GROVE

A computationally grounded model for rational intention revision in BDI agents

by

Sam Leask

Thesis submitted to the
University of Nottingham
for the degree of
Doctor of Philosophy

Abstract

A fundamental aspect of Belief-Desire-Intention (BDI) agents is intention revision. Agents revise their intentions in order to maintain consistency between their intentions and beliefs, and consistency between intentions. A rational agent must also account for the optimality of their intentions in the case of revision. To that end I present GROVE, a model of rational intention revision for BDI agents. The semantics of a GROVE agent is defined in terms of constraints and preferences on possible future executions of an agent's plans. I show that GROVE is weakly rational in the sense of Grant et al. [36] and imposes more constraints on executions than the operational semantics for goal lifecycles proposed by Harland et al. [38]. As it may not be computationally feasible to consider all possible future executions, I propose a bounded version of GROVE that samples the set of future executions, and state conditions under which bounded GROVE commits to a rational execution.

Acknowledgements

I would like to offer my deepest gratitute to my supervisors Brian Logan and Natasha Alechina for sticking with me, and having my back through thick and thin. The circumstances of the past few years have been difficult and unexpected, but they both went above and beyond the responsibilities of their supervisory roles in supporting me as their student. I couldn't have asked for better mentors.

It would not have been possible for me to have written this thesis, or anything else for that matter, without the tireless efforts of my parents Andrew and Myra to support me throughout my education and otherwise. They put a roof over my head, stoked my curiosity and ambition to rise above my limitations, and put an internet connection in the wall.

I want to thank my close friend Lisa for her selfless support and warm company, and for feeding Yuki and Mochi while I've been away.

Finally, I want to pay my respects to my brother Will and close friend Jon, who were here cheering me on when I started but didn't make it to the end. We're all gonna make it.

Contents

1 Introduction

Intelligent agents are an abstraction of autonomous, intelligent entities, that interact with their environment in pursuit of high-level objectives. Agents maintain representations of beliefs and desires, corresponding to their (potentially incomplete) knowledge of their environment, and the high-level motivations that guide their behaviour. Agents rely on their beliefs to determine realistic courses of action, while their desires represent world states that agents act to bring about. In general, an agent's desires need not be mutually realisable, i.e., consistent, nor necessarily compatible with beliefs, i.e., realistic. For this reason, agents adopt *goals* corresponding to a consistent set of chosen desires, that represent the desired world state the agent chooses to achieve. Intuitively, a rational agent pursues a course of action that achieves its goals. In the literature, the meaning of "goal" varies between corresponding closely to desires [20], to corresponding to a course of action that an agent has chosen to perform in pursuit of its desires [71]. Moreover, goals are not necessarily a subset of desires, as the desires themselves may be too high-level for an agent to act directly on them, meaning that goals may be viewed as less abstract approximations of desires.

In the decision-theoretic literature [2, 62], rational agents seek to maximise utility corresponding to satisfying their desires, by acting to achieve their goals. Each action of a rational agent in that perspective is viewed as the one that gives the greatest benefit to the agent in terms of cost and value among the alternatives. This approach to rationality yields a definition of rationality that is suitable for an ideal agent, but it ignores the computational limits of practical, realisable agents. Practical agents cannot always determine the highest utility action to execute, and cannot deliberate over alternatives indefinitely to do so. Not only are practical agents' resources limited with respect to time, but also computation.

Bratman [10, 11] addresses the issue of rationality in resource-bounded agents by introducing the notion of *intentions* as a means to constrain deliberation and make rational agency tractable for practical agents. Intentions are (partial) courses of action that an agent commits to, avoiding repeated deliberation over the next action to take. An agent's intentions form what Bratman calls a filter of admissibility, meaning that the courses of action an agent can consider are limited to those that are compatible with its intentions. Rational agents must reconsider their intentions rather than being blindly committed to them, as when circumstances change, reflected by changing beliefs and desires, a change of focus in the form of revising intentions may be rational. Although agents should be able to reconsider their intentions, they represent a kind of commitment that is stronger than that of an agent toward goals, leading to the notion that intentions are resistant to reconsideration. Resistance to reconsideration means that an agent should not abandon intentions without rational cause to do so. Rational cause might be that intentions are deemed impossible, or no longer necessary. Because the cases where an agent can rationally reconsider intentions are limited in this way, reconsideration is not necessary unless such a case occurs, allowing to avoid unnecessary reconsideration. Moreover, committing to intentions constrains the means-end reasoning that an agent must do when determining how to act. Rather than selecting from all alternative (complete) means to satisfy desires, agents progressively plan for their intentions as necessary to preserve their feasibility. This avoids planning up-front for circumstances that may never actually occur, which is especially important when agents have incomplete beliefs and inhabit unpredictable, changing independently evolving environments. Bratman argues convincingly that the notion of intention is irreducible to beliefs and desires, and develops a theory of resource-bounded rational agency centred around the notion of intention: Belief-Desire-Intention.

The Belief-Desire-Intention (BDI) [10] model of agency has enjoyed wide success not only as a methodology for specifying and implementing practical intelligent agents [6, 118], but as a framework for investigating rational agency [20, 75, 80].

Intention revision is the process by which an agent amends its existing set of intentions by adding intentions, dropping intentions, or altering existing intentions. This process corresponds to a realignment of an agent's intentions with its beliefs and desires, and is a rational response to changes to desires, and certain changes in the environment, i.e., updated beliefs. For instance, it would be irrational to continue to pursue a course of action that is deemed impossible, even if the intended result is still desirable. In such a case, either the intention should be abandoned altogether, or revised in some way to make it possible to achieve, such as by replanning. Another rational cause of intention revision is loss of motivation, such as the corresponding desire being abandoned. It is clearly irrational for an agent to persist with satisfying a desire it no longer has. Rational intention revision is the adherence to a notion of rationality when revising intentions, for instance satisfaction of rationality postulates [74, 75], or maximisation of utility or benefit [36, 80]. While intention revision results in potentially many different reasonable sets of intentions, rational intention revision corresponds to choosing the "best" among the reasonable alternatives. An informal but clear definition of rational intention revision is given by Grant et al. (quoted verbatim), in terms of a rational agent state that is reached by revision [36]:

> We can think of rationality as being an "ideal" mental state for an agent: the agent has a consistent model of the environment, and has selected intentions that are mutually consistent and compatible with this model, and that are in addition optimal.

To illustrate instances where intention revision is rational, I introduce a simplified version of the Mars rover scenario that I discuss in more detail in Chapter 3. A rover situated on the Martian surface is tasked with collecting rock samples and returning them to a collection bin for layer analysis. The Martian surface is littered with obstacles, and subject to difficult weather conditions, potentially obscuring the rover's sensors. The rover intends to perform a round-trip of several rock samples that it has identified, before depositing them together in the collection bin. However, while en-route to the first sample, the rover observes an interesting rock that wasn't detected previously. It then has the choice between either continuing as it was, or trying to gather the newly discovered rock as well. This corresponds to adopting a new desire: collect the observed rock. Due to this new desire being adopted, it is possible for the rover to consider this new option in the context of its existing intentions. While it may seem straightforward for it to simply collect this rock as well as the others, the rover needs to determine that it is *capable* of doing so. Is the rock accessible? Will there be enough room in the collection vessel for it to carry this rock and the others it was already going to collect? If not, should it collect this rock but abandon its intention to collect one of the others? In addition, does collecting this rock versus another lead to a *more preferable* execution overall, taking into account any progress already made? Let us assume that collecting this rock poses no such problems for the agent and it adopts an intention that is compatible with its existing intentions. Now suppose that the rover moves toward the rock and discovers that it is actually the tip of an enormous boulder, and it is impossible to collect it after all. This impossibility corresponds to a loss of capability for the agent, meaning its intention to collect the rock is no longer feasible. In such a case, it may attempt an alternative plan, but it is unlikely it has an applicable plan for moving a buried boulder, so the only rational revision that can be made to its intentions is to abandon the intention to collect it altogether. Note that depending on how sophisticated the reasoning of the rover is, it may also abandon the desire to collect the rock, given that it is not just temporarily impossible, but permanently impossible. Now suppose that after collecting the remaining rocks, the rover backs up to the collection bin to deposit them, but its wheels slip on the loose dirt and one of the rocks falls into the bin unintentionally. A rational agent should at that point abandon its intention to deposit that rock in the bin, as it has already been achieved, albeit as an unintended side-effect. It would be irrational to persist with its original intention to deposit the rock, or to seek an alternative means to achieve it, leaving the only rational option of accepting that it achieved its goal despite failing to achieve it the way it intended. Meanwhile, another rock sample has fallen onto the ground. The original intention of depositing that rock is also no longer possible, but still unachieved. However, a rational agent should resist abandoning its intentions without rational cause [11]. Therefore, the rover should revise its intentions to collect the rock from the ground and deposit it, if possible to do so, and at the very least abandon its intention to deposit the rock if not. One last case to consider is that of *preference*. If there are multiple possible ways (i.e., joint means) for the agent to collect rocks, it should identify which is the most preferable of the reasonable choices. Moreover, when choosing between sets of rocks to collect, it should intend to collect the set that maximises preference. In the case where circumstances change and intentions may need to be revised, the revision should also account for preference in order to ensure that the resulting set of intentions is most preferred. Note that this may also involve reasoning about trade-offs, such as when intentions are in conflict. An intention should be rationally abandoned even if the agent has made progress toward it, if the progress toward it is deemed less preferable than achieving an alternative incompatible intention.

When considering more complex agent architectures than the simplified Mars rover, such as those with subgoals [78], complex goal types [23] failure handling [82], detection and avoidance of conflicts between intentions [77, 105, 108], detection and exploitation of synergies between intentions [47, 106, 122] and notions of preference [66] or utility [47], the intention revision process becomes more complicated. This follows from the increased number of causes for rational revision involved, and from the complexity involved in identifying the occurrence of those causes and how to revise intentions accordingly. Moreover, there may be multiple alternative revised intention sets that can be derived, and possibly consequential revisions to intentions and other mental attitudes such as beliefs following intention revision. It should be noted that the interdependencies between different mental attitudes constitute a significant source of complexity in the intention revision process [24, 96]. In addition, optimality must be accounted for, and the most preferable set of intentions selected of the reasonable alternatives, i.e., those that are mutually compatible and compatible with beliefs.

I now discuss the state of the literature with respect to rational intention revision. This discussion serves as an example to illustrate the gap in knowledge. A more in-depth discussion is given in Chapter 2 and Chapter 3. There have been many attempts to define, formalise, and operationalise rational intention revision. This work mainly falls into two general approaches: theory-based and practice-based. The theory-based approaches include the BDI agent model and rationality postulates of Rao and Georgeff [75], the metareasoning model of Russell and Wefald [80, 88], and the database perspective of Shoham [96] and adjacent work [27, 36, 45, 127]. The theory-based approaches are high-level and theoretical, specifying models of rational intention revision in BDI logics. While they specify and define rational intention revision, they are typically not operationalised, tractable, nor amenable to agent programming. On the other hand there are more practice-oriented approaches, adopting assumptions and a conceptual framework more consistent with traditional BDI agent programming, typically giving operational semantics and focusing on practical application. These approaches include work on goal interactions [47, 103, 123], failure handling [81], and goal deliberation [13, 38, 61, 79]. Due to the more programming-oriented conceptual framework and focus on practicality, these approaches typically do not consider rationality beyond partial satisfaction of rationality postulates, and applying BDI theoretical results to the proposed models is not straightforward.

This incongruence between the two major approaches has led to what is sometimes referred to as the "theory-practice gap" in the literature [9, 38, 61, 71, 104, 115]. A body of work has developed aiming to address the theory-practice gap by integrating BDI theory concepts with agent-programming languages [9, 24, 40, 54, 92, 117]. Although these approaches constitute significant progress toward bridging the theory-practice gap, there are some aspects of rational intention revision that they do not adequately address. Primarily, they do not account for the executability of intentions in their mutual context, i.e., their joint achievement, at the level of both intentions to-be and intentions to-do. Moreover, they do not consider subgoaling, planning, failure handling, or decision-theoretic notions of rationality, dealing mainly with rationality postulates. Because of this, the definition of a rational agent state preserved by intention revision (from Grant et al. [36]) is not satisfied in the context of common agent programming concepts and assumptions, leaving a remaining theory-practice gap.

In this thesis I provide a new, computationally grounded, model of rational intention revision for BDI agents. The model, GROVE, shares a conceptual framework with BDI agent-programming approaches, yet makes rational agency central. Executable traces are derived from goal-plan trees, and combined to give interleaved executions of multiple intentions. Interleavings are filtered by executability and ordered by preference in order to determine a most preferred execution of an agent's goals. The idealised, unbounded model gives an account of rational intention revision and rational BDI agency in general, presented as an operational semantics. A bounded version of GROVE is also presented, giving an account of bounded-rational BDI agency that requires no significant changes to the model, only the assumptions made. GROVE addresses the theory-practice gap by operationalising rational intention revision in the context of a model based on standard BDI agent programming abstractions. In addition, an account of bounded-rational BDI agency is given, bridging the gap between intractable BDI theory and practical agent programming.

The remainder of this thesis is structured as follows. In Chapter 2 I review the intention revision literature along the lines of the theory-practice dichotomy. In Chapter 3 I define the rational intention revision problem, elaborate on the limitations of the literature in addressing this problem, sketch the outline of a solution, and introduce a running example. In Chapter 4 I introduce GROVE and its operational semantics as a solution to the rational intention revision problem. In Chapter 5 I propose assumptions for bounding GROVE, formalise the notion of bounded rationality, and describe how GROVE can be bounded to give an account of bounded-rational BDI agency that is realisable in practical agents. I conclude with a summary of the contributions made and discussion in the context of existing work, suggest extensions to GROVE to further bridge the theory-practice gap, and identify avenues for future work.

2 Literature Review

> "AI makes philosophy honest."
> — Daniel Dennett

In this chapter I review the literature and identify the limitations with respect to rational intention revision. I present the literature around the idea that the approaches to intention revision and related work can be classified at a high-level as either theory-based, practice-based, or hybrid approaches. The classification of theory-based and practice-based approaches follows the identification of a "theory-practice" gap in the literature, which is addressed by the hybrid approaches. The theory-based approaches primarily consider rational behaviour of ideal BDI agents, and provide specification that provides a logical background for informing implementation of agents. However most of the theory-based approaches are not operationalised or even tractable for practical agents. On the other hand, the practice-based approaches consider tractable and realistic agents first and foremost. While these approaches are more amenable to practical agent programming, they are limited in their consideration of rationality and adopt a different conceptual framework from that of the theory-based approaches. The differences in terminology and conceptual framework of the theory-based and practice-based approaches constitute a gap in knowledge. Bridging this gap corresponds to realising practical agents that are amenable to agent programming, yet exhibit rational behaviour in line with the principles postulated by BDI theory. The hybrid approaches share a common aim of addressing the theory-practice gap.

The theory-based approaches I review consist of the founding work on BDI theory following Bratman [10], including models of BDI agency and rationality principles [20, 75], a strand of work investigating rational intention reconsideration [80, 88, 119], and the literature surrounding Shoham's database perspective [27, 36, 45, 96, 127].

The practice-based approaches I review include work on goal semantics [104, 115], goal and plan interactions [93, 103, 123], deliberation and goal management [38, 68], and preferences [42, 66, 113].

The main approaches that I consider within the hybrid class are the development of sophisticated agent-programming languages [14, 39, 40, 54, 76, 92], and a strand of work investigating the rationality of agents in traditional BDI agent-programming languages [8, 49, 117].

The structure of this chapter is as follows: I review the theory-based approaches in Section 2.1, the practice-based approaches in Section 2.2, the hybrid approaches in Section 2.3, and in Section 2.4 I elaborate on the BDI theory-practice gap in the context of the reviewed literature, and discuss how the problem of rational intention revision can be approached by addressing the limitations of the existing work.

2.1 Theory-based approaches

In this section, I discuss work that takes a more theoretical approach to the problem of intention revision in BDI agents. I first consider key work in philosophy that forms the foundation of much of the work on intention revision, before discussing attempts to formalise these ideas in logic.

The intentional stance is a folk psychological abstraction of intelligent behaviour put forth by Dennett [26]. By viewing an entity as a rational agent, one can then ascribe beliefs and desires to it. Beliefs representing its knowledge of the world, and desires being states of the world it would like to achieve. Then intentions are the chosen courses of action that the agent pursues in order to realise its desires in light of its beliefs. This perspective facilitates reasoning about, predicting, and ascribing motivation to an agent's actions. The intentional stance is the highest of three levels of abstraction proposed by Dennett, the others being the physical stance, corresponding to abstraction at the level of physical laws, and the design stance, corresponding to a teleological view focusing on purpose and function. Although Dennett is concerned primarily with explaining human behaviour, the intentional stance has found some traction in the field of artificial intelligence, particularly as a vehicle for designing and reasoning about autonomous and proactive artificial agents.

Following the development of theories of intention in the philosophical literature, Bratman [10, 11] outlines a general approach for a rational agent architecture in which intentions are treated as first-class mental attitudes alongside beliefs and desires. Bratman makes a case for the distinct role of intentions in resource-bounded rational agents, and argues that not only are intentions irreducible to beliefs and desires, but that they are instrumental in making deliberation tractable for resource-bounded agents in dynamic environments. The primary tasks of the architecture outlined by Bratman are means-end reasoning, deciding between alternative courses of action, and reasoning about interactions between intentions.

Bratman identifies several roles of intentions in constraining the reasoning performed by a rational agent. Firstly, intentions act as an input to means-end reasoning, meaning that means-end reasoning is focused on generating means by which to achieve intentions rather than toward arbitrary goals or in direct reaction to environmental change. Secondly, intentions act as constraints on further reasoning, such that a rational agent need not consider courses of action that are incompatible with its intentions. Lastly, intentions influence beliefs such that some beliefs are consequential of intentions, and agents believe to some extent that those consequences will follow. The constraining role can be further characterised by the notion of *consistency*. Intentions should be internally consistent, consistent with beliefs, and mutually consistent such that they can be jointly executed. Thus the requirement of consistency limits what intentions can be considered for adoption by the agent, what means can be employed to achieve them, and circumscribes the relevant information for determining when changes should be made to the intentions.

In a dynamic environment where circumstances can change unpredictably, in addition to the assumption that agents are knowledge-bounded and do not have complete knowledge of their environment, an agent's intentions are necessarily based on incomplete information that may render them unsuitable as beliefs are updated or revised. Bratman addresses this by proposing that intentions are stable and revocable. Stability implies that intentions resist reconsideration, and are abandoned only with rational cause to do so. On the other hand, revocability implies that intentions can ultimately be abandoned when they are no longer useful to the agent. These opposing properties give rise to what Bratman calls "tension", where a rational agent must balance stability of intentions with reconsideration. This balance is determined by the circumstances under which an agent can consider adopting an intention that is incompatible with its existing intentions, i.e., the circumstances under which it reconsiders its intentions.

In some cases it may payoff to reconsider intentions, and in other cases it may not, and unnecessary reconsideration is wasted effort. On the other hand, failing to reconsider intentions when it pays off to do so means the agent has less than optimal intentions. Bratman states that a rational agent aims to minimise this wasted effort.

Pollack [70] makes a similar case to Bratman, arguing for the centrality of the notion of intention in constraining deliberation and practical reasoning in resource-bounded agents.

Cohen and Levesque [20] give the first formal treatment of Belief-Desire-Intention agents. They introduce a possible worlds model based on temporal logic, in which mental attitudes such as beliefs and goals are modal operators. Their model captures Bratman's roles of intentions and gives an account of how intentions are adopted with respect to beliefs, goals, and existing intentions. Cohen and Levesque give several properties of goals that are desirable as a basis for defining intentions of rational agents: goals should be persistent, possible and unachieved. The property of persistence means that goals are not dropped without rational cause, possibility means that goals are required to be consistent with beliefs, and unachieved means that a rational agent does pursue goals that have already been achieved.

Intentions are defined in their model as persistent goals (referred to as P-GOALs), which are chosen goals that an agent believes it will act toward, and believes it will no longer pursue after doing so. They define an additional type of intention, a PR-GOAL, which also has an explicit motivation condition such that the goal can be dropped if the agent no longer needs to achieve it. The definition of intentions as P-GOALs or PR-GOALs gives two different kinds of commitment strategy toward intentions. The former is "fanatical" or blind commitment, where an agent pursues an intention until it is achieved or believed impossible, and the latter is referred to as "relativized" commitment where an intention can additionally be abandoned if no longer needed.

Although it constitutes a major first step, the model proposed by Cohen and Levesque has some limitations. Singh [99] argues that the definition of P-GOAL given by Cohen and Levesque is too strong as it implies an agent eventually believes a goal is achieved if the agent is competent. Singh notes that this cannot be guaranteed, because as long as the agent believes the goal is not impossible it can attempt it arbitrarily many times and still not achieve it, and does not necessarily need to act towards the goal in order to believe it will eventually be achieved either. Moreover, Singh argues that the definition of P-GOAL is problematic as it gives a definition of intention that is incompatible with multiple intentions. In addition, P-GOALs and PR-GOALs are based on criteria external to the agent and their success is not contingent on actions of the agent. Therefore a rational agent should not commit to these goals as intentions. Singh suggests that the issues with the model of Cohen and Levesque stem from the merging of the semantics of intentions and policies of intention revision, which should be separated. These problems are further addressed by Creel et al. [21].

Rao and Georgeff [73] investigate Bratman's asymmetry thesis [10] in a branching time logic of intention. They extend Bratman's rational properties of intention-belief consistency and incompleteness to analogous properties of intention-goal consistency, goal-belief consistency, and intention-goal incompleteness. Intention-goal consistency means that intentions are consistent with goals, and goal-belief consistency means that goals are consistent with beliefs, i.e., goals are realistic. Intention-goal incompleteness is argued to be a property of rational agents as intention-goal completeness would imply that an agent must adopt every goal as an intention, which is irrational if intentions are incompatible or the resources of the agent are limited. A rational agent therefore commits to a subset of goals and is not forced to commit to all of its goals, there are cases where an agent should not necessarily commit to a goal. For instance, goals may be logically consistent but have incompatible means. This is analogous to the idea of intention-belief incompleteness where it is rational for an agent can commit to an intention provided it is not impossible, rather than requiring that it is certainly possible. Moreover goal-belief completeness is irrational as an agent should not be forced to adopt inevitable beliefs as goals, which Rao and Georgeff identify as an issue with the realism constraint of Cohen and Levesque [20].

Rao and Georgeff [75] propose an alternative framework resembling branching-time computation tree logic (CTL). Possible scenarios in their model are represented as belief-accessible, goal-accessible, and intention-accessible worlds. They identify several rationality properties for agents. Belief-compatibility means that if an agent has a goal stating that some state is eventually reached, then it must believe that it is possible, i.e., there must be at least one path in all belief-accessible worlds where the goal is realised. They refine this notion to give a definition of strong realism, where an agent can only adopt goals that are consistent with beliefs. Goal-intention compatibility means that a rational agent only commits to intentions corresponding to its goals. Indefinite procrastination is avoided by the requirement that agents are committed to attempting their intended actions, although not necessarily successfully, and the requirement that agents eventually abandon their intentions (eg., due to being achieved or impossible). Additionally agents must be aware of their attempted actions, and whether or not they succeeded, and know their goals and intentions. These rationality properties are defined as a set of axioms for their model. Rao and Georgeff define three commitment strategies for agents: blind commitment, single-minded commitment, and open-minded commitment. A blindly committed agent maintains its intentions until they are actually believed to be achieved and thus never reconsiders its intentions. A single-minded agent can abandon intentions that are no longer achievable, and otherwise maintains them. An open-minded agent can additionally abandon intentions when they no longer correspond to its goals, such as when an intention is no longer useful. Rao and Georgeff note the similarity between single-minded commitment and fanatical commitment, and open-minded commitment and relativized commitment of Cohen and Levesque [20]. In addition they point out that a rational agent should not be forced to intend the side-effects of its intentions even if they are inevitable (the side-effect problem), and similarly should not be forced to adopt inevitable beliefs as goals (non-transference). Due to the axiomatisation and definition of intention they introduce, their model avoids these problems. Their model differs from that of Cohen and Levesque in that it treats intentions as basic attitudes of the agent rather than axiomatic commitment to goals, and the definition of future commitment of an agent is focused on intention revision, rather than the definition of intention itself.

van der Hoek et al. [46] formalise capabilities of a rational agent as an operator in the context of a logic of knowledge and action. They define a capability operator in terms of "can" and "cannot" predicates, which are conditioned on actions and goals. The "can" predicate expresses that the agent knows that performing a given action brings about a particular goal, and that it is able to perform the action, i.e., the action is consistent with the currently believed state of the environment. On the other hand, the "cannot" predicate expresses that the action definitely cannot be performed to bring about the goal, either due to inconsistency with beliefs or because the action brings about a state other than the desired goal state. van der Hoek et al. point out that the "cannot" predicate is not simply the negation of "can", because the agent's knowledge is potentially incomplete. Therefore these predicates allow to classify the agent's actions into three classes with respect to a particular goal. For a goal with respect to a particular action, either the agent "can" achieve it, "cannot" achieve it, or it is unknown due to incomplete beliefs.

Padgham and Lambrix [64, 65] refine the notion of capabilities of rational agents further, by requiring only that a goal is possibly achievable rather than necessarily achievable as in the definition given by van der Hoek et al. [46]. Moreover, they distinguish between strong and weak capability based on the two main interpretations of capability that they identify in the literature, which they argue correspond to different degrees of commitment. Their notion of weak capability corresponds to a rational intention to achieve a goal, while their notion of strong capability corresponds to a rational commitment to a means to achieve a goal. The difference between the two is that weak capability requires that a goal or intention is realistic and desired, while strong capability additionally requires that the agent has an applicable means to achieve the goal or intention. This reflects the intuition that capability is the combination of ability and opportunity [65], which implies that it is irrational for an agent to commit to an intention that it does not believe it has the opportunity to achieve at present or in the future, nor any means to achieve it. In addition, Padgham and Lambrix note that if an agent has no means to achieve a goal, then opportunity (applicability of means) is irrelevant to deliberation.

2.1.1 Intention Reconsideration

Russell and Wefald [80] address limitations of the standard decision-theoretic account of rational behaviour with respect to bounded rationality. They develop an optimal model of bounded-rational agency based on the assumptions that agents are resource-bounded and situated in real-time environments where deliberation has a cost in terms of potential environmental change during deliberation. As deliberation leads to revised intentions, and intentions in a decision-theoretic model lead to a desired environmental state with an associated utility, the utility of deliberation in their model is derived from the utility of the revised set of intentions it results in. Because deliberation has both utility and cost, this leads to a notion of payoff for revising intentions when the utility is greater than the cost. The authors point out that while an ideal (unbounded) agent can deliberate until it has obtained an optimal set of intentions, a bounded agent must avoid deliberation that does not lead to a payoff, and engage in deliberation when it does. Rather than giving an account of how an agent selects the maximal utility action, their model is an account of an agent maximising expected net utility, i.e., maximal payoff. Russell and Wefald base their model on a notion of rational metareasoning, where an agent selects either an (object-level) external action to perform using its current intentions, or else executes a meta-level deliberation function that may revise its intentions and thus potentially lead to executing an external action with greater utility. The external action that an agent in their model performs based on its intentions is the “known best” so far, rather than deriving the best action of all possible actions it may take. In order to make the model tractable they introduce a probability distribution as a means for predicting the utility of deliberation, without reasoning about the exact outcomes of deliberation, which may incur a computational cost similar to simply deliberating.

The model proposed by Russell and Wefald [80] is investigated further and expanded on by Kinny and Georgeff [55], Wooldridge and Parsons [119], and Schut et al. [85, 86, 87, 88].

Kinny and Georgeff [55] conduct an empirical investigation of the impact of commitment on agent efficacy. They carried out a series of experiments on agents with various commitment strategies, and additionally vary environmental properties including dynamism and time budget for planning. The commitment strategies were varied in terms of degree of commitment and deliberation strategy. They found that when circumstances change agents are more effective if they reconsider in response to change, and that for greatest efficacy commitment to goals should be tailored to environmental dynamism, i.e., rate of environmental change. In addition they found that agents that abandon impossible intentions immediately outperform blindly committed agents that do not reconsider their intentions in most cases, while the most effective deliberation strategy was reconsideration in response to certain environmental changes.

Following these results, Schut and Wooldridge [85] incorporate the metareasoning model of deliberation from Russell and Wefald [80] into a BDI architecture. Using this architecture they carry out an empirical analysis to evaluate static and dynamic intention reconsideration policies in environments with varied dynamism. The dynamic intention reconsideration policy computes the expected value of deliberation by determining the expected utility of the next action after revising intentions, minus the time cost. They found that the degree of commitment was inversely proportional to environmental dynamism in agents with a dynamic reconsideration policy, meaning that the agent was more likely to reconsider goals in dynamic environments and avoid doing so in less dynamic environments. They suggest that this is because the computation of expected utility of deliberation in their model accounts for predictions about the evolution of the environment (using a probability distribution), therefore estimated utility of deliberation depends on environmental dynamism.

Schut and Wooldridge [86] conduct an empirical evaluation of intention reconsideration strategies in different environments, building on the work of Kinny and Georgeff. The environmental parameters varied were dynamism, accessibility (agent knowledge), and determinism. They found that dynamism had the strongest influence on agent effectiveness overall. The value of reconsideration increases with environmental dynamism, as with greater world change agents need to reconsider more often to keep their intentions consistent with changing circumstances.

Schut et al. [87, 88] describe an approach to intention reconsideration based on the architecture developed previously in [85, 86]. Their approach relies on solving a partially-observable Markov decision process (POMDP) to derive an optimal reconsideration policy for an agent. The POMDP captures the domain knowledge of the agent, including the expected rewards of achieving states through deliberation or action, and evolution of the environment. The authors argue that although the construction and solution of the POMDP is potentially intractable, it can be done offline and cheaply executed at run-time, making it a suitable approach for determining a reconsideration policy, i.e., it is computationally cheaper to estimate expected payoff of deliberation than it is to deliberate. Schut et al. compare the POMDP-based approach to their previous architecture using metareasoning [85, 86] and note that an agent with a POMDP-derived policy has a more consistent level of commitment to intentions when compared to a metareasoning-based agent, potentially due to their lesser dependency on predictions about environmental dynamism. However, they also note that metareasoning-based agents have lower overall cost of action. They found that the POMDP-based approach gave agents that executed more actions with greater efficacy, but had greater cost overall.

Wooldridge and Parsons [119] develop a formal model of rational BDI agency based on the metareasoning model of Russell and Wefald [80]. Their model allows formal definition of the optimality of an agent with respect to its environment, and optimality of a meta-level control function that corresponds to the metareasoning component of the model. They define an agent as optimal with respect to an environment if there is no other agent which achieves a computational run with greater utility, in line with the standard decision-theoretic notion of agent optimality. Similarly, they define a meta-level control function as optimal with respect to an agent and environment if no other meta-level control function gives a computational run with greater utility. An optimal meta-level control function then corresponds to a policy that makes an optimal series of choices between deliberation and action in a computational run. Wooldridge and Parsons suggest that a meta-level control function could determine the best reconsideration policy by reasoning about environmental conditions, potentially choosing from a set of deliberation strategies such as those identified by Kinny and Georgeff [55].

2.1.2 Database perspective

The database perspective [96] refers to a strand of work [16, 27, 36, 45, 50, 57, 58, 96, 124, 125, 127] that aims to give a formal account of intention revision in BDI logics that is analogous to work on belief revision. The work as of 2009 and the motivation behind it are surveyed in an article by Castro-Manzano [16], to which I direct the interested reader. Work in this sphere views the mental state of a BDI agent as a database to be updated and revised in order to preserve consistency within and between mental attitudes. A central concern is the impact of revising mental attitudes on the consistency of other mental attitudes, due to their inherent interdependencies. In order to preserve consistency of an agent's mental state inconsistencies between mental attitudes must be resolved, and their resolution by revision can prompt further revision. The main difficulty addressed by the database perspective work is the definition of postulates for rational revision of mental attitudes, and specification of models that satisfy those postulates.

Hoek et al. [45] introduce a model of rational intention revision based on temporal logic of beliefs and intentions. They distinguish between strong beliefs, which are independent of intentions, and weak beliefs, which are contingent on intentions. Weak beliefs define a notion of realism where an agent's intentions must be consistent with strong beliefs such that the weak beliefs contingent on intentions are also consistent with strong beliefs. This means that both the preconditions and postconditions of intended actions must be consistent with how the agent believes the world currently is or will evolve, based on both strong beliefs and weak beliefs, i.e., intended actions must be realistic in the context of the environment and other intended actions. The distinction of strong and weak beliefs is motivated by the intuition that a rational agent's intentions must be revised to be consistent with strong beliefs (the believed state of the world), rather than the other way around, and consequently weak beliefs may be revised due to intention revision. Similarly, intentions may be revised in order to restore consistency to weak beliefs, but this cannot affect the strong beliefs. Hoek et al. argue that this distinction along intention-dependency of beliefs is necessary in order to model rational intention revision.

Shoham [96] develops the database perspective of intention revision through investigation of the notion of intention within a logical framework of belief and action. He aims to make progress toward bridging the "theory-practice gap" by adopting a practical view of intention rather than a philosophical one, leading to a precise interpretation of the notion of intention revision. Shoham argues that a notion of capability is necessary for relating beliefs and actions such that agents can reason about intentions. Because intentions and beliefs are required to be internally and mutually consistent, intended actions must also be consistent in the sense that they are jointly executable in some temporal ordering, i.e., their preconditions are satisfied and postconditions consistent with beliefs at each time point that an intended action is executed. When beliefs are revised, this can result in inconsistencies which must then be resolved by revision to intentions.

Ditmarsch et al. [27, 58] explore the interaction of belief and intention revision, with a focus on intention dynamics, in a dynamic logic. Their dynamic logic allows to represent how an agent's attitudes change over time. An agent in their logic committing to or abandoning an intention corresponds to a choice operator, the application of which is justified by practical reasoning. An intention is generated when practical reasoning rules determine that a desire can be satisfied by some means, which the agent can choose to adopt as an intention provided it has not already done so. Similarly, the choice to reconsider an intention is made on the basis of practical reasoning rules determining that an intention is no longer useful or is hindering progress in some way e.g., if it is no longer possible, or conflicts with other intentions.

Grant et al. [36] propose a model of BDI agency to investigate revision of mental attitudes. Their approach is similar to the approach outlined by Shoham [96] in that it makes central the connection between consistency of mental attitudes and the influence of their interdependencies on intention revision. It differs from other approaches that explicitly adopt the database perspective in that intentions are treated atemporally, giving a weaker definition of consistency between intentions and beliefs. However, Grant et al. consider optimality of intentions in addition to consistency, which other approaches do not. Their definition of optimal intentions is based on cost-value analysis and resembles decision-theoretic approaches to defining agent rationality. Grant et al. give rationality axioms for mental states of BDI agents. A weakly rational agent has consistent beliefs, desires, and intentions, while a strongly rational agent is weakly rational and has optimal intentions also. An agent is weakly rational if its mental state is belief-rational and intention-rational. Belief-rationality corresponds to consistency between intentions and beliefs, and implies that an agent is capable of its intentions. Intention-rationality corresponds to mutual consistency of intentions and also requires that already satisfied desires are not pursued by intentions. Grant et al. identify the correspondance between their notions of belief-rationality and intention-rationality and the rationality principles of Cohen and Levesque [20] and Rao and Georgeff [75]. Provided that an agent's mental state is weakly rational, it is also strongly rational if its intentions are at least as beneficial as any alternative intention set that it could adopt while remaining weakly rational. Benefit is derived by subtracting the cost of intended actions from the value of the desires they satisfy. While weak rationality defines rationality with respect to consistency of a mental state, as in Shoham's [96] definition of consistency as central to the database perspective, strong rationality is a further constraint of optimality as a choice between weakly rational mental states. Grant et al. specify postulates for rational revision of a mental state. Abstractly, their postulates specify that revisions to strongly rational mental state should result in a strongly rational mental state, preserving consistency and optimality.

Icard et al. [50] propose a formal model of belief and intention revision that adopts Shoham's database perspective. A database, corresponding to a BDI agent's mental state, maintains consistency and coherence between beliefs and intentions, and formalises Shoham's notion of consistency for mental attitudes. A database is coherent if the joint preconditions of intended actions are consistent with beliefs, meaning that it is possible to execute the intended actions together in a temporal ordering. If intentions and beliefs are coherent then all intended actions can be performed, which Icard et al. identify as a minimal requirement of rational balance between beliefs and intentions. Their model is based on a logic incorporating paths, which are possible evolutions of the database. For the property of coherence, preconditions are required to hold on at least one belief-consistent path, giving an optimistic interpretation of coherence as circumstances may not necessarily evolve along any particular path. However this definition of coherence does imply that the intentions are at least no impossible according to the agent's beliefs. Icard et al. give postulates for intention revision that require that the result of a revision is a consistent and coherent database, even if this requires abandoning existing intentions to restore consistency and coherency. Although they note that only beliefs contingent on intentions should be revised following intention revision, following the ideas of van der Hoek et al. [45], they do not make a formal distinction between strong and weak beliefs in their model.

van Zee et al. [125, 126, 127] consider rational interaction between intentions and beliefs, parameterised by time points as in [50]. They identify and address problems with the definition of coherence and incomplete axiomatisation given by Icard et al. [50]. Icard et al.

define coherence as the joint executability of intended actions, and account for temporality of intentions and beliefs in propositions but not modalities, leading to an overly strong definition. The model proposed by van Zee et al. incorporates a stronger notion of coherence, based on a logic where modalities are parameterised by time points. This gives a semantics where if the preconditions of an intended action hold at a given time point, then it is possible to perform that action at that time point. Moreover, they distinguish between strong and weak beliefs in their model, allowing to formally postulate that revision of intentions and weak-beliefs cannot cause revision of strong beliefs. An agent in their model weakly believes that the postconditions of its intended actions hold at the time points after they are to be executed, and intentions and beliefs are coherent if the agent believes that it is possible to execute all of its intentions, i.e., preconditions of intended actions are not inconsistent with strong beliefs. Their definition of coherence then allows for the possibility of intended actions to make the preconditions of other intended actions true, i.e., intentions are not necessarily incoherent if the agent only weakly believes the preconditions of intended actions are satisfied.

2.2 Practice-based approaches

In this section, I review work that adopts a practical approach to the problem of intention revision. The conceptual framework more or less common to the literature I review here differs from the logical framework of BDI theory, where desires and intentions are viewed as first-class primitives. Instead, the primitive concepts here are goals and plans, which roughly correspond to desires and intentions. The exact correspondence between these concepts depends largely on the work in question. The degree of commitment to goals is either implicit in an agent programming language's semantics or explicit in the form of programmed commitment strategies, or representations of agent attitude such as goal state flags that denote role of goal in deliberation. Here I limit consideration of goals and plans to their role in intention revision from an agent programming perspective, a wider discussion of agent programming in general can be found in [118] and [6].

Due to this incongruence between the perspectives, I begin by discussing goal semantics, types, and properties, in order to establish the agent programming terminology and concepts that are common to the practice-based approaches. Next, I introduce the ideas of subgoals and partial planning as they are integral to several of the approaches I review here. Following that I consider work related to detecting and handling situations where intention revision is warranted, and deliberation over goals and goal lifecycle-based models. Lastly I discuss preferences, which are related to rational choice between alternatives and thus rational intention revision.

2.2.1 Goal semantics, types, and properties

According to Winikoff et al. [115], goals are dual-facted as they have both a declarative and procedural aspect. The declarative aspect corresponds to a desired state of the world to achieve, while the procedural aspect corresponds to action that must be taken on the part of the agent to achieve the goal, such as means. Winikoff et al. argue that the declarative aspect is necessary for agents to reason about goals, thus rational behaviour is predicated on declarative representation of goals. They further argue that the procedural aspect is necessary for agents to reason about the achievability of goals, as in agent programming the assumption is made that the agent must carry out some means in order to achieve the goal. Moreover, they argue that the procedural aspect of goals gives agents a problem to solve, echoing Bratman [11]. Both the declarative and procedural aspects of goals are argued as being necessary for rational behaviour in goal-directed agents.

The need for distinction between declarative and procedural aspects of goals, and the necessity of representation of both for rational behaviour, is also argued for by Dastani et al. [22].

Thangarajah et al. [104] argue that without explicit representation of declarative goals, agents cannot reason about the consistency of their goals. They note that while the BDI theoretical notion of rationality requires that adopted goals are a consistent set of chosen desires, some agent programming languages cannot be consistent with this definition of rationality as they have no explicit representation of goals, or only procedural goals are represented.

van Riemsdijk et al. [79] aim to give a general and unifying definition of what is meant by a goal, and explores how they are pursued. In addition they give definitions and semantics for common goal types in agent programming. They define a goal pragmatically as a preferred progression of an agent.

Governatori et al. [35] define a general notion of an outcome as a fundamental concept that agents reason about, of which goals are a sub-type. While goals are preferred outcomes, desires are acceptable outcomes.

Properties

Winikoff et al. [115] identify several properties of goals of rational agents and relate them to the properties of rational commitment from Cohen and Levesque [20], and Rao and Georgeff [75]. They argue that goals of a rational agent are persistent, unachieved, possible, consistent and known. Persistence means that goals are retained unless there is good cause to abandon them, unachieved means an agent should only have goals it believes are not achieved, possible means the goal is consistent with beliefs in some sense, consistent means that goals are mutually consistent, and known means that the agent knows what goals it has. With respect to persistence, Winikoff et al. give the example of decoupling of goal success/failure and plan success/failure. If a plan fails, the goal it pursues should not necessarily fail as well, and similarly if a goal is achieved this may not necessarily occur as a result of plan completion. A plan can complete execution without its goal being achieved, and a goal can be achieved serendipitously by environmental change in which case its plan should no longer be pursued.

Braubach and Pokahr [12] extend the properties for goals proposed by Winikoff et al. to account for those of long-term goals. They suggest that goals of a rational agent should also be producible, suspendable, and variable duration. Producible means that a goal can be invoked or revoked by the agent, i.e., added or removed at runtime, suspendable means that goals can be suspended and resumed, and variable duration means making a distinction between goals that are short-term and immediately require means-end reasoning, and long-term or strategic goals that do not directly control action but still influence the behaviour of the agent.

Types

Dastani et al. [22] discuss three types of declarative goal that can be incorporated into agent programming languages. They identify three goal types in the literature, procedural (perform) goals, achievement goals, and maintenance goals, which they argue are sub-types of declarative goals. They argue for and justify the integration of these goals in logic-based agent-programming languages.

van Riemsdijk et al. [79] propose a unifying semantics for several goal types that they identify as common in agent programming. The goal types are achievement, performance, query, and reactive maintenance. Achievement goals are representing a desired state to be achieved, performance goals are procedural goals corresponding to action to be taken, query goals represent knowledge the agent aims to acquire, and reactive maintenance goals represent a state that the agent aims to preserve and act to restore if necessary.

Dastani et al. [23] give an operational semantics for temporal goals, building conceptually on the work of van Riemsdijk [79]. Temporal goals refer to multiple states, representing conditions that an agent aims to achieve, maintain, or avoid over periods of time. In their semantics temporal goals are LTL formulas that are modelled as achievement goals and maintenance goals.

Duff et al. [28, 29] consider maintenance goals and distinguish types of maintenance goals corresponding to different attitudes of an agent toward maintenance of a state. Reactive maintenance goals invoke achievement goals to restore maintenance conditions that have been violated, while proactive maintenance goals invoke action to prevent predicted violation of their maintenance condition. Thus both of these types of maintenance goal directly control agent behaviour. They additionally discuss passive maintenance goals, which correspond to constraints on agent behaviour and do not directly control action.

2.2.2 Subgoaling and partial plans

Bratman [10, 11] and Pollack [70] argue for the suitability of partial planning to BDI agents in dynamic environments, as they facilitate plan selection at runtime, aid in guiding deliberation, and are filled in as necessary, giving some resilience to changing circumstances. An agent can commit to a partial plan to pursue an intention, without committing to exactly how it is completed until necessary. This style of planning can be contrasted with reactive planning (as in the Procedural Reasoning System (PRS) [33, 34]), where beliefs are mapped directly to actions, i.e., actions are selected rather than (partial) plans.

Subgoals are abstract steps toward a goal, sometimes referred to as subsidiary goals [61]. Goals can be decomposed into subgoals, allowing a complex task to be broken down into smaller parts. In addition, partial plans can be specified using subgoals to represent conditions that must be satisfied for completion of plan, and can be satisfied by assigning and executing plans for subgoals. Subgoals are suited to partial planning as they only need to be planned for and achieved if they are deemed necessary by the agent for pursuit of the parent goal [78]. Moreover, agents can reason about subgoals to detect opportunities for synergistic execution, such as only needing to achieve a common subgoal of intentions once [106].

2.2.3 Causes of intention revision

Goal interactions

Thangarajah et al. [103, 105, 106, 108] develop a strategy for detection and handling of interactions between goals and plans, using summary information [17, 18, 19]. Their approach uses summary information to determine the necessary and possible resources used, requirements and effects of, and common subgoals of paths through goal-plan trees corresponding to a possible way for an agent to achieve a goal. Avoidance of conflicts is a quality of rational behaviour, as it allows to avoid unnecessary failure and thus abandonment of intentions. Moreover, the BDI theoretical literature [75] emphasises that an agent's intentions should be consistent so that it is not working at cross-purposes. A goal-plan tree is a hierarchical data structure that denotes the relationships between goals and plans, and represents all ways for an agent to achieve the root goal. A path through a goal-plan tree corresponds to the execution of a series of plans and potentially the achievement of subgoals, which ultimately achieves the root goal. Paths can also be viewed as complete plans corresponding to the "filling in" of partial plans to achieve the root goal. The approach of Thangarajah et al. allows agents to compile summary information about goal-plan trees statically or offline, and update the summary information dynamically at run-time. Summary information is used to schedule the execution of intentions in order to avoid conflicts due to resource usage or effects, and to exploit synergies that arise due to common subgoals between intentions. Their distinction of the notions of possible/potential and necessary/definite conflicts between goals allows to avoid revising intentions unnecessarily, which would correspond to undercommitment, in cases where goals are possibly/potentially in conflict. A possible conflict between goals corresponds to conflict between at least one path for each goal representing their future evolution. A necessary conflict occurs when all paths corresponding to future evolutions of some goals are in conflict. In the case of conflicts between preconditions and effects of plans, it may be possible to schedule intentions to avoid conflicts. Thangarajah et al. introduce the notion of a preparatory effect characterised by an effect of a step in pursuit of a goal upon which the precondition of a later step depends. If a preparatory effect is threatened by the execution of other intentions, then there is a conflict. They also identify the case where intentions are mutually inconsistent meaning that scheduling cannot avoid a conflict, i.e., all schedulings of some intentions results in conflict. Similarly, common subgoals between intentions can be exploited by scheduling intentions such that the common subgoal only needs to be achieved once.

An alternative approach to using summary information for reasoning about goal interactions is proposed by Shaw et al. [94, 95]. Shaw et al. use Petri nets to represent dependencies between goals and plans, allowing for detection of conflicts and synergies. In their approach, causal links (dependencies) between goals and plans are modelled using Petri nets, which account for the necessary information to reason about preconditions and effects of goals with respect to goal interactions. In addition to detection of conflicts, common effects of plans can be detected. This type of synergy can be exploited by plan merging, such as suggested by Horty and Pollack [47].

Winikoff et al. [114] define temporal goal types and consider conflicts between temporal goals. They identify both logical conflicts (satisfiability) and conflicts over protected conditions of temporal goals. Their approach assumes requirements modeling protected conditions (or in-conditions) of a goal are available. By reasoning about mutual satisfiability and mutual consistency of requirement sets, goals can be scheduled to avoid conflicts.

Zatelli et al. [123] propose a method for run-time conflict detection and resolution between intentions in Jason agents. Their aim is to maximise the internal concurrency of agents by executing as many non-conflicting plans concurrently as possible, which implies identifying and scheduling (non-)conflicting plans and sub-plans. Plans are annotated with conflict sets by an agent programmer. If any elements of a conflict set are referenced by another plan, then those plans potentially conflict. Conflict sets can contain a rich set of identifiers, including events, goals, and resource identifiers. Rather than scheduling plans as atomic units as in the approach of Thangarajah et al. [103], Zatelli et al. suggest scheduling at the sub-plan level in order to maximise concurrency. They observe that some parts of plans may conflict while others do not, and therefore can be conceptually split into safe and conflicting parts with respect to other plans. The conflict sets of intended plans are checked in order to detect conflicts, and potentially conflicting intentions are scheduled in order to avoid conflict. Zatelli et al. propose several strategies for deciding which of a set of conflicting intentions takes priority when scheduling.

Horty and Pollack [47] develop a framework for exploring rational merging of plans in the context of plan costs and values for desires. Their framework addresses limitations of the standard decision-theoretic view of cost-value analysis of intention revision by considering the adoption of incompatible intentions in the context of an agent's existing intentions, taking into account the cost-value tradeoff of abandoning existing intentions. Horty and Pollack argue that it is sensible to require that goals and their means are evaluated together rather than in isolation, as the value of adopting an intention is determined by the value of the desire it satisfies and the cost of the means to do so. Existing intentions in their framework act as a filter of admissibility [10], not only in the sense that they must be compatible, but also that it must be worthwhile to adopt the intention in terms of the cost-value tradeoff of the resulting intention set versus the existing intention set. This goes beyond the interpretation of admissability as being identified with mutual capability, i.e., in addition to limiting what an agent can consider doing, limiting what it is worthwhile for the agent to consider doing. Horty and Pollack identify a type of synergy where plan steps can be merged, allowing for lower cost overall compared to executing plans in isolation. However, they consider only intention revision with respect to plan merging, and do not consider intention revision in general.

Yao et al. [122] propose an approach to failure recovery for BDI agents based on exploitation of positive interactions between intentions to re-establish preconditions of plan steps. Intentions are scheduled such that the preconditions of non-progressable intentions are restored by the effects of other intentions.

Xu et al. [120] propose a framework for interleaving intentions based on first-principles planning. Their framework allows to avoid conflicts and exploit synergies by scheduling the concurrent execution of intentions. The possible execution traces of a set of intentions are derived and overlapping parts of executions identified as possible candidates for merging in concurrent executions.

Failure handling

Sardina et al. [81, 82, 84] define a semantics of goals in the context of standard BDI handling of plan failures. Declarative and procedural (event) goals are distinguished in their semantics, allowing to decouple goal success/failure from plan success/failure. They consider a case where failure handling involves rationally relaxing commitment to subgoals. When a plan for a subgoal fails or is blocked and cannot progress (and there are no applicable plans for that subgoal), it may be rational for an agent to abandon the subgoal if there is an applicable alternative plan which can be adopted for a goal higher in the goal hierarchy. Sardina et al. note the similarity between rational abandonment of a subgoal in this particular case, and the abandonment of impossible goals as required by BDI theoretical definitions of rational commitment. In their semantics, failure is propagated up the goal hierarchy only when an alternative means is available, which avoids abandoning progress only for the agent to end up unable to progress the blocked intention anyway. Sardina et al. identify the commitment strategy of their semantics as being between the single-minded and open-minded commitment strategies of Rao and Georgeff [75], as goals are dropped only when achieved or in the particular case mentioned.

Unruh et al. [109, 110] propose an approach to semantic compensation in failure handling of BDI agents. Central to their approach is the idea of semantic compensation for intentions, where an agent handles plan failures by adopting a goal to achieve a recovery state. Achievement of a recovery state is considered a practical compromise between making no compensation for failed actions, and the impossible ideal of rolling back or "undoing" actions that were part of a failed plan. In the complex and dynamic environments that BDI agents inhabit, it is not always possible to roll back or undo actions. When plans fail, an agent's environment can be left in an undesirable state that hinders further progress. For instance resources in use by the failed plan may be unavailable for re-attempting the plan or for use in pursuing other intentions. If other intentions were suspended due to conflicts with the effects of the failed plan, those effects may be in place while plan is no longer being pursued, preventing the agent from making progress with respect to those intentions. The approach proposed by Unruh et al. uses domain knowledge specified in terms of goals to determine how best to compensate for failed goals and plans. The goals specified for compensation refer to the effects, resources, and actions involved in the execution of plans and goals, and it is these aspects that are compensated for by adopting a declarative goal corresponding to a recovery state. This is in contrast to executing some fixed sequence of recovery steps defined by an agent programmer, which would correspond to a procedural interpretation of compensation. The magnitude of compensation required, corresponding to the amount of progress that must be abandoned, is proportional to the height in the goal hierarchy at which the failure is handled. Handling the failure higher in the goal hierarchy allows for more general compensation, but at the risk of losing more progress than handling it at a lower level. Failure handling is then the combination of semantic compensation and re-attempting failed tasks.

Bordini & Hübner [7] extend the semantics of AgentSpeak to account for the plan failure handling behaviour of the Jason interpreter. In Jason, goal deletion events may trigger plans that handle clean-up behaviour, or other operations prior to backtracking, i.e., selecting an alternative plan. This is in contrast to the standard AgentSpeak approach to failure handling, where intentions are abandoned if no applicable plan can be found. Moreover, Bordini & Hübner identify cases where plan failure can occur that are consequential to practical implementation of agents (beyond the abstract nature of AgentSpeak), such as failure of actions within plans, and when intentions are suspended while actions are completing. They show how these types of failure are handled in Jason by the failure handling mechanism, how more advanced operations on goals can be performed using Jason's internal actions (allowing more sophisticated failure handling), and give semantics that extend AgentSpeak to facilitate this.

2.2.4 Deliberation over goals

Deliberation strategies

Pokahr et al. [68], introduce a deliberation strategy that assumes the generic goal lifecycle from [67]. Their deliberation strategy identifies when a change to goal states inhibits or uninhibits goals, and deliberate over goals in order to determine whether to activate or suspend goals in light of the change to goal states. The incompatibility relationships between goals are represented by inhibition links forming a directed acyclical graph. The deliberation strategy ensures that uninhibited goals are pursued by the agent (it does not procrastinate) and that conflicting goals are not simultaneously pursued. As changes to goal states are caused primarily by changes to beliefs, the deliberation strategy implicitly relates belief revision to intention revision.

Leask and Logan [56] show how several simple deliberation strategies can be implemented in meta-APL, an agent programming language with reflective capabilities. The reflective capabilities of meta-APL allow for the encoding of deliberation strategies purely at the meta-level, without altering the object-level agent program.

Goal lifecycle

A strand of work [38, 61, 79] building on the seminal goal life-cycle framework of Braubach et al. [13, 67] aims to give a generic operational semantics for common types of goals. In the goal lifecycle models, an agent's attitude toward goals are represented by explicit goal states, which characterise the role of goals in the agent's execution. The goal lifecycle can be viewed as a state transition system that generically describes the states goals may occupy and the transitions that may be made between states corresponding to a change of attitude toward goals or change of focus of the agent.

Braubach et al. [13, 67] describe and formalise the lifecycle of goals in the context of the Jadex [69] agent platform.

Riemsdijk et al. [79] define "active" and "suspended" states for leaf goals, i.e., subgoals and planning are not considered. The active state characterises goals the agent is actively pursuing, i.e., intentions, while the suspended state characterises goals the agent is not currently pursuing. Goals in the active state can have plans assigned to them and executed by the agent. Commitment to goals is defined by success and failure conditions. When either is true, the corresponding goal is dropped by the agent. The success condition corresponds to achievement of the goal, which for declarative achievement goals is identified with the goal formula itself. The failure condition corresponds to impossibility of achieving a goal, which is rational cause to abandon it.

Morandini et al. [61] extend the model proposed by Riemsdijk et al. [79] to account for non-leaf goals, i.e., subgoals and failure handling. Goal state changes may trigger state changes in related goals, for instance if a goal is suspended then its subgoals are also suspended. Likewise for goals that are dropped.

Harland et al. [38] give an operational semantics for BDI agents, accounting for semantics of goals, plans, and actions. Their model unifies the previous goal lifecycle work within a complete operational semantics for BDI agents. They identify additional states and operations for goals, allowing to cleanly incorporate both proactive and reactive maintenance goals in their semantics. In addition they consider states and operations corresponding to suspending, resuming, and aborting goals [101]. Their model is described in detail in Chapter 4.

Castelfranchi and Paglieri [15] investigate the role of supporting beliefs in determining degrees of commitment to goals. Although their approach is theoretical and from a philosophical viewpoint, the stages of filtering by beliefs passed by goals as they progress from representing desires to representing intentions bears a strong similarity to the notion of a goal lifecycle. In their model, goals acquire supporting beliefs that allow them to progress from one stage to the next, moving closer to the degree of commitment corresponding to intentions, i.e., maximal commitment. When supporting beliefs are lost, goals revert back to lower stages corresponding to lesser degrees of commitment, closer to the level of desires, i.e., minimal commitment.

Preferences

Rational BDI agents must derive a consistent set of goals from their desires. However, there may be many possible consistent sets of goals that can be derived. In order to choose between consistent goal sets, an agent can use preferences over goals or sets of goals.

Castelfranchi and Paglieri [15] describe preference beliefs as a type of goal-supporting belief, which can be used for filtering goals during deliberation. Intuitively this means that for a goal in conflict with other goals to progress to being intended, it must be supported by preference beliefs, i.e., it is part of a most preferred execution, although they do not refer to executions explicitly. On the other hand, if a goal loses the support of preference beliefs, this would imply a more preferred goal it is in conflict with would be intended instead, i.e., it would not be pursued in any most preferred execution. This corresponds to using preferences to resolve conflicts between goals.

Thangarajah et al. [104] use preference orderings over consistent goal sets to determine which goals the agent adopts in cases of conflict between goals. They introduce a notion of a ruleset for goal generation (from desires) that determines when goals are adopted or abandoned, by choosing a consistent goal set. Goals are adopted if they are in the chosen set, and abandoned if not. Consequently, the specified ruleset implicitly encodes a commitment strategy for the agent. They give a basic ruleset that corresponds to the open-minded commitment of Rao and Georgeff [75], and extend it to a more flexible form of commitment by adding rules to permit resolving conflict between goals by adopting alternative plans, and preferring goals the agent is already pursuing when conflicts cannot be resolved in that way (corresponding to Bratman's property of stability of intentions [10]). Preference is given as a relation over consistent goal sets that gives rise to an ordering over the consistent goal sets the agent might adopt. A rational agent in their framework repeatedly chooses the most preferred consistent set of goals over the course of execution. The notion of preference employed by Thangarajah et al. is similar in purpose to value (from decision theory) or goal priorities (from agent programming). Although it allows to decide which consistent set of goals an agent should adopt, it is not a general formulation that permits reasoning about the preference of alternative courses of action. Commitment to goals in their model is determined by the rulesets (corresponding to a commitment strategy), rather than a consequence of reasoning about preference over possible courses of action and the past actions (or commitments) of the agent.

In addition to choosing which goals to adopt, BDI agents must choose plans to execute to achieve their goals. There may be several applicable plans an agent could use to achieve a given goal. Executing any of those plans might achieve the goal, however a rational agent should select the applicable plan that best satisfies the agent's preferences, i.e., the maximally preferred plan.

Padgham et al. [66] use plan preferences are to determine the suitability of applicable plans. In agent programming, plan contexts are used to denote when a plan is applicable, but may also be used to constrain the choice of plan further, accounting for suitability. The authors point out that this leads to a burden on the programmer to ensure the plan contexts correctly encode plan suitability and lead to the agent selecting the "correct" plan from among the applicable plans for a goal. They suggest that the concepts of applicability and suitability of plans should be kept separate, allowing the agent to reason about them separately. This is useful in cases such as plan failure, where an alternative plan might need to be used. If the plan contexts are also required to account for plan suitability, they may be so strong that alternative plans are deemed inapplicable and cannot be used in that case. The solution proposed is to use preference formulae to denote suitability of plans. Each preference formula captures a numeric measure of an independent attribute of a plan, according to both the importance of the attribute as a component of preference and the degree to which the plan satisfies it. By taking the sum of these preference measures, a measure of the suitability of a plan is derived. This gives an ordering over applicable plans allowing a most preferable applicable plan to be chosen for a goal, and allows the agent to use less preferrable applicable plans in case of plan failure.

Nunes et al. [63], present a Tropos-inspired meta-model approach to modelling agents with preference-based plan selection over softgoals. Agents seek to maximise the expected utility of chosen plans, and the utility of a plan is determined by its expected contribution to softgoals, taking into account preferences over softgoals. Softgoals are akin to values in decision theory, and represent independent attributes of plans that should be maximised when possible. Plans are selected based on softgoal satisfaction and preferences, in the context of uncertain plan success. Each plan makes positive or negative quantitative contributions to each softgoal. Preferences correspond to a measure of importance of the satisfaction of each softgoal. The plan utility is then the sum of the expected contributions weighted by preferences. Contributions of subgoals to a plan are derived from the plans for those subgoals. The plan utilities are quantitative, giving a definitive ordering over plans, and the agent selects the plan with highest expected utility. As the preferences dictate the relative importance of satisfaction of softgoals by plans in this approach, they are used for informing plan selection only.

Visser et al. [112, 113] extend an existing language for preferences [1, 3] to allow specification of preferences over properties of goals in addition to preferences over plan properties and resource usage. The properties of a goal allow to specify preferences over the effects of achieving it, independently of the plans used to achieve it. For instance, it may be useful to specify preferences over goals in terms of the value of achieving a goal, or the effects brought about by achieving it, independently of the plans used in doing so. This might also include properties such as the priorities of goals. Goal-plan trees are annotated such that plans are assigned properties and resource usage, and this information is propagated upwards through the goal-plan tree by summary information. Agents must select the most preferable applicable plan taking into account preferences over resource usage and properties of plans. This approach is inspired by the summary information work of Thangarajah et al. [105, 106, 108], and extends the notion of necessary and possible resource usage of plans and goals to properties of plans and goals. The degree to which a plan satisfies the agent's preferences (given as preference formulae) is determined by deriving a numeric meausre of preference satisfaction and then ordering the plans by preference satisfaction.

Horty and Pollack [47] consider the problem of assessing candidate plan suitability in the context of an agent's existing plans. The authors present a framework of rational choice for resource-bounded agents in dynamic environments, where choice is not only which plans to execute, but how they are executed in a mutual context to maximise utility. They note that the desirability of achieving a goal may depend on the way the agent achieves it in the context of the agent's existing commitments, and point out that the standard approach to using utilities for plan selection neglects this. Intuitively, executing a plan may be more or less desirable in the context of the agent's existing plans, than it would be in isolation. They consider the compatibility of plans and the impact this has on utility of candidate plans, and take into account the cost and benefit of plans. Cost and benefit can be seen as negative and positive preference respectively. The agent selects the plan that maximises the benefit minus the cost in the context of existing plans, if one exists. This approach goes beyond using preferences for plan selection, as it considers the existing commitments the agent has, and the impact of scheduling plans on their utility. Thus, utility is used to inform a choice of execution of plans (in a mutual context), rather than simply which plans to use.

Hindriks et al. [44] present an architecture for rational action selection taking into account preferences. The agent programming language GOAL is extended with temporal operators for goals and preferences. Both goals and preferences are represented by LTL formulae. The agent uses lookahead to ensure that it selects an action from an execution trace that satisfies the goals, which are treated as hard constraints. In addition, traces are ordered using preference formulae, so that a most preferred trace can be executed. Preferences are treated as soft constraints, which the agent satisfies as much as possible, but is not strictly committed to them unlike goals. Achieving goals is strictly preferred to satisfying preferences, so the preferences can be viewed as a means to decide between possible executions that achieve the agent's goals. Intuitively, the most preferred trace does not have to satisfy all preference formulae, it only needs to dominate the alternative traces. As the lookahead is bounded by the length of the trace prefixes it can inspect, the agent cannot always follow a globally most preferred trace, nor can it guarantee that all goals are achieved on a trace, but instead makes a bounded-rational choice with respect to goals and preferences. This approach goes beyond preferences over plans and goals, as the preference formulae are LTL formulae and can refer to the agent's mental state in a flexible way. For instance, a preference formula can specify that the agent should achieve one goal before pursuing another, or possibly make assertions about the actions that are executed corresponding to plan preferences or scheduling. Therefore preferences in this architecture are at the level of executions.

2.3 Hybrid approaches

In this section, I consider work that explicitly addresses the theory-practice gap. These approaches aim to bring agent programming closer to BDI theory, especially satisfaction of rationality principles. These approaches are "hybrid" in the sense that they take the practicality of agents seriously, while avoiding compromising on correctness and rationality.

Firstly, I discuss agent-programming languages that explicitly address the theory-practice gap, including 3APL [39], GOAL [40], Dribble [76], and BOID [24]. Secondly, I review a strand of work extending the situation calculus for programing rational agents. Lastly, I discuss work that investigates to what extent event-based, PRS-like agents satisfy the rationality principles put forward in the BDI theory literature.

Hindriks et al. [39, 41, 60] specify an abstract agent programming language, 3APL, that makes the connection between commitment to procedural goals and action central and explicit. 3APL agents are endowed with plan revision rules that are used to revise an agenda containing the current plans for procedural goals in its goal base. The agenda therefore corresponds to an agent's intentions. A 3APL agent reasons about its goals and plans by applying plan revision rules to revise its agenda. This represents a reflective reasoning capability of 3APL agents, as goals can be selected for execution by plan revision rules based on reasoning about the beliefs, goal base, and agenda of an agent. Moreover, the execution state of goals and plans can be monitored and accounted for during execution. Hindriks et al. suggest that plan revision rules can be used to program complex deliberation such as commitment strategies, priority ordering over goals, failure handling, and plan optimisation (such as merging or scheduling), which are all useful techniques for enhancing the rationality of agents.

Hindriks et al. [4, 40, 42, 43] identify the lack of declarative goals in agent programming languages as a reason for a conceptual gap between theoretical approaches such as BDI logics, and practical approaches such as agent programming languages and platforms. They note that while agent programming languages focus on procedural goals as their primary programming abstraction, in BDI theory and logics the concepts of desires and intentions are taken as primitive attitudes. This distinction is especially important when considering commitment strategies and rationality principles of agents specified in agent programming languages, where the necessary concepts may not be supported. To address this, Hindriks et al. specify semantics for GOAL, an agent programming language with declarative goals. A GOAL agent has a logical goal base and a notion of capabilities that relates beliefs and goals to actions. Goals are adopted if the agent is capable of them, and the adopted goals form a logical relation from goals in the goal base. While the adopted goals must be consistent, the goals in the goal base are not necessarily so. As the semantics adopts a logical interpretation of goals, Hindriks et al. develop a logic that corresponds to the GOAL semantics allowing to reason about and prove properties of GOAL agents in terms of their program text, rather than program traces. Hindriks et al. extend GOAL with temporal goals and define a notion of satisfiability for temporal goals with deadlines. They employ a lookahead horizon based on the deadlines of temporal goals in order to bound the scope of the lookahead and make satisfiability tractable. As goals may not necessarily be achieved within the lookahead scope, they focus on ensuring that goals are not made impossible. Because the lookahead horizon is based on the deadlines of temporal goals, this notion of satisfiability only ensures temporal goals are not unsatisfiable if they have deadlines, i.e., they correspond to temporal formulas containing “until” or “before”. Moreover, it is unclear what the lookahead bound would be at any given point in execution solely by examining the program text, as it depends on the maximal deadline of goals the agent has adopted, and is thus dynamic and is limited only by the extent of the deadlines imposed by the goals. Although their architecture treats goals as hard constraints, achievement goals have no selective force, i.e., they are not used for any filtering beyond avoiding waste by achieving goals redundantly. Because of this, no treatment of conflicts between goals (including their plans or actions) is given, and achievability of goals is only considered for maintenance goals as they must be satisfied in all states and can be readily tested for satisfaction within a finite prefix of a trace. The finite prefix of a trace may not include achievement of all achievement goals in the trace, so the satisfiability of achievement goals cannot always be determined due to the bounded lookahead. There is no consideration of planning in GOAL agents, and the default commitment strategy given by the semantics is blind commitment, as GOAL agents do not drop goals unless they are achieved.

Similar approaches aim to improve on 3APL and GOAL. van Riemsdijk et al. [76] combines features from 3APL and GOAL into a programming language with beliefs, declarative/procedural goals, and plans. In addition they establish correspondence between their proposed language, Dribble, with a dynamic logic. Dastani et al. [14, 24] specify an agent architecture for reasoning about the relationships between mental attitudes, based on defeasible logic. Their proposed language, BOID, generates goals and plans and filters them by checking feasibility, coherence, and conflicts. Deliberation is then viewed as selecting valid sets of mental attitudes from an acceptable set of alternatives, and the choice over alternatives corresponds to a conflict resolution strategy. However, the task of detecting and resolving conflicts is left to the programmer.

Another strand of work extends the situation calculus for programming rational agents. Shapiro et al. [83, 89, 90, 91, 92] propose an extension to the situation calculus that formalises rational behaviour of agents. They define a notion of capability that formalises rational action of an agent toward its goals. Goals are characterised by the paths an agent can take by acting such that the goals are achieved. An action strategy, dictating what action should be taken in a given situation, induces all possible paths an agent can take to achieve its goals. Shapiro et al. define a rational search operator that takes an IndiGolog program and a set of prioritised goals, and produces a plan that respects the meaning of the program, the goals, and their priorities. The rational search corresponds to finding a legal execution of a potentially non-deterministic IndiGolog program, resulting in a plan that corresponds to a rational execution of an IndiGolog agent.

Khan et al. [51, 52, 53, 54] prove several rationality properties for a Simple Rational APL (SR-APL) with prioritised goals. They consider the concurrent execution of plans of a BDI agent, and the impact of this on capability. Moreover, their language accounts for subgoals and planning, bringing their approach conceptually closer to common BDI agent programming. Goals in SR-APL are temporally extended, allowing to reason about the consistency of intentions, in terms of plans. The temporal extension of goals corresponds to a limited lookahead horizon, although it does not extend to subgoals of plans. Khan et al. prove several rationality properties for SR-APL. The first property states that the agent's beliefs (knowledge) and chosen goals are internally consistent, with respect to the theory $\mathcal{D}$ which describes the world and also the agent's declarative and procedural goals and their dynamics. The other two properties hold in a static environment and essentially state that any action performed by the agent is consistent with the agent's intentions (with respect to the theory $\mathcal{D}_{\bar{Exo}}$ stating that there are no exogenous actions).

Bordini et al. [8, 9] investigate which of the nine asymmetry thesis principles from Rao and Georgeff [74] are satisfied by AgentSpeak(L) [71] agents. They find that AgentSpeak(L) agents satisfy intention-belief incompleteness and belief-intention incompleteness, but not intention-belief consistency. This is because AgentSpeak(L) agents can intend impossible things, but do not necessarily believe the consequences of everything they intend nor intend everything they believe possible. Bordini et al. determine that while intention-desire consistency is satisfied, intention-desire incompleteness is not, as AgentSpeak(L) agents desire that which they intend. Desire-intention incompleteness is satisfied, as AgentSpeak(L) agents can have desires that they do not intend, e.g., if there are no applicable plans for those desires. Desire-belief consistency is not satisfied, but desire-belief incompleteness and belief-desire incompleteness are. This is because AgentSpeak(L) agents do not assert the possibility of satisfying their desires and do not desire everything they believe. Bordini et al. note that the combination of asymmetry thesis principles satisfied by AgentSpeak(L) does not correspond to any of the BDI logics described by Rao and Georgeff in [74].

Wobcke [117] develops a framework for modeling PRS-like agents based on CTL and dynamic logic. He investigates which of the rationality postulates of Rao and Georgeff [75] are satisfied by PRS-like agents. In order to make precise what is meant by “PRS-like”, Wobcke adopts an abstract architecture for PRS-like agents [116] that is intended to extend the BDI architecture of Rao and Georgeff [72] such that it captures the essential properties of the PRS-like family of architectures. Wobcke determines that the postulates of belief-goal compatibility, goal-intention compatibility, beliefs about intentions, and beliefs about goals are satisfied, while the others are invalid for PRS-like agents. The axiom of intention to action is not satisfied, as it requires that agents eventually act on their intentions, but PRS agents can execute their highest value plan and so may never activate lower value plans, permitting indefinite procrastination. Similarly, the “no infinite deferral” axiom is not satisfied as it requires that agents eventually abandon their intentions, which is not the case if an agent indefinitely procrastinates. Moreover, PRS agents may continually unsuccessfully attempt their intentions yet never abandon them despite never achieving them. Wobcke also determines that the axiom of awareness of primitive events is not satisfied by PRS-like agents, as PRS-like agents do not in general track attempted actions and their outcome. Overall PRS-like agents satisfy some of the rationality postulates of Rao and Georgeff, but not all of them. Wobcke notes that those that are not satisfied correspond primarily to the ability to represent declarative goals explicitly, and the distinction between goals, plans, and intentions, both of which PRS-like agents lack.

Hübner et al. [48] present programming patterns for defining declarative goals using plans in AgentSpeak(L). Although goals are a central component of AgentSpeak, they are primarily implicit in plans. Hübner et al. describe plan patterns that transform the plans for goal events in AgentSpeak(L) to permit declarative treatment of goals. Their patterns facilitate encoding several degrees of commitment to goals, that resemble the different types of commitment described by Cohen and Levesque [20], including P-GOALs, single-minded commitment, and open-minded commitment. They emphasise that the addition of the `.dropGoal` internal action (from the Jason [5] interpreter for AgentSpeak) is sufficient for a true declarative treatment of goals without resorting to explicit representation of goals e.g., in a goal base, and without extending the AgentSpeak semantics.

2.4 Rational Intention Revision

A rational agent must not only rationally choose a set of intentions, but also rationally update them in response to changes. These changes are either changes in the environment, reflected by changes to beliefs, or changes in high-level motivation, reflected by changes to desires e.g., to accommodate requests from other agents. Moreover, changes to intentions should be minimal, so as to preserve the stability of the agent's intentions over time, in accordance with Bratman [10].

Grant et al. [36] propose a model of mental state revision for BDI agents, and define a notion of rationality for the model. Their definition of rationality incorporates notions of value and cost for intentions, inspired by decision theoretic treatment of rationality. The model they propose defines ideal rational intention revision, i.e., the properties of intention revision that an ideal rational agent would exhibit.

They define the mental state of an agent as a BDI structure with value defined over subsets of desires, and cost defined over subsets of intended actions. The rational balance of mental attitudes is characterised by axioms that a rational BDI structure satisfies. Altogether, they give a theory of rational revision of propositional mental attitudes. I introduce these axioms informally before presenting them formally below.

Grant et al. distinguish a notion of belief rationality from intention rationality. Belief rationality is the requirement that the beliefs of the agent are mutually consistent and the intentions are consistent with the beliefs. Consistency of intentions with beliefs is determined by a notion of capability for intentions. If an agent believes it is capable of performing an intention then that intention is consistent with beliefs. Capability is determined by satisfaction of preconditions of the intended action. Belief rationality is formalised by axiom A1 [36].

Intention rationality relates beliefs and intentions by requiring that the agent's intentions are consistent (in a limited sense, as I will explain later) and that the intentions do not achieve desires that are already believed to be achieved. Intention rationality is formalised by axioms A2, A3, and A4 [36].

If a BDI structure satisfies axioms A1 through A4, i.e., it is belief and intention rational, then it is said to be *weakly rational*. A weakly rational agent can successfully execute its intentions and achieve a subset of desires, however it may not do so in an optimal fashion. For instance, the subset of desires achieved by its intentions may not be maximally valuable, and the cost of its intended actions may be not be minimal. The difference between the value of a set of intentions and its cost is referred to as the benefit of a set of intentions, and gives a measure of quality of a set of intentions in terms of the tradeoff between value and cost. Note that this resembles a notion of preference for sets of intentions. Grant et al. highlight the compatibility of their model with decision-theoretic approaches as an advantage.

An optimal set of intentions maximises benefit such that there is no alternative set of intentions that could be adopted that has superior benefit, i.e., it is at least as beneficial as all alternative intention sets. Note that this must be taken in the context of a weakly rational agent in order to restrict the alternative intention sets to those compatible with a belief rational and intention rational agent. Thus, Grant et al. define a rational agent as one that is weakly rational and satisfies the optimality axiom A5 [36], which requires that the agent's intentions are maximally beneficial.

The axioms given by Grant et al. in [36] are summarised as follows:

A1 B is consistent, i.e., $B \not\vdash \bot$

A2 I is feasible in the context of B (for every $(\alpha, \theta) \in I$, $B \vdash r_{\alpha,\theta}$, where $r_{\alpha,\theta}$ says that α's preconditions are true, and α terminates and makes θ true)

A3 $goals(I)$ is consistent

A4 For every $\theta \in goals(I)$, $B \not\vdash \theta$

A5 There is no I' such that $S' = \langle B, D, I', v, (c, C)\rangle$ satisfies A1 - A4 and $ben(I') > ben(I)$, where $ben(I) = v(goals(I)) - c(actions(I))$; that is, there is no other set of intentions the agent can select which achieves more valuable goals by cheaper means.

where $\langle B, D, I, v, (c, C)\rangle$ is a BDI structure, B is a set of beliefs, D is a set of desires, I is a set of intentions represented by chosen recipes, i.e., action-desire pairs, v is a valuation function for subsets of D, and c is a cost function for actions in the set C, of which the set of actions within I is a subset, i.e., intended actions are assumed to have known cost. Note that actions in I are stipulated to have known cost. The function $goals$ extracts the desires from the pairs in I, while the function $actions$ extracts the actions from the pairs in I. The capability proposition $r_{\alpha,\theta}$ (corresponding to preconditions of α) denotes that executing the action α can be executed to bring about the desire θ with certainty. Note that from the perspective of these axioms there are no sequences of actions, only singular actions, which may be complex, i.e., correspond to entire plans.

The main contribution of Grant et al. [36] is the specification of revision of a rational BDI structure. They consider the revision of beliefs, desires, intentions, cost, and value. Although revising any of these components of a BDI structure may be straightforward in their model (simply replace the component), they point out that a revision may trigger revisions of other components in order to preserve the rationality of the BDI structure. For instance, adding or removing desires may mean the intentions need to be revised in order to be rational with respect to the updated desires.

In addition to the requirement that revisions preserve the rationality of a BDI structure, Grant et al. stipulate that changes to intentions should be minimal, i.e., follow the principle of minimal change which they term *parsimony*. A change is parsimonious if it gives a rational BDI structure that is at least as beneficial as alternatives, but is also at least as "close" to the previous structure as alternatives. A rational agent in their model chooses the closest updated BDI structure corresponding to a rational change. They define a notion of closeness for intentions in terms of the number of common intentions between a BDI structure and an updated BDI structure, and the number of intentions that are added in the updated BDI structure.

I will now briefly summarise the rational revision operations of Grant et al.. As my focus is on rational intention revision, I limit the discussion to intention revision and operations that can cause it.

The primary effects of belief update of a rational BDI structure that threaten the rationality of the intentions in the structure are effects on the feasibility of intentions (axiom A2), whether or not intentions are necessary (axiom A4), and whether there are more beneficial means by which to pursue an existing intention under the revised beliefs (axiom A5). Note that with respect to axiom A5, intentions may also be added or removed due to changes to their benefit under the revised beliefs e.g., if availability of means has changed.

Adding a desire can affect the satisfaction of axiom A5, as there may then be a more beneficial set of intentions that could be adopted that satisfy the added desire. Removing a desire may prompt the agent to revise its intentions if it had intended to achieve that desire.

Grant et al. distinguish exogenous causes of intention revision from endogenous causes. The exogenous cause they describe is an external agent making commands to a rational BDI agent to add or remove an intention, and that agent revising its intentions accordingly. It is unnecessary to describe the exogenous case here. In the endogenous case, adding or removing an intention corresponds to revising the set of intentions to recover intentional rationality and optimality when other components have changed. Grant et al. also consider revisions to cost and value and their effect on intentions, which can be summarised as revising the set of intentions to ensure satisfaction of axiom A5, i.e., the intention set must be maximally beneficial. Intention revision can then be classified as either due to a change to beliefs, desires, or cost/value.

3 Rational Intention Revision

In this chapter I define the problem of *rational intentional revision*, and compare it to related problems introduced in my analysis of the literature on BDI agents in Chapter 2, including goal deliberation, intention reconsideration, and goal reasoning. I then identify limitations in the existing approaches that constitute a gap in knowledge. To address the limitations in the state-of-the-art approaches in the literature, I generalise the rational intention revision problem and define the sub-problem of bounded-rational intention revision, which can be practically solved to give a realisable model of bounded-rational BDI agency. Finally, I introduce an example to illustrate the main ideas, which I return to throughout the thesis.

3.1 Rationality

I define rationality for BDI agents as follows. An agent executes actions, resulting in an execution trace. Assume a notion of preference over execution traces, such that they can be partially ordered. Then, a BDI agent is rational if it executes a *maximally preferred* sequence of actions. Since the actions a BDI agent may take are defined by intentions, the execution of a BDI agent can be viewed as the joint execution of a set of intentions, resulting in an execution of a sequence of actions.

Not only must the actions that a rational agent executes in pursuit of its intentions be executable, but the future actions of the agent should also be executable when the agent comes to perform them. It stands to reason that a rational agent only intends what it can realistically do. This leads to a notion of *capability* of an agent with respect to intentions. An agent is capable of its intentions if it can execute them fully, i.e., there is a joint execution of the intentions that is consistent with the agent's beliefs.

The capability of an agent towards its intentions is essential to rational behaviour. An intention in isolation may or may not be executable, but that executability may change when part of a joint execution. That is, intentions may interact and this can affect the capability of the agent toward them. A rational agent takes advantage of this by choosing intentions such that they have a most preferred joint execution. This implies that the selection of means for intentions is central to rational behaviour, as the capability of an agent towards an intention depends on the means for that intention in the joint context.

The notion of choosing an execution that accounts for future actions is in stark contrast to approaches that consider only the next steps that are taken toward intentions. By requiring that the full (joint) execution of a set of intentions is most preferred, rationality is defined in terms of what is achieved and how rather than simply the immediate next step an agent takes.

A rational agent must not only choose a set of intentions that it is capable of, but also choose the most preferred among these. Intuitively, the possible executions circumscribe what an agent can realistically intend to do, but a rational agent must execute the most preferred among them. Then rational intention selection can be viewed as filtering sets of intentions by capability, and then by preference, i.e., what is rational to do is a subset of what an agent is capable of.

3.1.1 The Rational Intention Revision problem

When circumstances change, what an agent is capable of and what is most preferred may change. Therefore the set of intentions may need to be revised to ensure that a most preferred execution is followed. Changing circumstances correspond to changes to beliefs and desires. As a corollary of rationality defined as a most preferred execution of a set of intentions, *rational intention revision* implies revising intentions such that a most preferred execution *in the context of updated beliefs or desires* is pursued. A rational BDI agent then intends to follow a most preferred execution of its intentions, and revises this in response to changing circumstances. Not only must the set of intentions be revised, but the manner in which they are pursued may need to be revised. This poses the problem of how to revise intentions such that a most preferred execution is obtained.

Work on BDI theory typically distinguishes between the problems of intention selection and intention revision. In practice, however, agents maintain a set of intentions during execution which they resist abandoning, revising them as necessary, rather than selecting a set of intentions either each time they act or when circumstances change. Therefore intention revision (and its problems) subsume intention selection (and its problems) by viewing intention selection as a special case of intention revision, where the agent currently has no intentions, such as in the initial state.

The rational intention revision problem can be decomposed into sub-problems which correspond to problems addressed in the literature. These sub-problems are: means-end reasoning (choice of plans or actions), goal deliberation, detection of and avoidance of conflicts, exploiting synergy, failure handling, accounting for preferences, and addition or removal of desires. Each sub-problem pertains to a change of relevant beliefs or desires that may alter what the most preferred execution is, and thus how the agent should revise its intentions. See Chapter 2 for a detailed discussion of each sub-problem with respect to the literature.

Means-end reasoning is a sub-problem of intention revision as the availability of applicable means directly impacts the formation of intentions and their mutual executability. In addition, some means may be more preferable to others, and this may depend on the mutual context e.g., to exploit synergy between intentions or avoid conflicts between intentions.

Goals can conflict logically in the case that their combination brings about an inconsistent state, but also in the way they are achieved, including the effects of their associated plans and resources that are used. The idea that an agent should adopt a set of goals that are consistent with its beliefs traditionally falls under the moniker of deliberation. Generally work on deliberation considers only one component of intention revision or another, and is usually at a high level that describes the set of goals the agent should adopt. This usually does not account for the scheduling of steps, the incompatibility of goals and plans, or any notion of preference. Moreover, goals typically have hierarchical structure which makes this more complex. For instance, a subgoal representing a partial execution of an intention may no longer have an applicable plan. Essentially the full execution of goals (including subgoals) must be taken in account in order to determine whether their joint execution is rational, which is usually not what is meant by deliberation.

If an agent is no longer capable of an intention, for instance due to plan failure, the agent may revise its intentions to remedy this and restore capability. In classic BDI failure handling, an intention is abandoned if the agent is no longer capable of it, i.e., if the chosen plan is no longer feasible. An alternative approach is to replace a failed plan with an alternative applicable plan, and propagate the failure upwards through the goal hierarchy if none is available. The classic approach makes the implicit assumption that capability toward an intention is contingent on applicability of the plan currently being executed for it, for instance by using plan contexts or in-conditions. However, an applicable plan or true plan context does not necessarily guarantee that an agent (believes it) is capable of a complete execution of such a plan. For instance, a plan may have a true context but be inconsistent with the agent's other intentions, or in the case of a partial plan, the agent may be incapable of later steps or subgoals, which may not necessarily have applicable plans. Furthermore, a rational agent may abandon a plan in order to adopt a more preferable alternative, even if the existing plan is still executable. This is not accounted for in existing approaches, but is a logical consequence of my definition of rationality.

Preferences are used to determine the best option between the available options, such as when assigning plans to goals, and when deciding which of a set of conflicting goals are pursued. In addition, the notion of preference is potentially more convenient for agent programmers than utility as in decision-theoretic approaches, as it can account for qualitative properties of goals and plans, and their executions, which may be simpler than specifying the utility of a desired state.

Desires may be added or removed from an agent's set of desires. In either case the most preferred execution may change. When a desire is added, an agent may adopt it as an intention, and it may conflict with the agent's existing intentions, prompting further revision. When a desire is removed, an agent should no longer pursue it, and this may allow previously conflicting desires to be intended, or synergies the agent was counting on to be no longer possible.

3.2 Summary of existing approaches

As shown in Chapter 2, a range of approaches to the rational intention revision problem have been proposed [36, 38, 54]. I now summarise the state-of-the-art approaches.

3.2.1 Summary of Grant et al.

I now summarise the work of Grant et al. [36], which comprises a state-of-the-art definition of rational revision of mental attitudes (including intentions) in BDI agents. While I summarise here the essential aspects of how Grant et al. define rational intention revision, I refer the reader to the discussion of their work in the general context in Chapter 2.

The model proposed by Grant et al. defines ideal rational intention revision, i.e., the properties of intention revision that an ideal rational agent would exhibit. Their model tends towards the perspective of BDI logics, and as such they give no operationalisation. They give axioms for rational BDI structures, which represent the state of a rational BDI agent. These axioms correspond to properties of rational BDI agents, and define a weak form of rationality, and a stronger form of rationality which subsumes it. The notion of belief rationality described by their axioms requires that a BDI agent's beliefs are consistent. The notion of intention rationality described by their axioms requires that a BDI agent is capable of its intentions, that its intentions are logically consistent, and that its intentions are not already achieved. Their optimality axiom requires that an agent's intentions are maximally beneficial with respect to their cost and value compared to alternative sets of intentions. If a BDI agent is both belief rational and intention rational, it is said to be weakly rational. If a BDI agent is belief rational, intention rational, and has optimal intentions, it is said to be rational.

In addition to definition of a rational BDI structure, Grant et al. define rational revision of a BDI structure in terms of postulates. Their postulates define rational revision as revision of a rational BDI structure that preserves rationality.

3.2.2 Summary of Harland et al.

As discussed in Chapter 2, Harland et al. [38] give an operational semantics for BDI agents that is centred around the idea of a life-cycle for goals. The model they propose constitutes the state-of-the-art model of goal management, and tends towards the perspective of agent programming. Their semantics accounts for the management of achievement goals, performance goals, and both reactive and proactive maintenance goals. In their model, the goal life-cycle is defined in terms of states that goals occupy, and the transitions between them. The state a goal is assigned represents the attitude of the agent toward the goal, and circumscribes the relevant beliefs that could trigger a change of attitude toward that goal, by goal state transitions conditioned on beliefs. The model defined by Harland et al. allows the agent to select a set of active goals, which constitute the agent's intentions, and to suspend goals in order to facilitate scheduling of intentions. Active goals are considered executable and thus have applicable plans assigned to them. Intention revision in their model is fulfilled by revising the set of active goals, i.e., changing goal states. Their model also accounts for the semantics of plans and subgoals.

3.2.3 Summary of Khan and Levesque

Khan & Levesque [54] specify idealised BDI agent behaviour in terms of a model that accounts for some principles of rationality. Their model aims to bridge the gap between agent theories and practical agent-programming languages, and is based on ConGolog [25]. Intentions are fulfilled by goals in their model, which are assigned plans and may have subgoals. Goals are temporally extended in the sense that a limited form of lookahead permitted. However, the lookahead does not extend to full expansion of subgoals, and only accounts for plans that are already adopted by the agent and goal dynamics e.g., dropping a goal. They define consistency between goals in terms of the consistency of their plans. Goals are consistent if there is a concurrent execution of their plans. Goals are prioritised in their model, meaning that agents adopt a consistent set of goals that is at least as prioritised as other consistent sets of goals. In [54], three rationality properties for their model are proven. The first property they prove states that beliefs are consistent with intentions with respect to a domain theory that accounts for knowledge of the world and the agent's intentions, including goal dynamics. The second and third properties they prove state that any action performed by the agent is consistent with the agent's intentions. The latter two properties hold under the assumption of a static environment, i.e., exogenous actions are not permitted.

3.3 Limitations of existing approaches

I now explain how the existing approaches that I have summarised only partially address the rational intention revision problem.

3.3.1 Critique of Grant et al.

While the model proposed by Grant et al. [36] constitutes a step forward for rational intention revision, it relies on certain assumptions that limit the scope of their solution.

Firstly, they represent intentions as chosen recipes, which are pairs of actions and intended effects, i.e., desires. Although they note that actions may be complex e.g., involving sequence and non-deterministic choice, these actions are treated as effectively atomic and cannot be inspected beyond the associated capability proposition, cost, and value. This means that their model does not account for temporality of intentions, which they acknowledge [36]. Because of this, the agent cannot reason about interactions between complex actions, other than the simplistic notion of mutual capability and consistency of achieved desires. They further assume that executing an intended action does not alter the feasibility of other intended actions. Both of these assumptions, related to capability within intentions and mutual capability between intentions, appear to be due to their more general assumption that the rationality of the agent is atemporal. If actions here are instead assumed to have some limited temporality e.g., intentions are a sequence of actions such as a plan, rather than an opaque complex action, then their notion of intention rationality must be extended to account for this.

In agent programming, plans are typically partial in the sense that they allow means-end reasoning to be deferred, in accordance with Bratman [10]. Partial plans may contain subgoals which have further potentially partial plans selected to achieve them. In that case, determining the agent's capability to execute an intention (corresponding to axiom A2 in [36]) becomes more complex, as the choices of selected plans for an intention may not be known in the current state. In addition, we may relax the assumption that intended actions (or plans here) do not alter the capability of other intended actions (or plans), in which case executing a plan may alter the agent's capability with respect to other plans, even within a single intention in the context of partial plans. In that case, axiom A2 cannot be preserved without reasoning about the preconditions and effects of intended plans to ensure a partial plan has a concrete execution the agent is capable of not only in isolation, but also in the context of other intended plans.

One solution to this is to facilitate reasoning about the executability of intended partial plans containing subgoals using summary information [106, 108], where a partial plan for a goal is represented by a *goal-plan tree*. Then, any complete execution of a partial plan corresponds to a path through a goal-plan tree. The executability of such a path can be determined, allowing to recover axiom A2, i.e., we can ensure that the set of intentions is feasible and execution achieves the desired effects.

Another issue is that the assumptions made so far constrain the agent to achieving intentions serially, i.e., they cannot be pursued concurrently. This is another consequence of the assumption of atemporality. If we relax this assumption further to allow the agent to pursue intended plans concurrently, then this introduces the potential for interactions between intentions which might affect their capability. For instance, executing one intention partially might put the agent in a state where it is no longer capable of pursuing other intentions temporarily. This would constitute a conflict between intentions. Once again, summary information [108] can be used to determine the mutual capability of intentions and schedule the execution of intended plans in order to preserve capability as required by axiom A2.

Rational intention revision in [36] is defined as revision of the set of intended action-desire pairs to produce a BDI structure which satisfies their axioms. Implicit in their choice of representation of intentions is the assumption that intentions cannot fail during execution, and the assumption that intentions are revised in whole rather than in part, i.e., the agent cannot revise only part of an intention as the intended action is opaque and cannot be reasoned about beyond capability and outcome. The former assumption corresponds to the idea that if an agent is capable of an intention then executing it brings about the desired effect. This a strong assumption that implies that the agent's actions cannot fail. The latter assumption eschews, for instance, plan revision as an important aspect of intention revision. Both of these assumptions are consequences of the atemporal treatment of intentions in their model.

3.3.2 Critique of Harland et al.

The assumptions made by Harland et al. [38] preclude a precise treatment of rational intention revision in their model. Although they account for some aspects of rational behaviour, such as dropping an achieved goal (and any subgoals) without further deliberation, in general the goal state transitions are assumed to be mandated by an external deliberation component, which is left unspecified. Moreover, plans in their model are assigned to goals by an external means-end reasoning function, which is also unspecified. They make clear that their intention behind leaving these components unspecified is for the model to be generally applicable.

The notion of capability in [38] is limited to checking the preconditions of the next action in a plan for a goal are satisfied. This means that although the agent may have a rich set of intentions with subgoals, plans, and actions, the capability of an agent with respect to an intention is limited to whether the next action for that intention is executable. This notion of capability does not account for the (possible) future executions of an intention, and precludes reasoning about interactions between actions toward intentions except in the case of the immediate next action for each intention. In the case that capability toward an intention is lost, by the next action's preconditions not being satisfied, the plan fails and must be replaced by the means-end reasoning function. If no alternative plan is found then the failure is propagated upwards by dropping the goal. Thus, the means-end reasoning function is assumed to ensure that suitable plans are assigned to goals. Because this component is unspecified and assigns plans to single goals, the model cannot account for making plan choices in order to facilitate scheduling of intentions or preserve capability of other intentions (avoid conflicts), maximise preference, or a combination of these. Simply, a goal is activated and thus intended before a plan is even assigned to it, so the choice of plan cannot directly affect the choice of whether the goal is intended in the first place. Choices of plans affect the preference of the set of intentions overall, so this model cannot guarantee maximal preference of intentions. Also consider the case where the means-end reasoning function assigns an applicable plan to a goal, yet there is no way to schedule this plan with the other intentions. In that case, the agent has a set of inconsistent intentions. Similarly, if a plan is not applicable in isolation but *is* applicable in the context of the other intentions, by positive interaction, the agent cannot ensure this plan would be assigned

In the goal life-cycle model proposed by Harland et al. [38], the transitions between goal states are conditioned on decisions made by an unspecified deliberation function, which are assumed to be reflected in the beliefs by deliberation facts. Aside from interaction via deliberation facts, it is unclear how the deliberation function interacts with the model, and deliberates over goals and plans. Except for the cases where a goal is dropped if achieved or failed, the transitions between goal states are conditioned on deliberation facts, which implies that the deliberation function is a necessary component of the model. In fact, because the deliberation function determines which goals are active or otherwise, it is responsible for intention revision. In theory the deliberation function may activate or suspend goals arbitrarily, but in practice the responsibility of ensuring that intention revision is rational falls to the agent programmer. Randomly activating goals is permitted, but is unlikely to result in a set of maximally preferred intentions. Moreover it is unlikely to result in a non-conflicting set of intentions, let alone maximise preference. Because of this their model does not account for or stipulate rational intention revision.

3.3.3 Critique of Khan & Levesque

The model proposed by Khan & Levesque [54] accounts for a limited form of rational intention revision. Their model does not account for plan preferences, only strict priority over goals. Although they consider the impact of concurrent executions of plans on the consistency of intentions, they do not define consistency in terms of full executions of goals. Instead, they make the weaker requirement that an adopted plan does not make the already adopted goals impossible. To account for cases where plans are no longer viable, such as in the case where a conflict is unforeseen, they allow plans to be repaired.

3.4 A new model of Rational Intention Revision

In order to address the limitations identified in the existing work, I elaborate on how these limitations can be overcome and propose a solution to the rational intention revision problem.

Firstly, I explain how the complete execution of a goal can be defined. Secondly, I explain how the interleaving of executions for goals allows an agent to reason about the mutual execution of a set of goals. Then, I explain how a maximally preferred execution is defined in terms of interleaved executions of goals. Finally, I show how rational intention revision is defined in terms of a most preferred future execution of a set of goals.

3.4.1 Goal Reification

Full executions of goals can be derived from subgoals and plans, by the process of *goal reification*. For the sake of this explanation, assume goals correspond to goal-plan trees. Goal reification results in a set of traces corresponding to a complete execution of a goal. Consider the case of achieving a single goal by determining a sequence of external actions that realise that goal. This sequence of external actions can be viewed as a concrete execution of a goal.

The behaviour of an abstract agent can be characterised as an execution of a series of external actions that ultimately achieves a set of goals. Then, given a set of goals to achieve, the agent determines a sequence of external actions that achieves them.

Reifying a goal

Reification of a goal completely removes any ambiguity about how it will be achieved, and corresponds to complete means-end processing of a goal. For partial plans, we can replace any subgoals with an execution for each subgoal, i.e., we reify the subgoals. By doing this recursively, we can derive a complete plan.

When a goal has several (partial) plans for it, the reification of a goal yields a set of executions or complete plans. Each execution corresponds to a unique set of plan choices, one plan choice for each (sub)goal that is reified. Note that for a goal-plan tree reification yields a set of executions that correspond to all complete paths through the tree.

Reifying a goal set

A set of goals is interpreted as a set of objectives that any execution of an agent should aim to achieve. In an ideal execution the agent achieves all goals in the set (although not necessarily at the same time). Sometimes an execution that achieves all of the goals may not be possible due to incompatibility between goals.

We can extend the notion of goal reification to a set of goals by interleaving the executions of a set of reified goals. While reifying a goal gives executions that achieve that goal, reifying a set of goals gives interleavings of those executions that achieve the entire set of goals. Each interleaving represents a combination of executions of all the goals in the set, potentially concurrently (interleaved). Note that we can derive interleavings for subsets of the abstract goal set as well.

In order to achieve a set of goals, the agent carries out a goal set execution. The execution of a goal corresponds to a complete plan, while the execution of a set of goals corresponds to an interleaving of complete plans. The sequence of actions that achieves a set of goals together (although not necessarily at the same time) can be viewed as an interleaving of a set of action sequences that achieve the goals in the set.

Extending goal reification to a set of goals introduces several interesting problems that are central to rational intention revision.

Firstly, some executions of goals may not be applicable in the current state of the agent's environment. We could require that the executions are compatible with the agent's beliefs, as in pure planning approaches, but choose instead to allow that some executions are not currently applicable for reasons that will be clear in the sequel.

Secondly, goals may negatively interact when their executions are interleaved, i.e., the executions of goals may be incompatible with each other. This may always be the case e.g., if two executions require a consumable resource and there is not enough for both, or only when the executions are interleaved in certain ways, e.g., when the effects of actions toward one clobber the preconditions of actions toward another. This means that in some cases although the goal executions that are interleaved are applicable in isolation, their interleaving may not.

Lastly, goals may positively interact when their executions are interleaved. If the goal executions being interleaved are not applicable, this does not necessarily mean their interleaving is also not applicable. Consider the case where the effects of actions in one goal execution enable the preconditions of actions in another goal execution which would in isolation be inapplicable. Then there may be an interleaving which combines these goal executions which is applicable despite the goal executions it is made up of not all being applicable.

3.4.2 Most preferred interleaving

Under the assumption that intentions are traces, and a rational set of intentions is captured by a most preferred interleaving, the more general representation introduced permits revision of parts of intentions, and permits revision of intentions during their execution if circumstances change.

An executable most-preferred interleaving, constituting a rational agent's intentions, may be threatened by changes to beliefs. If beliefs change such that the interleaving is no longer executable (i.e., the intentions are no longer feasible), then the agent must seek an alternative most preferred interleaving, i.e., revise its intentions. Likewise, if beliefs change such that an active goal of the interleaving the agent was pursuing is believed to be achieved, then the agent should avoid pursuing it. In that case, the agent should seek an alternative interleaving where any plans for that goal are skipped. Skipping over plans for achieved goals corresponds to avoiding adopting goals that have already been achieved. Similarly, if a goal is achieved while an agent is executing a plan for it, the remainder of the plan can be skipped over. In either case the agent avoids executing steps toward goals that are already achieved.

As for addition of a desire, the agent may seek an alternative interleaving that also achieves the added desire if there is one that is more preferred than the one it is already pursuing. Similarly, if a desire is removed then the agent must seek an alternative interleaving if it was an active goal of the most preferred interleaving it was pursuing. In the addition case, the set of interleavings can be extended using the set of traces for the added desire. Intuitively, this means giving the agent the option to select that desire as an intention and pursue it concurrently with other intentions. In the removal case, the set of interleavings is contracted such that there are no interleavings where the removed desire is selected as an intention.

One issue with the correspondence of intention revision with selecting an alternative interleaving is that interleavings are *complete* executions of a set of intentions. If circumstances change during execution it would be irrational to start over from scratch if progress has been made toward some intentions, and they are pursued in the most preferred alternative interleaving, i.e., they would still be in the revised intention set. Note that this problem does not occur in [36], as under their assumptions there is no notion of progress toward an intention.

To resolve this, one solution is to allow the agent to pursue a partial interleaving corresponding to a continuation of execution from a given point. Then rather than selecting an alternative most preferred interleaving to pursue, a rational agent would have the option to preserve the progress it has made toward its intentions if they are unaffected by revision. To represent this, I define the notions of an interleaving *prefix* and *suffix*. Any interleaving can be split into a prefix and suffix, where the prefix corresponds to actions the agent has already executed, and the suffix corresponds to the remaining actions to be executed. A prefix is permitted to be empty, allowing the suffix to be any interleaving from the set of interleavings, capturing the initial state of an agent. It is important to note that in the set of interleavings there may be many interleavings which share a given prefix, but which differ in their suffix. Thus, an agent can pursue an interleaving up to a point, and adopt a different suffix from the one originally intended if circumstances change. This corresponds to executing a different interleaving overall.

To illustrate, consider Singh's cafeteria agent [99], which serves drinks to customers. One customer might order a coffee, while another orders tea. In that case, the cafeteria agent forms an intention to make coffee and an intention to make tea, and sets about executing these. The agent might pick up a cup as the first action toward its intentions. At that point, the agent's remaining actions are to either serve coffee in the cup and then make tea, or serve tea in the cup and then make coffee. If at that moment either of the customers cancels their order, the agent should not put down the cup and start from scratch, but instead continue with its remaining uncancelled order, ensuring that its future actions fulfil its revised intentions. For instance, if the customer ordering coffee decided against it just as the agent picked up a cup, the agent would then no longer consider interleaving suffixes where it serves coffee to that customer (either before or after the tea), and instead executes a suffix where it just serves tea to the other customer. On the other hand, if the agent had started to pour tea into the cup, it would still be able to continue with that intention, and discard its intention to serve coffee to the customer. Of course, if the customer cancelled their order after the agent had already started pouring coffee into the cup, then revising intentions is not as simple as deciding to make tea instead.

Representing the agent's progress toward intentions so far by a prefix is insufficiently flexible to capture intention revision entirely. Consider the case where a desire is removed, and the agent had already made some progress toward it as an intention, captured by actions on an interleaving prefix (as with the cafeteria agent that has poured an unwanted coffee). Then, it is impossible to derive a suffix that avoids executing the remaining actions toward the removed desire, as the interleavings that match the prefix are those where progress has already been made toward the removed desire. In order to avoid this problem, rather than maintaining a prefix I stipulate that agents maintain a *history* of their executed actions, which does not necessarily correspond to the prefix of any interleaving. An agent derives a prefix, corresponding to the progress it preserves from its history, and uses it to derive suffixes from the interleavings.

Given a choice of interleaving suffixes, a rational agent executes the most preferred. This recovers the optimality axiom A5 from [36].

As the choice of prefix is not fixed (constrained by the history), agents have a choice of how much progress toward intentions is preserved, dictated by the preference of the suffix that can be derived using the chosen prefix. This allows to capture typical BDI failure handling. In typical BDI failure handling, if a plan is no longer executable during execution, the plan is dropped and an alternative plan is selected for the parent goal. This corresponds to removing progress made toward the plan in order to pursue an alternative. Plan failure can propagate through a goal-plan hierarchy if the plan failure causes the parent plan to fail also, e.g., if there are no alternative plans for the failed plan. This corresponds to removing even more progress in order to pursue an alternative. Removing progress to allow pursuing an alternative is captured by the notion of deriving a shorter interleaving prefix from the history, in order to select a more preferable suffix. BDI agents are well suited to dynamic environments due to their robust failure handling and ability to respond to changing circumstances, so this is essential behaviour to capture.

As long as an agent is always pursuing a most preferred interleaving suffix consistent with its history, interleavings, beliefs, desires, then there is by definition no other interleaving suffix the agent could adopt that is more preferred. This is the case even when progress must be removed in order to pursue a more preferable interleaving suffix, when beliefs change such that executability is threatened, when goals can conflict, when goals are achieved by external forces, and when desires change. Therefore, an agent executing a most preferred interleaving suffix under the assumptions I have made is rational, i.e., if it starts in a rational state satisfying the axioms A1-A5 of Grant et al. [36], then it preserves this even when circumstances change. As I have described how the interleavings correspond to executions of intention sets, and how an interleaving suffix is chosen (and an alternative chosen if circumstances change), rational intention revision is captured.

The degree to which a rational agent revises its intentions in reaction to a belief update should be proportional the significance of the changes to the beliefs. Significant changes imply a significant change of course is necessary to retain a rational course of action under the changed circumstances. Likewise, a less significant change should prompt a less significant change of course. One of the central problems of rational intention revision is defining

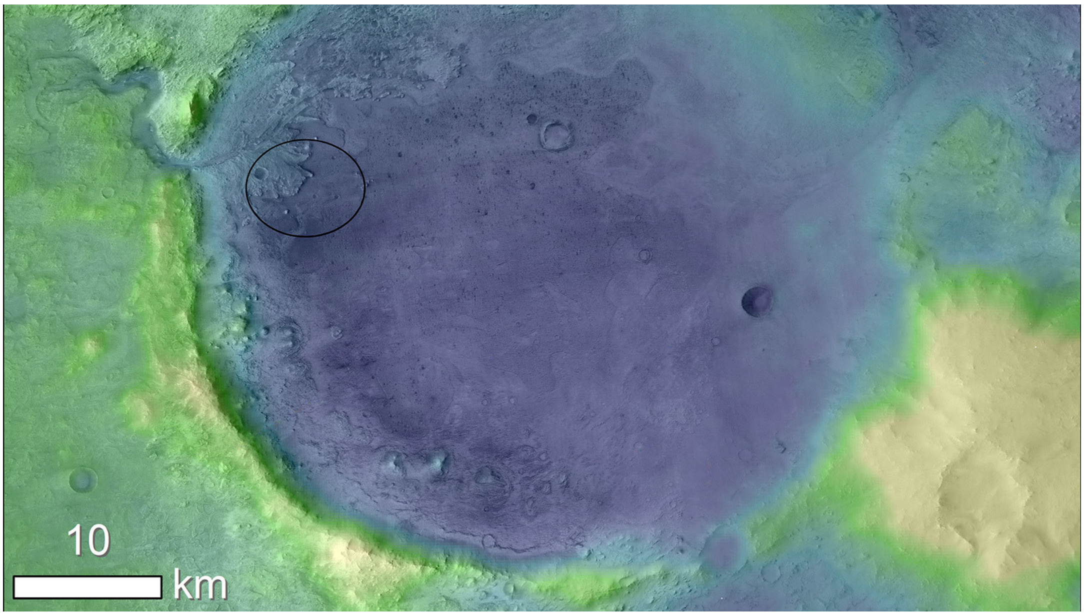

Figure 3.1: Jezero Crater, landing site for Perseverance mission shown circled. Lighter colours show higher elevation. Credit: NASA/JPL-Caltech/MSSS/JHU-APL/ESA

what makes a belief update significant in this sense. A corollary of this is how the intentions should be revised in proportion to the change. Defining what constitutes a relevant belief update and a significant change are central to the problem of rational intention revision, as is determining how an agent should revise its intentions proportionately in reaction to these changes. In fact, solving this corresponds to solving the classical problem of intention reconsideration.

3.5 Mars rover scenario

In order to aid with explaining the behaviour of agents throughout the sequel, I will now introduce a practical example that will be revisited throughout. This scenario is derived from the garbage removal example used by Rao to introduce AgentSpeak(L) in [71], and inspired by the detailed Mars rover scenario described in [38].

3.5.1 Perseverance

In 2020, NASA launched the Perseverance mission. The main objective of Perseverance is to land a rover on the surface of Mars that is tasked with performing experiments, exploring the landing site, and collecting rock samples. The rock samples collected by the rover are intended to be returned to Earth by a later mission. Perseverance is the successor to the Curiosity rover, which is described as the inspiration for the example scenario given in [38].

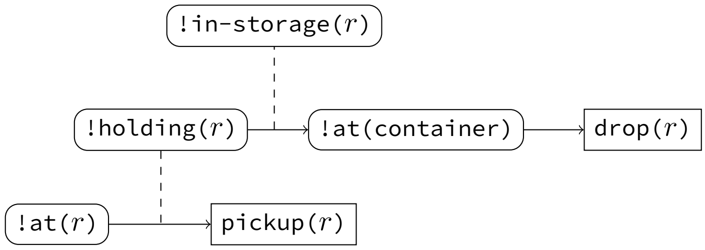

Figure 3.2: (Partial) goal-plan tree for Mars rover's "collect rock from location" behaviour

The planned landing site for Perseverance is the Jezero Crater (pictured in Figure 3.1), which is hypothesised to be the location of an ancient river delta in Mars's distant past. It is predicted that the area of most interest is the western rim of the crater, where the detected concentration of carbonate mineral deposits are highest. These carbonate deposits are of significant interest as they may have preserved signs of ancient life. However, the hills making up the western rim of the crater are reported to be up to 1600 feet in height (roughly 488 metres), which could make the terrain difficult for an autonomous rover to traverse. Therefore, it is essential that a rover travelling across such terrain is robust to failure, and able to tolerate a degree of unpredictability in its interactions with the environment.

3.5.2 Worked Example

Let us imagine a scenario similar to that of Perseverance, where a rover collects rock samples for later processing. The rover must travel to and collect rocks before returning to and depositing them in an immobile container. In transit, the rocks are carried on a trailer.

The objective of this rover is to collect as many of these rocks as possible. Since it can carry several rocks at a time, the rover may minimise the number of trips it makes by travelling directly between the locations of rocks rather than collecting and depositing one rock at a time. The environment is represented as a grid where each cell may be occupied by either the container, or potentially mutiple rocks to be collected. Let us further assume that the rover can occupy the cells where the container and rocks are located in order to deposit and pick up rocks respectively.

The rover is only operational during the Martian sol, as it must be able to use its sensors to navigate and manipulate objects accurately. Therefore, the time available to complete its task is limited. For simplicity, assume moving between adjacent cells has a fixed time cost, and that travelling between locations has a (predictable) linearly increasing cost with distance travelled. Also assume that actions like picking up a rock and dropping a rock have fixed costs that are invariant of any properties of the rock being manipulated.

The top-level goal for collecting a rock sample can be represented as a *goal-plan tree* (see Figure 3.2).

Goals are represented by rounded-corner nodes, and actions by rectangles. The relation between goals and plans is denoted by a dashed line. In Figure 3.2 the goal `in-storage`(r) has a single plan made up of two subgoal expressions and an external action, and denotes a goal to have deposited the rock r in the container. Two of the subgoals (subgoal expressions are denoted by the "!" prefix) are invocations of the `at`(r) goal which signifies that the agent should be at the location of the object r. The `holding`(r) goal denotes that the agent should be holding the rock r. After picking up a rock, the agent carries it until it drops it. The `drop`(r) action removes the rock r from the trailer and drops it at the rover's current location. If the current location is the container, the rock is deposited into the container when dropped.

Generally there will be multiple rocks in the environment that the rover is tasked with collecting, and thus it would adopt several (ground) top-level `in-storage`(r) goals, one for each rock it is tasked with collecting. Although the rover can carry multiple rocks in its trailer, it can only be in one location at a time. In addition, the trailer has a limited capacity for rocks so the rover can only carry so many, up to some maximum quantity known to the rover.

In the remainder of this thesis, I present a model of an idealised rational BDI agent, compare it to existing approaches, and propose a bounded version of the model.

4 Unbounded GROVE

In this chapter I present GROVE, a model of rational BDI agency that follows the outline for a new model addressing the rational intention revision problem outlined in Chapter 3. In GROVE, intentions are selected by interleaving their concrete executions to determine a most preferred way of achieving them.

I introduce the model gradually alongside explanations and intuition for key concepts, culminating in a formal semantics of the model. I then draw some comparisons with existing models of intention revision.

I show that GROVE is strongly rational as it complies with the postulates given by Grant et al. [36], and demonstrate that the executions of GROVE agents are a subset of possible executions of equivalent agents based on the lifecycle model of Harland et al. [38]. Finally, I show that there are irrational executions permitted by an agent using the model in [38] which are not permitted by the equivalent GROVE agent.

4.1 Preliminaries

In this section, I introduce and define the basic elements of GROVE.

I assume a set P of atoms, and denote by L the set of literals over P: $L = P \cup \{\neg l \mid l \in P\}$. The entailment relation $\models$ is defined as follows: for $P' \subseteq P$, $P' \models p$ iff $p \in P'$ and $P' \models \neg p$ iff $p \notin P'$, i.e., negation is interpreted as negation as failure. The complement of a literal l is denoted $\sim l$, and for a set of literals L the complement $\sim L$ is defined as: $\sim L = \{\neg p \mid p \in L\} \cup \{p \mid \neg p \in L\}$. $P' \models L$ iff $\forall l \in L\ P' \models l$.

4.1.1 Beliefs, Goals and Plans

The agent's *beliefs* $B \subseteq P$ represent the agent's information about the environment and itself. The agent's *possible goals* are denoted by $D \subseteq P$, where each $g \in D$ represents a state of affairs that the agent may want to bring about, and which it has the means to achieve in at least one environment state. A goal g is considered achieved iff $B \models g$.

The set of plans available to the agent are denoted by Π. Each plan $\pi \in \Pi$ consists of a sequence of plan steps. Each plan step is either an *action* $e \in Act$ or a *subgoal* $!g$, $g \in D$. Plans are defined by the grammar $\pi = (e \mid !g)^+ \mid \epsilon$, where ϵ denotes the empty plan, and the set of actions Act is the union of actions that appear in plans available to the agent. The function $plans : D \mapsto 2^{\Pi} \setminus \{\emptyset\}$ returns the (non-empty) subset of the agent's plans that achieve a goal, i.e., for each $g \in D$ the agent has at least one plan to achieve it (sometimes termed *relevant plans* in the BDI literature).

The *preconditions* of an action $e \in Act$ are a set of literals which must be true before the execution of the action, and the *postconditions* of the action are a set of literals that are expected to be true after the execution of the action. For an action e with preconditions pre(e) and postconditions pos(e), if $B \vDash$ pre(e) then $B \vDash$ pos(e) immediately after executing e. I stipulate that the pre- and postconditions of an action are consistent, i.e., pre(e) does not contain $l, \sim l$ for any l (and the same for pos(e)).

The relations between goals, plans and actions can be represented using *goal-plan trees* (GPT) [19, 103, 105].[1] The root of a GPT is a top-level goal (goal-node), and its children are the plans that can be used to achieve the goal (plan-nodes). In general, there are several alternative plans to achieve a goal, hence the plan-nodes forming the children of a goal-node are viewed as 'OR' nodes. In contrast, plan execution involves performing all the steps in the plan: hence, the children of a plan-node are viewed as 'AND' nodes. As in Yao et al. [121, 122], I consider goal-plan trees in which plans may contain primitive actions in addition to sub-goals.

Each goal g induces a goal plan tree $\tau = gpt(g)$ rooted at g. A goal plan tree thus represents all possible ways of achieving the goal g available to an agent. Consider an execution of a goal-plan tree $gpt(g)$ through to the achievement of the top-level goal g. This entails executing a plan that achieves the root goal, which might include subgoal steps that must be recursively evaluated, i.e., they constitute goal-plan (sub)trees. A complete path taken through the goal-plan tree can be identified by the sequence of plans that are executed, one chosen for each subgoal step. Each of these paths represents one way to achieve the top-level goal at the root of the tree.

I further assume that goal-plan trees are non-recursive, i.e., in a well-formed agent program g should not occur as a subgoal in any means to achieve g. For example, consider the declarative (to-be type) goal `at(`r`)` (from the Mars rover example) that describes the desired state of being at the location of the object r. In a purely declarative interpretation of goals, it makes no sense for `at(`r`)` to be instrumental in achieving itself. In addition, recursive goal-plan trees enable potentially infinite (non-terminating) agent executions. However, this interpretation may not reflect how agent programs are structured in some agent-programming languages, where goals are interpreted more procedurally and this behaviour is employed, e.g., for looping behaviour. For reasons that will become clear in the remainder of this chapter, recursive goal-plan trees would be problematic for determining whether an execution is feasible and whether an agent's goals will eventually be achieved.

Returning to the Mars rover scenario from Chapter 3, suppose that the rover has to collect a rock from a given location and deliver it to the container. This corresponds to a goal `in-storage(`r`)`, which involves accomplishing the subgoals `holding(`r`)` and

[1] The goal-plan trees corresponding to a BDI agent program can be derived in a straightforward way [98].

`at(container)`, and finally executing the drop action. However if there is no applicable plan for `holding(`r`)`, for instance because the trailer is already full, then there is no complete execution of `in-storage(`r`)` that is executable. In this case, `in-storage(`r`)` contains subgoal steps which themselves have plans, potentially containing further steps. In order to determine whether the goal can be achieved, it is necessary to know whether there is a viable complete path for it through the goal-plan tree, which can be executed.

4.1.2 Step Sequences, Traces and Interleavings

GROVE is based on possible executions of the plans an agent may use to achieve its goals. Executions of plans are induced by goal-plan trees. In this section, I introduce the key definitions that allow us to make this notion precise.

Step Sequences A *step sequence* is a sequence $\sigma = s_1, s_2, \ldots, s_n$ where each *step* s_i is a pair (A, e) consisting of a set of *active goals* $A \subseteq D$, and an action $e \in \mathit{Act}$. Intuitively, the active goals for an action e can be thought of as the ends for which the action forms (part of) the means. The set of active goals for a step sequence $\sigma = (A_1, e_1), \ldots, (A_n, e_n)$ is given by

$$\mathit{agoals}((A_1, e_1), \ldots, (A_n, e_n)) = \bigcup_{i=1}^{n} A_i$$

In the context of the Mars rover scenario, a step might look like:

$$(\ \{\texttt{in-storage(rock)}, \texttt{holding(rock)}\},\ \texttt{pickup(rock)}\ )$$

which represents picking up a rock, making progress towards achieving the goals `in-storage(rock)` (a top-level goal) and `holding(rock)` (a subgoal). In later steps, the active goal set might be simply `in-storage(rock)`, reflecting the fact that the goal `holding(rock)` has been achieved and dropped, while the goal `in-storage(rock)` remains to be achieved.

The *projection* of a step sequence $\sigma = s_1, \ldots, s_n$ with respect to a set of atoms E, σ_E, is defined by

$$\begin{aligned} \epsilon_E &= \epsilon \\ (A, e)_E \circ \sigma &= \sigma_E \text{ where } A \cap E \neq \emptyset \\ (A, e)_E \circ \sigma &= (A, e) \circ \sigma_E \text{ where } A \cap E = \emptyset \end{aligned}$$

where $\circ$ denotes concatenation of step sequences. That is, the projection of σ with respect to E is the sequence σ' in which all steps that have active goals in E are omitted. Note that projection preserves the ordering of steps in σ.

A *history* is a step sequence containing the steps executed by the agent so far. A history h' is a *subhistory* of a history h if there exists $E \subseteq agoals(h)$ such that $h' = h_E$. Let us call E the *elided goals* of h and the steps in h not appearing in h' *elided steps.* [2]

The pre- and postconditions of a step sequence are denoted by *prec* and *post* respectively:

$$\begin{aligned}
\text{prec}(A, e) &= \text{pre}(e) \\
\text{prec}(s_1, \ldots, s_n) &= \text{prec}(s_1) \cup \\
&\bigcup_{i=2}^{n} \Big[\text{prec}(s_i) \setminus \text{post}(s_1, \ldots, s_{i-1})\Big] \\
\text{post}(A, e) &= \text{pos}(e) \\
\text{post}(s_1, \ldots, , s_{n-1}, s_n) &= \text{post}(s_1, \ldots, s_{n-1}) \uplus \text{post}(s_n)
\end{aligned}$$

where $\uplus$ is defined as $X \uplus Y = \big(X \setminus \sim Y\big) \cup Y$. Note that the precondition of a step sequence excludes preconditions established by steps earlier in the sequence (and not undone).[3] On the other hand, the postconditions of a step sequence includes *all* literals that are established by actions in the sequence not undone by a later step.

A step sequence $s_1, s_2, \ldots, s_n$ is *coherent* if no step destroys the preconditions of later step(s) in the sequence, that is, at no step s_i there exists $l \in \text{post}(s_1, \ldots, s_i)$ such that $\sim l \in \text{prec}(s_{i+1}, \ldots, s_n)$. Note that any prefix and suffix of a coherent step sequence are themselves coherent. A coherent step sequence $s_1, s_2, \ldots, s_n$ is *executable* given beliefs B if its preconditions are true given B, that is if $B \vDash \text{prec}(s_1, s_2, \ldots, s_n)$. An executable step sequence $s_1, s_2, \ldots, s_n$ is *non-redundant* given beliefs B if

$$\forall s_i, s_j, i < j,\ B \cap agoals(s_j) \neq \emptyset \rightarrow \text{post}(s_i) \cap \sim agoals(s_j) \neq \emptyset$$

Intuitively, a non-redundant step sequence does not contain a step s_j with an active (sub)goal that is currently believed unless the negation of the goal is a postcondition of an earlier step s_i. Concatenation for step sequences is denoted by $\circ$. As usual, ϵ is identity for $\circ$, i.e., $\sigma \circ \epsilon = \epsilon \circ \sigma = \sigma$.

[2] Some steps may need to be executed even if an active goal has been elided, such as steps to release resources. It is straightforward to ensure such steps are not elided, but I omit this for brevity

[3] These are called *preparatory effects* in [103]. However I extend their notion to include the establishment of the precondition of an action by a previous action in the same plan.

4.1.3 Traces and Interleavings

Traces A *trace* is a step sequence corresponding to a possible execution of a plan for a goal. Traces are generated by expanding the subgoals in the plan and the plans for those subgoals recursively. The set of execution traces, $traces(g)$, induced by a goal $g \in D$ is given by:

$$\begin{aligned} traces(g) &= traces(\{g\}, g) \\ traces(A, g) &= \{\epsilon\} \cup \\ &\qquad \{\sigma \mid \sigma = expand(A \cup \{g\}, u_1), \ldots, \\ &\qquad\qquad expand(A \cup \{g\}, u_k) \\ &\qquad\quad \text{for some } u_1, \ldots, u_k \in plans(g)\} \\ expand(A, e) &= (A, e) \\ expand(A, !g') &\in traces(A, g') \end{aligned}$$

Note that $traces(A, g)$ contains the empty trace ϵ, corresponding to the case where a (sub)goal is achieved fortuitously and steps to achieve it do not need to be executed. I stipulate that each trace $\sigma \in traces(g)$ is coherent.

The definition of $traces(A, g)$ can be extended to account for richer goal and plan semantics. For instance, we could define trace generation for executing steps in parallel (as in PRS [34]). The parallel operator enforces no ordering on the steps it encloses, allowing the agent to execute those steps in arbitrary order provided that they are all completed. This could be accomplished in GROVE by expanding parallel sequences into a set of traces corresponding to the orderings of the constituent steps in the parallel step, however a formal realisation of this is future work.

Another possible extension to trace generation would be introducing *abort methods* similar to those in [37], where each plan has a (possibly empty) sequence of primitive actions that should be executed when a plan is cancelled, such as when the parent goal has been achieved unexpectedly, or the plan has failed. As plan cancellation may cascade, several abort methods might be invoked in sequence. Abort methods may be used to restore the environment to the state prior to executing the plan, or for releasing resources acquired by the plan but not released at the point of failure. In order to identify the abort method for a given plan that has been cancelled, steps must be annotated with the plan that they originated from. When steps toward active goals are elided from the history, abort methods must be executed for each plan to which those steps belong. Extending GROVE with abort methods is future work (see Chapter 6 for further discussion).

Interleavings An *interleaving* is a step sequence corresponding to a possible execution of the traces induced by a set of (top-level) goals $G \subseteq D$. The set of interleavings for a set of traces is generated by freely interleaving the steps comprising the traces whilst preserving the ordering of steps within the traces and their coherence. More precisely, the set of interleavings generated by a set of goals $G = \{g_1, g_2, \ldots, g_n\}$ is given by

$$\begin{aligned} \mathit{inters}(G) = \{\rho \mid \rho = \sigma_i \,||\, \ldots \,||\, \sigma_j \;\wedge \{g_i, \ldots, g_j\} \subseteq G \\ \sigma_i \in \mathit{traces}(g_i), \ldots, \sigma_j \in \mathit{traces}(g_j) \wedge \\ \rho \text{ is coherent } \} \end{aligned}$$

where $||$ is the interleaving operator. As interleavings are required to be coherent, each interleaving $\rho \in \mathit{inters}(G)$ is executable in some environment, i.e., for each $\rho \in \mathit{inters}(G)$ there is a set of beliefs $B' \subseteq P$ such that $B' \models \mathrm{prec}(\rho)$. Note that, a set of goals G may have no coherent interleavings. For example, achieving goals $g, g' \in G$ may each require the consumption of some non-renewable resource such as time, energy or money, so that it is possible to achieve either g or g' but not both.

A coherent interleaving is executable if its preconditions are believed true, while incoherent interleavings are not executable in any environment. The environmental states in which a coherent interleaving may be executed can be viewed as defined by those preconditions. Therefore executability is a tighter constraint than coherence, as to be satisfied it requires that the agent's environment is believed to be in one of these states corresponding to true preconditions. However, the environment may not necessarily be (believed to be) in one of those states, in which case an interleaving may be coherent but not executable.

In general, some interleavings will be preferred to others. For example, interleavings that achieve higher priority goals, or goals requiring fewer resources to achieve, may be preferred. Let us assume a *preference ordering* on interleavings specified by a relation $\mathit{prf}(B, G, \sigma, \sigma')$ which is true when the interleaving σ is strictly preferred to the interleaving σ' given beliefs B and goals G. I assume prf is a strict partial order: i.e., a relation that is irreflexive, asymmetric and transitive, and that all executable interleavings are preferred to all non-executable interleavings. I further assume that the preference ordering prf over interleavings applies to suffixes of interleavings. Incoherent interleavings are non-executable and can never be executed. Consequently they are assumed to have undefined preference, as they are never preferable to anything and shouldn't be executed even if there are no coherent alternatives.

Preferences are useful for determining which course of action to take when there is a choice, e.g., choosing which goals to achieve if some are in conflict. We might consider different types of preferences or preference beliefs taken into account by prf, such as those derived from goal-plan tree annotations written by the agent programmer as in [113], or in terms of benefit derived from the value of goals and the cumulative cost of steps to achieve them as in [36]. Castelfranchi & Paglieri [15] suggest dividing preference beliefs (for goals) into at least two sub-classes: value and urgency. Value beliefs prescribe a measure of value in achieving the goals they pertain to. Urgency beliefs refer to temporal limits on goal achievement, such as deadlines (as in [111]).

The set of possible future executions of an agent with beliefs B, goals G and history h is the set of most preferred suffixes of interleavings in $inters(G)$ that have a subhistory of h as a prefix. More precisely,

Definition 4.1 (Possible Future Executions)**.** *The set of* possible future executions $pexecs(B, G, h)$ *is given by:*

$$\begin{aligned} pexecs(B, G, h) \ = \ & \\ \{\sigma \mid \exists X \ ((h_X \circ \sigma) \in inters(G)) \ \wedge \ \sigma \textit{ is non-redundant } \wedge \\ \neg\exists X' \exists\sigma'((h_{X'} \circ \sigma') \in inters(G) \ \wedge \\ prf(B, G, \sigma', \sigma) \ \wedge \ \sigma' \textit{ is non-redundant })\} \end{aligned}$$

Intuitively, the possible future executions of an agent with goals G are the most preferred, non-redundant suffixes of interleavings in $inters(G)$ that form a continuation of a (projection of) the history of actions executed so far. That is, the agent will continue to pursue a course of action unless the situation (and the agent's beliefs) changes in such a way that a different interleaving suffix becomes more preferred. Note that while the agent will start executing the more preferred interleaving at a point that is consistent with the actions already executed in the history, the choice of possible future executions is based solely on preferences over the actions yet to be executed. In particular, a longer suffix of an inteleaving in $inters(G)$ that "reuses" fewer actions from the history may be preferred (e.g., have lower cost) than a future execution that reuses more actions from the history.

In general, the possible future executions in $pexecs(B, G, h)$ may achieve different subsets of G using different plans with differing costs and execution times. However, from the point of view of the agent, they are all equivalent. For example, the Mars rover agent may consider an execution that achieves the (single) delivering its current cargo of rocks, and an interleaving that achieves the goals of collecting several rocks, equally preferable. The choice of which goals in G to pursue is implicit in the choice of most preferred future execution, so depends on what courses of action are consistent with beliefs, and how prf is defined.

Recall that the definition of $traces$ generates an empty sequence ϵ in addition to traces for each expanded plan for a particular goal. This gives rise to interleavings where steps toward that particular goal have been omitted, so there is always an interleaving that allows to skip plans for goals that have already been achieved. Note also that the steps for the one instance of the redundant plan that is permitted may come from any of the traces where those steps were expanded, so there would be an interleaving that executes the plan once where the steps "belong" to a given trace for each of the traces where that plan could be executed.

4.2 GROVE Semantics

In this section, I give the operational semantics of GROVE in terms of a transition system. Each transition transforms one agent configuration into another, and corresponds to a single computation/execution step. I first define the configurations of a GROVE agent before presenting the transition rules.

An agent configuration is a 4-tuple $\langle B, G, h, f \rangle$ where $B \subseteq P$ is a set of beliefs, $G \subseteq D$ is a set of top-level goals, h is a history of steps executed so far and f is a *phase flag* from the set $\{s, m, a\}$.

4.2.1 Execution cycle

Each cycle of a GROVE agent consists of three phases: the belief update phase (s), the goal update phase (m), and the execution (a) phase.

Belief update phase

In the *belief update* (s) phase the agent's beliefs are updated based on sensory information to reflect changes resulting from the agent's most recently executed action and exogenous changes to the environment.

$$\frac{B' = \mathit{sense}(B)}{\langle B, G, h, s \rangle \rightarrow \langle B', G, h, m \rangle} \tag{4.1}$$

The function $\mathit{sense}(B)$ takes the agent's current beliefs B as an argument, and returns an updated set of beliefs $B' = \mathit{sense}(B)$ reflecting the environment state at this cycle. For the sake of simplicity in the semantics, the set of beliefs B is updated by sense rather than explicitly handling update of beliefs. This permits an agnostic attitude with respect to the specifics of updating beliefs.

Goal update phase

In the *goal update* (m) phase, the agent's goals are updated in response to requests from users or other agents to adopt or drop goals and when goals are achieved.

$$\frac{(G^+, G^-) = \mathit{mesg}(G) \quad G' = ((G \cup G^+) \setminus G^-) \setminus B}{\langle B, G, h, m \rangle \rightarrow \langle B, G', h, a \rangle} \tag{4.2}$$

The function, $\mathit{mesg}(G)$ takes the current goals G as an argument and returns a pair $(G^+, G^-) = \mathit{mesg}(G)$ consisting of the set of goals to be adopted, G^+, and the set of goals to be dropped, G^-. I stipulate that $G^+ \subseteq D$ and $G^+ \cap G^- = \emptyset$. The agent's updated goals for this cycle, G', are then given by $G' = ((G \cup G^+) \setminus G^-) \setminus B$.

The $mesg(G)$ function may also be used to model addition of goals in response to changes in the environment (or changes in beliefs), for instance top-level goals may be adopted in response to events generated by rules. For the sake of simplicity (and generality of the semantics) the top-level goals are updated based on the output of $mesg$ and achievement with respect to B, similar to the role of $sense$ in the belief update phase.

Execution phase

In the *execution* (a) phase, the set of possible future executions $pexecs(B', G', h)$ are (re)computed, the first step of a possible future execution is executed, and the history of executed actions is extended with the executed step.

$$\frac{\sigma \in pexecs(B, G, h) \quad \sigma = (A, e) \circ \sigma'}{\langle B, G, h, a\rangle \to \langle B, G, h \circ e, s\rangle} \tag{4.3}$$

$$\frac{pexecs(B, G, h) = \emptyset}{\langle B, G, h, a\rangle \to \langle B, G, h, s\rangle} \tag{4.4}$$

Note that, when an agent adopts one or more new goals, the possible future executions may or may not achieve the new goals in G^+ or the old goals in G, depending on the agent's preference relation prf. For example, one or more of the new goals in G^+ may not be jointly achievable with one or more goals in G, and the agent may prefer executions that achieve the goals in G. Conversely, the newly adopted goals G^+ may be of higher priority/more preferred than the goals in G. Similarly, when the agent drops one or more goals giving a new set of goals $G' \subseteq G$, the goals achieved by possible future executions in $pexecs(B, G', h)$ may or may not be a subset of the goals achieved by the executions in $pexecs(B, G, h)$. For example, if a high priority goal that is not jointly achievable with other goals is dropped, the agent may be able to pursue a larger number of goals.

The cycle then returns to the *belief update* (s) phase.

4.2.2 Commitment to Intentions in GROVE

The GROVE execution cycle has some similarities with the execution or deliberation cycles found in BDI architectures, however there are important differences. A BDI agent commits to a set of top-level goals (ends), which form the basis of the agent's intentions. The choice of how to achieve the agent's (sub)goals (means) is deferred for as long as possible, so that the most appropriate plan can be selected based on the current state of the environment. In contrast, a GROVE agent commits to a subset of *both* top-level goals and subgoals while deferring commitment to other top-level goals (and their subgoals). The current intentions of a GROVE agent are those top-level goals achieved in all possible future executions, $I = \bigcap\{agoals(\sigma) \mid \sigma \in pexecs(B, G, h)\} \cap G$. Goals achieved in some but not all possible future executions, i.e., $J = \bigcup\{agoals(\sigma) \mid \sigma \in pexecs(B, G, h)\} \cap G \setminus I$, are top-level goals the agent can achieve and to which the agent may commit in the future. While achievable, such goals may not remain so depending on which possible future execution the agent executes. Therefore, they reflect a lesser degree of commitment than to goals that are achieved in all possible future executions. As this depends on the state of the environment and its future evolution (reflected in B and coherence of future executions), commitment to intentions in GROVE is predicated on beliefs. If J is non-empty, either not all goals in G are jointly achievable or achieving them all is not preferred (e.g., achieving them all is too expensive).

Note that, while a GROVE agent has no firm commitment to goals in J, if the environment is static and the agent's goals are not updated (i.e., the agent's preferences over possible future executions do not change), its commitment to a set of achievable goals is stable. The set I grows monotonically as execution progresses (by addition of goals from J). Conversely, some goals from J will cease to be achievable, as the interleavings on which they are achieved no longer match the history, i.e., they are inconsistent with the course the agent has followed. For intuition, the top-level goals in I and J can be viewed as disjoint sets of necessary and potential goals respectively (analogous to necessary/definite and potential effects/resources in [105, 106, 108]).

In the case of a dynamic environment, I may change because some interleavings become non-executable or less preferred for some other reason (e.g., an interleaving achieving a higher value goal becomes executable), or because the agent's goals are updated. As a result, the history may contain steps from plans that were dropped, possibly even plans that were attempted and failed multiple times. The process of matching the history to the set of interleavings to derive the set of possible future executions involves 'masking' such redundant steps (otherwise the history will not match any interleaving). Note that the least number of steps in the history is not required to be masked, but it is reasonable to assume that interleavings that match more steps in history will be preferred, as they will involve executing fewer actions in the future. Essentially, a GROVE agent prefers to preserve progress because it means that less work must be done in the future than not doing so. Assuming that actions have a cost, and that, all other things being equal, an agent prefers future executions of lower cost, it is irrational for the agent to repeat steps already performed if these steps also occur in a prefix of an interleaving in $pexecs(B, G, h)$. Moreover, some steps may not be repeatable, e.g., if a step has consumed a non-renewable resource.

This behaviour of GROVE has two consequences: one is relative stability of commitments even in the dynamic case, and another is the ability to resume failed plans at the point where they failed.

4.3 Rationality of GROVE

In this section, I show that the agent executions specified by GROVE are rational. We show that GROVE conforms to the rationality postulates proposed by Grant et al. [36], and that, in a static environment, GROVE executions are a subset of those generated by the goal life-cycle semantics of Harland et al. [38], i.e., GROVE agents are 'more rational' than agents conforming to the Harland et al. model. I also consider the rationality properties proposed by Khan and Lespérance [54].

4.3.1 GKPW Rationality Postulates

I begin by briefly summarising the model of rationality proposed by Grant et al. [36] (GKPW) and comparing it to GROVE. GKPW define a high-level model of the mental state of BDI agents, called a BDI structure. A *BDI structure* S is a tuple $\langle B, D, I, v, (c, C)\rangle$, where B is a set of beliefs (all consequences of a finite belief base B_0), D is a set of declarative goals (in the same language as beliefs), I is a set of intentions (pairs (action, goal), with functions $goals(I)$ and $actions(I)$ returning respectively the set of goals occurring in I and the set of actions occurring in I), v is a function from sets of goals to non-negative real numbers representing the value of achieving a set of goals to the agent (satisfying the condition that a superset has at least the same value as its subset), $C \supseteq actions(I)$ and c is a function from subsets of C to non-negative real numbers representing the cost of executing this set of actions (c also satisfies the condition that a superset of a set of actions costs at least as much as the set of actions).

GKPW define five postulates on *rational* BDI structures:

A1 B is consistent, i.e., $B \nvdash \perp$

A2 I is feasible in the context of B (for every $(\alpha, \theta) \in I$, $B \vdash r_{\alpha,\theta}$, where $r_{\alpha,\theta}$ says that α's preconditions are true, and α terminates and makes θ true)

A3 $goals(I)$ is consistent

A4 For every $\theta \in goals(I)$, $B \nvdash \theta$

A5 There is no I' such that $S' = \langle B, D, I', v, (c, C)\rangle$ satisfies A1 - A4 and $ben(I') > ben(I)$, where $ben(I) = v(goals(I)) - c(actions(I))$; that is, there is no other set of intentions the agent can select which achieves more valuable goals by cheaper means.

BDI structures satisfying postulates A1 – A4 are referred to as *weakly rational* BDI structures (WRBDI) while structures satisfying A1 – A5 are *rational* BDI structures (RBDI). GKPW state several complexity results concerning WRBDI and RBDI structures, but give no algorithms for revising (W)RBDI structures in a rational way. Our work can be seen as a step towards providing a computationally grounded approach to this problem.

Lemma 4.2. *A GROVE agent is weakly rational in the sense of [36], i.e., satisfies postulates A1 - A4.*

Proof. A1 holds because beliefs are atomic, any belief is either in B or not, therefore B must be consistent (a belief cannot simultaneously be and not be in B). A2 holds because all actions in a possible future execution are executable (from Definition 4.1) and guaranteed to achieve the goal if executed. A3 holds because the active goals of a possible future execution are consistent (from the coherence of interleavings). A4 holds because achieved goals are dropped (Rule (4.2)) for top-level goals, and as a consequence of Definition 4.1 (non-redundancy). □

GROVE does not assume that a numerical value can be assigned to each set of goals or a cost to each set of actions. However, if this is possible, then these values can be used to derive a preference order on interleavings that chooses the optimal (in the sense of GKPW) set of interleavings for execution. The relative value of an execution is captured by a notion of "benefit" (adapted from [36]), derived from the sum value of the (top-level) goals it achieves minus the cost of the steps to achieve them.

Theorem 4.3. *A GROVE agent is rational in the sense of [36], i.e., satisfies postulates A1 - A5, if the preference relation prf gives an ordering by benefit $ben(\sigma) = v(agoals(\sigma) \cap G) - c(\sigma)$, where $v(G')$ is the value of achieving the top-level goals $G' \subseteq G$, and $c(\sigma)$ is the cost of executing the step sequence σ.*

Proof. A1-A4 hold by Lemma 4.2. A5 holds because the preference of each step sequence matches its benefit, and GROVE executes a maximally preferred step sequence, i.e., $\rho \in pexecs(B, G, h)$ is maximally preferred if $\neg\exists\sigma' \in pexecs(B, G, h)$ such that $prf(B, G, \sigma', \sigma)$, which is true by Transition (4.3) of the GROVE operational semantics and Definition 4.1. The maximally preferred step sequence ρ achieves the set of top-level goals G' corresponding to $agoals(\sigma) \cap G$, also following the GROVE operational semantics. □

While GROVE is weakly rational under the assumptions of Grant and Perlis, some of these assumptions are stronger than those made by the GROVE semantics. Particularly, actions in their model are assumed to be independent of each other (although possibly complex), as are goals. These assumptions preclude p-effects, such as within plans, and subgoal relationships between goals. We might consider what kind of rationality, in the sense of Grant and Perlis, that a GROVE agent would exhibit if these assumptions were relaxed. Relaxing the assumption of action independence corresponds to allowing intermediate steps toward a goal between adopting it and achieving it. Additionally, relaxing the assumption of goal independence corresponds to allowing subgoals within plans. In both of these cases, it is possible for an agent to fail to achieve a goal it has adopted (in a dynamic environment). In a static environment, GROVE agents remain weakly rational under these relaxed assumptions as the axioms A1-A4 are still satisfied, because the notions of goal and plan failure are irrelevant in a static environment. However in the case of a dynamic environment, while a GROVE agent cannot be guaranteed to achieve its intentions, it always adopts a set of feasible and consistent intentions. This gives a kind of 'weaker' rationality that is consistent with axioms A1, A3, and A4, while partially satisfying axiom A2. This is because a GROVE agent always executes (part of) an executable possible future execution in each execution cycle, although it is not required to follow that particular execution in later cycles. Therefore, in a dynamic environment a GROVE agent's intentions are feasible and achievable in each execution cycle, even if they are not guaranteed to be achieved.

4.3.2 HMTY Goal Life-Cycle Semantics

I now consider the goal life-cycle semantics of Harland et al. [38] (HMTY). I present the relevant parts of their semantics here before proving equivalence of the HMTY goal life-cycle model with GROVE under certain assumptions.

The HMTY semantics unifies previous work on goals of monitoring and accomplishment [28, 61, 79] and on aborting, resuming and suspending goals [100, 102, 103].

HMTY consider both achievement and maintenance goals. As GROVE currently does not encompass maintenance goals, I focus on achievement goals here. Each goal is assigned a state.

The goal states:

Pending Goal is inactive, awaiting further consideration. Can be either activated, suspended, or dropped.

Active Goal is being actively pursued by the agent, may have a plan (or several) associated with it. A plan must be assigned, or else the goal is dropped if no plan can be found.

Suspended Goal is paused, possibly with plans associated, awaiting either reconsideration or reactivation.

There are also Monitoring and Abort states, which I do not consider here as the former is exclusively for maintenance goals, and the latter is functionally equivalent to dropping the goal but allows for executing abort methods before dropping the goal.

Goal operations:

- $consider$. Adopt the goal. (Assign the Pending state)
- $activate$. Start pursuing plans for the goal. Goals must be activated in order for plans to be selected and executed (only Active goals can execute actions or initiate planning). (Transition to the Active state)

– $suspend$. Disallow execution of actions toward the goal until resumed. (Transition to the Suspended state)

– $reconsider$. Transition the goal to the Pending state, abandoning any plan it might have associated with it (and executing abort methods), if it had one prior to suspension.

– $reactivate$. Transition a suspended goal to the Active state and permit its associated plan to be re-assigned following execution of resume methods (goal must have a plan associated with it).

– $drop$. Drop/discard a goal, such as when achieved or failed (determined impossible). This includes dropping any plans associated with it.

A configuration in HMTY is a tuple $\langle \mathcal{B}, \mathcal{G} \rangle$ where $\mathcal{B}$ is a set of beliefs and $\mathcal{G}$ is a set of goal contexts of the form $\langle I, \text{ach}(\kappa, S, F), Rules, State, \pi \rangle$, where I is a goal context identifier, κ is a goal context condition, S is a success condition, F is a failure condition, $Rules$ is a set of condition-action pairs for goal update, $State$ is a state flag, and π is a plan body. If either of a goal's success condition S or failure condition F are true, the goal is dropped. As each condition-action pair is triggered by the state of the beliefs, state transitions are triggered by changes to beliefs. The state flag represents the current state of the goal in the agent's deliberation. Goals in the Pending state have no plan associated with them, and are not currently being executed. A Pending goal may be activated (transition to the Active state) as a consequence of deliberation if the context condition κ is true. Goals in the Active state must have a plan body associated with them, and are considered executable. Plans are assigned to goals by a means-end reasoning function, mer, which allows for both pre-written plans and online generation of plans.

The beliefs of the agent are updated in each cycle to reflect the goal state changes dictated by a *deliberation function*, by the addition of facts detailing the operation and identifier of the goal. For instance, adding the fact $activate(I)$ to the beliefs signals that the goal instance with identifier I should be activated. The semantics does not distinguish between operations that are triggered by these facts or by internal triggers. The decisions about which transitions to perform and when are primarily made by the deliberation function. The cases where internal triggers can cause goal state transitions are limited to when a subgoal is added as a consequence of executing a subgoal step in a plan (for achievement goals), and dropping a goal when the success or failure conditions are true (or no plans can be found for the goal). However, in the first case the activation of subgoals depends on the deliberation function regardless. The deliberation function is assumed to be consistent with the HMTY operational semantics, but is not further specified.

The execution cycle of an HMTY agent is made up of the repeated execution of three phases: goal update, plan update, and execution. This execution is not necessary cyclical, with goal transition rules taking precedence whenever they are applicable, and planning and execution rules being applied otherwise. The semantics is consequently given as three sets of transition rules: goal transition rules, planning rules, execution rules. These transition rules are based on CAN [4] transition rules.

The general form of a goal transition is as follows. Given the condition c, the action A takes achievement goal $g = \{I, \text{achieve}, Rules, S_1, P_1\}$ from state S_1 to state S_2 and plan P_1 to P_2.

$$\frac{\langle c, A\rangle \in Rules \quad B \models c}{\langle B, \mathcal{G} \cup \{I, \text{achieve}, Rules, S_1, P_1\}\rangle \longrightarrow \langle B, \mathcal{G} \cup \{I, \text{achieve}, Rules, S_2, P_2\}\rangle} \tag{4.5}$$

I follow the original semantics [38] in abbreviating the general form of a goal transition (4.5) where $g \in \mathcal{G}$ (and $A \neq \texttt{abort}$) as:

$$\frac{\langle c, A\rangle \in Rules \quad B \models c}{\langle I, \text{achieve}, Rules, S_1, P_1\rangle \longrightarrow \langle I, \text{achieve}, Rules, S_2, P_2\rangle} \tag{4.6}$$

The standard (common) set of parameterised rules for goals, denoted by $standard(I, Succ, Cond)$, is as follows:

$$\{\langle s, \texttt{drop}\rangle | s \in Succ\} \cup \tag{4.7}$$

$$\{\langle \text{drop}(I), \texttt{drop}\rangle, \langle \text{abort}(I), \texttt{abort}\rangle, \langle \text{suspend}(I), \texttt{suspend}\rangle, \tag{4.8}$$
$$\langle Cond \wedge \text{activate}(I), \text{activate}\rangle\} \cup$$

$$\{\langle \text{reactivate}(I), \texttt{reactivate}\rangle, \langle \text{reconsider}(I), \texttt{reconsider}\rangle\} \tag{4.9}$$

The initial state of an achievement goal g in HMTY is represented by a goal context $g = \langle I, \text{achieve}, Rules, \text{Pending}, \epsilon\rangle$, i.e., goals are initially in the Pending state with an empty plan.

The transition rule for goal activation:

$$\frac{\langle c, \texttt{activate}\rangle \in Rules \quad B \models c}{\langle I, \text{achieve}, Rules, \text{Pending}, \epsilon\rangle \longrightarrow \langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle} \tag{4.10}$$

The transition for dropping a goal:

$$\frac{\langle c, \texttt{drop}\rangle \in Rules \quad B \models c}{\langle B, \mathcal{G} \cup \{I, \text{achieve}, Rules, State, \pi\}\rangle \longrightarrow \langle B, \mathcal{G}\rangle} \tag{4.11}$$

[4]See [84, 101, 115] for a detailed description of the CAN operational semantics.

The transition for suspending a goal:

$$\frac{\langle c, \texttt{suspend}\rangle \in \mathit{Rules} \quad B \models c \quad \mathit{State} \in \{\text{Pending}, \text{Active}\}}{\langle I, \text{achieve}, \mathit{Rules}, \mathit{State}, \pi\rangle \longrightarrow \langle I, \text{achieve}, \mathit{Rules}, \text{Suspended}, \pi\rangle} \tag{4.12}$$

Adding a subgoal corresponds to executing a subgoal plan. The transition for adding a subgoal:

$$\frac{\mathit{stable}}{\begin{array}{c}\langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, \mathit{Rules}, \text{Active}, \mathit{SG}\rangle\}\rangle \qquad \longrightarrow \\ \langle B, \mathcal{G} \cup \{\langle I_P, \text{achieve}, \mathit{Rules}, \text{Active}, \mathit{SGP}\rangle\} \cup \ \{\langle I_C, \text{achieve}, \mathit{Rules}_1, \text{Pending}, \epsilon\rangle\}\rangle\end{array}} \tag{4.13}$$

where

- SGP is $S_C \vee F_C \vee \mathit{drop}(I_C)$:$?S_C$
- ':' denotes a guard condition (left-hand side) which must be true in order to execute the plan (right-hand side). The plan ceases progression without failure until the guard condition is true. The '?' prefix denotes a test action, such that if S_C is true the action succeeds, otherwise it fails.
- Rules_1 is $\mathit{standard}(I_C, \{S_C, F_C\}, \mathit{true}) \cup \{\langle \mathit{drop}(I_P), \texttt{drop}\rangle, \langle \mathit{suspend}(I_P), \texttt{suspend}\rangle\}$ $\cup \{\langle \mathit{reactivate}(I_P), \texttt{reactivate}\rangle, \langle \mathit{reconsider}(I_P), \texttt{reconsider}\rangle\}$
- stable denotes that no goal transition rules are applicable, i.e., planning rules and execution rules may be executed.
- The success and failure conditions of the subgoal SG are S_C and F_C respectively.

The SGP plan ensures the parent goal waits for the child goal to be dropped, failed, or achieved, while the extra rules added to *Rules₁* ensure that if the parent goal is dropped, suspended, reactivated, or reconsidered, the child goal is also dropped or suspended accordingly.

The transition for `reconsider`:

$$\frac{\langle c, \texttt{reconsider}\rangle \in \mathit{Rules} \quad B \models c}{\langle I, \text{achieve}, \mathit{Rules}, \text{Suspended}, \epsilon\rangle \longrightarrow \langle I, \text{achieve}, \mathit{Rules}, \text{Pending}, \epsilon\rangle} \tag{4.14}$$

The transition for `reactivate`:

$$\frac{\langle c, \texttt{reactivate}\rangle \in \mathit{Rules} \quad B \models c}{\langle I, \text{achieve}, \mathit{Rules}, \text{Suspended}, \pi\rangle \longrightarrow \langle I, \text{achieve}, \mathit{Rules}, \text{Active}, \epsilon\rangle} \tag{4.15}$$

For any goal in the Active state which has an empty plan, a plan must be found via means-end reasoning. If a plan cannot be found, the goal is dropped.

Planning rules:

$$\frac{stable \quad \Pi = mer(\text{achieve}, B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle\}) \quad \Pi \neq \epsilon}{\langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle\}\rangle \longrightarrow \langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \Pi\rangle\}\rangle} \tag{4.16}$$

$$\frac{stable \quad \Pi = mer(\text{achieve}, B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle\}) \quad \Pi = \epsilon}{\langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle\}\rangle \longrightarrow \langle B, \mathcal{G}\rangle} \tag{4.17}$$

$$\frac{\begin{array}{c} stable \quad \pi \neq \epsilon \\ \langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \pi\rangle\}\rangle \longrightarrow \langle B', \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \text{fail}\rangle\}\rangle \end{array}}{\langle B, \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \pi\rangle\}\rangle \longrightarrow \langle B', \mathcal{G} \cup \{\langle I, \text{achieve}, Rules, \text{Active}, \epsilon\rangle\}\rangle} \tag{4.18}$$

Rule 4.18 means that if a plan has failed it is replaced with the empty plan to allow re-planning.

The execution rules of HMTY are based on the standard CAN rules [84]. As these rules apply only to goals in the Active state (as I do not consider the Aborting state here), the rules are given in an abbreviated form that refers to a configuration by the beliefs and particular plan being executed.

Plan transition rules:

$$\frac{stable \quad \langle B, P_1\rangle \longrightarrow \langle B', P'\rangle}{\langle B, P_1; P_2\rangle \longrightarrow \langle B', P'; P_2\rangle} \tag{4.19}$$

$$\frac{stable}{\langle B, nil; P\rangle \longrightarrow \langle B, P\rangle} \tag{4.20}$$

$$\frac{stable}{\langle B, P; nil\rangle \longrightarrow \langle B, P\rangle} \tag{4.21}$$

$$\frac{stable}{\langle B, fail; P\rangle \longrightarrow \langle B, fail\rangle} \tag{4.22}$$

An HMTY agent's beliefs are updated by executing actions, although it is noted in [38] that this is merely a design choice and a more complex treatment of beliefs, such as sensory update, is possible in CAN.

I now show that under suitable assumptions an HMTY agent can produce the same execution as a GROVE agent.

Recall that the deliberation function is assumed to be consistent with the HMTY operational semantics but otherwise left unspecified. I address this issue in GROVE by defining rational deliberation in terms of preferences over possible future executions.

Theorem 4.4. *Let the environment be static and actions infallible. Then for any GROVE agent with initial configuration $\langle B, G, \epsilon, s\rangle$ and an execution history h, there is an HMTY agent with the same goals and plans which produces the same execution history.*

Proof. Consider a history h generated by the GROVE agent from initial configuration $\langle B, G, \epsilon, s\rangle$. Since the environment is static and actions are infallible, without loss of generality, I can assume that the agent is executing a single interleaving $h = \sigma_0$ such that $pexecs(B, G, \epsilon) = \{\sigma_0\}$.

For the sake of simplifying the proof, I assume that *expand* in Section 4.1.2 is implemented so that each step (A, e) is annotated with plans and goals respectively. In particular, each active goal $a \in A$ is annotated with the plan π_a that was selected to achieve a, and the action e is annotated with the goal g_e that it achieves. I denote an annotated goal by $g{:}\pi_g$ and an annotated action by $e{:}g_e$. The step annotations in σ_0 are used to determine what should happen in the HMTY agent. When the goals of an HMTY agent are activated and plans are assigned to them, the annotations on A inform the choice of plan for each goal. That is, if a goal g is activated then $mer = \pi_g$, such that $g{:}\pi_g \in A$. Subgoal steps $!g'$ require adding g' to $\mathcal{G}$ in the HMTY agent (I also omit some details to do with deliberation facts and subgoal plans in HMTY which do not present any complications to the argument).

I define a relation of *matching* between configurations of GROVE and HMTY agent, where in both configurations the next transition is executing an action. Instead of the initial configuration $\langle B, G, \epsilon, s\rangle$, I consider $\langle B, G, \epsilon, a\rangle$ reached from it by internal transitions.

The matching initial HMTY configuration is $\langle \mathcal{B}, \mathcal{G}\rangle$ where $\mathcal{B} = B$, and $\mathcal{G}$ contains goal contexts of the form $\langle I, \text{achievement}, standard(I, \{g\}, true), \text{Pending}, \epsilon\rangle$ for each $g \in G$. This means that the goals in $\mathcal{G}$ are achievement goals with tautological context conditions. Each goal context corresponds to a goal $g \in G$, which is thus dropped when g is believed (achieved). As the context conditions are tautological, these goal contexts can be activated by simply adding a deliberation fact activate(I) where I is the identifier of the goal context to be activated. We assume these deliberation facts are added as necessary to activate any Pending goals in $\mathcal{G}$.

For an arbitrary GROVE configuration $\langle B, G, h', a\rangle$ where a subhistory h' of h has been executed and $pexecs(B, G, h') = \{\sigma_{h'}\}$ is a suffix of σ_0, the matching HMTY configuration $\langle \mathcal{B}, \mathcal{G}\rangle$ corresponds to removing the actions in h' from the plans for goals and subgoals adopted so far; $\mathcal{B}$ is B with additional 'deliberation facts' which are records of goal adoption, and $\mathcal{G}$ corresponds to G plus subgoals g with currently executing plans π_g (the prefix of π_g is a sub-sequence of h' and the suffix is a sub-sequence of $\sigma_{h'}$).

In order to show that at any point in the history h, the HMTY agent can execute the same action as that executed by the GROVE agent, I need to show that in matching configurations the action in the first element of $\sigma_{h'}$ is executable by the HMTY agent, and the resulting configurations (after some internal transitions) again match. Employing an inductive strategy, I first demonstrate that an HMTY agent can match the first action of a GROVE agent when they have equivalent initial configurations. Secondly, I demonstrate that the HMTY agent can match the action chosen by the GROVE agent at each point in the execution history h' provided they have matched up to that point.

In the initial configuration, the action $e{:}g_e$ in the first element of σ_0 is the first action of a plan $e \circ \pi$ for some $g_e \in G$. It can be executed by the HMTY agent by assumption, since there is a goal $\langle I, \text{achieve}, \mathit{standard}(I, \{g\}, \text{true}), \text{Pending}, \epsilon\rangle$ in the HMTY configuration, which can be activated and assigned a plan $e \circ \pi$ to become $\langle I, \text{achieve}, \mathit{standard}(I, \{g\}, \text{true}), \text{Active}, e \circ \pi\rangle$. For the inductive step, we need to consider two cases for the action $e{:}g_e$ in the first element $(A, e{:}g_e)$ in $\sigma_{h'}$. The first case is when e belongs to a plan for $g_e \in G$ (a top-level goal), which is as in the initial configuration. The second case is when e is the first action in the suffix of a plan for a subgoal g_e. Since by the definition of a matching configuration, there is a goal of the form $\langle I, \text{achieve}, \mathit{standard}(I, \{g_e\}, \text{Active}, e \circ \pi\rangle$ in the HMTY configuration it can be chosen for execution, and in the resulting configuration there is a goal $\langle I, \text{achieve}, \mathit{standard}(I, \{g_e\}, \text{true}), \text{Active}, \pi\rangle$ so the HMTY configuration again matches the GROVE configuration corresponding to $\sigma_{h' \circ e}$. □

The converse of Theorem 4.4 is not the case; there are executions permitted by HMTY that are not possible in GROVE. I give an example of such an irrational (from GROVE point of view) execution in the proof of the theorem below.

Theorem 4.5. *Let the environment be static and actions infallible. There exists an execution h of an HMTY agent with initial beliefs B and goals G that cannot be generated by a GROVE agent with initial configuration $\langle B, G, \epsilon, s\rangle$.*

Proof. The behaviour of HMTY that GROVE cannot reproduce is caused by the fact that HMTY does not check for coherence of plans, while GROVE does. Consider the following example. Suppose both agents have two goals, g_1 and g_2 with plans π_1 for g_1 and π_2 for g_2, and both π_1 and π_2 are executable given current beliefs B. Suppose the HMTY agent's deliberation function implements serial execution of goals, i.e., π_1 is executed first, and one of the actions in π_1 makes one of the actions in π_2 unexecutable. Then there is an execution of HMTY agent $h = \pi_1 \circ \pi_2'$, where π_2' is a prefix of π_2, for which there is no corresponding GROVE execution, since $\pi_1 \circ \pi_2 \notin \mathit{inters}(\{g_1, g_2\})$ since it is not coherent. □

The analogue of Theorem 4.4 does not hold for dynamic environments, where plans may stop being executable because of the environment changing.

Theorem 4.6. *Let the environment be dynamic and actions infallible. There exists an execution h of a GROVE agent with initial configuration $\langle B, G, \epsilon, s\rangle$ that cannot be generated by an HMTY agent with beliefs B and goals G.*

Proof. Consider the following example, which demonstrates the difference between the GROVE and HMTY approaches to 'backtracking' when a plan fails. Assume both agents have a goal g with a single applicable plan e_1, e_2, and that both agents executes the first action e_1, and some environmental event makes e_2 unexecutable. As a result, the remainder of the plan becomes non-executable, so the parent goal g becomes non-achievable. In this case, the HMTY agent would drop the goal, but would potentially re-adopt it later if it becomes achievable again, and start executing the plan from the beginning. It is not possible for the HMTY agent to resume a plan for a dropped goal from the point where it was abandoned. Meanwhile, a GROVE agent never 'drops' goals in this sense (progress remains accessible on the history) and can therefore resume executing the plan when it later becomes executable. Rather than dropping a goal, a GROVE agent selects actions to execute from a most preferred interleaving that matches (a projection of) its history. Since the history contains e_1, it may select an future execution that contains e_2 as the next step. □

I argue that the ability to pick up execution of a plan at the point where it was dropped previously is a useful and rational behaviour (provided the environment is amenable to it). This is the only behaviour of GROVE that a HMTY agent cannot match in a dynamic environment. In fact, I can prove an analogue of Theorem 4.4 for dynamic environments, *provided all plans consist of a single action.*

Lemma 4.7. *Let the environment be dynamic, actions infallible, and plans contain only single steps. Then a GROVE agent with initial configuration $\langle B, G, \epsilon, s\rangle$ yields an execution history h corresponding to an interleaving of top-level goals $G' \subseteq G$, i.e., $h \in inters(G')$*

Proof. The assumption that plans are single steps implies that the traces induced by G are also single steps. This follows from the single step in each plan either being a subgoal or an action, which gives at most one action per trace.

In the initial state, if there is a most preferred execution available then the agent executes its first step e. The execution of e corresponds to the complete execution of a single-step trace, which achieves one or more top-level goals G'' (actions are infallible). The step e is added to the agent's history. Any interleaving suffixes that match with the prefix corresponding to e are themselves interleavings of traces for some set of top-level goals G''' where $G''' \subset (G \setminus G'')$. Therefore any possible future execution of the agent in the state following the initial state is an interleaving of traces induced by G'''.

For the state $\langle B, G, h', s\rangle$ where h' is a subhistory of h, h' is the sequence of steps executed so far, which corresponds to an interleaving of (single-step) traces for the set of top-level goals achieved so far. Executing a step e' of a future execution in that state extends h'. The resulting history, $h' \circ e'$, is also an interleaving of h' (an interleaving of single-step traces) and e' which is a single-step trace.

Therefore each execution cycle results in a history that is an interleaving of single-step traces, and any future executions are also interleavings of single-step traces. The concatenation of an interleaving of single-step traces with another is also an interleaving of single-step traces. The complete execution h of the agent corresponds to an interleaving of single-step traces for a subset of top-level goals of the agent.

□

The Lemma 4.7 establishes that, under the assumption that plans are single steps, the execution of a GROVE agent is an interleaving of the single-step traces induced by the top-level goals G. This means that equivalence with an HMTY agent can be established in the same manner as in Theorem 4.4, assuming that deliberation facts are available.

Theorem 4.8. *Let the environment be dynamic and actions infallible, and plans contain only single steps. Then for any GROVE agent with initial configuration $\langle B, G, \epsilon, s \rangle$ and an execution history h, there is an HMTY agent with the same goals and plans which produces the same execution history.*

Proof. By Lemma 4.7, the execution h is an interleaving $h \in inters(G')$ where $G' \subseteq G$. In the initial state, the GROVE agent executes an action e, which is matched by the HMTY agent following the strategy in Theorem 4.4. The HMTY agent selects the correct plans and instantiates any subgoals as necessary until it reaches a plan containing e, by using the available deliberation facts. After executing e, both agents have the same set of top-level goals $G''' \subset G \setminus G''$ where G'' is the set of top-level goals achieved by e. This is because executing an external action achieves one or more top-level goals (G'') which are dropped by both agents when achieved (following their operational semantics).

In any successor state, the GROVE agent executes the first step e' of a possible future execution that achieves a subset of the remaining top-level goals. The HMTY agent matches this as in the initial state and any achieved goals are dropped by both agents, resulting in both agents having the same set of top-level goals.

□

4.3.3 KL Rationality Postulates

Lastly, we consider the rationality properties proposed by Khan and Lespérance [54] (KL). In [54] KL prove three rationality properties for a Simple Rational APL (SR-APL) with prioritised goals. The first property states that the agent's beliefs (knowledge) and chosen goals are internally consistent, with respect to the domain theory $\mathcal{D}$ which describes the world and also the agent's declarative and procedural goals and their dynamics. This is similar to postulates A1 and A3 of Grant et al. [36]. The other two properties hold in a static environment and essentially state that any action performed by the agent is consistent with the agent's intentions (with respect to the theory $\mathcal{D}_{\bar{Exo}}$ stating that there are no exogenous actions). These two properties trivially hold for GROVE, since any actions executed by GROVE come from plans for the goals of the agent, as part of executing an interleaving which is coherent and consistent with beliefs.

5 Bounded GROVE

The model of rational intention revision presented in Chapter 4 selects a maximally preferred, coherent interleaving to execute. While executing a maximally preferred interleaving is rational, it assumes that the set of interleavings corresponding to the agent's currents adopted goals is available. However, generating the set of (most preferred) interleavings may not be feasible in many real-world scenarios where computational resources are limited and/or the agent's goals change frequently. In this section, I propose a bounded version of GROVE that samples the set of future executions, and I state conditions under which bounded GROVE commits to a bounded rational execution.

While the unbounded version of GROVE in Chapter 4 defined rational behaviour of an idealised agent, further assumptions must be made in order to make the model realisable for practical, real-life agents.

Firstly, I stipulate that the traces for the agent's top-level goals are induced by goal-plan trees, and are static and can be computed offline. This requirement ensures that the agent has access to the full set of traces for each of its top-level goals in constant time. A similar assumption is made in the summary information work of Thangarajah et al. [105, 106, 108], where the summary information derived from goal-plan trees is assumed to be computed offline and kept up-to-date during execution, and the goal-plan trees are static. Their summary information may seem similar in purpose to traces here, however the corresponding aspects of GROVE that are updated at runtime are the history of the agent and the set of top-level goals, which are separate from the traces. As traces are computed offline, the bounding of the agent does not need to account for steps required to compute traces, which are effectively a static component of the agent. A consequence of this assumption is that when adding a new top-level goal, the traces are already available and therefore adding a new top-level goal entails a constant amount of computation, and does not affect the time complexity of the model overall.

Secondly, I assume that the set of executable traces can be computed in linear time from the agent's beliefs B given suitable indexing. This is possible by computing the preconditions of traces offline, and checking their consistency with beliefs at run-time. An alternative approach would be to compute the probability that a randomly selected trace is executable using a measure of plan coverage for goals as in [107]. Furthermore, I assume that all incoherent interleavings have the same minimal preference (e.g., 0). This is a stronger assumption than in unbounded GROVE (see 4.1.2), where I simply assumed that all executable step sequences are preferred to all non-executable sequences, but seems reasonable. An incoherent interleaving cannot be successfully executed, and it seems reasonable to discount any utility that may accrue from a partial execution prior to failure. Second, I assume that the distribution of preferences over coherent interleavings follows a normal distribution. Following this assumption, there will be a few very bad/good interleavings, and a much larger number with around average preference. Note that although the preference distribution of interleavings is domain-dependent, if it is known then the proportions of interleavings above and below a given preference bound can be determined. Here the normal distribution is chosen for the sake of simplicity in demonstrating the strategy in this chapter for bounding GROVE.

Finally, in the bounded model (Chapter 4), I assumed that at each execution cycle a GROVE agent has an unbounded amount of computation available to select the next action to execute. Here I will instead assume that the computation available to select the next action is bounded.

5.1 Bounded search for an ϵ-preferred interleaving

Bounded computation constitutes a particular type of resource-bounding. All realistic agents are necessarily bounded in this way, as explained by Bratman [11]: "for real agents it takes time to do such computations - the more complicated they are, the more time it takes.", and "All this must be done in a way that recognizes the fact that agents [humans or robots], are resource bounded: they are unable to perform arbitrarily large computations in constant time [as pointed out by Herbert Simon (1957)]". Thus, a computationally bounded agent is limited in how much reasoning it can do before it must act, and therefore must use its limited resources efficiently in order to maximise its efficacy.

An ideal or computationally unbounded agent (as in 4) can be characterised as searching an arbitrarily large solution-space to find an optimal solution. Therefore bounding GROVE implies identifying this solution-space, bounding the search, and revisiting our assumptions about the quality of solution that can be found given the limited search.

The search can be bounded in two ways: by restricting the solution-space that is to be searched (limiting the number of candidate solutions to consider), and by weakening what qualifies as an optimal solution (increasing the number of candidate optimal solutions).

The former corresponds to sampling the solution-space, rather than exhaustively searching it, while the latter corresponds to accepting a solution that is "good enough" with respect to some minimum acceptable preference bound. The minimum acceptable preference bound forms an interval that contains the optimal solution (maximal preference solution that an unbounded model would find), but does not insist upon it. Note that both of these approaches decrease the number of solutions that must be considered.

Let us suppose a bounded agent has a limited number of future steps it can consider before acting. The agent uses this quota to randomly sample the set of interleavings, before executing the first step of a most preferred interleaving in the sample.

Bounding the number of future steps the agent considers in each execution cycle is similar to approaches that use a notion of bounded lookahead horizon (see [43], where a lookahead horizon is defined for each goal and used to ensure maintenance conditions will not be violated). However while those approaches bound how far the agent can lookahead with respect to each goal, here the total number of steps examined is limited for a set of goals. This allows to strongly restrict the number of steps considered for a set of goals, while the number of steps considered in alternative approaches grows with the number of goals. A step budget puts a fixed and precise bound on computation, whereas the bound given by a fixed lookahead horizon (considering a fixed number of steps *in each* possible execution) depends on factors like the number of goals and plans to consider.

For the sake of providing intuition, we can view the set of possible future executions of a bounded GROVE agent, analogous to that of an unbounded GROVE agent from Chapter 4, as defined in terms of a randomly generated set of candidate interleavings J composed of randomly generated traces for a randomly selected subset of goals in G. The candidate interleavings J are assumed to be generated within a bound b, i.e., they are a set of interleavings that potentially achieve a subset of G and the sum of lengths of interleavings in J is less than or equal to b. The set of possible future executions for a bounded GROVE agent is then given by:

$$\begin{aligned} pexecs_b(B,G,h) \; = \{\sigma \mid \exists E\; h_E \circ \sigma \in J \;\wedge \\ \neg\exists E', \sigma'\; h_{E'} \circ \sigma' \in J \;\wedge \\ prf(B,G,\sigma',\sigma)\} \end{aligned}$$

where E is a projection on the history h to derive a prefix. The agent executes the first action of an interleaving $\rho_b \in I_b = pexecs_b(B,G,h)$. Such a model can be seen as approximating an agent that does a bounded amount of lookahead before selecting a next action to execute. Clearly such randomly generated interleavings are not guaranteed to be either coherent, executable or most preferred. However, we can compute the probability that at each cycle an agent with a given computation bound executes an action from an interleaving that is within a preference bound of a most preferred interleaving.

5.2 ϵ-PREFERRED INTERLEAVINGS

Since the most preferred interleaving is unknown prior to searching, the upper preference bound of the interleaving set is also unknown. In order to determine that an interleaving is most preferred requires exhaustively generating the set of interleavings. This exhaustive search is closely related to the complexity results regarding identification of a maximally beneficial intention set in the model of rational BDI revision proposed by Grant et al. [36]. Grant et al. prove that determining whether an intention set is maximally beneficial is NP-hard and the proof employs exhaustive search of possible sets of intentions. As an interleaving corresponds to a set of intentions, and preference corresponds to benefit, identifying a most preferred interleaving corresponds to identifying a maximally beneficial intention set.

Rather than requiring that a rational agent executes a most preferred interleaving, let us instead require that it executes an interleaving that is *at least ϵ-preferred*. An interleaving is ϵ-preferred if it is at least as preferable as a hypothetical interleaving corresponding to a lower preference bound, ϵ. This means that a proportion of the set of interleavings is considered "acceptable" for execution by a rational agent with respect to ϵ. Thus the requirement of a most preferred interleaving is relaxed to that of ϵ-preferred interleaving, of which there may be many depending on the proportion of the interleavings that are within the bound defined by ϵ. Note that ϵ need not be a numeric value, and a bounded GROVE agent does not require access to or knowledge of an ϵ-preferred interleaving. The ϵ-preferred interleaving is (potentially) hypothetical and used only to identify the proportion of possible interleavings that are "acceptable", and consequently determines the probabilistic success of a bounded GROVE agent, under the bound b. This is explained in more detail below.

The set of ϵ-preferred interleavings for a set of goals G given beliefs B is defined by

$$I_\epsilon = \{\rho \mid \rho \in \mathit{inters}(G) \wedge \mathit{prf}(B, G, \rho, \rho_\epsilon)\}$$

where ρ_ϵ is an interleaving with preference ϵ. Recall that I assume that interleavings are executable, so assume ρ_ϵ is executable.

The definition of I_ϵ is a straightforward restriction of the definition of $\mathit{inters}(G)$ from Chapter 4 to require that interleavings in I_ϵ are at least as preferable as the hypothetical interleaving ρ_ϵ.

I refer to the set of all possible interleavings, corresponding to $\mathit{inters}(G)$, as I^*. Note that I^* includes incoherent and non-executable interleavings. Given I^*, the probability of randomly generating an ϵ-preferred interleaving is equivalent to the proportion of ϵ-preferred interleavings in I^*, which is denoted $|I_\epsilon|/|I^*|$.

5.2.1 Coherency of interleavings

In order to determine $|I_\epsilon|/|I^*|$, we must first determine the number of coherent interleavings in I^*. The coherent interleavings are those without conflicts, therefore I make some assumptions about the possibility of conflicts between traces in order to model the proportion of coherent interleavings.

Average number of steps and traces

Let us assume that the properties of an agent's goal-plan trees are described by a 4-tuple $(\tau_g, \tau_p, \tau_a, \tau_d)$, where τ_g is the average number of subgoals in each plan, τ_p is the average number of plans for each goal, τ_a is the average number of (non-subgoal) steps in each plan, and τ_d is the average depth of a tree, depth is indexed from zero. For example, a tree with $\tau_d = 0$ corresponds to a single root goal with a set of τ_p leaf plans, each containing τ_a actions.

The average number of traces induced by a goal-plan tree τ is then:

$$\tau_p^{\tau_d \tau_g + 1}$$

Traces correspond to paths through a goal-plan tree. Any path through a tree described by $(\tau_g, \tau_p, \tau_a, \tau_d)$ can be characterised by a series of goal-plan choices, i.e., plan selection. For any goal there are τ_p plans that could be selected, and the number of goals that are achieved is determined by the number of goals per plan τ_g and the depth of the tree τ_d. For a goal-plan tree consisting of a top-level goal with only leaf plans, i.e., $\tau_d = 0$ the number of traces is exactly τ_p (simplification of $(\tau_p)^1$). Intuitively the depth τ_d corresponds to the number of layers in the tree which contain subgoals. For a plan with τ_g subgoals, each with τ_p leaf plans, there are $\tau_g \tau_p$ traces that represent completions of the plan. For a goal with plans that have subgoals with leaf plans, there are then $\tau_g + 1$ goals being achieved in each trace. The number of traces in that case is then $\tau_p^{\tau_g + 1}$ as each trace corresponds to a choice of plan for each goal, each plan contains τ_g subgoals, and there is a single top-level goal for which a plan is chosen, corresponding to selecting a plan $\tau_g + 1$ times, giving $\tau_p^{\tau_g + 1}$ possible combinations.

The number of subgoals achieved in a trace increases by τ_g for each additional layer in the goal-plan tree, i.e., it increases with the depth of the tree. Then, the exponent $\tau_d \tau_g$ corresponds to the number of subgoals achieved in a trace. If the agent has only one possible plan for its subgoals, then the number of traces is simply $\tau_d \tau_g$. To account for the plan choices for each subgoal we apply the exponent to τ_p, the number of plans for each goal.

The number of steps in each trace is given by:

$$\tau_a(\tau_d \tau_g + 1)$$

Note that the number of goals achieved by a trace is $\tau_d \tau_g + 1$. Each layer of subgoals adds τ_g additional goals achieved to a trace, and for each additional goal τ_a steps are introduced. The number of subgoal layers is defined by τ_d. An additional τ_a steps are added to account for the plan for the top-level goal.

Number of possible interleavings

The length of an interleaving of a set of traces $T = \{\sigma_1, \ldots, \sigma_n\}$ is:

$$\sum_{i=1}^{n} \|\sigma_j\|$$

which is just the sum of the lengths of the traces in T.

For a trace σ_i, the number of possible positions it can be fitted into the sequence of length $\sum_{i=1}^{n} \|\sigma_j\|$ are[1] [2] :

$$\binom{\sum_{j=1}^{n} \|\sigma_j\|}{\|\sigma_i\|}$$

That is, for each trace its steps are placed into the step sequence until all steps of all traces in T have been allocated positions in the step sequence and the interleaving is complete.

Then the number of interleavings for a set of traces T, $ninters(T)$ is:

$$ninters(T) = \prod_{i=1}^{|T|} \binom{\sum_{j=i}^{|T|} \|\sigma_j\|}{\|\sigma_i\|}$$

The intuition for this is that an interleaving of two traces must be a step sequence that is the length of both combined, and each step in either trace is assigned a unique index corresponding to a position in the interleaving. However, once a step is designated an index, the following step from that trace cannot occur earlier in the interleaving, as the ordering of steps in traces is preserved. This naturally extends to more than two traces.

Number of coherent interleavings

An interleaving is incoherent if a dependency (p-effect) between steps is not protected, resulting in the preconditions of a step established by an earlier step(s) being undone by an intervening step or steps.

My aim is to model conflicts abstractly in terms of properties of goal-plan trees, in order to establish the proportion of coherent interleavings. The proportion of coherent interleavings represents the probability of an interleaving being coherent when traces are interleaved. If traces do not use the same resources, or have overlapping preconditions or postconditions at any point, then they cannot interact and thus any interleaving of them will be coherent.

[1] I use $|S|$ to denote the cardinality of a set S and $\|\tau\|$ to denote the length of a trace τ.

[2] This is a generalisation to multiple sequences of the formula for the complexity of merging or shuffling two sequences in [59].

Similarly, if traces require the same resources, or have overlapping preconditions or post-conditions at any point, then their interleavings may be incoherent in the case of negative interaction.

One approach to determining whether traces interact is to use summary information [105, 108]. Summary information allows to determine the possible and necessary resources and effects of goals, and thus whether those goals necessarily or possibly conflict. If goals necessarily conflict, then there is no coherent interleaving (scheduling in [105]) of their traces. On the other hand, if goals possibly conflict, then this implies that some interleavings will be incoherent while others are coherent.

Another approach is to use the notion of plan compatibility from Horty et al. [47], where p-effects are modelled as causal dependency links in plans, which can be threatened under some schedules if plans are incompatible. Their approach identifies the cases where plans can be merged without compromising dependency links, in order to achieve a lower overall cost. However they do not consider goals, and plans are assumed to be complete.

The existing approaches consider only how to detect possible interactions between goals, i.e., qualitatively determine possibility of interaction, rather than quantitatively computing the likelihood that a conflict occurs. One way to do this might be enumerate the schedulings of goals that possibly conflict, and those where no conflict occurs. However that corresponds to an exhaustive search of interleavings which is to be avoided. It may be possible to identify the critical sections in the traces of a real agent, and use those to derive a model of the coherent and incoherent interleavings of an agent based on its goal-plan trees. Such an approach would allow building a probabilistic (domain-specific) model of the interactions between traces when interleaved. Although that approach may be practically useful, the modelling I propose here assumes random distribution of critical sections in interleavings for simplicity of exposition.

Here I adopt a simple approach to modelling conflicts, where p-effects are modelled by critical sections. The approach is based on enumerating the possible interleavings and of those the coherent interleavings, and determining the proportion of possible interleavings that are coherent. The main advantage of this approach is that it does not stipulate exhaustively generating all possible interleavings of a set of traces in order to determine the proportions of coherent and incoherent interleavings, and it permits precise non-probabilistic formulation of these proportions.

I stipulate that a conflict occurs if a *critical section* in a trace σ_i overlaps with a critical section in a trace σ_j. Intuitively, incoherent interleavings are those where critical sections are interleaved. I assume that each critical section consists of a single *critical step* from a set C of critical steps. Critical steps in C are randomly distributed throughout the traces in T, and an interleaving is coherent iff it contains no adjacent critical steps, i.e., there are no continguous blocks of two or more steps from C in the interleaving. The number of critical steps in C is assumed to be proportional to the sum of the length of traces in T (so that average length of traces does not affect the proportion of incoherent interleavings). Note that randomly distributing critical steps throughout traces is equivalent to assuming that a critical step occurs on average every so many steps in an interleaving.

The strategy for computing the proportion of incoherent interleavings can then be summarised as follows. Given a set of coherent traces (recall that traces are computed offline and assumed to be coherent), when the traces are interleaved with a set of critical steps the incoherent interleavings are those where critical steps are adjacent. Determing the proportion of coherent interleavings in the set, i.e., those where conflict steps are adjacent, equates to determining the probability of a randomly chosen interleaving from the set being coherent. The probability that a randomly generated interleaving of T is coherent is determined by the proportion of interleavings of T that are coherent for a set of critical steps C.

First, note that any interleaving containing all of the critical steps in C (once) is equivalent to interleaving a set of traces with a step sequence corresponding to an ordering of C. Interleaving the traces in T with each of the critical step sequences induced by C gives the set of coherent and incoherent interleavings of traces in T with critical steps C. More precisely, consider a single step sequence σ_c corresponding to an arbitrary order of the critical steps in C. Then the set of step sequences to interleave is $T' = T \cup \{\sigma_c\}$.

Deriving the number of coherent and incoherent interleavings of T' equates to counting the number of interleavings of T', given by $ninters(T')$ and multiplying it to account for possible configurations of σ_c. We multiply by $|C|!$ as this is the number of possible orderings of C. Then, the number of possible interleavings N, including both coherent and incoherent interleavings, is:

$$N = ninters(T')|C|!$$

Note that $|C|$ must be less than the combined length of traces in T, otherwise only incoherent interleavings are produced. This is because when $|C|$ is equal to or greater than the combined length of traces, it is impossible to avoid critical steps being adjacent to one another.

In order to determine the number of coherent interleavings of T', it is useful to make several observations about the form that any coherent interleavings in T' take.

To begin with, observe that any coherent interleaving of T' can be viewed as an interleaving of T interleaved with some σ_c, such that every step in σ_c is either separated by steps from T, or occurs as the first or last step. Intuitively, there must be at least one step from T between each consecutive critical step, and there may or may not be steps from T preceding the first critical step, or following the last. This follows from the requirement that the critical steps in a coherent interleaving are not adjacent.

For example, consider the case where $\sigma_c = c_1, \ldots, c_k$, and for simplicity T is represented by a step sequence $t_1, \ldots, t_n$. The general form of a coherent interleaving of σ_c and T in this case is $t_1, \ldots, t_n$ with c_1 and c_k inserted such that they are not adjacent.

Note that only when c_1 occurs before t_1, c_k occurs after t_n, or both of these is true, is it possible that a critical step has steps from T on only one side of it in a coherent interleaving. Otherwise, each critical step is required to have steps from T either side of it.

Then the problem of determining how many valid arrangements meet these criteria is equivalent to the classic combinatorics problem of determining how many ways there are to distribute a number of indistinguishable items between distinguishable bins such that every bin has at least one item (see Chapter 2 of [30]). Then any coherent interleaving of T' corresponds to a surjective mapping of steps from T to the *gaps* between critical steps of σ_c, such that each gap contains at least one step from T in order to meet the requirement that critical steps are not adjacent.

The number of such mappings is:

$$\binom{|T| - 1}{\|\sigma_c\|}$$

where $\binom{n}{k}$ denotes binomial choice, i.e., n choose k. Here $n-1$ (as in $\binom{n-1}{k-1}$) becomes $|T|-1$ because of the number of spaces between $|T|$-many steps that can be occupied by critical steps. The $k-1$ (as in $\binom{n-1}{k-1}$) becomes $\|\sigma_c\|$ as this corresponds to the number of critical steps that must be placed. Thus the number of mappings corresponds to the number of ways to assign critical steps to spaces between steps from traces.

We must also account for the cases where no steps from T occur before the first critical step or after the last. In either of those two cases the number of mappings is:

$$\binom{|T| - 1}{\|\sigma_c\| - 1}$$

The intuition for this is that one "gap" (at either end) is left empty and does not need to be filled (and thus exactly one critical step is not "between" trace steps, only adjacent to them at one end).

This corresponds to the case where either the first step of a coherent interleaving is a critical step, or the last step is a critical step. Note that since both of these cases are symmetrical, we double this number in the final total.

The final case to consider is where both gaps at each end are empty:

$$\binom{|T| - 1}{\|\sigma_c\| - 2}$$

This corresponds to the case where the first and last steps of a coherent interleaving are both critical steps (and thus exactly two critical steps are not "between" trace steps, only adjacent to them at either end).

Then, the number of coherent interleavings of a set of traces T and a step sequence σ_c is the sum of the numbers of coherent interleavings in each of these cases:

$$\binom{|T|-1}{\|\sigma_c\|} + 2\binom{|T|-1}{\|\sigma_c\|-1} + \binom{|T|-1}{\|\sigma_c\|-2}$$

By application of Pascal's rule this reduces to:

$$\binom{|T|+1}{\|\sigma_c\|}$$

Finally, we multiply this by the orderings in C to derive N_c, the number of coherent interleavings of a set of traces T and a set of critical steps C (as opposed to just σ_c, which is one ordering of C):

$$N_c = |C|!\binom{|T|+1}{|C|}$$

The proportion of coherent interleavings is then N_c/N.

5.2.2 Preference Distribution

As the preference of interleavings is assumed to be normally distributed, we can relate ϵ-preference to an interval of a normal distribution, i.e., ϵ-preference is a lower bound on an interval that captures acceptably preferred interleavings. The proportion of interleavings within the interval then corresponds to the probability of randomly sampling an ϵ-preferred interleaving.

A normal distribution is represented by a tuple (μ, σ) where μ is the mean and σ is the standard deviation. The preference distribution determines the range of possible values for ϵ. ϵ determines which interleavings are sufficiently preferable, and depends on the distribution of preference because it must be set at an appropriate value to capture a high enough proportion of interleavings in order to find an ϵ-preferred interleaving within b simulated steps.

In order to abstract ϵ-preference from absolute values, we take ϵ to be a function of the preference distribution e.g., the 95^{th} percentile.

Finally, the probability p is given by:

$$p = p_\epsilon N_c/N$$

where p_ϵ is the proportion of the preference distribution with lower bound ϵ.

5.3 Probability of an ϵ-preferred interleaving

I now prove that at least one coherent ϵ-preferred interleaving can be found for a set of goal-plan trees G within the bound b with probability $1 - (1-p)^{b/l}$. The intuition for this is that the probability that at least one coherent ϵ-preferred interleaving is found within a bound b can be computed by subtracting the probability of finding only incoherent or sub-ϵ-preferred interleaving, which is $(1-p)^{b/l}$, from one. The bound b permits generating on average b/l interleavings, as l is the average length of an interleaving and b is a bound on steps generated. Thus, at least one coherent ϵ-preferred interleaving is found within bound b with confidence $1 - (1-p)^{b/l}$.

Theorem 5.1. *The set of goals G induces a set of goal-plan trees following the assumptions established in Section 5.2.1. Due to my simplifying assumption that the traces induced by a goal-plan tree have uniform length, we can refer to the average length of an interleaving for traces induced by G trees as l. Given a bound b on the number of simulated steps that are considered per execution cycle, a bounded GROVE agent $\langle B, G, \epsilon, s\rangle$ finds an ϵ-preferred interleaving with confidence $1 - (1-p)^{b/l}$, where $p = p_\epsilon N_c/N$ where p_ϵ is the proportion of preferences above the threshold ϵ, under the assumption of a normal preference distribution (μ, σ).*

Proof. The set of goals G induces a set of goal-plan trees. I assume that conflicts (and thus coherency) of interleavings can be modelled by interleaving of traces with a set of critical steps C. I further assume that the cardinality of the set of critical steps C for some set of traces T is proportional to l, such that $|C|$ is less than the combined length of induced traces in T, and can then be treated as a property of T and thus omit C without loss of generality. Then, for a set T of randomly selected traces such that each $\sigma_g \in T$ corresponds to a goal $g \in G$, i.e., there is an injective partial mapping of top-level goals to traces in T.

We assume a fixed budget of steps b may be simulated per execution cycle, which puts an upper bound on the number of interleavings that can be generated during the search. The number n of interleavings this permits to be generated given b is $n = b/l$ where l is the average length of interleavings. We treat generation of interleavings as sampling with replacement.

The probability of generating an incoherent or sub-ϵ-preferred interleaving is $1-p$ (inverse of p). The probability of generating only incoherent or sub-ϵ-preferred interleavings in b/l simulations is $(1-p)^{b/l}$. Thus the inverse of this is the probability of generating at least one coherent and ϵ-preferred interleaving.

Then, the probability that of b/l randomly generated interleavings at least one is ϵ-preferred, is given by:

$$1 - (1-p)^{b/l}$$

where p is the probability of randomly generating an ϵ-preferred interleaving. $\square$

The practical application of this result is that by adjusting the bound b, the probability of finding a coherent ϵ-preferred interleaving can be adjusted. Turning this around, given a desired probability of finding a coherent ϵ-preferred interleaving, the necessary bound can be computed.

5.3.1 Stability

I now consider the question of stability of a bounded GROVE agent. The simplest interpretation of an execution cycle for a bounded GROVE agent is to search for an executable interleaving and execute the most preferred in each execution cycle. This results in execution of an epsilon-preferred interleaving in each execution cycle with confidence defined by Theorem 5.1. However, this interpretation implies that the agent essentially discards the results of searches in previous cycles, stipulating only that it executes the best executable interleaving that it generates in each cycle. Under this interpretation, an agent may partially execute an interleaving with preference that is *lower* than those that resulted from earlier searches, as the most preferred interleaving among those generated in the *current* cycle is executed. resulting from the search in the current execution cycle. This differs from the behaviour of the unbounded model in Chapter 4, in which the agent always executes the maximally preferred executable interleaving. This principle can be satisfied in the execution cycle of a bounded GROVE agent by retaining the most preferred interleaving from the previous search cycle and continuing with it if it is more preferred than any interleavings found by search in the following cycle. If any of the executable interleavings found in the current execution cycle have greater preference than the best from the previous cycle, then the agent executes the most preferred of those instead. This interpretation of the execution cycle gives a bounded GROVE agent that always executes the most preferred interleaving available to it. Note that in the case that the previous most preferred interleaving is no longer executable, a bounded GROVE agent executes the most preferred interleaving found in the current cycle, following the assumption that non-executable interleavings have minimal preference. Lastly, retaining interleavings from previous cycles may represent a cost to the agent in terms of space, however this quantity does not grow with the number of interleavings considered (as induced by the bound b) and corresponds to a multiple of l (the average length of an interleaving). This cost can be limited to a constant factor by retaining either the most preferred interleaving of the previous cycle, or some maximum number of interleavings that can be retained, in either case giving a predictable cost. Moreover, this interpretation of the execution cycle is not necessary to guarantee ϵ-preference, only consistency with the behaviour of unbounded GROVE in the sense of executing the most preferred executable interleaving available across execution cycles.

The possibility of retaining the interleavings from previous execution cycles raises the question of whether a bounded GROVE agent should avoid executing retained interleavings in some cases. Unlike the case where the agent does not retain interleavings and there is only a single set of interleavings found by search to consider, the agent may consider retained interleavings (from previous cycles) in addition to those found by search in the current cycle. As preferences may change with the evolving environment, it is possible that the preference of retained interleavings 'decays' over time, even if they remain executable. The retained interleavings may decay to the point where a rational agent may opt to execute several search cycles if it is likely to find an execution with greatly superior preference to the most preferred of those it has retained. Exploring this question further is future work.

5.3.2 Robustness

Let us consider the related problem of robustness to failure and changing circumstances. While a bounded GROVE agent searches for an epsilon-preferred interleaving in each execution cycle, the stability policy of retaining the previous most preferred interleaving suggests further extensions that can be made along the lines of retaining interleavings. Consider an agent that retains several interleavings as a kind of "backup" for when circumstances change and it must seek an executable interleaving. Interleavings that were previously non-executable may be executable in the changed circumstances, and could have greater preference than those found by search in a given execution cycle. Therefore, these previously generated interleavings may be valuable to retain in some kind of finite cache. The cached interleavings may be useful in situations where circumstances have changed, making the agent more robust to change. For instance, the most preferred of these interleavings may be superior to those found in later cycles, such as in the case where an executable interleaving could not be found during search, but there are cached interleavings that are executable in the changed circumstances. Such an agent would still be rational with respect to Theorem 5.1, but with the advantage of access to the interleavings generated in previous execution cycles. This is advantageous to the robustness of a bounded GROVE agent as it effectively increases the pool of interleavings available to select from in each execution cycle, thereby giving a wider base of possible courses of action for the agent.

Lastly, there is the general problem of determining which interleavings should be retained in a finite cache, i.e., determining a policy for preferring caching one interleaving over another. This is similar to the problem of plan coverage [107], where the preconditions of plans are used to determine coverage of different situations that may develop. For a caching policy that retains only most preferred interleavings, there may be situations that are difficult for an agent programmer to anticipate when encoding the preferences for certain domains. For instance, preference may need to take into account factors related primarily to caching and robustness, such as the failure rate of the agent, i.e., frequency of having to abandon progress, the amenability of the environment to different types of executions, coverage (in the sense of plan coverage) of interleavings in the cache, and preference of non-executable interleavings that may become executable later. Strategies for caching and other extensions are outside of the scope of this thesis, see Chapter 6 for discussion of future work.

6 Conclusion

In this thesis I have presented a novel model of rational intention revision. By analysing the state-of-art approaches with respect to a definition of the rational revision problem, I identified a gap in knowledge which I have addressed by giving operational semantics for a model of rational BDI agency. Moreover, I presented a bounded version of the model that under certain assumptions accounts for bounded-rational behaviour that is realisable in real-life, practical agents.

Although the model I have presented addresses the rational intention revision problem under the assumptions made, there are limitations which correspond to differences between the assumptions and some approaches in the literature. For instance, I assume a notion of preference in my definition of rationality, rational intention revision, and the semantics of GROVE, however I do not elaborate in detail how this may be derived using existing techniques or propose a novel technique for deriving preference over executions.

6.1 Future Work

One avenue for future work is investigation of how GROVE might be extended to account for a richer variety of goal types, procedural operations on goals, a richer language of goals, and derivation of preferences over executions which account for the kind of complex deliberation discussed in the literature.

While GROVE assumes achievement goals, the literature accounts for a richer variety of goals, including maintenance goals [28, 29] (and temporal goals in general), and soft goals [44] as a kind of qualitative preference. Moreover, goals in GROVE are assumed to be atoms, while BDI logic approaches to agent programming [40] often assume richer languages for their representations of goals, permitting specification of complex goals such as conjunctive goals and temporal goals.

The concepts of traces and interleavings, central to the semantics of GROVE, bear similarity to the computational traces employed in agent semantics based on temporal logic incorporating temporal goals, such as the semantics for temporal goals proposed by Dastani et al. [23], the directly-executable temporal logic agent framework of Fisher and Hepple [31, 32], and the approach to agent programming with temporal goals of Hindriks and van Riemsdijk [42, 44]. Moreover, a temporal interpretation of goals as "desired progressions" returns us to the motivating intuitions behind the seminal goal lifecycle work of van Riemsdijk et al. [79], and the interpretation of goals as “preferred outcomes” proposed by Governatori et al. [35].

One sub-type of maintenance goals, the passive maintenance goal, could be accounted for by generalising hard constraints on interleavings. Recall that interleavings are filtered first by executability, then by preference. Passive maintenance goals constrain execution such that the agent refrains from committing to a course of action that will violate maintenance conditions. This is similar to the notion that an agent refrains from committing to courses of action that cannot be executed. A simple extension would be to extend the notion of executability to account for a set of maintenance conditions, such that an interleaving is not executable if it violates any of the maintenance conditions when executed. Note also that maintenance goals are a sub-class of temporal goals, so extending GROVE with passive maintenance goals would constitute a first step toward an extension of GROVE to account for more complex temporal goals.

Finally, the semantics of GROVE does facilitate some procedural operations on goals that are standard in agent programming. The suspension and resumption of intentions is implicit in the ordering of steps in interleavings. The definition of traces induced by a goal-plan tree can be extended to include suspend methods and resume methods [102], and abort methods [100] for goals and plans.

A number of different extensions can be made to GROVE based on the general approach of altering how traces are induced by goal-plan trees, and how future executions are derived. These extensions include allowing abort, suspend, and resume methods to be executed where appropriate. While these methods are usually annotations on goal-plan trees and can therefore be captured by extending the derivation of traces to account for these, determining when these methods are actually executed is a higher level concern that coincides with switching between top-level goals at the interleaving level (in the case of suspend/resume methods), and abandoning progress or skipping plans (in the case of abort methods). Since a trace corresponds to a complete execution of a goal, extensions to the semantics with respect to the interpretation of goals fall under this umbrella. For instance, suspend and resume methods [101] can be incorporated into traces to respresent executions of a goal that involve execution of such methods. Note that GROVE already allows an implicit form of suspension and resumption of goals using interleavings. A top-level goal is essentially suspended when the agent has made progress toward it but executes steps for other goals before continuing with it, on the most preferred interleaving. This gives an account of concurrent execution of intentions as a consequence of the interleaved nature of the overall execution of goals. However, suspend and resume methods as they are used in agent programming are procedural operations used to free (on suspension) and recapture (on resumption) resources used by a goal or plan, and are specified as annotations on goal-plan trees. The way traces are derived from goal-plan trees can be extended to make use of these annotations, such that additional traces are available where the steps for suspend and resume methods are available. Determining when suspend or resume methods should be executed is at a level of control above that of traces, and likely requires extension of either how interleavings are generated, or $pexecs$. This is because suspension and resumption of a goal coincides with executing steps for other goals. Switching between which trace is being executed within an interleaving corresponds to a type of suspension and resumption of top-level goals. Because the traces reify the entirety of the executions of goals, the requirement that resumption follows suspension, and suspension does not follow suspension, can be represented directly in the generation of traces that encode this behaviour. The execution of suspend and resume methods then corresponds to a course of action that can be reasoned about and executed as with standard traces. A related possible extension is that of suspend and resume *conditions*, which dictate when suspension and resumption should occur. Suspend and resume methods proposed by Thangarajah et al. [101] may use conditions to prompt their execution, based on environmental circumstances, or in the model proposed by Harland et al. [38] it is suggested they may be conditioned on the state of other goals also. As with suspend and resume methods, these conditions can be encoded in traces by either requiring that the suspend method and resume methods capture the respective conditions in their preconditions. However, this does not capture the case where suspend and resume conditions correspond to aspects of the agent's mental state aside from beliefs, which is a direction for further investigation. In addition, some suspend and resume conditions potentially correspond to conditions that can be captured by temporal formulas, when a goal is suspended awaiting some external change in circumstance rather than to avoid conflicts, for instance.

The incorporation of abort methods [101], which correspond to executing a series of actions prior to dropping a goal or plan. Intuitively this means an agent may execute steps toward a goal that it has decided to drop without achievement, such as in case of plan failure or an alternative course of action being more preferable.

Extending GROVE to allow abort methods would require executing the steps for abort methods at points where a GROVE agent abandons progress, such as when dropping goals or plans. These cases coincide with deriving an alternative (shorter) prefix from the history. The most significant implication of abort methods for GROVE is that abandoning progress may introduce (abort) steps that must be executed on a possible future execution that immediately follows abandoning goals or plans that are in progress. Moreover, a rational agent should be able to reason about these as consequences of a decision to abandon progress, as they may incur additional costs or may be infeasible. Note that in the goal life-cycle model proposed by Harland et al. [38] abort methods are required to execute succesfully and are always feasible, i.e., they are assumed to never fail, so this aspect of rational behaviour is avoided entirely.

Reasoning about executability and preference of a set of intentions in the context of abort, suspend, and resume methods would constitute a novel contribution. Existing work does not consider cases where these operations can fail or conflict with other tasks, and does not consider how agents should determine when it is preferable to perform these operations, i.e., weigh them up as options.

Lastly, GROVE can be extended by incorporating reactive maintenance goals [29]. Reactive maintenance goals inhabit an inactive state until their maintenance condition is violated, in which case they initiate action to restore it. Rather than representing desired states to be achieved, reactive maintenance goals represent states that should be preserved and the steps that can be executed to restore them. This gives a dual-faceted notion of maintenance, combining a constraint that should be respected and an achievement goal that is invoked to restore it. Although the semantics of GROVE allow for the latter, the former constraint aspect not only needs to be accounted for, but must also prompt the adoption of the latter only when it is necessary. One approach to achieving this is a procedural-style encoding akin to that of maintenance goals in Jadex [13], where an achievement goal can be assigned a flag that prevents it from being dropped when achieved. In addition, the achievement goal can be made non-achievable until the maintenance condition is violated, by requiring that all traces corresponding to its goal-plan tree have the negation of the maintenance condition as a precondition. Then, upon adopting the achievement goal a GROVE agent actively pursues it (as an intention) only when it is executable, i.e., when the maintenance condition is violated. Upon violation of the maintenance condition the agent pursues the achievement goal and upon achieving it (and restoring the maintenance condition) it does not abandon the goal, instead "resetting" it as in the procedural approach described by Braubach and Pokahr [13], at which point it will be non-achievable provided the maintenance condition has been successfully restored. This approach suggests the possibility of GROVE agents reasoning about the relative preference of violating maintenance conditions, taking into account whether or not they are able to restore them in the context of their intentions. Permitting this corresponds to treating maintenance goals as soft constraints, rather than the hard constraints they are usually interpreted as, but potentially gives more autonomous behaviour as the agent is free to reason about and choose to abandon its maintenance goals temporarily when it is rational to do so.

Another avenue for future work is implementation of GROVE, following the proposed bounded-rational semantics in Chapter 5. Partial progress has been made on encoding bounded GROVE in meta-APL, for which an implemented interpreter exists. Other approaches to this may be to investigate how to achieve GROVE-like rational behaviour in existing agent-programming platforms, similar to the approach taken by de Silva [97] to incorporate HTN planning in a standard BDI interpreter.

There is also the question of verifying properties of GROVE agents. As the idea of an execution of an agent in terms of concrete steps is central to GROVE, it seems plausible that liveness and safety properties with respect to goals in a GROVE agent can be investigated without significant extension to the model.

Bibliography

1. J. A. Baier and S. A. McIlraith. “On Domain-Independent Heuristics for Planning with Qualitative Preferences”. In: *Logical Formalizations of Commonsense Reasoning, Papers from the 2007 AAAI Spring Symposium, Technical Report SS-07-05, Stanford, California, USA, March 26-28, 2007*. 2007, pp. 7–12. URL: http://www.aaai.org/Library/Symposia/Spring/2007/ss07-05-003.php.

2. J. L. Bermúdez. *Decision Theory and Rationality*. Oxford University Press, New York, NY, US, 2009. ISBN: 978-0199596249. DOI: 10.1093/acprof:oso/9780199548026.001.0001. URL: https://doi.org/10.1093/acprof:oso/9780199548026.001.0001.

3. M. Bienvenu, C. Fritz, and S. A. McIlraith. “Planning with Qualitative Temporal Preferences”. In: *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*. 2006, pp. 134–144. URL: http://www.aaai.org/Library/KR/2006/kr06-016.php.

4. F. S. de Boer, K. V. Hindriks, W. van der Hoek, and J. C. Meyer. “Agent Programming With Declarative Goals”. *CoRR* cs.AI/0207008, 2002. URL: http://arxiv.org/abs/cs.AI/0207008.

5. R. Bordini, J. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons, 2007. DOI: 10.1002/9780470061848. URL: https://doi.org/10.1002/9780470061848.

6. R. H. Bordini, M. Dastani, J. Dix, and A. E. Fallah-Seghrouchni, eds. *Multi-Agent Programming: Languages, Platforms and Applications*. Vol. 15. Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer, 2005. ISBN: 0-387-24568-5.

7. R. H. Bordini and J. F. Hübner. “Semantics for the Jason Variant of AgentSpeak (Plan Failure and some Internal Actions)”. In: *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*. 2010, pp. 635–640. DOI: 10.3233/978-1-60750-606-5-635. URL: https://doi.org/10.3233/978-1-60750-606-5-635.

8. R. H. Bordini and Á. F. Moreira. “Proving BDI Properties of Agent-Oriented Programming Languages”. *Ann. Math. Artif. Intell.* 42:1-3, 2004, pp. 197–226. DOI: 10.1023/B:AMAI.0000034527.45635.e5. URL: https://doi.org/10.1023/B:AMAI.0000034527.45635.e5.

9. R. H. Bordini and Á. F. Moreira. "Proving the Asymmetry Thesis Principles for a BDI Agent-Oriented Programming Language". *Electr. Notes Theor. Comput. Sci.* 70:5, 2002, pp. 108–125. DOI: 10.1016/S1571-0661(04)80591-7. URL: https://doi.org/10.1016/S1571-0661(04)80591-7.

10. M. Bratman. *Intention, plans, and practical reason.* Harvard University Press, Cambridge, MA, 1987. ISBN: 9780674458185. URL: http://books.google.de/books?id=I0nuAAAAMAAJ.

11. M. E. Bratman, D. J. Israel, and M. E. Pollack. "Plans and Resource-Bounded Practical Reasoning". *Computational Intelligence* 4, 1988, pp. 349–355. DOI: 10.1111/j.1467-8640.1988.tb00284.x. URL: https://doi.org/10.1111/j.1467-8640.1988.tb00284.x.

12. L. Braubach and A. Pokahr. "Representing Long-Term and Interest BDI Goals". In: *Programming Multi-Agent Systems - 7th International Workshop, ProMAS 2009, Budapest, Hungary, May 10-15, 2009. Revised Selected Papers.* 2009, pp. 201–218. DOI: 10.1007/978-3-642-14843-9_13. URL: https://doi.org/10.1007/978-3-642-14843-9_13.

13. L. Braubach, A. Pokahr, D. Moldt, and W. Lamersdorf. "Goal Representation for BDI Agent Systems". In: *Programming Multi-Agent Systems, Second International Workshop ProMAS 2004, New York, NY, USA, July 20, 2004 Selected Revised and Invited Papers.* 2004, pp. 44–65. DOI: 10.1007/978-3-540-32260-3_3. URL: https://doi.org/10.1007/978-3-540-32260-3_3.

14. J. M. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. W. N. van der Torre. "The BOID architecture: conflicts between beliefs, obligations, intentions and desires". In: *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS 2001, Montreal, Canada, May 28 - June 1, 2001.* 2001, pp. 9–16. DOI: 10.1145/375735.375766. URL: https://doi.org/10.1145/375735.375766.

15. C. Castelfranchi and F. Paglieri. "The Role of Beliefs in Goal Dynamics: Prolegomena To a Constructive Theory of Intentions". *Synth.* 155:2, 2007, pp. 237–263. DOI: 10.1007/s11229-006-9156-3. URL: https://doi.org/10.1007/s11229-006-9156-3.

16. J. M. Castro-Manzano. "An Introduction to Intention Revision: Issues and Problems". In: *Proceedings of the Fifth Latin American Workshop on Non-Monotonic Reasoning 2009 (LANMR'09) , Facultad de Ciencias Básicas, Ingeniería y Tecnología , Universidad Autónoma de Tlaxcala (UAT) , Apizaco, Tlaxcala, México, November 5-6, 2009.* 2009, pp. 76–89. URL: http://ceur-ws.org/Vol-533/07_LANMR09_04.pdf.

17. B. J. Clement and E. H. Durfee. *Identifying and resolving conflicts among agents with hierarchical plans.* Tech. rep. In AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities, 1999.

18. B. J. Clement and E. H. Durfee. "Performance of Coordinating Concurrent Hierarchical Planning Agents Using Summary Information". In: *4th International Conference on Multi-Agent Systems, ICMAS 2000, Boston, MA, USA, July 10-12, 2000*. 2000, pp. 373–374. DOI: 10.1109/ICMAS.2000.858481. URL: https://doi.org/10.1109/ICMAS.2000.858481.

19. B. J. Clement and E. H. Durfee. "Theory for Coordinating Concurrent Hierarchical Planning Agents Using Summary Information". In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA*. 1999, pp. 495–502. URL: http://www.aaai.org/Library/AAAI/1999/aaai99-071.php.

20. P. R. Cohen and H. J. Levesque. "Intention Is Choice With Commitment". *Artif. Intell.* 42:2-3, 1990, pp. 213–261. DOI: 10.1016/0004-3702(90)90055-5. URL: https://doi.org/10.1016/0004-3702(90)90055-5.

21. J. Creel, C. Menzel, and T. R. Ioerger. "Intention Is Commitment with Expectation". In: *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA*. 2007, pp. 50–55. URL: http://www.aaai.org/Library/FLAIRS/2007/flairs07-011.php.

22. M. Dastani, M. B. van Riemsdijk, and J. C. Meyer. "Goal types in agent programming". In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*. 2006, pp. 1285–1287. DOI: 10.1145/1160633.1160867. URL: https://doi.org/10.1145/1160633.1160867.

23. M. Dastani, M. B. van Riemsdijk, and M. Winikoff. "Rich goal types in agent programming". In: *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*. 2011, pp. 405–412. URL: http://portal.acm.org/citation.cfm?id=2030530\&CFID=69153967\&CFTOKEN=38069692.

24. M. Dastani and L. W. N. van der Torre. "Programming BOID-Plan Agents: Deliberating about Conflicts among Defeasible Mental Attitudes and Plans". In: *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19-23 August 2004, New York, NY, USA*. 2004, pp. 706–713. DOI: 10.1109/AAMAS.2004.10082. URL: https://doi.org/10.1109/AAMAS.2004.10082.

25. G. De Giacomo, Y. Lespérance, and H. J. Levesque. "Congolog, a Concurrent Programming Language Based on the Situation Calculus". *Artif. Intell.* 121:1-2, 2000, pp. 109–169. DOI: 10.1016/S0004-3702(00)00031-X. URL: https://doi.org/10.1016/S0004-3702(00)00031-X.

26. D. C. Dennett. *The Intentional Stance*. The MIT Press, Cambridge, MA, 1987.

27. H. van Ditmarsch, T. de Lima, and E. Lorini. “Intention Change via Local Assignments”. In: *Languages, Methodologies, and Development Tools for Multi-Agent Systems - Third International Workshop, LADS 2010, Lyon, France, August 30 - September 1, 2010, Revised Selected Papers.* Ed. by M. Dastani, A. E. Fallah-Seghrouchni, J. Hübner, and J. Leite. Vol. 6822. Lecture Notes in Computer Science. Springer, 2010, pp. 136–151. DOI: 10.1007/978-3-642-22723-3_8. URL: https://doi.org/10.1007/978-3-642-22723-3_8.

28. S. Duff, J. Harland, and J. Thangarajah. “On proactivity and maintenance goals”. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems - AAMAS ’06.* 2006, pp. 1033–1040. DOI: 10.1145/1160633.1160817. URL: https://doi.org/10.1145/1160633.1160817.

29. S. Duff, J. Thangarajah, and J. Harland. “Maintenance Goals in Intelligent Agents”. *Comput. Intell.* 30:1, 2014, pp. 71–114. DOI: 10.1111/coin.12000. URL: https://doi.org/10.1111/coin.12000.

30. W. Feller. *An Introduction to Probability Theory and Its Applications.* Vol. 1. Wiley, 1968. ISBN: 0471257087.

31. M. Fisher. “Agent Deliberation in an Executable Temporal Framework”. *J. Appl. Log.* 9:4, 2011, pp. 223–238. DOI: 10.1016/j.jal.2011.09.004. URL: https://doi.org/10.1016/j.jal.2011.09.004.

32. M. Fisher and A. Hepple. “Executing Logical Agent Specifications”. In: *Multi-Agent Programming, Languages, Tools and Applications.* 2009, pp. 1–27. DOI: 10.1007/978-0-387-89299-3_1. URL: https://doi.org/10.1007/978-0-387-89299-3_1.

33. M. P. Georgeff and F. F. Ingrand. “Decision-Making in an Embedded Reasoning System”. In: *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989.* 1989, pp. 972–978. URL: http://ijcai.org/Proceedings/89-2/Papers/020.pdf.

34. M. P. Georgeff and A. L. Lansky. “Reactive Reasoning and Planning”. In: *Proceedings of the 6th National Conference on Artificial Intelligence. Seattle, WA, USA, July 1987.* 1987, pp. 677–682. URL: http://www.aaai.org/Library/AAAI/1987/aaai87-121.php.

35. G. Governatori, F. Olivieri, S. Scannapieco, A. Rotolo, and M. Cristani. “The Rationale Behind the Concept of Goal”. *Theory Pract. Log. Program.* 16:3, 2016, pp. 296–324. DOI: 10.1017/S1471068416000053. URL: https://doi.org/10.1017/S1471068416000053.

36. J. Grant, S. Kraus, D. Perlis, and M. J. Wooldridge. “Postulates for Revising BDI Structures”. *Synthese* 175:Supplement-1, 2010, pp. 39–62. DOI: 10.1007/s11229-010-9735-1. URL: https://doi.org/10.1007/s11229-010-9735-1.

37. J. Harland, D. N. Morley, J. Thangarajah, and N. Yorke-Smith. "Aborting, Suspending, and Resuming Goals and Plans in BDI Agents". *Autonomous Agents and Multi-Agent Systems* 31:2, 2017, pp. 288–331. DOI: 10.1007/s10458-015-9322-4. URL: https://doi.org/10.1007/s10458-015-9322-4.

38. J. Harland, D. N. Morley, J. Thangarajah, and N. Yorke-Smith. "An Operational Semantics for the Goal Life-Cycle in Bdi Agents". *Autonomous Agents and Multi-Agent Systems* 28:4, 2013, pp. 682–719. DOI: 10.1007/s10458-013-9238-9. URL: https://doi.org/10.1007/s10458-013-9238-9.

39. K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J. C. Meyer. "Agent Programming in 3apl". *Autonomous Agents and Multi-Agent Systems* 2:4, 1999, pp. 357–401. DOI: 10.1023/A:1010084620690. URL: https://doi.org/10.1023/A:1010084620690.

40. K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J. C. Meyer. "Agent Programming with Declarative Goals". In: *Intelligent Agents VII. Agent Theories Architectures and Languages, 7th International Workshop, ATAL 2000, Boston, MA, USA, July 7-9, 2000, Proceedings*. 2000, pp. 228–243. DOI: 10.1007/3-540-44631-1_16. URL: https://doi.org/10.1007/3-540-44631-1_16.

41. K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J. C. Meyer. "Formal Semantics for an Abstract Agent Programming Language". In: *Intelligent Agents IV, Agent Theories, Architectures, and Languages, 4th International Workshop, ATAL '97, Providence, Rhode Island, USA, July 24-26, 1997, Proceedings*. 1997, pp. 215–229. DOI: 10.1007/BFb0026761. URL: https://doi.org/10.1007/BFb0026761.

42. K. V. Hindriks, W. van der Hoek, and M. B. van Riemsdijk. "Agent programming with temporally extended goals". In: *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1*. 2009, pp. 137–144. URL: https://dl.acm.org/citation.cfm?id=1558031.

43. K. V. Hindriks and M. B. van Riemsdijk. "Satisfying Maintenance Goals". In: *Declarative Agent Languages and Technologies V, 5th International Workshop, DALT 2007, Honolulu, HI, USA, May 14, 2007, Revised Selected and Invited Papers*. 2007, pp. 86–103. DOI: 10.1007/978-3-540-77564-5_6. URL: https://doi.org/10.1007/978-3-540-77564-5_6.

44. K. V. Hindriks and M. B. van Riemsdijk. "Using Temporal Logic to Integrate Goals and Qualitative Preferences into Agent Programming". In: *Declarative Agent Languages and Technologies VI, 6th International Workshop, DALT 2008, Estoril, Portugal, May 12, 2008, Revised Selected and Invited Papers*. 2008, pp. 215–232. DOI: 10.1007/978-3-540-93920-7_14. URL: https://doi.org/10.1007/978-3-540-93920-7_14.

45. W. van der Hoek, W. Jamroga, and M. J. Wooldridge. "Towards a Theory of Intention Revision". *Synthese* 155:2, 2007, pp. 265–290. DOI: 10.1007/s11229-006-9145-6. URL: https://doi.org/10.1007/s11229-006-9145-6.

46. W. van der Hoek, B. van Linder, and J. C. Meyer. "A Logic of Capabilities". In: *Logical Foundations of Computer Science, Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11-14, 1994, Proceedings.* 1994, pp. 366–378. DOI: 10.1007/3-540-58140-5_34. URL: https://doi.org/10.1007/3-540-58140-5_34.

47. J. F. Horty and M. E. Pollack. "Evaluating New Options in the Context of Existing Plans". *Artif. Intell.* 127:2, 2001, pp. 199–220. DOI: 10.1016/S0004-3702(01)00060-1. URL: https://doi.org/10.1016/S0004-3702(01)00060-1.

48. J. F. Hübner, R. H. Bordini, and M. J. Wooldridge. "Plan patterns for declarative goals in AgentSpeak". In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006.* 2006, pp. 1291–1293. DOI: 10.1145/1160633.1160869. URL: https://doi.org/10.1145/1160633.1160869.

49. J. F. Hübner, R. H. Bordini, and M. J. Wooldridge. "Programming Declarative Goals Using Plan Patterns". In: *Declarative Agent Languages and Technologies IV, 4th International Workshop, DALT 2006, Hakodate, Japan, May 8, 2006, Selected, Revised and Invited Papers.* 2006, pp. 123–140. DOI: 10.1007/11961536_9. URL: https://doi.org/10.1007/11961536_9.

50. T. F. I. III, E. Pacuit, and Y. Shoham. "Joint Revision of Beliefs and Intention". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010.* 2010. URL: http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1297.

51. S. M. Khan. "Rational agents: prioritized goals, goal dynamics, and agent programming languages with declarative goals". In: *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3.* 2010, pp. 1653–1654. URL: https://dl.acm.org/citation.cfm?id=1838526.

52. S. M. Khan and Y. Lespérance. "A logical framework for prioritized goal change". In: *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3.* 2010, pp. 283–290. URL: https://dl.acm.org/citation.cfm?id=1838246.

53. S. M. Khan and Y. Lespérance. "Handling prioritized goals and subgoals in a logical account of goal change". In: *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 2.* 2009, pp. 1155–1156. URL: https://dl.acm.org/citation.cfm?id=1558187.

54. S. M. Khan and Y. Lespérance. "SR-APL: a model for a programming language for rational BDI agents with prioritized goals". In: *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3.* 2011, pp. 1251–1252. URL: http://portal.acm.org/citation.cfm?id=2034511\&CFID=69154334\&CFTOKEN=45298625.

55. D. Kinny and M. P. Georgeff. "Commitment and Effectiveness of Situated Agents". In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991.* 1991, pp. 82–88. URL: http://ijcai.org/Proceedings/91-1/Papers/014.pdf.

56. S. Leask and B. Logan. "Programming Agent Deliberation Using Procedural Reflection". *Fundamenta Informaticae* 158:1–3, 2018, pp. 93–120. DOI: 10.3233/FI-2018-1643. URL: https://doi.org/10.3233/FI-2018-1643.

57. E. Lorini, M. Dastani, H. van Ditmarsch, A. Herzig, and J. C. Meyer. "Intentions and Assignments". In: *Logic, Rationality, and Interaction, Second International Workshop, LORI 2009, Chongqing, China, October 8-11, 2009. Proceedings.* 2009, pp. 198–211. DOI: 10.1007/978-3-642-04893-7_16. URL: https://doi.org/10.1007/978-3-642-04893-7_16.

58. E. Lorini, H. van Ditmarsch, and T. D. Lima. "A Logical Model of Intention and Plan Dynamics". In: *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings.* 2010, pp. 1075–1076. DOI: 10.3233/978-1-60750-606-5-1075. URL: https://doi.org/10.3233/978-1-60750-606-5-1075.

59. A. Mansfield. "An Algorithm for a Merge Recognition Problem". *Discrete Applied Mathematics* 4:3, 1982, pp. 193–197. ISSN: 0166-218X. DOI: https://doi.org/10.1016/0166-218X(82)90039-7. URL: https://doi.org/https://doi.org/10.1016/0166-218X(82)90039-7.

60. J. C. Meyer, W. van der Hoek, and B. van Linder. "A Logical Approach To the Dynamics of Commitments". *Artif. Intell.* 113:1-2, 1999, pp. 1–40. DOI: 10.1016/S0004-3702(99)00061-2. URL: https://doi.org/10.1016/S0004-3702(99)00061-2.

61. M. Morandini, L. Penserini, and A. Perini. "Operational semantics of goal models in adaptive agents". In: *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1.* 2009, pp. 129–136. URL: https://dl.acm.org/citation.cfm?id=1558030.

62. P. K. Moser, ed. *Rationality in Action: Contemporary Approaches.* Cambridge University Press, 1990. ISBN: 978-1-55860-856-6.

63. I. Nunes and M. Luck. “Softgoal-based plan selection in model-driven BDI agents”. In: *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS ’14, Paris, France, May 5-9, 2014*. 2014, pp. 749–756. URL: http://dl.acm.org/citation.cfm?id=2615852.

64. L. Padgham and P. Lambrix. “Agent Capabilities: Extending BDI Theory”. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA*. 2000, pp. 68–73. URL: http://www.aaai.org/Library/AAAI/2000/aaai00-011.php.

65. L. Padgham and P. Lambrix. “Formalisations of Capabilities for Bdi-Agents”. *Autonomous Agents and Multi-Agent Systems* 10:3, 2005, pp. 249–271. ISSN: 1573-7454. DOI: 10.1007/s10458-004-4345-2. URL: https://doi.org/10.1007/s10458-004-4345-2.

66. L. Padgham and D. Singh. “Situational preferences for BDI plans”. In: *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS ’13, Saint Paul, MN, USA, May 6-10, 2013*. 2013, pp. 1013–1020. URL: http://dl.acm.org/citation.cfm?id=2485080.

67. A. Pokahr, L. Braubach, and W. Lamersdorf. “A BDI architecture for goal deliberation”. In: *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*. 2005, pp. 1295–1296. DOI: 10.1145/1082473.1082740. URL: https://doi.org/10.1145/1082473.1082740.

68. A. Pokahr, L. Braubach, and W. Lamersdorf. “A Goal Deliberation Strategy for BDI Agent Systems”. In: *Multiagent System Technologies, Third German Conference, MATES 2005, Koblenz, Germany, September 11-13, 2005, Proceedings*. 2005, pp. 82–93. DOI: 10.1007/11550648_8. URL: https://doi.org/10.1007/11550648_8.

69. A. Pokahr, L. Braubach, and W. Lamersdorf. “Jadex: A BDI Reasoning Engine”. In: *Multi-Agent Programming: Languages, Platforms and Applications*. 2005, pp. 149–174.

70. M. E. Pollack. “The Uses of Plans”. *Artif. Intell.* 57:1, 1992, pp. 43–68. DOI: 10.1016/0004-3702(92)90104-6. URL: https://doi.org/10.1016/0004-3702(92)90104-6.

71. A. S. Rao. “AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language”. In: *Agents Breaking Away, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996, Proceedings*. 1996, pp. 42–55. DOI: 10.1007/BFb0031845. URL: https://doi.org/10.1007/BFb0031845.

72. A. S. Rao and M. P. Georgeff. “An Abstract Architecture for Rational Agents”. In: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR’92). Cambridge, MA, USA, October 25-29, 1992*. 1992, pp. 439–449.

73. A. S. Rao and M. P. Georgeff. "Asymmetry Thesis and Side-Effect Problems in Linear-Time and Branching-Time Intention Logics". In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991.* 1991, pp. 498–505. URL: http://ijcai.org/Proceedings/91-1/Papers/077.pdf.

74. A. S. Rao and M. P. Georgeff. "Decision Procedures for BDI Logics". *J. Log. Comput.* 8:3, 1998, pp. 293–342. DOI: 10.1093/logcom/8.3.293. URL: https://doi.org/10.1093/logcom/8.3.293.

75. A. S. Rao and M. P. Georgeff. "Modeling Rational Agents within a BDI-Architecture". In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991.* 1991, pp. 473–484.

76. B. van Riemsdijk, W. van der Hoek, and J. C. Meyer. "Agent programming in dribble: from beliefs to goals using plans". In: *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings.* 2003, pp. 393–400. DOI: 10.1145/860575.860639. URL: https://doi.org/10.1145/860575.860639.

77. M. B. van Riemsdijk, M. Dastani, and J. C. Meyer. "Goals in Conflict: Semantic Foundations of Goals in Agent Programming". *Autonomous Agents and Multi-Agent Systems* 18:3, 2009, pp. 471–500. DOI: 10.1007/s10458-008-9067-4. URL: https://doi.org/10.1007/s10458-008-9067-4.

78. M. B. van Riemsdijk, M. Dastani, and J. C. Meyer. "Subgoal Semantics in Agent Programming". In: *Progress in Artificial Intelligence, 12th Portuguese Conference on Artificial Intelligence, EPIA 2005, Covilhã, Portugal, December 5-8, 2005, Proceedings.* 2005, pp. 548–559. DOI: 10.1007/11595014_54. URL: https://doi.org/10.1007/11595014_54.

79. M. B. van Riemsdijk, M. Dastani, and M. Winikoff. "Goals in agent systems: a unifying framework". In: *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2.* 2008, pp. 713–720. URL: https://dl.acm.org/citation.cfm?id=1402323.

80. S. J. Russell and E. Wefald. "Principles of Metareasoning". *Artif. Intell.* 49:1-3, 1991, pp. 361–395. DOI: 10.1016/0004-3702(91)90015-C. URL: https://doi.org/10.1016/0004-3702(91)90015-C.

81. S. Sardiña and L. Padgham. "A BDI Agent Programming Language With Failure Handling, Declarative Goals, and Planning". *Auton. Agents Multi Agent Syst.* 23:1, 2011, pp. 18–70. DOI: 10.1007/s10458-010-9130-9. URL: https://doi.org/10.1007/s10458-010-9130-9.

82. S. Sardiña and L. Padgham. “Goals in the context of BDI plan failure and planning”. In: *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*. 2007, p. 7. DOI: 10.1145/1329125.1329134. URL: https://doi.org/10.1145/1329125.1329134.

83. S. Sardiña and S. Shapiro. “Rational action in agent programs with prioritized goals”. In: *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*. 2003, pp. 417–424. DOI: 10.1145/860575.860642. URL: https://doi.org/10.1145/860575.860642.

84. S. Sardiña, L. de Silva, and L. Padgham. “Hierarchical planning in BDI agent programming languages: a formal approach”. In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*. 2006, pp. 1001–1008. DOI: 10.1145/1160633.1160813. URL: https://doi.org/10.1145/1160633.1160813.

85. M. C. Schut and M. J. Wooldridge. “Intention reconsideration in complex environments”. In: *Proceedings of the Fourth International Conference on Autonomous Agents, AGENTS 2000, Barcelona, Catalonia, Spain, June 3-7, 2000*. 2000, pp. 209–216. DOI: 10.1145/336595.337377. URL: https://doi.org/10.1145/336595.337377.

86. M. C. Schut and M. J. Wooldridge. “Principles of intention reconsideration”. In: *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS 2001, Montreal, Canada, May 28 - June 1, 2001*. 2001, pp. 340–347. DOI: 10.1145/375735.376326. URL: https://doi.org/10.1145/375735.376326.

87. M. C. Schut, M. J. Wooldridge, and S. Parsons. “Reasoning about Intentions in Uncertain Domains”. In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 6th European Conference, ECSQARU 2001, Toulouse, France, September 19-21, 2001, Proceedings*. 2001, pp. 84–95. DOI: 10.1007/3-540-44652-4_9. URL: https://doi.org/10.1007/3-540-44652-4_9.

88. M. C. Schut, M. J. Wooldridge, and S. Parsons. “The Theory and Practice of Intention Reconsideration”. *J. Exp. Theor. Artif. Intell.* 16:4, 2004, pp. 261–293. DOI: 10.1080/09528130412331309277. URL: https://doi.org/10.1080/09528130412331309277.

89. S. Shapiro and G. Brewka. “Dynamic Interactions between Goals and Beliefs”. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. 2007, pp. 2625–2630. URL: http://ijcai.org/Proceedings/07/Papers/422.pdf.

90. S. Shapiro, Y. Lespérance, and H. J. Levesque. “Goal Change”. In: *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh,*

Scotland, UK, July 30 - August 5, 2005. 2005, pp. 582–588. URL: http://ijcai.org/Proceedings/05/Papers/1114.pdf.

91. S. Shapiro, Y. Lespérance, and H. J. Levesque. "Goal Change in the Situation Calculus". *J. Log. Comput.* 17:5, 2007, pp. 983–1018. DOI: 10.1093/logcom/exm050. URL: https://doi.org/10.1093/logcom/exm050.

92. S. Shapiro, Y. Lespérance, and H. J. Levesque. "Goals and Rational Action in the Situation Calculus - A Preliminary Report". In: *In Working Notes of the AAAI Fall Symposium on Rational Agency: Concepts, Theories, Models, and Applications*. 1995, pp. 117–122.

93. P. H. Shaw. "Reasoning about goal-plan trees in autonomous agents : development of petri net and constraint-based approaches with resulting performance comparisons". PhD thesis. Durham University, UK, 2010. URL: http://etheses.dur.ac.uk/125/.

94. P. H. Shaw and R. H. Bordini. "Towards Alternative Approaches to Reasoning About Goals". In: *Declarative Agent Languages and Technologies V, 5th International Workshop, DALT 2007, Honolulu, HI, USA, May 14, 2007, Revised Selected and Invited Papers*. 2007, pp. 104–121. DOI: 10.1007/978-3-540-77564-5_7. URL: https://doi.org/10.1007/978-3-540-77564-5_7.

95. P. H. Shaw, B. Farwer, and R. H. Bordini. "Theoretical and experimental results on the goal-plan tree problem". In: *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 3*. 2008, pp. 1379–1382. URL: https://dl.acm.org/citation.cfm?id=1402877.

96. Y. Shoham. "Logical Theories of Intention and the Database Perspective". *J. Philos. Log.* 38:6, 2009, pp. 633–647. DOI: 10.1007/s10992-009-9116-8. URL: https://doi.org/10.1007/s10992-009-9116-8.

97. L. de Silva. "BDI Agent Reasoning with Guidance from HTN Recipes". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*. 2017, pp. 759–767. URL: http://dl.acm.org/citation.cfm?id=3091233.

98. L. de Silva and L. Padgham. "A Comparison of BDI Based Real-Time Reasoning and HTN Based Planning". In: *AI 2004: Advances in Artificial Intelligence, 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004, Proceedings*. 2004, pp. 1167–1173. DOI: 10.1007/978-3-540-30549-1_118. URL: https://doi.org/10.1007/978-3-540-30549-1_118.

99. M. P. Singh. "A Critical Examination of Use Cohen-Levesque Theory of Intentions". In: *10th European Conference on Artificial Intelligence, ECAI 92, Vienna, Austria, August 3-7, 1992. Proceedings*. 1992, pp. 364–368.

100. J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. “Aborting tasks in BDI agents”. In: *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*. 2007, p. 6. DOI: 10.1145/1329125.1329133. URL: https://doi.org/10.1145/1329125.1329133.

101. J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. “Operational Behaviour for Executing, Suspending, and Aborting Goals in BDI Agent Systems”. In: *Declarative Agent Languages and Technologies VIII - 8th International Workshop, DALT 2010, Toronto, Canada, May 10, 2010, Revised, Selected and Invited Papers*. 2010, pp. 1–21. DOI: 10.1007/978-3-642-20715-0_1. URL: https://doi.org/10.1007/978-3-642-20715-0_1.

102. J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. “Suspending and resuming tasks in BDI agents”. In: *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 1*. 2008, pp. 405–412. URL: https://dl.acm.org/citation.cfm?id=1402443.

103. J. Thangarajah and L. Padgham. “Computationally Effective Reasoning About Goal Interactions”. *J. Autom. Reasoning* 47:1, 2011, pp. 17–56. DOI: 10.1007/s10817-010-9175-0. URL: https://doi.org/10.1007/s10817-010-9175-0.

104. J. Thangarajah, L. Padgham, and J. Harland. “Representation and Reasoning for Goals in BDI Agents”. In: *Computer Science 2002, Twenty-Fifth Australasian Computer Science Conference (ACSC2002), Monash University, Melbourne, Victoria, Australia, January/February 2002*. 2002, pp. 259–265. URL: http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV4Thangarajah.html.

105. J. Thangarajah, L. Padgham, and M. Winikoff. “Detecting & Avoiding Interference Between Goals in Intelligent Agents”. In: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. 2003, pp. 721–726. URL: http://ijcai.org/Proceedings/03/Papers/105.pdf.

106. J. Thangarajah, L. Padgham, and M. Winikoff. “Detecting & exploiting positive goal interaction in intelligent agents”. In: *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*. 2003, pp. 401–408. DOI: 10.1145/860575.860640. URL: https://doi.org/10.1145/860575.860640.

107. J. Thangarajah, S. Sardiña, and L. Padgham. “Measuring plan coverage and overlap for agent reasoning”. In: *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*. 2012, pp. 1049–1056. URL: http://dl.acm.org/citation.cfm?id=2343847.

108. J. Thangarajah, M. Winikoff, L. Padgham, and K. Fischer. "Avoiding Resource Conflicts in Intelligent Agents". In: *Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002.* 2002, pp. 18–22.

109. A. Unruh, J. Bailey, and K. Ramamohanarao. "A Framework for Goal-Based Semantic Compensation in Agent Systems". In: *Safety and Security in Multiagent Systems - Research Results from 2004-2006.* 2009, pp. 130–146. DOI: 10.1007/978-3-642-04879-1_10. URL: https://doi.org/10.1007/978-3-642-04879-1_10.

110. A. Unruh, H. Harjadi, J. Bailey, and K. Ramamohanarao. "Semantic-compensation-based recovery management in multi-agent systems". In: *In Proc. of Second IEEE Symposium on Multi-Agent Security and Survivability (IEEE MAS&S'05.* 2005.

111. K. Vikhorev, N. Alechina, and B. Logan. "Agent programming with priorities and deadlines". In: *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3.* 2011, pp. 397–404. URL: http://portal.acm.org/citation.cfm?id=2030529\&CFID=69153967\&CFTOKEN=38069692.

112. S. Visser, J. Thangarajah, and J. Harland. "Reasoning about Preferences in Intelligent Agent Systems". In: *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011.* 2011, pp. 426–431. DOI: 10.5591/978-1-57735-516-8/IJCAI11-079. URL: https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-079.

113. S. Visser, J. Thangarajah, J. Harland, and F. Dignum. "Preference-Based Reasoning in BDI Agent Systems". *Auton. Agents Multi Agent Syst.* 30:2, 2016, pp. 291–330. DOI: 10.1007/s10458-015-9288-2. URL: https://doi.org/10.1007/s10458-015-9288-2.

114. M. Winikoff, M. Dastani, and M. B. van Riemsdijk. "A unified interaction-aware goal framework". In: *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings.* 2010, pp. 1033–1034. DOI: 10.3233/978-1-60750-606-5-1033. URL: https://doi.org/10.3233/978-1-60750-606-5-1033.

115. M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. "Declarative & Procedural Goals in Intelligent Agent Systems". In: *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002.* 2002, pp. 470–481.

116. W. Wobcke. "An Operational Semantics for a PRS-Like Agent Architecture". In: *AI 2001: Advances in Artificial Intelligence, 14th Australian Joint Conference on Artificial Intelligence, Adelaide, Australia, December 10-14, 2001, Proceedings.* Ed. by M. Stumptner, D. Corbett, and M. J. Brooks. Vol. 2256. Lecture Notes in Computer Science. Springer, 2001, pp. 569–580. DOI: 10.1007/3-540-45656-2_49. URL: https://doi.org/10.1007/3-540-45656-2_49.

117. W. Wobcke. “Intention and Rationality for PRS-Like Agents”. In: *AI 2002: Advances in Artificial Intelligence, 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia, December 2-6, 2002, Proceedings.* 2002, pp. 167–178. DOI: 10.1007/3-540-36187-1_15. URL: https://doi.org/10.1007/3-540-36187-1_15.

118. M. J. Wooldridge. *Introduction to multiagent systems.* Wiley, 2002. ISBN: 978-0-471-49691-5.

119. M. J. Wooldridge and S. Parsons. “Intention Reconsideration Reconsidered”. In: *Intelligent Agents V, Agent Theories, Architectures, and Languages, 5th International Workshop, ATAL ’98, Paris, France, July 4-7, 1998, Proceedings.* 1998, pp. 63–79. DOI: 10.1007/3-540-49057-4_5. URL: https://doi.org/10.1007/3-540-49057-4_5.

120. M. Xu, K. McAreavey, K. Bauters, and W. Liu. “Intention Interleaving Via Classical Replanning”. In: *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019.* 2019, pp. 85–92. DOI: 10.1109/ICTAI.2019.00021. URL: https://doi.org/10.1109/ICTAI.2019.00021.

121. Y. Yao and B. Logan. “Action-Level Intention Selection for BDI Agents”. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016.* 2016, pp. 1227–1236. URL: http://dl.acm.org/citation.cfm?id=2937103.

122. Y. Yao, B. Logan, and J. Thangarajah. “Robust Execution of BDI Agent Programs by Exploiting Synergies Between Intentions”. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.* 2016, pp. 2558–2565. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12148.

123. M. R. Zatelli, J. F. Hübner, A. Ricci, and R. H. Bordini. “Conflicting goals in agent-oriented programming”. In: *Proceedings of the 6th International Workshop on Programming Based on Actors, Agents, and Decentralized Control, AGERE 2016, Amsterdam, The Netherlands, October 30, 2016.* 2016, pp. 21–30. DOI: 10.1145/3001886.3001889. URL: https://doi.org/10.1145/3001886.3001889.

124. M. van Zee, M. Dastani, D. Doder, and L. W. N. van der Torre. “Consistency Conditions for Beliefs and Intentions”. In: *2015 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 22-25, 2015.* 2015. URL: http://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10282.

125. M. van Zee and D. Doder. “AGM-Style Revision of Beliefs and Intentions”. In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016).* 2016, pp. 1511–1519. DOI: 10.3233/978-1-61499-672-9-1511. URL: https://doi.org/10.3233/978-1-61499-672-9-1511.

126. M. van Zee, D. Doder, M. Dastani, and L. W. N. van der Torre. "AGM Revision of Beliefs about Action and Time". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Q. Yang and M. J. Wooldridge. AAAI Press, 2015, pp. 3250–3256. URL: http://ijcai.org/Abstract/15/458.

127. M. van Zee, D. Doder, L. van der Torre, M. Dastani, T. F. I. III, and E. Pacuit. "Intention As Commitment Toward Time". *Artif. Intell.* 283, 2020, p. 103270. DOI: 10.1016/j.artint.2020.103270. URL: https://doi.org/10.1016/j.artint.2020.103270.