)$, where F is some proposition, are 'x sees to it that F', 'x brings it about that F', or 'x is an agent of F'. Kanger made the following assumptions regarding the logic for Shall and Do:[7]

[4] There are also other kinds of normative sentences, e.g., value judgments.

[5] Note that von Wright (1968) argues that deontic concepts should be studied as part of *praxeology*, the study of the notions of agency and activity.

[6] Possibility and necessity are the traditional *alethic* modalities, or modalities of truth. Other modalities are the *temporal*, *deontic*, *epistemic*, and *doxastic* modalities.

[7] Kanger assumed the following rules ($\Rightarrow$ denotes the relation of logical consequence):

1. If F and $F \Rightarrow G$, then G.
2. If $F \Rightarrow G$, then $\neg G \Rightarrow \neg F$.
3. If $F \Rightarrow G$ and $G \Rightarrow H$, then $F \Rightarrow H$.

Table 3.1: One-agent types of normative positions

$\mathbf{T}_1$	$\text{May}\,\text{Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_2$	$\text{May}\,\text{Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_3$	$\text{May}\,\text{Do}(x,F)\,\&\,\neg\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_4$	$\neg\text{May}\,\text{Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_5$	$\text{May}\,\text{Do}(x,F)\,\&\,\neg\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_6$	$\neg\text{May}\,\text{Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May}\,\text{Do}(x,\neg F)$
$\mathbf{T}_7$	$\neg\text{May}\,\text{Do}(x,F)\,\&\,\neg\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\text{May}\,\text{Do}(x,\neg F)$

1. If $F \Rightarrow G$, then $\text{Shall}\,F \Rightarrow \text{Shall}\,G$.
2. $(\text{Shall}\,F \wedge \text{Shall}\,G) \Rightarrow \text{Shall}(F \wedge G)$.
3. $\text{Shall}\,F \Rightarrow \neg\text{Shall}\neg F$.
4. If $F \Leftrightarrow G$, then $\text{Do}(x,F) \Leftrightarrow \text{Do}(x,G)$.
5. $\text{Do}(x,F) \Rightarrow F$.

The last assumption tries to capture the notion of successful action; if x 'sees to it that' or 'brings it about that' F, then F is indeed the case. The operator May can be defined in terms of Shall in the following way:

$$\text{May}\,F \text{ if and only if } \neg\text{Shall}\neg F.$$

The result of combining Shall/May, Do, and external and internal negation of sentences is a powerful language for expressing normative sentences. A conditional norm formulated in this way may, for example, have the form

$$c(x,y) \rightarrow \text{Shall}\,\text{Do}(x,\neg d(x,y)),$$

which can be interpreted as 'if $c(x,y)$, then it shall be the case that x sees to it that not $d(x,y)$' or, more concisely, 'if $c(x,y)$, then x shall see to it that not $d(x,y)$'. Let, e.g., c stand for *Diff* and d for Lap_9 (see Sect. 3.1.1). Then the sentence above can be a concise representation of the sentence 'if x and y are different agents, then x shall see to it that the protected spheres of agents x and y do not overlap with 9 squares'.

3.2.2 The Kanger-Lindahl theory of normative positions

The language of deontic action-logic was used by Kanger as a basis for a theory of *rights*. Kanger's system contained six types of 'legal' or 'normative' positions. The theory was further developed by Lindahl (1977) in his three systems of types of normative positions. The simplest of these systems is a system of seven *one-agent types of normative positions*, based on the logic of Shall and Do. Given the underlying logic,

$$\neg\text{May Do}(x,F)\,\&\,\neg\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May Do}(x,\neg F)$$

is logically false. The types are formally defined as sets of tuples $\langle x,F\rangle$ defined by the statements in Table 3.1. For example, $\mathbf{T}_2 =$

$$\{\langle x,F\rangle : \text{May Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May Do}(x,\neg F)\}.$$

The intuition behind the notion of a normative 'position' is perhaps best explained by reference to its 'liberty space', and the relation 'less free than' (see below). Lindahl defines three *basic types of one-agent liberties* as follows:

$L_1 = \{\langle x,F\rangle : \text{May Do}(x,F)\}$
$L_2 = \{\langle x,F\rangle : \text{May}\,[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\}$
$L_3 = \{\langle x,F\rangle : \text{May Do}(x,\neg F)\}$

The *liberty space* of a given type $\mathbf{T}$ is the set of basic types of one-agent liberties corresponding to $\mathbf{T}$. For example, the liberty space of $\mathbf{T}_2$ is $\{L_1,L_2\}$, meaning that an agent x having this type of normative position has the liberties L_1 and L_2, but not L_3, with respect to the state of affairs F. Informally, x may 'see to it that' F and may refrain from 'seeing to it that' F and 'seeing to it that' not F, but may not 'see to it that' not F.

The one-agent types $\mathbf{T}_i$ are mutually disjoint and constitute a partition, which means that their union exhausts the set of logical possibilities.[8] In the following, I shall often allow myself to use the type names as shorthand for the expressions in Table 3.1; e.g., $\mathbf{T}_2(x,F)$ will often be used as an abbreviation of

$$\text{May Do}(x,F)\,\&\,\text{May}[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]\,\&\,\neg\text{May Do}(x,\neg F).$$

When the context permits, I will freely switch between regarding the types as sets of tuples and as syntactic elements. Note that $\mathbf{T}_5$ - $\mathbf{T}_7$ can be abbreviated as follows:

$\mathbf{T}_5(x,F) : \text{Shall Do}(x,F)$
$\mathbf{T}_6(x,F) : \text{Shall}\,[\neg\text{Do}(x,F)\,\&\,\neg\text{Do}(x,\neg F)]$
$\mathbf{T}_7(x,F) : \text{Shall Do}(x,\neg F)$

In Lindahl's terminology, $\mathbf{T}_2$ and $\mathbf{T}_4$ are each other's *converses*, and the same holds for $\mathbf{T}_5$ and $\mathbf{T}_7$. $\mathbf{T}_1$, $\mathbf{T}_3$, and $\mathbf{T}_6$, are their own converses, and thus *neutral* types. Each of Kanger's types is logically equivalent to one of Lindahl's types, except one which is equivalent to the disjunction of $\mathbf{T}_1$ and $\mathbf{T}_3$.

A relation 'less free than' is defined on the set $\{\mathbf{T}_1,...,\mathbf{T}_7\}$, such that if $\mathbf{T}$ is less free than $\mathbf{T}'$, then removing $\langle x,F\rangle$ from $\mathbf{T}'$ to form $\mathbf{T}$ reduces x's basic liberties with respect to F. The 'less free than' relation is a strict partial ordering on $\{\mathbf{T}_1,...,\mathbf{T}_7\}$, graphically represented by the Hasse diagram in Fig. 3.3. Note that some types, e.g., $\mathbf{T}_2$ and $\mathbf{T}_4$, are incomparable as regards this sense of freedom. Informally, a *deontic path* between two types $\mathbf{T}$ and $\mathbf{T}'$ is an unbroken chain of edges from $\mathbf{T}$ to $\mathbf{T}'$ in the Hasse diagram. The *deontic*

[8] Note that Lindahl uses italic T in the type names, whereas I use boldface roman $\mathbf{T}$ in order to adhere to the notation commonly used in the papers (where italic T is used instead for norm-building operators on conditions).

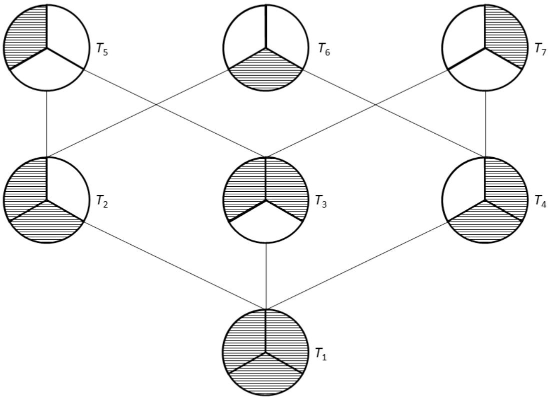


Figure 3.3: Hasse diagram for the 'less free than' relation. (Reiteration of the figure by Lindahl (1977, p. 105). A shaded left sector represents L_1, a shaded lower sector represents L_2, and a shaded left sector represents L_3.)

distance between $\mathbf{T}$ and $\mathbf{T}'$ is one plus the number of edges in a minimal deontic path between $\mathbf{T}$ and $\mathbf{T}'$.

Returning once again to the Waste Collectors world, we note that the sentence 'if x and y are different agents, then x shall see to it that the protected spheres of agents x and y do not overlap with 9 squares' can be concisely represented by

$$\mathit{Diff}(x, y) \to \mathbf{T}_7(x, \mathit{Lap}_9(x, y)).$$

3.2.3 Sergot's extended system

The Kanger-Lindahl theory of normative positions has been further developed by Jones and Sergot. A summary of their work is given in Paper IV; a more comprehensive overview is (Sergot, 2013). Jones and Sergot perform a similar analysis to that of Lindahl, by first generating a set of 64 *maxi-conjunctions* and then showing that 57 of the 64 conjunctions are internally inconsistent; the remaining seven conjunctions are precisely Lindahl's one-agent types of normative positions. Sergot (2001) performs a finer-grained analysis, which results in an extended set consisting of 15 types of normative positions, shown in Paper IV, Table 2.[9] According to Sergot (2001), each of Lindahl's $\mathbf{T}_3$, $\mathbf{T}_5$ and $\mathbf{T}_7$ is equivalent to one of Sergot's types, while each of $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_4$ and $\mathbf{T}_6$ is equivalent to a disjunction of three of Sergot's types. Thus, just as

[9] In the table, E_x (where x is regarded as a place-holder) corresponds to the Kanger-Lindahl Do operator, and F is some proposition. The table uses O ('obligatory') for Shall and P ('permissible') for May. The same logical properties are assumed of O, P, and E_x as in Lindahl's analysis.

Lindahl attains a finer-grained analysis than Kanger, Sergot attains a finer-grained analysis than Lindahl. Sergot (2001, Sect. 4) formalizes the idea of finer-grained analyses through the notions of *elaboration* and *refinement*. It is shown in Sergot (2001) that Sergot's set of 15 types is a refinement, in this technical sense, of Lindahl's set of seven types.

3.2.4 Normative systems as algebraic structures

In a series of papers, that are comprehensively summarized in (Lindahl & Odelstad, 2013), Lindahl and Odelstad have developed an algebraic framework for the treatment of, among other things, normative systems. An important application of their theory, the so-called theory of joining-systems, is to represent normative systems as so-called joining-systems, where conditional norms are joinings (represented by ordered pairs of conditions) from a structure of grounds to a structure of (normative) consequences, where the latter are defined in terms of an algebraic version of the theory of normative positions. Their idea is to use the one-agent types of normative positions as operators on descriptive *conditions* to get deontic conditions. A ν-ary condition d can be true or false of ν agents $x_1, ..., x_\nu$. Thus, $d(x_1, ..., x_\nu)$ is a statement which may be true or false. In the special case when the sequence of agents is empty, i.e. $\nu = 0$, d represents a proposition which may be true or false. It is possible to construct Boolean algebras of conditions, since negations d', conjunctions $(c \wedge d)$, and disjunctions $(c \vee d)$ can be formed in the following way:

$d'(x_1, ..., x_\nu)$ iff $\neg d(x_1, ..., x_\nu)$,
$(c \wedge d)(x_1, ..., x_\nu)$ iff $c(x_1, ..., x_p)$ and $d(x_1, ..., x_q)$, and
$(c \vee d)(x_1, ..., x_\nu)$ iff $c(x_1, ..., x_p)$ or $d(x_1, ..., x_q)$

where $\nu = \max(p, q)$. It is not necessary that $p = q$, but the free variables in $c(x_1, ..., x_p)$ must be the same, and in the same order, as the free variables in $d(x_1, ..., x_q)$. $\perp$ is the ν-ary empty condition such that for no $x_1, ..., x_\nu$, $\perp(x_1, ..., x_\nu)$, and $\top$ is the ν-ary universal condition such that $\top(x_1, ..., x_\nu)$ for all $x_1, ..., x_\nu$.

An algebraic version of the theory of normative positions is obtained by transforming each one-agent type of normative positions into a norm-creating operator operating on descriptive conditions. Consider, for example the following norm for the now familiar Waste Collectors world: 'the moving agent shall see to it that it does not stand in the same square as another agent'. We first reformulate: 'if x and y are different agents, and x is the moving agent, then x shall see to it that the protected speres of x and y do not overlap with 9 squares'. The consequent of this norm may be expressed as the normative condition $(T_7 Lap_9)$, obtained by applying the norm-building operator T_7 to the binary condition Lap_9 (see Sect. 3.1.1). For each one-agent type $\mathbf{T}_i$ of normative positions there is a corresponding operator T_i, and $(T_i Lap_9)$ is the ternary condition such that

$$(T_i Lap_9)(y, z, x) \text{ iff } \mathbf{T}_i(x, Lap_9(y, z)).$$

More generally, if d is a ν-ary (descriptive) condition, then $(T_i d)$ is the $\nu+1$-ary condition such that

$$(T_i d)(x_1, ..., x_\nu, x_{\nu+1}) \text{ iff } \mathbf{T}_i(x_{\nu+1}, d(x_1, ..., x_\nu)).$$[10]

Thus, each T_i is an operator on descriptive conditions, defined in terms of $\mathbf{T}_i$, and the result of applying each T_i to the condition d is a set $\{T_1 d, ..., T_7 d\}$ of normative conditions. Boolean compounds of these conditions are then formed by $\wedge, '$, and $\vee$. Together with an implicative relation R, which fulfils certain conditions, the Boolean algebra thus obtained forms a so-called *Boolean quasiordering* (*Bqo*). Lindahl and Odelstad define the notion of a *normative position condition-implication structure*, abbreviated *np-cis*, which is based on a *Bqo* on descriptive conditions and a *Bqo* on deontic conditions, so-called *cis-Bqo*'s. The *np-cis*, constructed to fulfil the requirements of the theory of normative positions,.was originally presented in (Lindahl & Odelstad, 2004). Let $\mathcal{B} = \langle B, \wedge, ', R\rangle$ be a *cis-Bqo* with a domain B of descriptive conditions $d_1, d_2, ...$, let

$$B_{\mathcal{T}} = \{T_i d | d \in B - \{\bot, \top\}, 1 \leq i \leq 7\},$$

and let $B^*_{\mathcal{T}}$ be the closure of $B_{\mathcal{T}}$ under $\wedge$ and $'$. A 'Boolean normative position-algebra' $\langle B^*_{\mathcal{T}}, \wedge, '\rangle$ is obtained, from which a *cis-Bqo* $\langle B^*_{\mathcal{T}}, \wedge, ', R\rangle$ is constructed: A *cis* $\langle B^*_{\mathcal{T}}, \wedge, ', R\rangle$ is a *normative position cis* with regard to $\mathcal{B}$ if for any $c, d \in B$ it holds that

(1) if $i \neq j$, then $(T_i d \wedge T_j d) R \bot$ (for $i, j \in 1, ..., 7$),

(2) $\top R (T_1 d \vee ... \vee T_7 d)$,

(3) $T_1 d Q T_1(d')$, $T_3 d Q T_3(d')$, $T_6 d Q T_6(d')$, $T_2 d Q T_4(d')$, $T_5 d Q T_7(d')$,

(4) if cQd then $T_i c Q T_i d$ (for $i \in 1, ..., 7$),

(5) $T_i(\top) Q \bot$ if $i = 1, 3, 4, 7$, and

(6) $T_i(\bot) Q \bot$ if $i = 1, 2, 3, 5$.

In the definition above, Q represents the indifference part of the relation R. Requirements (1)–(3) express restrictions on R in a normative position algebra, *viz.* that the types are mutually incompatible, jointly exhaustive, and that some types are the converses of others and some are neutral. Requirements (4)–(6) follow from the logic of Shall and Do. For more details on Boolean quasiorderings and condition implication structures, see, for example, (Lindahl & Odelstad, 2013; Odelstad & Boman, 2004). Two structures which are related to (or, more precisely, derivatives of) the *np-cis* will be presented in Sect. 4.4.

[10] In the following, the parentheses around $T_i d$ will be dropped to facilitate the presentation. Thus, $T_i d$ is a normative condition, and should be read $(T_i d)$. I adopt the convention that negation ($'$) binds tighter than the operator. Thus, $T_i d'$ should be read $T_i(d')$, not $(T_i d)'$.

The T_i operators transform ν-ary conditions on states s to $\nu+1$-ary conditions on 'situations' $\langle x, s\rangle$, where x is considered as the 'acting' or 'moving' agent. (Note the extra argument $x_{\nu+1}$ which is introduced by each T_i to separate the moving agent x from the agent $x_{\nu+1}$ to which the normative condition applies. The purpose of this separation is to allow for normative systems which can give agents other than the moving agent the right to perform immediate 'reaction acts', e.g., rewards or sanctions.)

The Lindahl-Odelstad algebraic approach to normative systems treats norms as *joinings*, represented as ordered pairs, from a *cis-Bqo* of *grounds* to a *cis-Bqo* $\langle B^*_{\mathcal{T}}, \wedge, ', R\rangle$ of *consequences*. Since the T_i operators introduce an extra agent argument $x_{\nu+1}$ for consequences, where ν is the arity of the operand, it is practical to introduce a similar operator (or, rather, family of operators) for grounds. The 'move operator' M transforms a ν-ary condition c on a state s to a $\nu+1$-ary condition (Mc) on a situation $\langle x, s\rangle$.[11] For example, M transforms the binary state condition *Diff* (see Sect. 3.1.1) to the ternary condition *MDiff*, such that

$$M\mathit{Diff}(x_1, x_2, x_3; x, s) \text{ iff } \mathit{Diff}(x_1, x_2; s) \text{ and } x_3 = x.$$

In many cases, it is desirable to formulate grounds based on conditions which specifically involve the acting agent. This can be accomplished by the M_1 operator, which is similar to the M operator, but also identifies x_3 with the first agent argument of *Diff*:

$$M_1\mathit{Diff}(x_1, x_2, x_3; x, s) \text{ iff } \mathit{Diff}(x_1, x_2; s) \text{ and } x_3 = x, \text{ and } x_3 = x_1.$$

More generally, M_κ (where $1 \le \kappa \le \nu$) transforms a ν-ary condition c on a state s to a $\nu+1$-ary condition $(M_\kappa c)$ on $\langle x, s\rangle$:

$$M_\kappa c(x_1, ..., x_\nu, x_{\nu+1}; x, s) \text{ iff } c(x_1, ..., x_\nu; s) \text{ and } x_{\nu+1} = x \text{ and } x_{\nu+1} = x_\kappa.$$

The M operator, and the notion of an *m-cis* based on this operator, was introduced by Odelstad and Boman (2004, Sect. 4.3), and its 'sibling' M_κ was introduced in (Hjelmblom, 2008).

The ordered pair $\langle M_1\mathit{Diff}, T_7\mathit{Lap}_9\rangle$ thus represents the norm

$$\forall x_1, x_2, x_3, x \in \Omega : \forall s \in S : M_1\mathit{Diff}(x_1, x_2, x_3; x, s) \rightarrow T_7\mathit{Lap}_9(x_1, x_2, x_3; x, s)$$

where Ω is a set of agents and S is a set of states. The representation of such sentences as joinings in so-called *Boolean joining-systems* makes possible reasoning about notions such as weakest ground, strongest consequence, and minimal norms (Lindahl & Odelstad, 2013, pp. 575ff).

Assuming a setting in which one agent x is distinguished as the moving agent, $\langle M_1\mathit{Diff}, T_7\mathit{Lap}_9\rangle$, where $T_7\mathit{Lap}_9 \in B^*_{\mathcal{T}}$, is the algebraic version of the single norm in Sect. 3.2.2. This norm is 'triggered' in the situation $\langle x, s\rangle$ for each agent y such that $\mathit{Diff}(x, y)$, in which case the normative condition $T_7\mathit{Lap}_9(x, y, x; x, s)$ is 'in effect'. From this follows $\mathbf{T}_7(x, \mathit{Lap}_9(x, y))$ in state s, i.e., that x has a normative position of type $\mathbf{T}_7$ with respect to the state of affairs $\mathit{Lap}_9(x, y)$ in s. Thus, if $\mathit{Diff}(x, y)$, then $\mathrm{Shall\,Do}(x, \neg\mathit{Lap}_9(x, y))$.

[11] Like before, the parentheses around Mc are dropped in the following.

3.3 The Dalmas architecture

Dalmas (Odelstad & Boman, 2004) is an abstract architecture for a class of norm-regulated MAS. A *deterministic* Dalmas is a simple MAS in which the actions of an agent are connected to transitions between system states. The agents in the system take turns to act; only one agent at a time (the 'moving' or 'acting' agent) may perform an action.

A specific Dalmas is given its unique features by various sets, operators and functions, which are the components of an ordered 9-tuple. Two components are of particular interest:

1. The *deontic structure-operator*, which for each situation of the system determines an agent's deontic structure (i.e., the set of permissible acts) on the feasible acts in the current situation.
2. The *preference structure-operator*, which for each situation determines the preference structure (i.e., the set of the most preferable acts) on the permissible acts.

A special kind of Dalmas is the *norm-regulated simple (deterministic)* Dalmas, which employs what is often referred to as 'negative permission', since its deontic structure consists of all acts that are not explicitly prohibited by a normative system. The preference structure consists of the most preferable of the acts in the deontic structure. In short, a Dalmas agent's behaviour is regulated by the combination of a normative system and a utility function; the agent chooses the most desirable act, according to the utility function, within the 'room for manoeuvre' determined by the norms. Thus the Dalmas architecture employs the 'agent oeconomicus norma' model (see Sect. 1.2) for MAS. The normative framework of the Dalmas is based on the Lindahl-Odelstad algebraic version of the Kanger-Lindahl theory of normative positions, in which normative consequences are formulated by applying norm-creating operators to descriptive conditions, as described in Sect. 3.2.4. From these general normative conditions follow normative sentences regarding specific states of affairs. In the next step, prohibition of individual actions in a specific situation is derived from such normative sentences (Odelstad, 2008; Odelstad & Boman, 2004).

The mechanism for deriving action prohibitions from normative statements will be explained by example: suppose that a normative system for the Waste Collectors world contains the norm $\langle M_1 \mathit{Diff}, T_4 \mathit{Lap}_9 \rangle$, and that the system contains two distinct agents e_1 and e_2. Now consider a state s of the system such that e_1 is in square $(2, 3)$ and e_2 is in $(3, 3)$. Then $\mathit{Lap}_6(e_1, e_2; s)$, i.e., $\mathit{Lap}_6(e_1, e_2)$ holds in s. In the situation $\langle e_1, s \rangle$, we have $M_1 \mathit{Diff}(e_1, e_2, e_1; e_1, s)$, since e_1 is the acting agent. Thus $T_4 \mathit{Lap}_9(e_1, e_2, e_1; e_1, s)$ is 'in effect', which means that e_1 has a normative position of type $\mathbf{T}_4$ with regard to $\mathit{Lap}_9(e_1, e_2)$ in s. In other words,

$$\begin{gathered}\neg\text{May}\,\text{Do}(e_1, \mathit{Lap}_9(e_1, e_2))\ \& \\ \text{May}[\neg\text{Do}(e_1, \mathit{Lap}_9(e_1, e_2)) \wedge \neg\text{Do}(e_1, \neg \mathit{Lap}_9(e_1, e_2))]\ \& \\ \text{May}\,\text{Do}(e_1, \neg \mathit{Lap}_9(e_1, e_2)).\end{gathered}$$

The analysis in (Odelstad & Boman, 2004, pp. 160f) yields the following interpretation of this statement: the action a is prohibited for e_1 in s if $\neg Lap_9(e_1, e_2)$ holds in s, and $Lap_9(e_1, e_2)$ would hold in $a(x, s)$, the resulting state when e_1 performs the action a in s. The other types were analyzed in a similar fashion, and to facilitate the presentation, a set of operators E_i^a was defined. Illustrating with the Lap_9 condition, the definitions are shown below (with quantifiers omitted):

2. $(E_2^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $[Lap_9(x_1, x_2; s) \wedge \neg Lap_9(x_1, x_2; a(x, s))]$
3. $(E_3^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $[Lap_9(x_1, x_2; s) \leftrightarrow Lap_9(x_1, x_2; a(x, s))]$
4. $(E_4^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $[\neg Lap_9(x_1, x_2; s) \wedge Lap_9(x_1, x_2; a(x, s))]$
5. $(E_5^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $\neg Lap_9(x_1, x_2; a(x, s))$
6. $(E_6^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $\neg[Lap_9(x_1, x_2; s) \leftrightarrow Lap_9(x_1, x_2; a(x, s))]$
7. $(E_7^a Lap_9)(x_1, x_2, x_3; x, s)$ iff $Lap_9(x_1, x_2; a(x, s))$

If $T_4 Lap_9(x_1, x_2, x_3; x, s)$ is in effect, and $E_i^a Lap_9(x_1, x_2, x_3; x, s)$ holds, then a is prohibited for x_3 in $\langle x, s\rangle$. It can be seen that $T_1 Lap_9(x_1, x_2, x_3; x, s)$ means that

$$\begin{gathered}\text{May}\,\text{Do}(x_3, Lap_9(x_1, x_2))\,\& \\ \text{May}[\neg\text{Do}(x_3, Lap_9(x_1, x_2)) \wedge \neg\text{Do}(x_3, \neg Lap_9(x_1, x_2))]\,\& \\ \text{May}\,\text{Do}(x_3, \neg Lap_9(x_1, x_2)).\end{gathered}$$

From this type follows no restriction on x_3's actions; hence there is no need for an E_1^a operator.

In the particular situation described above, if $E_4^a Lap_9(e_1, e_2, e_1; e_1, s)$ holds for some action a, then a is prohibited for e_1. Now suppose that, in the specific state s, there is an action n that moves e_1 one step north, from $(2, 3)$ to $(3, 3)$. Then $E_4^n Lap_9(e_1, e_2, e_1; e_1, s)$, since $\neg Lap_9(e_1, e_2)$ holds in s but $Lap_9(e_1, e_2)$ would hold in the state $n(x, s)$, the state which is reached if x performs n in s. Thus, from $T_4 Lap_9(e_1, e_2, e_1; e_1, s)$, together with $E_4^n Lap_9(e_1, e_2, e_1; e_1, s)$, it follows that n is prohibited for e_1.

To make comparisons between Dalmas and other architectures for (norm-regulated) multi-agent systems, it should first be noted that it is somewhat misleading to talk about *the* Dalmas architecture. As remarked by Odelstad and Boman (2004, p. 163), Dalmas is merely intended as the first step towards a theory of MAS architectures that use norms for restricting agent behaviour. A Dalmas is formalized as a number of set-theoretical predicates characterizing a multi-agent system, and by defining a number of such predicates which are specifications of the Dalmas, the theory can be developed in different directions. Thus, a hierarchy of set-theoretical predicates is obtained. In the remainder of this section, 'the Dalmas' refers to the class of Dalmases described in (Odelstad & Boman, 2004).

Criado et al. (2011, Table 2) compare a number of proposed languages for norm specification and implementation. The aspects considered in the comparison are (a) the deontic modalities employed, if any; (b) what is being controlled; (c) the enforcement mechanisms employed, if any; (d) the type of activation conditions employed, if any; and (e) the type of temporal constraints employed, if any. Adding to the comparison the language for norms used in the Dalmas architecture, we first note that (a) the normative consequences of the (conditional) norms are based on the seven one-agent types of normative positions, and thus implicitly on the deontic modalities O/Shall and P/May. As explained earlier, the Dalmas norms (b) control actions in an implicit way, by determining for an acting agent types of normative positions regarding specific states of affairs, from which prohibitions of specific actions may follow. Activation conditions for norms (d) are represented by the norms' antecedents, which are based on state conditions. As regards (c) and (e), there is no special support for temporal constraints, and the Dalmas agents always comply with the norms.

Boella et al. (2008) discuss different levels[12] in the development of NMAS. It could be argued that Dalmas is at the lowest level, consisting of closed systems in which norms are defined offline and imposed on agents, i.e., automatically enforced or *regimented*. However, since Dalmas norms are explicitly represented, the agents could be said to be aware not only of their own behaviour, but also of the norms. Therefore, an alternative viewpoint is that the Dalmas architecture employs *self-enforcement* of norms.[13] With the explicit representation of norms already at hand, it is, for example, straightforward to develop the architecture to reach the second level, in which agents can be aware of norms, autonomously reason about whether or not to comply with them, and use them in, e.g., communication and negation. Furthermore, the idea of letting some agents function as 'legislators' which have the power to add, remove or change norms (Odelstad & Boman, 2004, p. 163), aims for the third level, in which norms can be manipulated by agents. This would address some of the challenges for norm-regulated MAS identified by Boella et al. (2008). It would also embed evolutionary qualities in the MAS, and so enable studies into on-line norm emergence, agent adaptation and adjustable autonomy. In particular, it would enable unsupervised machine learning, allowing for the agents to adapt in what could be called an intelligent fashion to the environment, as well as to the actions by other agents in that environment.

According to the first guideline by Balke et al. (see Sect. 3.1.3), the developer of norm-regulated MAS should justify which definition of normative MAS is used. The Dalmas is not based on only one of the definitions reproduced in Sect. 3.1.3, but borrows characteristics from a number of them. It lies perhaps closest to the first definition, the so-called 'social definition', since the Dalmas is "governed by restrictions on patterns of behaviour of the agents" and these restrictions "dictate what actions are permitted or prohibited under a given set of conditions" (Balke et al., 2013, pp. 8f). On the other hand, since

[12] Table 1 in (Criado et al., 2011) gives a quick overview.

[13] See Fig. 1 with discussion in (Criado et al., 2011).

they are automatically enforced, DALMAS norms do not specify effects of compliance or non-compliance with norms. Furthermore, to follow the guideline, the developer should also explain choices made regarding the representation of norms. Norms in the DALMAS architecture have (i) a conditional structure, and are (ii) agent-specific behavioural norms. Regarding (iii) basic features such as norm application and norm change, it should be noted that the correct application of norms to concrete cases is an integrated part of the architecture, and that DALMAS norms cannot be changed on-line from 'within' the system.

3.4 Previous work: dnrDALMAS

A generic Prolog instrumentalization, in the form of a SICStus Prolog module, of the DALMAS architecture was presented and discussed in (Hjelmblom, 2008). The Prolog module, `dnrDALMAS`, may be used to create standalone text-based implementations of specific DALMASes. The `dnrDALMAS` module contains a set of predicates for creating and initializing a DALMAS implementation, including a normative system, querying the state of the DALMAS, and performing runs of the system. A DALMAS implementation consists of a set of (user-defined) primary predicates, which correspond to the primary functions characterizing the system, written in Prolog. A central part of the implementation is the `prohibited/3` predicate, which determines if an act is permissible or not according to the normative system, by checking for each norm and each available act whether or not a prohibition of the act follows from the norm. This mechanism for testing norms resembles a forward chaining rule interpreter of a production system, which, for each available action and each combination of agents in the system, sequentially tests the ground of each norm. If the ground of a norm is fulfilled (is 'triggered', in the terminology of production systems), then the consequence of the norm, a normative position for some agent regarding some state of affairs, is added to the set of normative positions which hold in the current situation. (The term 'conflict set' is avoided here, since the grounds of several norms may be fulfilled without their consequences necessarily being in 'conflict'.) It is then immediately tested if the normative position leads to a prohibition of an action. Thus, it can be said that each triggered 'production' (norm) fires, in the sense that it is tested if its consequence prohibits an action. If it does, then the algorithm moves on to test the next available action.

Using the `dnrDALMAS` module, two implementations of example systems (the *Colour and Form system* and the *Waste Collectors system*) were created and presented in (Hjelmblom, 2008). These applications function both as standalone programs with simple text-based user interfaces, and as logic servers. The code is available for download and publicly and freely disseminated.[14]

[14] `http://drpa.se/norms/nrtssit/`

Chapter 4

Contribution

4.1 Preamble

This chapter gives an account of the main contribution of the thesis. Most, but not all, of its content consists of references to papers I, II, III, IV, and V. The material is, when necessary, rearranged to form a coherent exposition, and sometimes augmented with unpublished material and extended remarks or discussions. The latter includes, for example, the definition of the relation 'at least as restrictive as' and its interpretation as an 'operationalization' of the relation 'less free than' on types of normative positions. To facilitate for the reader, an overview of the notation and the relationships between various operators occuring in this chapter is given in the appendix (pp. 63).

4.2 Transition type prohibition

4.2.1 Overview

Papers II & III present an approach to normative systems for MAS modelled as transition systems, where actions are associated with transitions between different system states. The basic idea is to create a set of norm-creating operators, based on prohibition of different types of state transitions with respect to some condition d on a number of agents x_1, ..., x_ν. Thus, the permission or prohibition of actions is related to the permission or prohibition of these types of state transitions. The paper introduces the notion of a *norm-regulated transition system situation*, which is intended to represent a single step in the run of a (norm-regulated) transition system. The representation of conditional norms in this framework is similar to the algebraic representation used in the norm-regulated DALMAS architecture (see Sect. 3.3), but not based on the Kanger-Lindahl one-agent types of normative positions. Instead, the set of norm-creating operators

$$\{P_1, P_2, P_{2\prime}, P_4, P_{4\prime}, P_5, P_6, P_{6\prime}, P_7\}$$

in papers II, III originates from a systematic exploration of the possible types of state transitions with respect to $d(x_1,...,x_\nu)$.[1] Thus, this is an example of an approach to norms that is not explicitly based on deontic logic. Sect. 4.4 discusses a potential connection between transition type prohibition and the theory of normative positions, *viz.* that prohibition of different types of state transitions may provide a semantics for (an extended set of) the types of one-agent normative positions.

4.2.2 Norm-regulated transition system situations

A norm-regulated transition system situation is represented by an ordered pair $\langle \mathbf{S}, \mathcal{N} \rangle$ where $\mathbf{S}$ is a transition system situation and $\mathcal{N}$ is a normative system. A *transition system situation* is an ordered 5-tuple $\mathbf{S} = \langle x, s, A, \Omega, S \rangle$ where the pair $\langle x, s \rangle$, consisting of a state s and an acting ('moving') agent x, is the 'situation' part of the transition system situation, and $A = \{a_1, ..., a_m\}$ is a set of available actions, $\Omega = \{x_1, ..., x_n\}$ is a set of agents, and S is a set of states. An action a may be regarded as a function such that $a(x, s) = s^+$ means that s^+ is the resulting state when x performs act a in state s.

Since, in s as well as in s^+, either $d(X_\nu)$ or $\neg d(X_\nu)$ holds, there are with regard to $d(X_\nu)$ four possible alternatives for the transition from s to a following state s^+:

I. $d(X_\nu; s)$ & $d(X_\nu; s^+)$

II. $\neg d(X_\nu; s)$ & $d(X_\nu; s^+)$

III. $d(X_\nu; s)$ & $\neg d(X_\nu; s^+)$

IV. $\neg d(X_\nu; s)$ & $\neg d(X_\nu; s^+)$

Each alternative represents a basic type of transition with regard to the state of affairs $d(X_\nu)$. The 'transition type conditions' $C_i^a d(X_\nu, x_{\nu+1}; x, s)$, $i \in \{2, 2', 4, 4', 5, 6, 6', 7\}$, are defined in Paper III as disjunctions of basic transition types, to indicate whether or not the transition which corresponds to agent $x_{\nu+1}$ performing action a in situation $\langle x, s \rangle$ has any of the corresponding basic transition types with regard to $d(X_\nu)$. Using the Lap_9 condition from Sect. 3.1.1 for illustration, C_4^a is defined as follows: For all $x_1, x_2, x_3, x \in \Omega$ and all $s \in S$,

$$C_4^a Lap_9(x_1, x_2, x_3; x, s) \text{ iff } \neg Lap_9(x_1, x_2; s) \,\&\, Lap_9(x_1, x_2; a(x_3, s)).$$

C_7^a is defined as follows: For all $x_1, x_2, x_3, x \in \Omega$ and all $s \in S$,

$$\begin{gathered} C_7^a Lap_9(x_1, x_2, x_3; x, s) \\ \text{iff} \\ [Lap_9(x_1, x_2; s) \,\&\, Lap_9(x_1, x_2; a(x_3, s))] \\ \text{or } [\neg Lap_9(x_1, x_2; s) \,\&\, Lap_9(x_1, x_2; a(x_3, s))]. \end{gathered}$$

[1] In the remainder of this section, the sequence $x_1, ..., x_\nu$ will be abbreviated as X_ν.

Note that the latter can be simplified to

$$C_7^a Lap_9(x_1, x_2, x_3; x, s) \text{ iff } Lap_9(x_1, x_2; a(x_3, s)).$$

Important relations between the different C_i^a conditions are summarized in Fig. 4.1. Conditions that are equivalent due to the symmetries listed in III are placed in the same box, and implicative relations are illustrated with arrows. The figure shows, for example, that for all a and all d,

$$C_4^a d(X_\nu, x_{\nu+1}; x, s) \text{ iff } C_2^a d'(X_\nu, x_{\nu+1}; x, s),$$

and that

$$C_4^a d(X_\nu, x_{\nu+1}; x, s) \text{ implies } C_7^a d(X_\nu, x_{\nu+1}; x, s).$$

For convenience, let $C_i^a \subseteq C_j^a$ denote that all disjuncts of C_i^a are also disjuncts of C_j^a. Hence, if $C_i^a \subseteq C_j^a$, then

$$C_i^a d(X_\nu, x_{\nu+1}; x, s) \text{ implies } C_j^a d(X_\nu, x_{\nu+1}; x, s).$$

As in the Dalmas architecture, a norm in the normative system $\mathcal{N}$ is represented by an ordered pair $\langle c, P_j d\rangle$, where the condition c on the situation $\langle x, s\rangle$ is the *ground* of the norm and the (normative) condition $P_j d$ is its *consequence*. If $\langle c, P_j d\rangle$ is a norm in $\mathcal{N}$, and there are conditions c, d and a sequence of agents $x_1, ..., x_\nu \in \Omega$ such that $c(x_1, ..., x_p)$ in the situation $\langle x, s\rangle$, then the normative condition $P_j d(x_1, ..., x_p, x_{\nu+1}; x, s)$ is in effect. In that case, each action $a \in A$ such that $C_j^a d(x_1, ..., x_q, x_{\nu+1}; x, s)$, where $\nu = \max(p, q)$, is prohibited for the agent $x_{\nu+1} \in \Omega$ in $\langle x, s\rangle$.

The intuition behind the norm-regulated transition system situation is that it represents a 'snapshot' of a transition system in which each transition is deterministic and represents the action of a single agent. It has been instrumentalized into a general-level tool, `NRTSSit`, for implementation of a wide range of specific norm-regulated agent systems. The tool, an adaptation of the `jDALMAS` tool for norm-regulated DalmaSes, has a client/server architecture consisting of a Java client library for creating graphical applications and a Prolog module for creating for normative system logic servers. An example system is presented in Sect. 4.5.2.

The relation 'at least as restrictive as'

Given the definition of *Prohibited* in Paper III (p. 87), a relation 'at least as restrictive as' can be defined on the set of operators P_i in the following way: 'P_i is at least as restrictive as P_j' if and only if 'the set of basic transition types prohibited by P_i is a superset of the set of basic transition types prohibited by P_j'. The relation is illustrated in the Hasse diagram in Fig. 4.2, where a filled square indicates that the corresponding basic transition type is not prohibited, whilst a white square indicates a prohibition. It follows that 'P_i is at least as restrictive as P_j' means that 'the set of actions prohibited by P_i applied to some condition in a specific situation is a superset of the set of actions prohibited

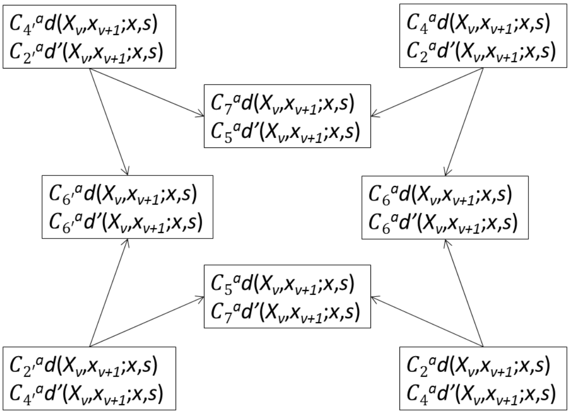

$C_{4'}{}^a d(X_v,x_{v+1};x,s)$
$C_{2'}{}^a d'(X_v,x_{v+1};x,s)$
$C_4{}^a d(X_v,x_{v+1};x,s)$
$C_2{}^a d'(X_v,x_{v+1};x,s)$
$C_7{}^a d(X_v,x_{v+1};x,s)$
$C_5{}^a d'(X_v,x_{v+1};x,s)$
$C_{6'}{}^a d(X_v,x_{v+1};x,s)$
$C_{6'}{}^a d'(X_v,x_{v+1};x,s)$
$C_6{}^a d(X_v,x_{v+1};x,s)$
$C_6{}^a d'(X_v,x_{v+1};x,s)$
$C_5{}^a d(X_v,x_{v+1};x,s)$
$C_7{}^a d'(X_v,x_{v+1};x,s)$
$C_{2'}{}^a d(X_v,x_{v+1};x,s)$
$C_{4'}{}^a d'(X_v,x_{v+1};x,s)$
$C_2{}^a d(X_v,x_{v+1};x,s)$
$C_4{}^a d'(X_v,x_{v+1};x,s)$

Figure 4.1: Relationships between transition type conditions.

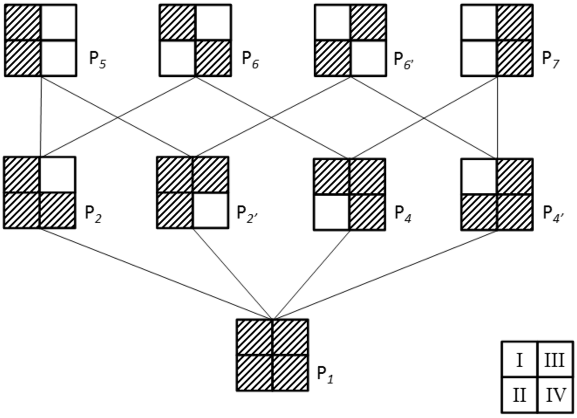


Figure 4.2: The ‘at least as restrictive as’ relation on transition type prohibition operators.

by P_j applied to the same condition in the same situation'. For example, let $x_1, x_2, x_3, x \in \Omega$, $s \in S$, and $a \in A$, and assume that $C_4^a Lap_9(x_1, x_2, x_3)$ holds in $\langle x, s \rangle$. Then $P_4 Lap_9(x_1, x_2, x_3; x, s)$ implies $Prohibited_{x,s}(x_3, a)$, i.e., that a is prohibited for x_3 in $\langle x, s \rangle$. Now, as Fig. 4.1 reveals, $C_4^a Lap_9(x_1, x_2, x_3; x, s)$ implies $C_7^a Lap_9(x_1, x_2, x_3; x, s)$, and hence $P_7 Lap_9(x_1, x_2, x_3; x, s)$ would also imply $Prohibited_{x,s}(x_3, a)$ in the same situation. More generally, given some condition d, if a is prohibited by the normative condition $P_4 d$ then a will also be prohibited by $P_7 d$. In other words, the set of actions prohibited by $P_4 d$ is a subset of the set of actions prohibited by $P_7 d$. Similar arguments can be constructed for $P_2 d$ and $P_5 d$, $P_2 d$ and $P_6 d$, $P_{2'} d$ and $P_5 d$, $P_{2'} d$ and $P_{6'} d$, $P_4 d$ and $P_6 d$, $P_{4'} d$ and $P_{6'} d$, and $P_{4'} d$ and $P_7 d$. To summarize,

$$[Prohibited_{x,s}^{P_i d}(x_{\nu+1}, a) \text{ implies } Prohibited_{x,s}^{P_j d}(x_{\nu+1}, a)] \text{ if } C_i^a \subseteq C_j^a,$$

where $Prohibited_{x,s}^{P_i d}(x_{\nu+1}, a)$ means that a is prohibited for $x_{\nu+1}$ in $\langle x, s \rangle$, according to the normative condition $P_i d$.

It should be noted that the 'at least as restrictive as' relation is a strict partial ordering on $\{P_1, P_2, P_{2'}, P_3, P_{3'}, P_5, P_6, P_{6'}, P_7\}$. Some operators, such as P_2 and $P_{6'}$, are incomparable as regards restrictiveness in this sense. In one sense, $P_{6'}$ is more restrictive than P_2, since there are two basic transition types that are prohibited by $P_{6'}$ but not by P_2, but in another sense P_2 is more restrictive than $P_{6'}$, since there is one basic transition type that is prohibited by P_2 but not by $P_{6'}$. As regards P_1, it is vacuously true that all other types are at least as restrictive as P_1, since the set of actions prohibited by P_1 is, in all cases, the empty set.

4.3 One-agent types of normative positions: extended systems

Paper IV further examines the approach to normative systems in which normative conditions are based on the theory of normative positions. One contribution is an interpretation of Sergot's extended system of one-agent types of normative positions from Sect. 3.2.3. It is argued that this interpretation can provide a semantics for system norms, but is less suitable as a basis for agent-specific norms. In pursuit of a representation of agent-specific norms, Paper IV examines other possible extensions of the theory of normative positions. The main contribution follows from an observation by Odelstad (2008) that the statement 'x does not see to it that F and does not see to it that not F' can be understood in two different ways: either 'passivity' with regard to F, or 'opposivity'. The idea is that

$$\neg\mathrm{Do}(x, F) \& \neg\mathrm{Do}(x, \neg F)$$

can be decomposed into two distinct cases. Using $\mathbf{\Delta}$ as an abbreviation for Do, and (with a special interpretation in mind) $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ as mnemonics for 'Leave' and 'Oppose', respectively,

$$\neg\mathbf{\Delta}(x,F)\&\neg\mathbf{\Delta}(x,\neg F)$$

is assumed to be logically equivalent to the disjunction of the two formulas $\mathbf{\Lambda}(x,F)$ and $\mathbf{\Omega}(x,F)$. Based on this idea, two extended sets of types of normative positions are constructed. One of the systems thus obtained is then given an interpretation, a semantics for agent-specific norms, in terms of transition type prohibition.

The result of decomposing

$$\neg\mathrm{Do}(x,F)\&\neg\mathrm{Do}(x,\neg F)$$

in the way described in Paper IV, Sect. 2.2 is another extended set of 15 types of normative positions, shown in Paper IV, Table 7. With the additional assumptions $\mathbf{\Lambda}(x,F)$ iff $\mathbf{\Lambda}(x,\neg F)$ and $\mathbf{\Omega}(x,F)$ iff $\mathbf{\Omega}(x,\neg F)$, Lindahl's notions of liberty types and liberty spaces (see Sect. 3.2.2) are generalized, and a 'less free than' relation is defined on the extended set of types.

The definition of the extended set of types of normative positions was motivated by the pursuit of a mapping between the nine transition type prohibition operators and the types of one-agent normative positions. By the additional assumption that

$$\begin{gathered}\mathrm{May}\,\mathbf{\Delta}(x,F)\,\&\,\mathrm{May}\,\mathbf{\Delta}(x,\neg F)\\ \text{iff}\\ \mathrm{May}\,\mathbf{\Lambda}(x,F)\,\&\,\mathrm{May}\,\mathbf{\Omega}(x,F),\end{gathered}$$

and elimination of self-contradictory types, the extended system of 15 types of normative positions is reduced to a system of nine types; see Paper IV, Table 8. Again, a 'less free than' relation is defined on this 'reduced extended' set of types. A Hasse diagram for this relation is shown in Paper IV, Fig. 1.

4.4 Algebraic versions of extended systems

A central contribution of Paper IV is the construction of an 'extended normative position *cis*' (Def. 1) and a 'reduced extended normative position *cis*' (Def. 2). These structures, abbreviated *np15-cis* and *np9-cis*, respectively, fulfil the requirements in the extended (resp., the 'reduced extended') theory of one-agent normative positions. In both cases, the procedure is similar to the one described in Sect. 3.2.4.

The chief aim of Paper IV is to build a bridge between the Kanger-Lindahl theory of normative positions, on the one hand, and the notion of transition type prohibition in Papers II, III, on the other. A comparison between the Hasse diagrams for the 'less free than' relation on the 'reduced extended' types of normative positions (Fig. 1 in Paper IV) and the 'at least as restrictive as' relation on transition type prohibition operators (Fig. 4.2) reveals a striking structural similarity. This supports the idea that 'at least as restrictive as' gives a kind of operational semantics to 'less free than', and that the transition type prohibition operators provide a semantics for the 'reduced extended' types of one-agent normative positions. Note that the stipulations made regarding the interpretation of the extended types are consistent with the logic of Shall and Do, but are certainly not the only possible; the generality of the theory of normative positions allow for many different interpretations.

Paper V investigates the use of an evolutionary algorithm for evolving normative systems for problem-solving MAS. The analysis of the example studied in the paper gives, as an interesting by-product, the insight that norms whose grounds and consequences are based on the same set of descriptive conditions can become 'operationally equivalent', in the sense that they have exactly the same effect in the same situations. In other words, norms can become operationally equivalent in cases where conditions belonging to the set of potential grounds of the norms also appear in the set of descriptive conditions underlying the set of potential consequences. Sect. 3.2 in Paper V discusses conditions for when two transition type prohibition operators P_i and P_j are 'equally prohibitive'. The intuition behind this notion is quite simple. Suppose, for example, that $Lap_9(x, y)$ holds in some state s. In this case, the result of x's compliance with $P_7 Lap_9(x, y, x)$ would be the same as the result of x's compliance with $P_{6\Omega} Lap_9(x, y, x)$, *viz.*, that $Lap_9(x, y)$ ceases to hold.[2] Hence, norms such as $\langle M Lap_9, P_7 Lap_9\rangle$ and $\langle M Lap_9, P_{6\Omega} Lap_9\rangle$ will be 'operationally equivalent', since if $Lap_9(x, y)$ holds in s then their normative consequences are equally prohibitive, and otherwise their normative consequences are not in effect.

The findings in Paper V can be generalized as follows. First, note that C^a_5, $C^a_{6\Lambda}$, $C^a_{6\Omega}$, and C^a_7, can be expressed as disjunctions of basic transition types:

- $C^a_5 d(X_\nu, x_{\nu+1}; x, s)$ iff
  $[d(X_\nu; s)\ \&\ \neg d(X_\nu; a(x_{\nu+1}, s))]$ or $[\neg d(X_\nu; s)\ \&\ \neg d(X_\nu; a(x_{\nu+1}, s))]$
- $C^a_{6\Lambda} d(X_\nu, x_{\nu+1}; x, s)$ iff
  $[d(X_\nu; s)\ \&\ \neg d(X_\nu; a(x_{\nu+1}, s))]$ or $[\neg d(X_\nu; s)\ \&\ d(X_\nu; a(x_{\nu+1}, s))]$
- $C^a_{6\Omega} d(X_\nu, x_{\nu+1}; x, s)$ iff
  $[d(X_\nu; s)\ \&\ d(X_\nu; a(x_{\nu+1}, s))]$ or $[\neg d(X_\nu; s)\ \&\ \neg d(X_\nu; a(x_{\nu+1}, s))]$
- $C^a_7 d(X_\nu, x_{\nu+1}; x, s)$ iff
  $[d(X_\nu; s)\ \&\ d(X_\nu; a(x_{\nu+1}, s))]$ or $[\neg d(X_\nu; s)\ \&\ d(X_\nu; a(x_{\nu+1}, s))]$

Now consider, for example, a normative system based on a *cis-Bqo* $\langle C, \wedge, ', R_C\rangle$ of descriptive grounds and a *cis-Bqo* $\langle D^*_{\mathcal{T}}, \wedge, ', R_{\mathcal{T}}\rangle$ of normative consequences, such that D is the set of descriptive grounds underlying $\langle D^*_{\mathcal{T}}, \wedge, ', R_{\mathcal{T}}\rangle$. Let $c \in C$ and $d \in D$, and suppose that there is an implicative relation between c

[2] Note that when associating the extended types of normative positions with the P_i operators, as in papers IV and V, $P_7 Lap_9(x, y, x)$ can be read as Shall Do$(x, \neg Lap_9(x, y))$ and $P_{6\Omega} Lap_9(x, y)$ can be read as Shall Oppose $Lap_9(x, y)$. Here, and in the following, C and P will be indexed by 2Λ, 2Ω, ... instead of 2, $2'$, ..., to highlight the connection between the reduced extended types of normative positions and the transition type prohibition operators. (See Paper IV.)

and d, i.e., that (i) c implies d.[3] Now suppose that, for some agents $x_1, ..., x_\nu$, $c(X_\nu)$ holds in s. We may then note the following:

1. $C^a_{2\Omega}d(X_\nu, x_{\nu+1}; x, s)$ and $C^a_{4\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ are false whenever $c(X_\nu; s)$, since by assumption $c(X_\nu; s)$ implies $d(X_\nu; s)$. It follows that, when $c(X_\nu; s)$, the set of actions that would be prohibited by the normative conditions $P_{2\Omega}d$ and $P_{4\Lambda}d$ is exactly the same as the set of actions that would be prohibited by P_1d, *viz.* the empty set. Informally, P_1, $P_{2\Omega}$, and $P_{4\Lambda}$ are 'equally prohibitive' with regard to d when $c(X_\nu)$ holds and c implies d.

2. The second disjunct of $C^a_5d(X_\nu, x_{\nu+1}; x, s)$ and the second disjunct of $C^a_{6\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ must be false. Hence, if $c(X_\nu; s)$, then $C^a_5d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{6\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{2\Lambda}d(X_\nu, x_{\nu+1}; x, s)$, which means that $P_{2\Lambda}d$, P_5d, and $P_{6\Lambda}d$ prohibit exactly the same set of actions when $c(X_\nu; s)$.

3. The second disjunct of $C^a_{6\Omega}d(X_\nu, x_{\nu+1}; x, s)$ and the second disjunct of $C^a_7d(X_\nu, x_{\nu+1}; x, s)$ must be false. Hence, if $c(X_\nu; s)$, then $C^a_{6\Omega}d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_7d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{4\Omega}d(X_\nu, x_{\nu+1}; x, s)$; so, when $c(X_\nu; s)$, $P_{4\Omega}d$, $P_{6\Omega}d$, and P_7d prohibit the same set of actions.

Now suppose instead that (ii) c implies d', and that $c(X_\nu)$ holds in s:

4. $C^a_{2\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ and $C^a_{4\Omega}d(X_\nu, x_{\nu+1}; x, s)$ are false whenever $c(X_\nu; s)$; by assumption $c(X_\nu; s)$ implies $\neg d(X_\nu; s)$. When $c(X_\nu; s)$, the set of actions that would be prohibited by the normative conditions $P_{2\Lambda}d$ and $P_{4\Omega}d$ is exactly the same as the set of actions that would be prohibited by P_1d, *viz.* the empty set.

5. The first disjunct of $C^a_{6\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ and the first disjunct of $C^a_7d(X_\nu, x_{\nu+1}; x, s)$ must be false. Hence, if $c(X_\nu; s)$, then $C^a_7d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{6\Lambda}d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{4\Lambda}d(X_\nu, x_{\nu+1}; x, s)$, which means that $P_{4\Lambda}d$, $P_{6\Lambda}d$ and P_7d prohibit exactly the same set of actions when $c(X_\nu; s)$.

6. The first disjunct of $C^a_5d(X_\nu, x_{\nu+1}; x, s)$ and the first disjunct of $C^a_{6\Omega}d(X_\nu, x_{\nu+1}; x, s)$ must be false. Therefore, if $c(X_\nu; s)$, then $C^a_5d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{6\Omega}d(X_\nu, x_{\nu+1}; x, s)$ iff $C^a_{2\Omega}d(X_\nu, x_{\nu+1}; x, s)$, which means that $P_{2\Omega}d$, P_5d, and $P_{6\Omega}d$ prohibit exactly the same set of actions when $c(X_\nu; s)$.

It is suggested in Paper V that the notion of equally prohibitive operators may be a suitable foundation for a notion of operational equivalence of norms. From 1 it follows that the set of actions that would be prohibited by $\langle Mc, P_1d\rangle$, $\langle Mc, P_{2\Omega}d\rangle$, and $\langle Mc, P_{4\Lambda}d\rangle$, is exactly the same, *viz.* the empty set. (If $\neg c(X_\nu; s)$, then the grounds are not fulfilled; if $c(X_\nu; s)$, then P_1, $P_{2\Omega}$,

[3] In other words, there is some overlap between C and D. As an illustration, suppose that c denotes the condition Lap_0 and d denotes *Diff*. Then it follows that c implies d.

Table 4.1: Operationally equivalent norms in the case of (i): c implies d.

$\langle Mc, P_1 d\rangle, \langle Mc, P_1 d'\rangle, \langle Mc, P_{2\Omega} d\rangle, \langle Mc, P_{4\Omega} d'\rangle, \langle Mc, P_{4\Lambda} d\rangle, \langle Mc, P_{2\Lambda} d'\rangle$
$\langle Mc, P_{2\Lambda} d\rangle, \langle Mc, P_{4\Lambda} d'\rangle, \langle Mc, P_5 d\rangle, \langle Mc, P_7 d'\rangle, \langle Mc, P_{6\Lambda} d\rangle, \langle Mc, P_{6\Lambda} d'\rangle$
$\langle Mc, P_{4\Omega} d\rangle, \langle Mc, P_{2\Omega} d'\rangle, \langle Mc, P_{6\Omega} d\rangle, \langle Mc, P_{6\Omega} d'\rangle, \langle Mc, P_7 d\rangle, \langle Mc, P_5 d'\rangle$

Table 4.2: Operationally equivalent norms in the case of (ii): c implies d'.

$\langle Mc, P_1 d\rangle, \langle Mc, P_1 d'\rangle, \langle Mc, P_{2\Lambda} d\rangle, \langle Mc, P_{4\Lambda} d'\rangle, \langle Mc, P_{4\Omega} d\rangle, \langle Mc, P_{2\Omega} d'\rangle$
$\langle Mc, P_{4\Lambda} d\rangle, \langle Mc, P_{2\Lambda} d'\rangle, \langle Mc, P_{6\Lambda} d\rangle, \langle Mc, P_{6\Lambda} d'\rangle, \langle Mc, P_7 d\rangle, \langle Mc, P_5 d'\rangle$
$\langle Mc, P_{2\Omega} d\rangle, \langle Mc, P_{4\Omega} d'\rangle, \langle Mc, P_{6\Omega} d\rangle, \langle Mc, P_{6\Omega} d'\rangle, \langle Mc, P_5 d\rangle, \langle Mc, P_7 d'\rangle$

and $P_{4\Lambda}$ are equally prohibitive with regard to d.) Informally, these norms are 'operationally equivalent'. From 2 it follows that $\langle Mc, P_{2\Lambda} d\rangle$, $\langle Mc, P_5 d\rangle$, and $\langle Mc, P_{6\Omega} d\rangle$ prohibit the same set of actions, and are thus also operationally equivalent, and from 3 it follows that $\langle Mc, P_{4\Lambda} d\rangle$, $\langle Mc, P_{6\Omega} d\rangle$, and $\langle Mc, P_7 d\rangle$ are operationally equivalent. Taking the equivalences in Fig. 4.1 into account, Table 4.1 summarizes case (i), and Table 4.2 summarizes case (ii). For each row in the tables, the norms in the row are operationally equivalent under the listed assumptions. What should be kept in mind here is that these assumptions are special cases. In the general case, the conditions c and d are not related by an implicative relation; they may, in fact, be members of wholly disjunct sets of conditions. In this case, the different P_i operators are not equally prohibitive.

4.5 Applications

4.5.1 The Waste Collectors system

Paper I further develops the Waste collectors system (Sect. 3.1.1) into a graphical application. The main purpose is to demonstrate how to develop specific Dalmas applications, using the general-level `jDALMAS` library as a Java client and the `dnrDALMAS` Prolog module as a logic server for the normative system. A variant of the Waste collectors system is used in Paper V to illustrate the ideas presented there; see Sect. 4.5.5.

4.5.2 The Rooms system

To illustrate the notion of a norm-regulated transition system situation and the use of the `NRTSSit` Java/Prolog tool, a variant of the *Rooms* example employed by Craven and Sergot (2008) was instrumentalized in Paper III. Albeit quite simple, the instrumentalization of the *Rooms* system illustrates some features of the iterated application of norm-regulated transition system situations, such as the ability to investigate the interplay between a normative system and agent utility functions, and thus demonstrates how to use the `NRTSSit` tool to develop norm-regulated utility-based MAS.

4.5.3 The Estates example

Paper IV illustrates the connection between the types of one-agent normative positions and transition type prohibition operators by a simplified version of the ‘ownership to an estate’ example employed by Lindahl and Odelstad (2013). The simple instrumentalization discussed in Paper IV is not a fully developed norm-regulated MAS with iterated application of norm-regulated transition system situations; it is merely intended to demonstrate the effect of applying a normative system in a single situation. The application can be used to experiment with different normative systems in different situations, to get an idea of how well the instrumentalized normative system captures the intention behind the system described by Lindahl and Odelstad. Such experiments can give valuable insights on the intricate interplay between the specification of actions and the specification of normative systems, as well as the reasonableness of the interpretation of types of normative positions in terms of transition type prohibition. Furthermore, the *Estates* application demonstrates how the normative framework handles norms with grounds and/or consequences that are Boolean combinations of simpler conditions.

4.5.4 The Forest Cleaner system

Automation of forest cleaning, the second example in Ch. 1, has been addressed by, among others, Ahonen-Jonnarth and Odelstad (2006, 2010); Odelstad (2007). The example can be represented by a DALMAS consisting of a single agent; in other words, a DALOAS. An instrumentalization of a simplified version, originally conceptualized by Ahonen-Jonnarth and Odelstad (2011), is presented in Sect. 4.2 of the technical report (Hjelmblom, 2011). This section is reproduced here with slightly changed notation.

The agent set Ω consists of a single forest cleaner agent x, operating in an environment consisting of a grid of forest squares ordered in rows and columns. The grid Y consists of $m \times n$ squares. The values of m and n $(m, n > 1)$ are fixed before a run of the system. The squares have a fixed size $z_1 \times z_2$, where z_1 and z_2 are integer values $(z_1, z_2 \geq 1)$. (c_1, c_2) denotes the coordinates of a position within the grid, where $0 \leq c_1 < mz_1$ and $0 \leq c_2 < nz_2$. The function $c : \mathbb{N} \to \mathbb{Z}^+$ returns the column containing x-coordinate c_1, and $r : \mathbb{N} \to \mathbb{Z}^+$ returns the row containing y-coordinate c_2. $\langle i, j \rangle$ denotes the square in column i and row j.

The simplification consists of the restriction that each square can contain at most one tree. A tree has a position (c_1, c_2), and can be of two *types*, conifer or broadleaf. Furthermore, a tree has a *diameter*, a *condition* (damaged or non-damaged), a *height* and a *quality* (low, medium or high). The current position of the agent and the current content of each square constitute a state of the system (or, more precisely, a state of the system's knowledge base). Let S be the set of possible states. A state descriptor has the form

$$\langle (c_1, c_2), \Upsilon \rangle \tag{4.1}$$

where (c_1, c_2) denotes the agent's position and

$$\begin{aligned}\Upsilon : \{\langle i,j\rangle \mid 1 \leq i \leq m,\ 1 \leq j \leq n\} \rightarrow \\ \{\langle p,k,d,e,h,q\rangle \mid p \in P, k \in K, d \in D, e \in E, h \in H, q \in Q\} \cup \{\Box\}\end{aligned}$$

where

$P = \left\{(c_1, c_2) \in \mathbb{N}^2 \mid c_1 < mz_1 \text{ and } c_2 < nz_2\right\}$ (tree position);
$K = \{\text{conifer, broadleaf}\}$ (type of tree);
$D = \{t_d \in \mathbb{Z}^+ \mid t_d \leq 150\}$ (tree diameter in mm);
$E = \{\text{damaged, undamaged}\}$;
$H = \{t_h \in \mathbb{Z}^+ \mid t_h \leq 100\}$ (tree height in dm);
$Q = \{\text{low, medium, high}\}$ (quality); and
$\Box$ means ‘no tree’.

For example,

$$\Upsilon_s(3,5) = \langle\text{(70,110), conifer, 75,non-damaged, 50, medium}\rangle$$

means that, in state s, square $\langle 3,5\rangle$ contains an undamaged medium-quality conifer tree located at position (70,110) and with diameter 75 mm and height 50 dm, whereas $\Upsilon_s(4,5) = \Box$ means that, in state s, there is no tree in square $\langle 4,5\rangle$.

Acts are functions with system states as arguments. The act set A consists of two acts:

1. $\textit{Remove} : S \rightarrow S$
2. $\textit{Leave} : S \rightarrow S$

$\textit{Remove}\left((c_1,c_2), \Upsilon\right)$ means that the agent, whose position is (c_1, c_2), removes the tree in square $(c(c_1), r(c_2))$, and then moves on to the next square, whereas $\textit{Leave}\left((c_1,c_2), \Upsilon\right)$ means that the agent leaves the tree (if any) located in square $(c(c_1), r(c_2))$, and moves on to the next square. More formally,

$$\textit{Leave}(\langle (c_1,c_2), \Upsilon\rangle) = \langle G(c_1,c_2), \Upsilon\rangle,$$

and

$$\textit{Remove}(\langle (c_1,c_2), \Upsilon\rangle) = \langle G(c_1,c_2), H(\langle (c_1,c_2), \Upsilon\rangle)\rangle,$$

where

$$G(c_1,c_2) = \begin{cases} \langle c_1 + z_1, c_2\rangle \text{ if } c(c_1) < m \text{ and } r(c_2) \text{ is odd} \\ \langle c_1, c_2 + z_2\rangle \text{ if } c(c_1) = m \text{ and } r(c_2) < n \text{ and } r(c_2) \text{ is odd} \\ \langle c_1 - z_1, c_2\rangle \text{ if } c(c_1) > 1 \text{ and } r(c_2) \text{ is even} \\ \langle c_1, c_2 + z_2\rangle \text{ if } c(c_1) = 1 \text{ and } r(c_2) \text{ is even and } r(c_2) < \ n \\ \langle c_1, c_2\rangle \text{ if } c(c_1) = m \text{ and } r(c_2) = n \text{ and } n \text{ is odd} \\ \langle c_1, c_2\rangle \text{ if } c(c_1) = 1 \text{ and } r(c_2) = n \text{ and } n \text{ is even} \end{cases}$$

and

$$H(\langle(c_1, c_2), \Upsilon\rangle) = \begin{cases} \Upsilon'(i,j) = \square \text{ where } i = c(c_1) \text{ and } j = r(c_2) \\ \Upsilon'(i,j) = \Upsilon(i,j) \text{ where } i \neq c(c_1) \text{ and } j \neq r(c_2). \end{cases}$$

In the following, we assume that the forest consists of 15 × 15 squares (i.e., $m = n = 15$) with a square size of 40 × 40 (i.e., $z_1 = z_2 = 40$) and that the initial state for the system is $\langle(10, 10), \Upsilon_0\rangle$ where Υ_0 can represent a randomly created forest.

Preference function and normative system

The behaviour of the forest cleaner agent is regulated by the combination of a normative system, which for every state determines which acts are permissible for the agent, and a preference function that determines which of the acts are the most preferable. As an example, let us consider the following simple rule system from (Ahonen-Jonnarth & Odelstad, 2010) (with x replaced with θ):

> Rule system 2: If two trees are less than θ meters from each other and one of them is damaged, remove the damaged tree. Otherwise remove the tree that has the smaller diameter.

There are several ways to formulate this rule system as a combination of a normative system and a utility function. Since the agent can only remove a tree if it is located in the agent's current square, one way is the following:

1. if the agent's current square contains a damaged tree and there is an undamaged tree within distance θ from this tree, then the agent shall see to it that this square[4] does not contain a tree;

2. if the agent's current square contains a damaged tree and there is another damaged tree with a larger diameter within distance θ, then the agent shall see to it that this square does not contain a tree;

3. if the agent's current square contains an undamaged tree and there is another undamaged tree with a larger diameter within distance θ, then the agent shall see to it that this square does not contain a tree; and

4. if permissible, the agent should always prefer to leave a tree over removing a tree.

The formal representation of these norms is based on the descriptive conditions C_{q,θ,d_1,d_2} and B_{q,θ,d_1,d_2}, where $q \in \{\text{curr}, \text{prev}\}$ and $d_1, d_2 \in \{\text{d}, \text{u}\}$. The intended meaning of, say, $C_{\text{curr},60,\text{d},\text{u}}(x_1; x, s)$ is that, in situation $\langle x, s\rangle$, x_1's current square contains a damaged tree and there is an undamaged tree within distance 60 from this tree. $B_{\text{prev},60,\text{u},\text{u}}(x_1; x, s)$ means that x_1's previous square contains an undamaged tree and there is a bigger undamaged tree (i.e., an undamaged tree with a larger diameter) within distance 60 from this tree.

[4] Note that 'this square' refers to the agent's current square *before* the agent acts. However, since both actions move the agent to the following square, when evaluating the effect of the act after it is performed, 'this square' will in fact refer to the agent's *previous* square.

The last rule may be represented by a utility function such that, in all states s, the utility of *Leave* is greater than the utility of *Remove*. Omitting the universal quantifier, rules 1–3 may be expressed in logical form in the following way:

1. $C_{\mathrm{curr,60,d,u}}(x_1; x, s) \wedge x_1 = x \rightarrow \text{Shall Do}(x_1, \neg C_{\mathrm{prev,60,d,u}}(x_1; x, s))$
2. $B_{\mathrm{curr,60,d,d}}(x_1; x, s) \wedge x_1 = x \rightarrow \text{Shall Do}(x_1, \neg B_{\mathrm{prev,60,d,d}}(x_1; x, s))$
3. $B_{\mathrm{curr,60,u,u}}(x_1; x, s) \wedge x_1 = x \rightarrow \text{Shall Do}(x_1, \neg B_{\mathrm{prev,60,u,u}}(x_1; x, s))$

(Note that, since the system contains a single agent x, it will always be the case that $x_1 = x$.) The algebraic representations of these three norms are

1. $\langle MC_{\mathrm{curr,60,d,u}}, T_7 C_{\mathrm{prev,60,d,u}} \rangle$,
2. $\langle MB_{\mathrm{curr,60,d,d}}, T_7 B_{\mathrm{prev,60,d,d}} \rangle$, and
3. $\langle MB_{\mathrm{curr,60,u,u}}, T_7 B_{\mathrm{prev,60,u,u}} \rangle$,

where M is the 'move operator' described in Sect. 3.2.4.

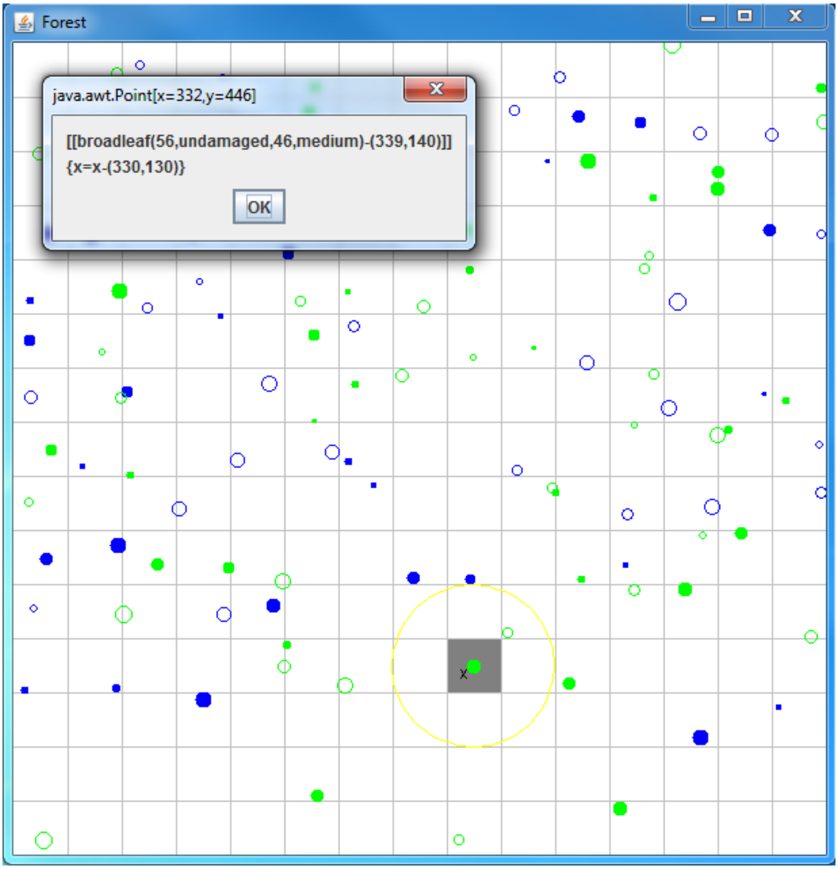
Implementation: CleanerDALOAS

`CleanerDALOAS` is a graphical Java instrumentalization of the Forest Cleaner system, built with the Java/Prolog DALMAS implementation described in Paper I. The graphical user interface consists of a client control panel and a graphical representation of the current state of the forest grid. In the state illustrated by Fig. 4.3, agent x has cleaned the lower part of the forest grid. The yellow circle shows the threshold distance $\theta = 60$ from the tree in the agent's current square, which is shown with a grey background. Filled circles represent undamaged trees, while unfilled circles represent damaged trees. Green colour means broadleaf and blue colour means conifer, and the size of the circle reflects the diameter of the tree.

The current version of the application can simulate cleaning of random forests, and save forest states on text files. It also has a simple tool for statistical analysis of forest stands, such as calculation of mean, median and standard deviation of the diameter or distance between trees. Several improvements are planned but not yet implemented, for example, the ability to load a saved forest from file. An interesting idea for the future is to drop the restriction of only one tree per forest square, to allow for simulations on real forest data, and use the ideas in Paper V to evolve a normative system for forest cleaner agents. The evaluation function described in (Ahonen-Jonnarth & Odelstad, 2010) could then be used as a fitness function in the genetic algorithm.

4.5.5 Pre-runtime evolution of normative systems

Paper V presents an approach to the pre-runtime design of normative systems for problem-solving MAS, in which evolutionary mechanisms play an important



java.awt.Point[x=332,y=446]

[[broadleaf(56,undamaged,46,medium)-(339,140)]]
{x=x-(330,130)}

Figure 4.3: Screenshot: Forest grid of a CleanerDALOAS.

part. The idea behind the suggested methodology is to use a top-down approach of selecting (a subset of) the most ‘efficient’ norms from an evolved normative system, rather than designing a normative system entirely from scratch in a bottom-up fashion. The approach is illustrated by evolving ever better normative systems for a class of problem-solving MAS called Explorer systems, which are represented as instances of the Waste Collectors system, by running a number of Explorer systems as part of the evaluation step in the genetic algorithm.

Chapter 5

Discussion and conclusion

In the introduction, I remarked that the thesis revolves around instrumentalizations on different levels of abstraction. This final chapter is structured as follows. The level of theory and theoretical concepts, is dealt with in Sect. 5.1.1. The levels of abstract and physical architecture, are discussed in Sect. 5.1.2, and Sect. 5.1.3 covers the level of specific multi-agent system applications. I will end with some concluding remarks.

5.1 Discussion and future work

5.1.1 Fundamental theoretical concepts

One of the features of the Lindahl-Odelstad algebraic approach to norms is its generality, in the sense that it makes possible the formulation of norms which do not explicitly refer to named actions. A clear separation between norms and actions allows for the specification of the normative system to be logically analysed and modified independently of the set of available actions. The logical and algebraic foundation of this approach makes possible formal and, at least to some extent, automated reasoning about various properties of normative systems. A key property is *coherence*, in the sense of absence of norm conflicts, which is the subject of norm *verification* (see, e.g., Alechina et al., 2013, p. 73). A basic kind of norm conflict is (logical) inconsistency. Two norms are contradictory in this sense if their normative consequences consist of logically inconsistent deontic propositions, e.g., if one norm prohibits something while the other permits the same thing. Consider, for example, a Waste Collector DALMAS (see Sect. 3.1.1) with a normative system containing the two norms $\langle \mathit{MDiff}, T_5 \mathit{Lap}_0 \rangle$ and $\langle \mathit{MDiff}, T_7 \mathit{Lap}_0 \rangle$, and suppose that MDiff holds of two agents x_1 and x_2 in the situation $\langle x, s \rangle$. Then both

$$\mathrm{T}_5(x, \mathit{Lap}_0(x_1, x_1)), \text{ i.e., Shall Do}(x, \mathit{Lap}_0(x_1, x_1)),$$

and

$$\mathrm{T}_7(x, \mathit{Lap}_0(x_1, x_1)), \text{ i.e., Shall Do}(x, \neg \mathit{Lap}_0(x_1, x_1)),$$

are in effect in this situation. From the logic of Shall and Do follows a contradiction. However, coherence cannot be reduced to logical consistency only. Even if a normative system is logically consistent, norm conflicts may still arise in a particular situation, given the set of actions available in that situation. Consider, as a simple example, a situation $\langle x, s\rangle$ of a Waste collector Dalmas in which there are three feasible acts for the acting agent x: g_E (for 'go east'), g_S ('go south') and p ('pass', i.e., do nothing). Suppose that the normative system contains the norm $\langle \mathit{MDiff}, T_7\mathit{Lap}_9\rangle$. Now, if three other agents x_1, x_2, and x_3 are standing in x's square, the square to the east of x, and the square to the south of x, respectively, then $\mathit{Lap}_9(x, x_1)$ will hold in the state $n(x, s)$, $\mathit{Lap}_9(x, x_2)$ will hold in $g_E(x, s)$, and $\mathit{Lap}_9(x, x_3)$ will hold in $g_S(x, s)$. Thus, no matter which of the three available actions x chooses, the transition type condition $E_7^a\mathit{Lap}_9$ will become true for x and some other agent. Hence, all three actions (including 'do nothing') will be prohibited for x. This situation, in which the Dalmas's deontic structure becomes empty, illustrates that a normative system may be incoherent (in an 'operational' sense) even though it is logically consistent. Thus the coherence of normative systems is determined by the interplay between the normative system and the set of actions. If the normative system is incoherent, it is not necessarily the normative system that is 'problematic'; for example, it could well be argued in the example above that there were too few available actions, or that the available actions were not powerful enough. In other words, according to this view, normative systems for MAS are coherent or incoherent only in relation to the specific multi-agent systems in which they are employed. The potential contribution to the field of normative verification and normative analysis (Criado et al., 2011, Sect. 6.2) is briefly mentioned here as ideas for the future, but is not further investigated in the thesis or the papers. Note that it may be somewhat misleading to talk about logical consistency of Dalmas norms, since strictly speaking they are represented as joinings in a Boolean joining-system (Bjs) correlating an algebraic structure of ground conditions to an algebraic structure (an *np-cis*) of normative consequences. Thus the normative conditions are algebraic entities, not logical, and 'inconsistency' of a *Bjs* refers to $\langle\top, \bot\rangle$ being a joining. However, since the norm-building operators T_i (resp., P_i) are defined in terms of the one-agent types of (extended) normative positions, and the requirements in the definition of the *np-cis* and its derivatives are justified by the Kanger-Lindahl logical postulates for Do, May, and Shall, it does make some sense to talk about (logical) consistency of Dalmas normative systems. An open *cis*-related question is how well the definition of the *np-cis* and its derivatives (see Sect. 3.2.4, 4.4) captures the underlying logic. Are there normative systems represented as consistent *Bjs*'s that still can give rise to logically inconsistent normative consequences? Should, for example, the definitions of the different variants of *np-cis* be modified to incorporate a second relation, corresponding to the 'less free than' relation on the types of normative positions?

It is remarked in Paper III that another feature of the generality of the algebraic approach is that it takes into account the idea that normative systems should express general rules where no individual names occur. The normative system correlates generic grounds with generic normative consequences, and is combined with a mechanism to instantiate generic grounds and consequences. Despite the expressive power of this approach, it still has its limitations. In a complex normative system, norms are general, with cross-references between variables in antecedents and consequents. As noted by Lindahl and Odelstad (2008, p. 230), the *cis* model (which is the basis of their algebraic approach to normative systems) has limitations, compared with predicate logic, if the primary aim is to provide an overall representation of such a complex normative system. For example, due to the way the arguments of conditions are treated, 'compound' conditions such as 'to be a grandmother of' cannot be expressed as a conjunction of 'simple' conditions like 'to be the mother of' and 'to be the father of'; they need to be introduced as separate primitives. An interesting line of future research is to develop a full algebraic treatment of conditions, having the same expressive power as predicate logic. According to Lindahl and Odelstad (2004), this requires mathematical concepts like Tarski's cylindric algebras (Henkin, Monk, & Tarski, 1971, 1985).

Paper III also discusses the issue of computational complexity. Scalability is a challenge for this approach as well as for most other approaches to normative systems (see, e.g., Alechina et al., 2013, Sect. 3.5). A central feature of the Lindahl-Odelstad algebraic treatment of norms, in which a normative system is represented as a Boolean joining-system, is the notion of *minimal* joinings. It is noted in Paper III that this notion might be employed to, at least to some extent, address the scalability issue. This is an interesting area of future research, which includes trying to develop algorithms for finding minimal joinings in a *Bjs* in general and a *Bjs* representing a normative system in particular. This might be a useful tool for formal verification of normative system consistency, and thus a valuable contribution to the area of normative reasoning, one of the open issues identified in Sect. 1.2.

A central assumption in the definition of the norm-regulated transition system situation, and thus in the interpretation of the Kanger-Lindahl one-agent types of normative positions in this setting, is that one agent is regarded as the 'acting' agent, and that there is no simultaneous action by other agents or the environment. A common objection is that this is a strong assumption, which limits the applicability of this device. To some extent, this objection is addressed in Paper V. In simulations, a close to asynchronous system behaviour may be obtained by the combination of short clock cycles and a 'do nothing' action. Nevertheless, it is true that this assumption rarely holds in true multi-agent settings with simultaneous agent actions and other events in the environment. One line of future work is to investigate the conditions for relaxing or circumventing the assumption. For example, suppose that s^+ is the resulting state when an agent x performs an action a in state s, while at the same time a set of other events (possibly including actions by other agents) occur. Suppose further that the transition from s to s^+ is of, say, type III as regards the state of affairs F, i.e., that F holds in s and $\neg F$ holds in s^+. Now, in some environments the contribution[1] of x is easily identified, for example

[1] Craven and Sergot (2008) use the term *strand* for the component that corresponds to x's contribution to an event.

when the effect of x's action is restricted to a part of the environment which is unaffected by the other events that occur simultaneously, but in general it is not a trivial problem to single out x's responsibility for the outcome. In fact, when actions are treated as functions, it may not even be clear in many settings what the resulting state is when two actions are performed simultaneously. One way to deal with the former issue is to not regard it as a problem at all. If, in this case, a transition type prohibition operator prohibits transition type III, then a is prohibited for x no matter what the contributions of the other simultaneous events are. (It could, for example, be argued that, to comply with the norm expressed by this transition type operator, x should have acted so that the effects as regards the truth of F of the other events were repealed.) One problem with this simple approach might be that, when interpreting the extended types of normative positions in terms of transition type prohibition operators, the interpretations of some concepts (e.g., Leave and Oppose) may appear less intuitive. Another obvious path ahead is to try to generalize the notion of transition system situations by exploring the applicability of Lindahl's system of *two-agent* types of normative positions, in environments where the agent acts simultaneously with other agents (possibly including settings in which the environment is treated as a special kind of 'agent'). The latter issue of non-deterministic actions (or combinations of actions) must be dealt with in some other way, perhaps involving counter-factual reasoning. A first step in this direction may involve trying to formally characterize settings in which simultaneous actions have deterministic effects.

Consider the following three sentences (see Paper IV, Sect. 1.3.1 for a background to the example):

(F_1) there is a fence between (the two estates) Whiteacre and Blackacre.

(F_2) x sees to it that a fence is erected between Whiteacre and Blackacre.

(F_3) x sees to it that x erects a fence between Whiteacre and Blackacre.

It is noted in the paper that there are subtle differences between the three kinds of sentences (1) 'x sees to it that F_1', (2) 'x sees to it that F_2', and (3) 'x sees to it that F_3'. (1) follows from (2), and (2) follows from (3), but not necessarily the other way round. For example, the latter two seem to imply either that there was no fence between the two estates before the action of x, or that a second fence was erected. In the theory of normative positions, all of these sentences are represented by statements of the form $\text{Do}(x, F_i)$. As remarked by Jones and Sergot, the Do approach to the logic of action

> ... abstracts away from considerations of state change and the temporal dimension, focusing essentially on the agent concerned and the states of affairs that he or she brings about. (Jones & Sergot, 1993, p. 292)

Let us now prefix (1), (2), and (3) with a deontic operator such as Shall. Assuming that there is already a fence between the estates, the only way for

x to comply with Shall (3) is to erect a second fence. This may well be the intention behind the normative system containing this normative sentence, but in many settings it is the 'end result' (in this case, that there is a fence between the estates) that is in focus. If there is already a fence between the estates, then the obligation Shall Do(x, F_1), i.e., 'x shall see to it that there is a fence between Whiteacre and Blackacre', can be fulfilled by a 'null action' (to follow Lindahl's terminology) in regard to the existence of the fence. In practice, in many MAS as well as legal contexts, norms based on sentences of this kind are sufficient. In the DALMAS setting, for example, if x fulfils the obligation Shall Do(x, F_1), then either x erects a fence, or there already is a fence between the estates and x performs a null action in regard to the fence's existence. Sentences of the kinds (2) and (3) are dealt with somewhat differently than sentences of the kind (1), since the former kinds refer to specific actions being performed or not performed, i.e., involve a relation between two consecutive states, while the latter refers to some state of affairs being or not being the case in a single state. As an example, Shall Do(x, F_3) can be dealt with by including in the DALMAS state descriptor the history of performed actions, and constructing norms whose consequences require the previous action performed by x to be the action 'erect a fence between Whiteacre and Blackacre'.

Paper IV raises the two related questions (A) if there is a 'system norms' interpretation of the extended system of 15 types of normative positions, and (B) if there is a reduction of Sergot's set of normative act positions that can be given an agent-specific norms interpretation. In relation to these issues, it is also natural to wonder if it is not possible to find an 'agent-specific norms' interpretation of the extended 15-type system. There are 15 combinations of basic transition types, but it is argued that some of the latter are not meaningful as the basis for agent-specific norms. Could more expressivity be gained by distinguishing transitions corresponding to the agent's 'activity' from transitions corresponding to the agent's 'inactivity'? At least it can be argued that some of the intuitions behind Do, Leave, and Oppose, seem to be linked with the notions of activity and inactivity. Hilpinen (1997) identifies four basic modes of action (and another four follow due to the symmetry between F and $\neg F$):

1. x brings it about that F ($\neg F$ to F, x active);
2. x lets it become the case that F ($\neg F$ to F, x inactive);
3. x sustains the case that F (F to F, x active);
4. x lets it remain the case that F (F to F, x inactive)

Note that 1 and 2 correspond to basic transition type II, while 3 and 4 correspond to I. To distinguish between, say, 1 and 2 in the case of II, one requires a model of what would happen in the case of 'inactivity' of x. If F becomes the case anyway, without any 'activity' of x, then x 'lets it become the case that' F (basic mode 2), otherwise x 'brings it about that' F (basic mode 1). Again, counterfactual reasoning seems necessary. It is, however, not a simple task to precisely define the meaning of being 'active', resp., 'inactive'.

The Lindahl-Odelstad theory of joining-systems was the result of a continuous development of their algebraic approach to norms. The application of the theory is, however, not restricted to representation of normative systems. One potential application area with a bearing on multi-agent systems is the field of representation and revision of beliefs, in which economy of expression and efficient ways of changing the systems are important desiderata. It seems plausible that the notion of minimal joinings, a central notion in the theory of joining-systems, could be well exploited within this field, as well as ideas similar to those dealing with norm addition and subtraction in (Lindahl & Odelstad, 2013, Sect. 4.3).

5.1.2 Architecture

One of the chief motivations of Lindahl and Odelstad for the development of the theory of joining-systems was their interest in so-called *intermediate* legal concepts[2], conceptually in between purely descriptive and purely normative concepts. Their aim was to "provide tools for a rational reconstruction of a legal system with intermediaries" (Lindahl & Odelstad, 2013, p. 625). This aspect of the theory has yet to be fully exploited in the MAS context. Paper IV suggests a development of `dnrDALMAS` and `nrtssit`, with their underlying architectures, to accommodate the representation of normative systems as networks of subsystems and relations between them, i.e., stratified normative systems with intermediate structures whose elements belong at the same time to both the set of consequences of one *Bjs* and the set of grounds of another *Bjs*. See, for example, the illustration in (Lindahl & Odelstad, 2013, Fig. 5). Such multi-layered systems could, for example, represent normative systems consisting of both behavioural and constitutive norms.

The future development of the Dalmas architecture can go in several other directions. In order to partially address one of the research challenges identified by Boella et al. (2008), the architecture could be developed to allow for agents playing *normative roles* such as legislators (cf. Sect. 3.3) or norm enforcers (cf. Sect 3.2.4). Support for legislator agents, for example, could be added to the Dalmas architecture by following the suggestions in papers II & III to let the normative system be a part of the state of the system, and provide some agents with the capability to choose actions that modify the normative system. Similarly, the Dalmas architecture could be augmented with support for norm-enforcing agents that obtain certain rights, represented as normative positions regarding some states of affairs, if other agents behave in a certain way. A first step in this direction could be to develop the normative framework to allow for ground conditions of norms based on generalized versions of the transition type operators applied to descriptive conditions. Such norms could be used for regulation of the behaviour of an agent with the role of a norm enforcer. An example of such a norm is the following: if an agent x performs action a in state s, such that the transition from s to $a(x, s)$ has type III (see

[2] Concepts often mentioned as examples of intermediate concepts are *ownership* ("being the owner of") and *citizenship* ("being a citizen"). The technical concept representing an intermediate concept in TJS is called an *intervenient*.

Sect. 4.2.2) with respect to some condition c, then the norm enforcer y has the normative position of type T_i with respect to some condition d, for example related to some kind of sanction.

Another possible direction is to pick up the idea in (Odelstad & Boman, 2004, p. 163) of addressing a minor limitation of the current version, *viz.* that norms are based on descriptive conditions on agents. In many applications, it is desirable to be able to deal with not only such 'agent norms', but also norms based on conditions on other entities in the environment. This could, for example, include groups of agents or physical objects such as terrain obstacles. For example, although it is possible in the Forest Cleaner system (see Sect. 4.5.4) to treat trees as agents of a very special kind, this makes the model cluttered and less intuitive. The possibility of formulating norms based on conditions on both an agent and a number of trees would make the model much more intuitive and readable, and potentially also lead to better performance.

The current versions of `dnrDALMAS` and `nrtssit` make no clear separation between (i) the derivation of normative positions from normative *np-cis* conditions, on the one hand, and (ii) prohibitions that follow from such normative positions, on the other. The `prohibited/3` predicate checks for each action if it is prohibited by the normative system, by first checking for each norm if its ground condition is met for some combination of agents. If so, the normative position that follows from the corresponding normative consequence is generated, and it is then checked whether or not a prohibition of the action follows from this normative position. An advantage of this 'one-pass' approach is speed, since it avoids checking more norms than necessary: if a prohibition of an action is found, then no more norms need to be checked for that particular action. A disadvantage, however, is that it complicates formal verification of the normative system. Separating the two 'deontic levels' represented by (i) and (ii) is desirable if norm verification is in focus. An idea for the future is to develop versions of `dnrDALMAS` and `nrtssit` which make this separation, by as a first step deriving the full set of normative positions that are 'in effect', and then as a second step deciding which prohibitions follow from this set.

Odelstad and Boman (2004) point out that their definition of *Prohibited* is based on elementary[3] norms, and that it should be investigated whether this is a substantial limitation. This is still an open issue. It should be noted, however, that the `prohibited/3` predicate in `dnrDALMAS` and `nrtssit` handles non-elementary norms as well, by making use of the remark by Odelstad and Boman (p. 162) that

$$(T_i d_1 \vee T_j d_2)(X_\nu, x) \rightarrow [(E_i^a d_1 \wedge E_j^a d_2)(X_\nu, x; x, s) \rightarrow \mathit{Prohibited}_{x,s}(a)],$$

and

$$(T_i d_1 \wedge T_j d_2)(X_\nu, x) \rightarrow [(E_i^a d_1 \vee E_j^a d_2)(X_\nu, x; x, s) \rightarrow \mathit{Prohibited}_{x,s}(a)].$$

It remains to be verified that this implementation handles all possible non-elementary norms in a correct and consistent manner. A related question is whether it is sufficient in the definition of *Prohibited* to regard only the so-called minimal norms, i.e., norms represented by minimal joinings. This is

[3] A norm is *elementary* if its ground and its consequence are both elementary, i.e., do not consist of Boolean combinations of simpler conditions.

interesting since minimal norms represent the specific normative content of a normative system. For example, is it possible to show that prohibitions following from a non-elementary norm also follow from a collection of norms that are both elementary and minimal? This question is still unanswered.

5.1.3 Applications

A number of application areas for Dalmas, such as education, edutainment, entertainment, and decision support, are mentioned in Paper I. For example, a norm-regulated Dalmas could be part of a game environment which lets the user act as a 'legislator', creating and playing with different 'laws' for the game world. A potential application within decision support could be to use norm-regulated systems as an analytical tool for the creation and/or evaluation of decision-theoretical expert systems, along with the ideas in (Boman, Åberg, et al., 1999). In a given situation, such a system could, for example, facilitate decision making by eliminating those feasible acts that are prohibited by some normative system.

It was suggested in the previous section to develop `dnrDALMAS`, as well as `nrtssit`, to allow for stratified normative systems with intermediate structures, potentially representing both behavioural and constitutive norms. As proposed in Paper IV, the example employed by Lindahl and Odelstad (2013, p. 631) is a natural candidate for testing the augmented versions and investigating various aspects of such architectures. Paper IV also suggests further analyses of experiments with the Estates application, presented in the paper, to shed some more light on the benefits and limitations of the instrumentalized algebraic approach to normative systems.

Paper V sketches an approach, based on evolving candidate normative systems using a genetic algorithm, to the pre-runtime design of normative systems for problem-solving MAS. An idea put forward in the paper is to apply the approach in other domains in which the grounds of the norms and the consequences are based on different sets of descriptive conditions, in order to further validate the suggested approach.

5.2 Conclusion

The idea of using norms as constraints on agent behaviour in a multi-agent system has been proposed for a host of applications, like forest cleaning, automatically negotiating environmental parameters in an intelligent building according to resident preferences, and autonomous robots governing greenhouses on Mars. By merging abstract algebra, deontic logic, and computer science methods, I have instrumentalized and further developed the Dalmas architecture for norm-regulated MAS. Furthermore, central norm-related parts of the Dalmas instrumentalization have been extracted into a lightweight version intended to show that the underlying approach to normative systems is applicable to a broader class of systems than the class of DalmasEs. The instrumentalizations consist of a Java libraries and Prolog modules that can be used together for creating specific runtime norm-regulated MAS. The underlying approach to norms is based on the Lindahl-Odelstad algebraic treatment of normative systems. Norms are represented as so-called joinings from a set of descriptive conditions on agents, constituting the set of potential grounds of the norms, to a set of normative conditions, constituting the set of potential consequences. The latter kind of conditions, which through instantiation give rise to normative sentences regarding specific states of affairs in specific situations, are formed by applying norm-building operators to descriptive conditions. I approached the construction of such norm-creating operators from two angles. I conducted a logical analysis based on the Kanger-Lindahl theory of normative positions, which resulted in two extended sets of types of normative positions. Based on algebraic versions of one of these extended systems, I created a set of operators for norms regulating the behaviour of agents. I also systematically explored the set of types of transitions with regard to some state of affairs. A set of norm-creating operators based on prohibition of such transition types was then constructed. I argued that in a basic class of multi-agent systems modelled as transition systems, the transition type prohibition operators specify a semantics for one of the extended systems of normative positions. Furthermore, I sketched an approach to the pre-runtime design of problem-solving MAS, involving the use of a genetic algorithm for finding suitable normative systems for norm-regulated DALMASes. I thus showed how to employ evolutionary mechanisms for instrumentalizing specific runtime systems into tools for solving problems.

I have thus addressed several open issues within the field of normative MAS, mainly regarding the explicit representation and implementation of normative systems and software tools based on formal representation of norms, making them useful for solving real-world problems. Furthermore, I have suggested and discussed potential future application areas and contributions to the open issues regarding, e.g., norm verification and analysis. The results, theoretical as well as practical, contribute to make possible theoretically sound norm-regulated MAS resting on transparently described and efficiently implemented normative systems.

5.3 And whither then?

It is a dangerous business going out your door. If you don't keep your feet, there's no knowing where you might be swept off to. As the account in Ch. 2 shows, my PhD road has been long and winding. When I stepped into it, I put up no other goals than to finish and defend my thesis. Now, as I draw closer to this goal, I realize that it is just another milestone.

The Road goes ever on and on
Down from the door where it began.
Now far ahead the Road has gone,
And I must follow, if I can,
Pursuing it with eager feet,
Until it joins some larger way
Where many paths and errands meet.
And whither then? I cannot say.

The Road Goes Ever On, a song by Bilbo Baggins
(from *The Fellowship of the Ring* by J.R.R. Tolkien)

Bibliography

Ahonen-Jonnarth, U., & Odelstad, J. (2006). Evaluation of simulations with conflicting goals with application to cleaning of young forest stands. In *Proc. of the 4th International Industrial Simulation Conference (ISC'2006)* (pp. 498–503). EUROSIS.

Ahonen-Jonnarth, U., & Odelstad, J. (2010). Simulation as a tool for the evaluation of forest management treatments. In *Proc. of the 2010 European Simulation and Modelling Conference (ESM'2010)* (pp. 426–433). EUROSIS.

Ahonen-Jonnarth, U., & Odelstad, J. (2011). *Röjningssystem 110518.* Unpublished manuscript.

Alechina, N., Bassiliades, N., Dastani, M., Vos, M. D., Logan, B., Mera, S., ... Schapachnik, F. (2013). Computational models for normative multi-agent systems. In G. Andrighetto, G. Governatori, P. Noriega, & L. W. N. van der Torre (Eds.), *Normative Multi-Agent Systems. Dagstuhl Follow-Ups* (Vol. 4, pp. 71–92). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi: 10.4230/DFU.Vol4.12111.71

Andrighetto, G., Governatori, G., Noriega, P., & van der Torre, L. W. (Eds.). (2013). *Normative Multi-Agent Systems. Dagstuhl Follow-Ups* (Vol. 4). Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH.

Arthur, W. B. (2006). Chapter 32 Out-of-equilibrium economics and agent-based modeling. In L. Tesfatsion & K. Judd (Eds.), *Handbook of Computational Economics* (Vol. 2, pp. 1551–1564). Amsterdam: Elsevier / North-Holland. doi: 10.1016/S1574-0021(05)02032-0

Artosi, A., Rotolo, A., & Vida, S. (2004). On the logical nature of count-as conditionals. In *Proc. of LEA 2004 Workshop.*

Balke, T., da Costa Pereira, C., Dignum, F., Lorini, E., Rotolo, A., Vasconcelos, W., & Villata, S. (2013). Norms in MAS: Definitions and related concepts. In G. Andrighetto, G. Governatori, P. Noriega, & L. W. N. van der Torre (Eds.), *Normative Multi-Agent Systems. Dagstuhl Follow-Ups* (Vol. 4, pp. 1–31). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi: 10.4230/DFU.Vol4.12111.1

Boella, G., & van der Torre, L. (2006). An architecture of a normative system: Counts-as conditionals, obligations and permissions. In *Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)* (pp. 229–231). New York, NY: ACM. doi:

10.1145/1160633.1160671

Boella, G., & van der Torre, L. (2008). Substantive and procedural norms in normative multiagent systems. *Journal of Applied Logic*, *6*(2), 152–171. (Selected papers from the 8th International Workshop on Deontic Logic in Computer Science) doi: 10.1016/j.jal.2007.06.006

Boella, G., van der Torre, L., & Verhagen, H. (2006). Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, *12*(2-3), 71–79. doi: 10.1007/s10588-006-9537-7

Boella, G., van der Torre, L., & Verhagen, H. (2008). Ten challenges for normative multiagent systems. In R. Bordini, M. Dastani, J. Dix, & A. E. Fallah-Seghrouchni (Eds.), *Programming Multi-Agent Systems. Dagstuhl Seminar Proceedings.* Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik. Retrieved from `http://drops.dagstuhl.de/opus/volltexte/2008/1636`

Bohm, D. (2004). *On creativity* (L. Nichol, Ed.). London: Routledge.

Boman, M. (1999). Norms in artificial decision making. *Artificial Intelligence and Law*, *7*(1), 17–35. doi: 10.1023/A:1008311429414

Boman, M., Åberg, H., Åhman, Å., Andreasen, J., Danielson, M., Jansson, C.-G., ... Walter, J. (1999). UBU: Utility-based uncertainty handling in synthetic soccer. In M. Asada & H. Kitano (Eds.), *RoboCup-98: Robot Soccer World Cup II. Lecture Notes in Computer Science* (Vol. 1604, p. 352-357). Berlin Heidelberg: Springer. Retrieved from `http://dx.doi.org/10.1007/3-540-48422-1_29` doi: 10.1007/3-540-48422-1_29

Boman, M., Davidsson, P., & Younes, H. L. (1999). Artificial decision making under uncertainty in intelligent buildings. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99)* (pp. 65–70). San Francisco: Morgan Kaufmann. Retrieved from `http://dl.acm.org/citation.cfm?id=2073796.2073804`

Broersen, J., Dastani, M., Hulstijn, J., & van der Torre, L. W. (2002). Goal generation in the BOID architecture. *Cognitive Science Quarterly*, *2*(3-4), 428–447. Retrieved from `http://homepage.tudelft.nl/w98h5/Articles/csq.pdf`

Chopra, S., & White, L. F. (2011). *A legal theory for autonomous artificial agents.* Ann Arbor, MI: University of Michigan Press.

Cohen, P. R., & Levesque, H. J. (2012). Persistence, intention, and commitment. In M. P. Georgeff & A. L. Lansky (Eds.), *Reasoning about actions & plans* (pp. 297–341). Amsterdam: Elsevier Science. Retrieved from `https://books.google.se/books?id=WjAw8zHbeegC` (Originally published in 1987 by M. Kaufmann Pub. Inc.)

Craven, R., & Sergot, M. (2008). Agent strands in the action language nC+. *Journal of Applied Logic*, *6*(2), 172–191. (Selected papers from the 8th International Workshop on Deontic Logic in Computer Science.) doi: 10.1016/j.jal.2007.06.007

Criado, N., Argente, E., & Botti, V. (2011). Open issues for normative multi-agent systems. *AI Communications*, *24*(3), 233–264. doi: 10.3233/AIC-2011-0502

Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B., & Schreckenghost,

D. (1999). Adjustable autonomy for human-centered autonomous systems. In *Working notes of the 16th International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems* (pp. 16–35). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.7765

Gaertner, D., Clark, K., & Sergot, M. (2007). *Ballroom etiquette:* a case study for norm-governed multi-agent systems. In P. Noriega et al. (Eds.), *Coordination, Organizations, Institutions, and Norms in Agent Systems II. Lecture Notes in Computer Science* (Vol. 4386, pp. 212–226). Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-74459-7_14

Garson, J. (2014). Modal logic. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Summer 2014 ed.). http://plato.stanford.edu/archives/sum2014/entries/logic-modal/.

Gibbs, J. P. (1965). Norms: The problem of definition and classification. *American Journal of Sociology*, *70*(5), 586-594. Retrieved from http://www.jstor.org/stable/2774978

Grossi, D., Dignum, F., & Meyer, J.-J. C. (2005). Contextual taxonomies. In J. Leite & P. Torroni (Eds.), *Computational Logic in Multi-Agent Systems. Lecture Notes in Computer Science* (Vol. 3487, p. 33-51). Berlin Heidelberg: Springer. doi: 10.1007/11533092_3

Grossi, D., & Jones, A. J. I. (2013). Constitutive norms and counts-as conditionals. In D. Gabbay, J. Horthy, X. Parent, R. van der Meyden, & L. van der Torre (Eds.), *Handbook of Deontic Logic and Normative Systems* (Vol. 1, pp. 407–441). London: College Publications.

Henkin, L., Monk, J. D., & Tarski, A. (1971). *Cylindric algebras, Part I. Studies in logic and the foundations of mathematics* (Vol. 64). Amsterdam: North-Holland. Retrieved from https://books.google.se/books?id=zuLuAAAAMAAJ

Henkin, L., Monk, J. D., & Tarski, A. (1985). *Cylindric algebras, Part II. Studies in logic and the foundations of mathematics* (Vol. 115). Amsterdam: North-Holland. Retrieved from https://books.google.se/books?id=-d7uAAAAMAAJ

Hilpinen, R. (1997). On action and agency. In E. Ejerhed & S. Lindström (Eds.), *Logic, Action and Cognition – Essays in Philosophical Logic. Trends in Logic, Studia Logica Library* (Vol. 2, pp. 3–27). Dordrecht, NL: Kluwer Academic. doi: 10.1007/978-94-011-5524-3_1

Hilpinen, R., & McNamara, P. (2013). Deontic logic: A historical survey and introduction. In D. Gabbay, J. Horthy, X. Parent, R. van der Meyden, & L. van der Torre (Eds.), *Handbook of Deontic Logic and Normative Systems* (Vol. 1, pp. 3–136). London: College Publications.

Hjelmblom, M. (2008). *Deontic action-logic multi-agent systems in Prolog* (Tech. Rep. No. 30). Gävle, Sweden: University of Gävle, Division of Computer Science, Gävle, Sweden. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-1475

Hjelmblom, M. (2011). *State transitions and normative positions within normative systems* (Tech. Rep. No. 37). Gävle, Sweden: University of Gävle, Department of Industrial Development, IT and Land Management, Gävle, Sweden. Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-10595`

Hjelmblom, M. (2013). Norm-regulated transition system situations. In J. Filipe & A. Fred (Eds.), *Proc. of the 5th International Conference on Agents and Artificial Intelligence (ICAART 2013)* (pp. 109–117). Portugal: SciTePress. Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-13987` doi: 10.5220/0004260801090117

Hjelmblom, M. (2014a). Instrumentalization of norm-regulated transition system situations. In J. Filipe & A. Fred (Eds.), *Agents and artificial intelligence. communications in computer and information science* (Vol. 449, pp. 80–94). Berlin Heidelberg: Springer. Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-17672` doi: 10.1007/978-3-662-44440-5_5

Hjelmblom, M. (2014b). Normative positions within norm-regulated transition system situations. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (Vol. 3, pp. 238–245). Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-17670` doi: 10.1109/WI-IAT.2014.173

Hjelmblom, M. (2015a). *Normative positions in multi-agent systems.* (Submitted to *Web Intelligence: An International Journal.* IOS Press)

Hjelmblom, M. (2015b). Offline evolution of normative systems. In S. Loiseau, J. Filipe, B. Duval, & J. van den Herik (Eds.), *Proc. of the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015)* (pp. 214–221). Portugal: SciTePress. Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-18956` doi: 10.5220/0005284102130221

Hjelmblom, M. (2015c). *Offline norm evolution.* (In Duval, B., van den Herik, J., Filipe, J., and Loiseau, S. (Eds.), *Revised selected papers from the 7th International Conference on Agents and Artificial Intelligence, ICAART 2015. Lecture Notes of Artificial Intelligence.* Berlin Heidelberg: Springer. In press.)

Hjelmblom, M., & Odelstad, J. (2009). jDALMAS: A Java/Prolog framework for deontic action-logic multi-agent systems. In A. Håkansson, N. Nguyen, R. Hartung, R. Howlett, & L. Jain (Eds.), *Agent and Multi-Agent Systems: Technologies and Applications. Lecture Notes in Computer Science* (Vol. 5559, pp. 110–119). Berlin Heidelberg: Springer. Retrieved from `http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-3963` doi: 10.1007/978-3-642-01665-3_12

Hollander, C. D., & Wu, A. S. (2011). The current state of normative agent-based systems. *Journal of Artificial Societies and Social Simulation*, *14*(2), 6. Retrieved from `http://jasss.soc.surrey.ac.uk/14/2/6.html`

Hume, D. (2000). *A treatise of human nature.* (D. Fate Norton & N. M. J., Eds.). Oxford, UK: Oxford University Press. (First published 1739-40.)

Jones, A. J. I., & Sergot, M. (1993). On the characterization of law and computer systems: The normative systems perspective. In J.-J. C. Meyer & R. J. Wieringa (Eds.), *Deontic logic in computer science: Norm-*

ative system specification (pp. 275–307). Chichester, UK: John Wiley and Sons. Retrieved from `http://portal.acm.org/citation.cfm?id=212501.212516`

Jones, A. J. I., & Sergot, M. (1996). A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, *4*(3), 427–443. doi: 10.1093/jigpal/4.3.427

Jonker, C. M. (2007). The pocket negotiator, synergy between man and machine. *NWO Grant proposal.*

Kanger, S. (1972). Law and logic. *Theoria*, *38*(3), 105–132. doi: 10.1111/j.1755-2567.1972.tb00928.x

Kitano, H., & Tadokoro, S. (2001). Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI magazine*, *22*(1), 39–52. doi: 10.1609/aimag.v22i1.1542

Knuth, D. E. (1997). *The art of computer programming* (3rd ed.). Reading, MA: Addison-Wesley.

Knuth, D. E. (2001). *Things a computer scientist rarely talks about.* Stanford, CA: CSLI Publications.

Lindahl, L. (1977). *Position and change: a study in law and logic. Synthese library* (Vol. 112). Dordrecht, NL: D. Reidel. Retrieved from `http://www.google.com/books?id=_QwWhOK8aYOC`

Lindahl, L., & Odelstad, J. (2004). Normative positions within an algebraic approach to normative systems. *Journal of Applied Logic*, *2*(1), 63–91. doi: 10.1016/j.jal.2004.01.004

Lindahl, L., & Odelstad, J. (2008). Intermediaries and intervenients in normative systems. *Journal of Applied Logic*, *6*(2), 229–250. (Selected papers from the 8th International Workshop on Deontic Logic in Computer Science.) doi: 10.1016/j.jal.2007.06.010

Lindahl, L., & Odelstad, J. (2013). The theory of joining-systems. In D. Gabbay, J. Horthy, X. Parent, R. van der Meyden, & L. van der Torre (Eds.), *Handbook of Deontic Logic and Normative Systems* (Vol. 1, pp. 545–634). London: College Publications.

Lloyd, J. W. (1987). *Foundations of logic programming* (2nd extended ed.). New York: Springer-Verlag New York.

Luck, M., & McBurney, P. (2008). Computing as interaction: Agent and agreement technologies. In *Proc. of The European Workshop on Multi-Agent Systems (EUMAS)* (pp. 1–6). Retrieved from `http://calcium.dcs.kcl.ac.uk/1319/1/dhms08.pdf`

Luger, G. F. (2005). *Artificial intelligence: structures and strategies for complex problem solving* (5th ed.). Harlow, UK: Addison-Wesley.

Lybäck, D., & Boman, M. (2004, August). Agent trade servers in financial exchange systems. *ACM Transactions on Internet Technology*, *4*(3), 329–339. doi: 10.1145/1013202.1013206

McNamara, P. (2014). Deontic logic. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Winter 2014 ed.). `http://plato.stanford.edu/archives/win2014/entries/logic-deontic/`.

Morris, R. T. (1956). A typology of norms. *American Sociological Review*, *21*(5), 610-613. Retrieved from `http://www.jstor.org/stable/`

2089098

Müller, T. (2006). A question of trust: Assessing the fulfillment of commitments in terms of strategies. In L. Goble & J.-J. Meyer (Eds.), *Deontic Logic and Artificial Normative Systems. Lecture Notes in Computer Science* (Vol. 4048, pp. 210–221). Berlin Heidelberg: Springer.

Odelstad, J. (2007). Agents, norms and forest cleaning. In G. Boella, L. van der Torre, & H. Verhagen (Eds.), *Normative multi-agent systems. dagstuhl seminar proceedings.* Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. Retrieved from http://drops.dagstuhl.de/opus/volltexte/2007/917

Odelstad, J. (2008). *Many-sorted implicative conceptual systems* (Doctoral dissertation, Royal Institute of Technology, Stockholm, Sweden). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-9624 (QC 20100901)

Odelstad, J., & Boman, M. (2004). Algebras for agent norm-regulation. *Annals of Mathematics and Artificial Intelligence*, *42*, 141–166. doi: 10.1023/B:AMAI.0000034525.49481.4a

Popper, K. R. (1959). *The logic of scientific discovery.* Hutchinson.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: a modern approach* (3rd [updated] ed.). Boston, MA: Pearson Education.

Sebesta, R. W. (2013). *Concepts of programming languages* (10th international ed.). Boston, MA: Pearson Education.

Segerberg, K., Meyer, J.-J., & Kracht, M. (2013). The logic of action. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Winter 2013 ed.). http://plato.stanford.edu/archives/win2013/entries/logic-action/.

Sergot, M. (2001, October). A computational theory of normative positions. *ACM Transactions on Computational Logic*, *2*(4), 581–622. doi: 10.1145/383779.383786

Sergot, M. (2008). Action and agency in norm-governed multi-agent systems. In A. Artikis, G. O'Hare, K. Stathis, & G. Vouros (Eds.), *Engineering Societies in the Agents World VIII. Lecture Notes in Computer Science* (Vol. 4995, pp. 1–54). Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-87654-0_1

Sergot, M. (2013). Normative positions. In D. Gabbay, J. Horthy, X. Parent, R. van der Meyden, & L. van der Torre (Eds.), *Handbook of Deontic Logic and Normative Systems* (Vol. 1, pp. 353–406). London: College Publications.

Simon, H. A. (1973). Does scientific discovery have a logic? *Philosophy of Science*, *40*(4), 471–480. Retrieved from http://www.jstor.org/stable/186282

Simon, H. A. (1996). *The sciences of the artificial* (Vol. 136). Cambridge, MA: MIT Press.

Singh, M. P. (1999). An ontology for commitments in multiagent systems:. *Artificial Intelligence and Law*, *7*(1), 97–113. doi: 10.1023/A:1008319631231

Steels, L. (1989). Cooperation between distributed agents through selforganization. In Y. Demazeau & J. Müller (Eds.), *Decentralized AI: Proceedings of MAAMAW89* (pp. 175–196). Amsterdam: Elsevier Science. doi: 10.1109/IROS.1990.262534

Sterling, L., & Shapiro, E. Y. (1994). *The art of Prolog: advanced programming techniques* (2nd ed.). Cambridge, MA: MIT.

Takahashi, T., Kitamura, Y., & Miwa, H. (2012). Organizing rescue agents using ad-hoc networks. In J. B. Pèrez et al. (Eds.), *Highlights on practical applications of agents and multi-agent systems. advances in intelligent and soft computing* (Vol. 156, pp. 139–146). Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-28762-6_17

Tedre, M. (2011, 08/01). Computing as a science: A survey of competing viewpoints. *Minds and Machines*, *21*(3), 361–387. doi: 10.1007/s11023-011-9240-4

Therborn, G. (2002). Back to norms! On the scope and dynamics of norms and normative action. *Current Sociology*, *50*(6), 863–880.

Tuomela, R. (1995). *The importance of us: A philosophical study of basic social notions* (Vol. 108) (No. 4). Redwood City, CA: Stanford University Press.

Turner, R. (2014). The philosophy of computer science. In E. N. Zalta (Ed.), (Spring 2014 ed.). Retrieved from `http://plato.stanford.edu/archives/spr2014/entries/computer-science/`

Verhagen, H., & Boman, M. (1999). Norms can replace plans. In *IJCAI'99 Workshop on Adjustable, Autonomous Systems.* Retrieved from `ftp://ftp.dsv.su.se/users/verhagen/IJCAI99WS.rtf`

von Wright, G. H. (1968). *An essay in deontic logic and the general theory of action: With a bibliography of deontic and imperative logic. Acta philosophica Fennica* (Nos. 21–22). Amsterdam: North-Holland.

Wooldridge, M. J. (2009). *An introduction to multiagent systems* (2nd ed.). Chichester, U.K.: John Wiley & Sons.

Web references were checked for availability in August, 2015.

Appendix

Table A.1 gives an overview of the reduced extended types of normative positions and the corresponding *np9-cis* operators, and Table A.2 gives an overview of the transition type operators and the corresponding transition type prohibition operators. When interpreting the reduced extended types of normative positions in terms of transition type prohibition, the ith row in Table A.1 corresponds to the ith row in Table A.2.

Table A.1: Reduced extended types and corresponding operators

Red. ext. type/op.	Reduced extended type (def.)
$\mathbf{P}_1/P_1$	$\text{May}\,\boldsymbol{\Delta}(x,F)\&\text{May}\,\boldsymbol{\Lambda}(x,F)\&\text{May}\,\boldsymbol{\Omega}(x,F)\&\text{May}\,\boldsymbol{\Delta}(x,\neg F)$
$\mathbf{P}_{2\Lambda}/P_{2\Lambda}$	$\text{May}\,\boldsymbol{\Delta}(x,F)\&\text{May}\,\boldsymbol{\Lambda}(x,F)\&\neg\text{May}\,\boldsymbol{\Omega}(x,F)\&\neg\text{May}\,\boldsymbol{\Delta}(x,\neg F)$
$\mathbf{P}_{2\Omega}/P_{2\Omega}$	$\text{May}\,\boldsymbol{\Delta}(x,F)\&\neg\text{May}\,\boldsymbol{\Lambda}(x,F)\&\text{May}\,\boldsymbol{\Omega}(x,F)\&\neg\text{May}\,\boldsymbol{\Delta}(x,\neg F)$
$\mathbf{P}_{4\Lambda}/P_{4\Lambda}$	$\neg\text{May}\,\boldsymbol{\Delta}(x,F)\&\text{May}\,\boldsymbol{\Lambda}(x,F)\&\neg\text{May}\,\boldsymbol{\Omega}(x,F)\&\text{May}\,\boldsymbol{\Delta}(x,\neg F)$
$\mathbf{P}_{4\Omega}/P_{4\Omega}$	$\neg\text{May}\,\boldsymbol{\Delta}(x,F)\&\neg\text{May}\,\boldsymbol{\Lambda}(x,F)\&\text{May}\,\boldsymbol{\Omega}(x,F)\&\text{May}\,\boldsymbol{\Delta}(x,\neg F)$
$\mathbf{P}_5/P_5$	$\text{Shall}\,\boldsymbol{\Delta}(x,F)$
$\mathbf{P}_{6\Lambda}/P_{6\Lambda}$	$\text{Shall}\,\boldsymbol{\Lambda}(x,F)$
$\mathbf{P}_{6\Omega}/P_{6\Omega}$	$\text{Shall}\,\boldsymbol{\Omega}(x,F)$
$\mathbf{P}_7/P_7$	$\text{Shall}\,\boldsymbol{\Delta}(x,\neg F)$

Table A.2: Transition type prohibition and transition type operators

Transition type prohibition op.	Transition type operator		Transition type operator (def.)
P_1	-	-	$\bot$
P_2	$C^a_{\{\text{III}\}}$	$C^a_{2\Lambda}/C^a_2$	$d(X_\nu;s)\wedge\neg d(X_\nu;a(x_{\nu+1},s))$
$P_{2'}$	$C^a_{\{\text{IV}\}}$	$C^a_{2\Omega}/C^a_{2'}$	$\neg d(X_\nu;s)\wedge\neg d(X_\nu;a(x_{\nu+1},s))$
P_4	$C^a_{\{\text{II}\}}$	$C^a_{4\Lambda}/C^a_4$	$\neg d(X_\nu;s)\wedge d(X_\nu;a(x_{\nu+1},s))$
$P_{4'}$	$C^a_{\{\text{I}\}}$	$C^a_{4\Omega}/C^a_{4'}$	$d(X_\nu;s)\wedge d(X_\nu;a(x_{\nu+1},s))$
P_5	$C^a_{\{\text{III,IV}\}}$	C^a_5	$\neg d(X_\nu;a(x_{\nu+1},s))$
P_6	$C^a_{\{\text{II,III}\}}$	$C^a_{6\Lambda}/C^a_6$	$\neg(d(X_\nu;s)\leftrightarrow d(X_\nu;a(x_{\nu+1},s)))$
$P_{6'}$	$C^a_{\{\text{I,IV}\}}$	$C^a_{6\Omega}/C^a_{6'}$	$d(X_\nu;s)\leftrightarrow d(X_\nu;a(x_{\nu+1},s))$
P_7	$C^a_{\{\text{I,II}\}}$	C^a_7	$d(X_\nu;a(x_{\nu+1},s))$

Part II

The Papers

Permissions

Papers I, III, and V are reprinted by kind permission from Springer Science and Business Media. The version of Paper V printed here contains minor corrections of the version currently in press with the publisher. Paper II is reprinted by kind permission from SciTePress. The version of Paper IV printed here contains some corrections of the version currently in press.

Errata

Paper II

Location	**Problem**	**Correction**
Sect. 2.1, p. 112, left col.	a condition c	a condition d
Sect. 2.1, p. 112, right col.	c(...)	d(...)

Paper III

Location	**Problem**	**Correction**
Sect. 1.2, p. 83	Reference [26]	[26]