UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

HENRIQUE MONTEIRO ROGICH JASINSKI

GERAÇÃO DE EXPLICAÇÕES CONTRASTIVAS PARA SELEÇÃO DE OBJETIVOS EM AGENTES BDI

CURITIBA

2022

HENRIQUE MONTEIRO ROGICH JASINSKI

GERAÇÃO DE EXPLICAÇÕES CONTRASTIVAS PARA SELEÇÃO DE OBJETIVOS EM AGENTES BDI

Generating Contrastive Explanations For Bdi-Based Goal Selection

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Orientador: Prof. Dr. Cesar Augusto Tacla

Coorientadora: Dra. Miriam Mariela Mercedes Morveli-Espinoza

CURITIBA
2022

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba

HENRIQUE MONTEIRO ROGICH JASINSKI

GERAÇÃO DE EXPLICAÇÕES CONTRASTIVAS PARA SELEÇÃO DE OBJETIVOS EM AGENTES BDI

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Data de aprovação: 26 de Abril de 2022

Dr. Cesar Augusto Tacla, Doutorado - Universidade Tecnológica Federal do Paraná

Dra. Jerusa Marchi, Doutorado - Universidade Federal de Santa Catarina (Ufsc)

Dr. Juan Carlos Nieves Sanchez, Doutorado - Umea University

Dra. Miriam Mariela Mercedes Morveli Espinoza, Doutorado - Universidade Tecnológica Federal do Paraná

ACKNOWLEDGEMENTS

I want to thank my mother, Soraia M. Vieira, and my partner, Carina C. D. Bittencourt, for supporting me in this journey and encouraging me to keep pushing through the hardships in it.

I want to express my gratitude to my advisers Prof. Dr. Cesar A. Tacla and Dra. M. Mariela M. Morveli Espinoza, for the guidance and assistance in this learning process. The many stimulating discussions helped to widen my view.

To the rest of my family and friends that dealt with my absence.

This study was financed in part by the Centro Nacional de Pesquisa - Brasil (CNPq), Finance Code 133320/2019-7.

RESUMO

Conforme o uso de sistemas baseados em agentes tem crescido, cada vez mais o público geral tem acesso a eles e é influenciado pelas decisões tomadas por tais sistemas. Isto torna mais necessário que tais sistemas sejam capazes de explicarem suas decisões para usuários leigos. *Beliefs-Desires-Intentions* é um modelo de agentes comumente utilizado. Ele pode ser bem complexo, uma vez que o agente possui um processo de deliberação interno para decidir quais objetivos ele irá perseguir baseado nas crenças que possui. Tal processo deliberativo é chamado de *seleção de objetivos*. Quando pessoas explicam coisas umas às outras, elas fazem uso de uma série de diferentes tipos de explicações. Um tipo de explicação muito comum é a *explicação contrastiva*, onde dois cenários são comparados e a explicação é apresentada como a diferença entre ambos. Desta forma, ao usar casos conhecidos e desconhecidos, é possível apresentar as causas que diferenciam os dois cenários. O interesse em inteligência artificial explicável tem crescido nos últimos anos, ainda assim poucos trabalhos são fundamentados em estudos das ciências sociais e cognitivas em como humanos geram e avaliam explicações. Assim sendo, esta dissertação busca identificar *quais informações precisam ser apresentadas para explicações contrastivas, baseado em achados das ciências sociais e cognitivas, e como gerar tais explicações no contexto da seleção de objetivos de agentes Beliefs-Desires-Intentions (BDI)*. Um método para gerar explicações contrastivas para seleção de objetivos baseada em BDI foi proposto, fundamentado nos trabalhos de Bouwel e Weber (2002) e Grice (1975). A estrutura das perguntas contrastivas propostas no trabalho de Bouwel e Weber (2002) servem de base para a forma das perguntas e respostas que o método proposto atende. O trabalho de Grice (1975) por sua vez estabelece requisitos para comunicação entre duas partes, no caso deste trabalho, um agente e um usuário, que estão cooperando. Tais requisitos, propostos por Grice na forma de quatro grupos de máximas (Quantidade, Qualidade, Relação e Modo), estabelecem restrições e boas práticas em relação às informações que são trocadas entre as partes. Ao basear-se nas máximas de Grice, espera-se que as explicações geradas sejam próximas a duas pessoas conversando e explicando algo entre si. O método gera um conjunto de *possíveis explicações*, onde cada uma é uma possível resposta, com seus respectivos conjuntos de informações relevantes. Um estudo de caso mostra como os cálculos das informações necessárias são feitos, e como os requisitos baseados no trabalho de Grice são abordados no método. O método atende a três das quatro máximas de Grice, uma vez que a máxima de Modo foi desconsiderada, já que depende da interação com o usuário, que está fora do contexto deste trabalho. As máximas de Qualidade, Relação e Quantidade são atendidas pelas formulações de cada tipo de pergunta utilizadas no primeiro procedimento. O segundo procedimento auxilia na satisfação da máxima de Quantidade. A seleção de uma única resposta precisa ser feito antes que a explicação possa ser apresentada para o usuário final. Tanto a seleção quanto a apresentação da explicação estão fora do escopo deste trabalho.

Palavras-chaves: agente bdi; seleção de objetivos; explicação contrastiva; inteligência artificial explicável.

ABSTRACT

As agent-based systems have been growing, more and more people have access to them and are influenced by decisions taken by such systems. This increases the necessity for such systems to be capable of explaining themselves to a lay user. The *Beliefs-Desires-Intentions* is a commonly used agent model. It can be fairly complex because an agent has an internal deliberation process to decide what goals it will pursue based on its beliefs. This deliberative process is called *goal selection*. When humans explain things to each other, they make use of a series of different types of explanations. One very common explanation type is the *contrastive explanation*, where two scenarios are compared and the explanation presents the differences between the cases. In such a way, by using a known case and an unexpected one, it is possible to present only the causes that differentiate both. Interest in explainable artificial intelligence has been increasing in recent years, yet few works are grounded on social and cognitive sciences studies on how humans generate and evaluate explanation. As such, this dissertation aims to identify *what information should be part of contrastive explanations, based on findings of social and cognitive sciences, and how to generate such explanations in the context of the Beliefs-Desires-Intentions (BDI) agent's goal selection*. A method for generating contrastive explanations for BDI-based goal selection was proposed, with groundings in the works of Bouwel and Weber (2002) and Grice (1975). The structure of contrastive questions proposed by Bouwel and Weber (2002) is used as a foundation for the questions and answers addressed by the proposed method. In turn, Grice's (1975) work provides requirements for the communication between two cooperating parties, in the context of this work, an agent and a user. Such requirements, proposed by Grice as four sets of maxims (Quantity, Quality, Relation and Manner), establish restrictions and good practices concerning the information exchanged between the parties. By basing the method on Grice's maxims, the generated explanations are expected to be closer to two people conversing and explaining some event among themselves. The method generates a set of *possible explanations*, such that each of them represents a possible answer, with its respective set of relevant information. A case study shows how the calculations of the required information are made and how requirements based on Grice's work are accounted for. The method addresses three out of four of Grice's maxims, as the maxim of Manner was disregarded since it is dependent on the user interaction, which is outside the scope of this work. The Quality, Relation, and Quantity maxims are addressed by each question type formulation used in the first procedure. The second procedure contributes to the satisfaction of the Quantity maxim. The selection of a single *explanation* needs to be done before presenting the answer to the final user. Both selection and presentation of the explanation are outside the scope of this work.

Keyword: bdi agent; goal selection; contrastive explanations; explainable artificial intelligence.

LIST OF FIGURES

LIST OF TABLES

LIST OF SOURCE CODE

LIST OF ABBREVIATIONS AND ACRONYMS

Acronyms

UTFPR	Universidade Tecnológica Federal do Paraná
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
AI	Artificial Intelligence
XAI	Explainable Artificial Intelligence
BDI	Beliefs-Desires-Intentions
FOL	First-Order Logic
FSM	Finite State Machine
DAG	Directed Acyclic Graph

LIST OF SYMBOLS

Notations

$\mathcal{L}$	A finite set of all literals defined in the language (in FOL)
$\mathcal{B}$	A subset of literals ($\mathcal{B} \subseteq \mathcal{L}$)
l	A single literal in $\mathcal{L}$
$\mathcal{R}$	A (finite) set of all rules in the language (in FOL), each being $\langle h, Body \rangle$
R	A finite set of rules (in FOL)
r	A single rule $\langle h, Body \rangle$
h	is the *head* of a rule, being a literal
$Body$	is the *body* of a rule, being a (finite) set of literals
$\mathcal{PN}$	The finite set of all predicate names in the language
$Pred$	A set of predicate names
$pred$	A predicate name
$term$	A term to be used in a predicate, being a constant or variable
$\mathcal{T}$	A theory (in FOL)
$\mathcal{F}$	The set of every possible formula constructed from $\mathcal{L}$
Γ	A set of sentences (or formulas, in FOL)
φ	A sentence (or formula, in FOL)
$\mathcal{I}$	A model of some Γ (in FOL)
A	An agent $A = \langle B, G, R, P \rangle$
B	A finite set of beliefs, each being a literal (in FOL)
b	A single belief
G	A finite set of goals, each being $\langle gid, st \rangle$
g	A goal $g = \langle gid, st \rangle$
GId	A finite set of goal identifiers, each being a literal
gid	is a unique literal (in FOL) identifying a goal
$\mathcal{S}$	A finite list of possible goal states
st	A single goal state
P	A finite set of plans, each being $\langle g, Gd, Act \rangle$
p	A single plan $p = \langle g, Gd, Act \rangle$
Gd	A finite set of guard clauses, each being a literal (in FOL)
Act	An ordered list of actions (undefined in this work)
I	A finite set of possible cycle indexes
i	A single cycle index
KB	A theory created with the agent's beliefs and rules
KB^+	The expanded knowledge base consisting of every sentence that entails from the agent's beliefs and rules

E	A finite set of events
e	A single event, being either a belief or a goal identifier
C	A finite set of causes
c	A single cause $\langle e, Cond \rangle$, where e is an event and $Cond$ a finite set of beliefs
H	The execution history
h	An entry in H, being $\langle i, G_H, B_H \rangle$
G_H	A finite set of changed goals, each being $\langle gid, st \rangle$, where gid is a goal identifier and st a goal state
B_H	is a finite set of changed beliefs, each being $\langle b, v \rangle$, where b is a belief and $v \in [ADD, REM]$ is the type of operation
RC	A finite set of related conditions, consisting of a set of presumptions or a single preference assertion
rc	a single related cause, being a single presumption or preference assertion
Gp	Is a non-directed graph, also called *explanation graph*
Ver	Is the finite set of vertices of Gp
Edg	Is the finite set of edges of Gp
$\mathcal{PE}$	Is the finite set of all *possible explanans* for a graph Gp
PE	Is a *possible explanans*, consisting of a set of related condition
Exp	Is the set of events in the *explanandum*

CONTENTS

1 INTRODUCTION

The usage of agents has been growing, ranging from simulation of natural and social phenomena (LIU *et al.*, 2014), autonomous robot controlling (ROBERTS *et al.*, 2015; FLOYD *et al.*, 2017), to agents on virtual environments (DANNENHAUER *et al.*, 2018; MOLINEAUX; DANNENHAUER; AHA, 2018). An agent is a computer system *situated* in some *environment* and capable of *autonomous action* in this environment to meet its delegated objectives (WOOLDRIDGE, 2009). With the growing complexity of the computational systems used in everyday life, comprehension and trust problems have increased, and agent-based systems are no exception.

Agent-based systems are concurrent, distributed, and often situated in dynamic environments (WINIKOFF, 2017). Agents can have different complexity levels, ranging from a reflexive agent that has a defined behavior for a given perception to agents that can deliberate about their perceptions and internal model of the world and, in doing so, choose the best course of action (BERMÚDEZ, 2014). This latter type of agent is what we call an *intelligent agent*. Intelligent agents can further increase the complexity of such systems since, in contrast to reflexive agents, their actions depend not just on their perceptions but also on their internal representation of their goals and environment. As such, the agent's ability to explain their practical reasoning – reasoning responsible for deciding what to do, thus, action-oriented – in its respective environment is advantageous for experts to model and evaluate the agent and users who interact or are affected by the agent.

An explanation can be described as the assignment of causal responsibility, as it presents possible causes for what is being explained (JOSEPHSON, 1994). What explanations are is further discussed in section 2.2.

Explanations need to take into account their targeted audience. When the audience is composed of people, basing the requirements of the explanation on how people generate and evaluate explanations for one another could improve their reception and understanding (MILLER, 2019). Why-questions are contrastive, following the form "*Why* P *rather than* Q*?*", where P is the *fact* to be explained, and Q is some *foil* case that was expected. Even when the question is posed as "*Why* P *?*", there is an implicit contrast case (MILLER; HOWE; SONENBERG, 2017). When people interact with a system, they will have different types of questions (HAYNES; COHEN; RITTER, 2009). Different explanations complement each other; under different circumstances, a different type of explanation is required.

Explanations of facts need to be differentiated from explanations of contrasts both in structure and in motivation (BOUWEL; WEBER, 2002). While explanations of plain facts are motivated by curiosity or to better understand the circumstances in which an event can happen, explanations of contrast present the differences in the causal history between the fact and the contrast case. Bouwel and Weber (2002) present a structure for three different types of contrastive questions: Property-contrast (P-contrast); Object-contrast (O-contrast); and Time-contrast (T-contrast). Together with these three structures, the authors present one instance of an adequate answer for each question type, with its respective structure. Van Bouwel and Weber's structures of contrastive questions and explanations are further discussed in the subsection 2.2.2.

It is worth mentioning that a contrastive explanation is not the same as a contrastive question. A contrastive question poses a comparison, but it does not require that such comparison to be used in the answer. An example of work that presents contrastive questions, but not contrastive answers, is (WINIKOFF, 2017), where why-questions are used, but no contrast is employed on the answer. A contrastive explanation/answer necessarily presents the differences between the elements being compared. Those explanations are usually more simple and shorter.

Since the explanation process is an exchange of information most likely involving a person, it should follow the rules of conversation. Grice (1975) presents the Cooperation Principle with four categories – Quantity, Quality, Relation and Manner –, each with a set of maxims that two entities in a cooperative dialogue are expected to follow. The "Quantity" category is of particular interest in this work. Its maxims state that one should be as informative as necessary, but not more than that. Further discussion about Grice's Cooperation Principle is presented in the section 2.2.5.

Interest in explainable artificial intelligence (XAI) grounded in social and cognitive sciences studies has grown recently (MILLER; HOWE; SONENBERG, 2017), as well as contrastive explanations (STEPIN *et al.*, 2021). Despite such growth, relatively few works approach explanations directed to the general public with such grounding (KAPTEIN *et al.*, 2017; STANGE; KOPP, 2020; HARBERS; BOSCH; MEYER, 2010). Besides, when focusing on an explainable agent's goal selection, the process by which an agent selects which goals it is going to pursue, to the best of our knowledge, no contrastive explanation generation approach was found. However, there are some related approaches for generating ontological and causal explanations (WINIKOFF, 2017; MORVELI-ESPINOZA; POSSEBOM; TACLA, 2019).

This work tackles the explainability problem in belief-based intelligent agents' goal selection process. It proposes a method for generating contrastive explanations constructed from the agent's execution log. The method is based on Van Bowel and Weber's structure of contrastive explanations and Grice's Cooperation Principles as requirements for the quality of explanations. The method does not present a final, user-ready explanation. It provides a set of possible explanations, where each possible explanation is a set of conditions required to attribute a causal relation to the contrastive question posed. The agent is considered to be cooperative and honest. It does not lie or withhold any information from the user (except for the simplicity of the explanation). Next is presented the motivation, followed by the problem and research goals, and finishes with this dissertation's structure.

1.1 Motivation

Miller et al. (2019) writes of how specialists have built explanations in artificial intelligence (AI) with little to no study of the requirements for a broader audience. They made an analogy of this situation as the “inmates running the asylum”. As such, many of the "explainable" and "transparent" approaches still require understanding some of the internal mechanics of the system in use.

Each day more commonly available systems employ complex AI techniques, and the decisions of those algorithms can have profound impacts on people’s lives. The explanations generated from these systems should aim at the general public.

Having the general public as consumers of the explanations of such systems raises new challenges in how to convey explanations in an accessible way. To that end, the studies of how people generate and evaluate explanations serve as a good foundation for explaining complex systems (MILLER; HOWE; SONENBERG, 2017).

Besides the lack of grounding in cognitive and social sciences, Haynes et al. (2009) show in their study that during people’s interaction with a system, different types of questions are posed, and those different types of questions require different types of answers.

Contrastive explanations are commonly employed by people and can bring benefits to the exchange process during the explanation. Contrastive questions provide an insight into the questioner’s mental model, allowing to have a better understanding of what they do not know, and contrastive explanations usually are more straightforward, more feasible, and less demanding both for the questioner and explainer (MILLER, 2021).

Although an increasing amount of works related to contrastive explanations have been published in the last few years, there are still few frameworks that account for contrastive explanations, especially those grounded on social and cognitive sciences findings (STEPIN *et al.*, 2021).

Given the high complexity of an agent’s goal selection process, non-contrastive explanations can become too complex for a lay user. On the other hand, contrastive explanations can provide simplicity by using the knowledge the user already has about the agent and only complementing the gaps in the explanation, like many humans explain things to one another. For example, when asked “Why the barn cough fire?” a human explainer will say that there was a short circuit and some inflammable material near it. However, the explainer will omit the fact that there was oxygen in the barn, as it is common knowledge, there is no necessity in stating it, even though without oxygen, there would be no fire. By comparing the barn that coughs fire to an ideal barn, the explainer can provide explanations targeted to the gaps of knowledge or misconceptions of the explainee.

1.2 Problem and Research Objectives

The Belief-Desire-Intention (BDI) is one of the most known agent models and is widely used. According to Georgeff *et al.* (1999), *beliefs* represent the state of the world, *desires* represent some end state that an agent deems as desirable, and *intentions* are desires that the agent is currently committed to, with their respective plans (means to achieve the desired state). The agent must express some resistance to dropping an intention out but also must be capable of such when the conditions of the world change. Another commonly used concept is *goals*, which are a state of affairs that is desirable and that the agent seeks to achieve. In that sense, both desires and intentions are goals, and when the term “goal” is employed, it refers to both.

People commonly attribute mental attitudes to complex systems to better understand their behavior. The BDI model already incorporates mental attitudes; these are the agent’s *beliefs*, *desires*, and *intentions*. This positively affects the interpretability of the system. Many works deal with various aspects of agent explanation, ranging from beliefs, plans, and actions; however, few of them have tackled the goal selection process (WINIKOFF, 2017; MORVELI-ESPINOZA; POSSEBOM; TACLA, 2019).

At least there are three types of different answers: ontological, causal, and contrastive. Different motivations for asking a question require different answers. As such, no single answer type can fulfill every need of the requested explanation. Again, many works on XAI provide ontological and causal answers (KAPTEIN *et al.*, 2019; STANGE; KOPP, 2020; HARBERS; BOSCH; MEYER, 2010; FAN, 2018). In turn, contrastive answers are just starting to grab the researcher’s attention (STEPIN *et al.*, 2021).

In order to bridge the gap of lack of grounding in social and cognitive sciences and lack of contrastive explanations, a contrastive explanatory model for BDI-based agents goal selection is required. From the social sciences area, Van Bouwel and Weber (2002) proposed a general structure of contrastive questions and the requirements for an adequate explanation, which can be adapted to explain an agent’s behavior. For this work, the agent is considered to be fully cooperative, and does not withhold any information or lies. As such, Grice’s (1975) Cooperation Principles are employed as guidelines for the explanation generation.

Still, the problem of answering a question is context-dependent. There are many valid explanations for a single event, as people do not list every cause for the event (which would be even impossible in many cases) but present a subset of causes. The subset is selected according to the explainee and explainer’s intentions, background knowledge, applicable roles, and many other aspects. Besides, how to present the explanation is another challenge, not just on how to structure a textual explanation but also if a purely textual approach is the best option.

As such, the following research question is posed:

Grounded on social and cognitive sciences works, what information is required to construct contrastive explanations for BDI-based agent’s goal selection process and how to generate such contrastive explanations?

The lack of social and cognitive science grounding on XAI approaches is addressed by employing Van Bouwel and Weber's work to derive the proposed explanation method in this work.

It is expected that by using the agent's beliefs and the causal relationship between them and their goals, contrastive explanations can be constructed.

Against this background, the main goal of this work is to: propose and evaluate a method for BDI-based agents capable of generating contrastive explanations for queries related to the goal selection process; such explanations are comprised of a set of beliefs that are causally related to the goal being questioned, grounded on Bouwel and Weber's work.

To achieve such a goal, the following tasks were set:

- Adapting Van Bouwel and Weber structure of contrastive questions to a BDI-based goal selection method.
- Proposing a method to satisfy Grice's Quantity maxims in the generated explanations.
- Evaluating the method in a case study.

1.3 Structure of This Work

The next chapter presents the previous works in the area, containing a brief review of agents and the Belief-Desire-Intention model and a review of explanations from works on social sciences that are used as the foundation for the method proposed.

Chapter 3 presents the proposed method of explanation generation, that is, how an *explanans* (the explanation for a given event) is generated for a given *explanandum* (the event, or events, to be explained)[1]. It starts with an overview of the method, followed by the formalization of the question types and their respective adequate answers, and lastly, the method for generating *possible explanans*. Chapter 4 follows up with a case study of the method, based on the Cleaner World Scenario.

Chapter 5 concludes with some of the limitations of the proposed method, the related work, and explores some future directions for improving the proposed method.

[1] Both *explanans* and *explanandum* are defined in subsection 2.2.3

2 LITERATURE REVIEW

This chapter presents the background upon which this work is built. First, core concepts about agents are presented. Next, explainability in artificial intelligence, focusing on explanations for agents and contrastive explanations.

2.1 Goal-Based Agents

Intelligent Agents is a well-studied topic in artificial intelligence. The following concepts were obtained from (WOOLDRIDGE, 2009). An agent is a computer system *situated* in some *environment* and capable of *autonomous action* in this environment to meet its delegated objectives. The core concepts are that an agent is inserted in some environment and performs actions autonomously in such environment to accomplish its objectives. In reasonably complex domains, the agent will not have complete control over the environment, being able simply to influence it.

In order to interact with the environment, the agent has some way of sensing the environment and some way of acting in it. Besides those two elements, there is some decision mechanism that rules the agent's behavior.

It is a common practice to attribute agents with mental states, like beliefs, desires, wishes, hopes, and so on. This approach is called *intentional stance* (WOOLDRIDGE, 2009). It helps us when dealing with complex behaviors that are hard to make sense of, by relating those agents to a human-like behavioral motivation.

The specific kind of agent that this work is interested in falls under the category of *practical reasoning* agents. Practical reasoning is the reasoning directed to action. There are at least two activities related to this type of reasoning for humans: deciding what to do, called *deliberation*; and deciding how to do it, called *means-end reasoning*. When an agent chooses a course of action and commits itself to its pursuit, it is called an *intention*. Intentions play the following roles in practical reasoning: they drive means-end reasoning; persist, in the sense that an intention is not easily dismissed; constrain future deliberation; influence beliefs upon which future practical reasoning is based.

The BDI model is a well-known and adopted model for practical reasoning agents. In (RAO; GEORGEFF, 1995), the authors present an abstract architecture for an interpreter of the BDI model. It consists of the initialization, three deliberative steps, the execution of a goal, and three knowledge revision steps. The interpreter pseudo-code can be seen in Figure 1.

The *initialize-state()* is responsible for initializing the data structures and other necessary elements. The *option-generator(event-queue)* is responsible for enumerating the possible options the agent has, based on the *event-queue*. The *deliberate(options)* selects a subset of the *options* that the agent must pursue. The *update-intentions(selected-options)* add the selected subset of options to the intention structure. In the *execute()* step, the agent executes an atomic action that is available (if any). The *get-new-external-events()* adds any events that happened

Figure 1 – BDI-interpreter pseudo-code.

```
BDI-interpreter
initialize-state();
repeat
    options := option-generator(event-queue);
    selected-options := deliberate(options);
    update-intentions(selected-options);
    execute();
    get-new-external-events();
    drop-successful-attitudes();
    drop-impossible-attitudes();
end repeat
```

Source: (RAO; GEORGEFF, 1995).

during the cycle execution to the event-queue. The *drop-successful-attitudes()* clear any satisfied desires and intentions. Lastly, the *drop-impossible-attitudes()* clear any impossible desires and intentions.

Castelfranchi and Paglieri (2007) proposed an extension of the BDI model, called Belief-Based Goal Processing (BBGP), that increases the dependency on beliefs. In it, the agent's decision to change the state of a goal can be defined entirely as a function of its beliefs. They define a four-stage model: a) **activation** stage is where the agent starts a goal based on its motivating beliefs; b) **evaluation** stage is where the agent evaluates if there are any beliefs that a goal should not be pursued; c) **deliberation** stage is where the agent checks for conflicting goals and selects a subset of preferred non-conflicting goals; and d) **checking** stage is where the agent evaluates if there are means-end beliefs to achieve a goal. The goal is required to pass through each stage, in order, to become ready to be executed.

Morveli-Espinoza et al. (2019) proposed a computational formalization of the BBGP model based on computational argumentation. It uses rules to decide when a goal should be allowed to advance its state. A rule in the context of the agent's stage rules is a set of premises and a conclusion. The set of premises must be composed entirely of beliefs. The conclusion of the rule is a single literal representing a goal. The conclusion holds if all premises also hold. Besides the stage rules, standard rules (with beliefs in the conclusion) are also part of the model; they just are not directly responsible for the goal progression, but can be chained with stage rules and interact with goals in that way. When a goal is selected to progress to the next stage, it is called *Active*, *Pursuable*, *Chosen*, or *Executive*, according to the stage. Before a goal becomes *Active*, it can be considered in a special state, that can be seen as the potential function of every possible goal. Any possible goal before its instantiation is in this state called *Sleeping*.

2.2 Explainable Artificial Intelligence

In (MILLER, 2019), the author presents a review of about 250 works from the social, psychology, and cognitive sciences areas, from the perspective of Explainable Artificial Intelligence (XAI). His work lays a foundation for building explainable systems that are based on the finds of how people make their explanations and their preferences. It is used as a guide for selecting the following works, where the criteria for a good explanation are defined.

2.2.1 Types of Questions

An explainee can have a wide range of motives for posing a question, and those different motives define the information that better fulfills its needs. In (HAYNES; COHEN; RITTER, 2009), the authors present an explanatory framework for intelligent agents, constructed from a cross-disciplinary review. The explanations are grouped by the kind of information required for answering them:

- Ontological explanations: these request information about the state of the event being explained, its properties, existing instances, or information about the concepts themselves (e.g., "What does X mean?");
- Mechanistic explanations: these request information about the causes and effects of the event being explained (e.g., "What are the causes of X?");
- Operational explanations: these present instrumental or procedural information related to goals and the means to achieve them (e.g., "What are the plans for achieving X?").
- Design rationale explanations: these seek to present the intended purpose of the event being explained (e.g., "What is the goal behind action X?").

In order to construct *ontological explanations*, it is necessary to provide information about “direct” properties. These properties are called *direct* because they are answered by accessing the agent’s data to provide instances, values, spatial relations, or information about some event. For *mechanistic explanations*, the agent’s causal history and model of causation are required. In turn, *operational explanations* require information on how a certain behavior can be achieved, which relates to a model of causation of the agent. Lastly, *design rationale explanations* require a model of the agent to answer what is the purpose of the agent, its plans and actions, what are the rules that it follows, and the relation between entities and/or events.

Haynes et al. (2009) evaluated how a user interacts with an intelligent agent and measured the frequency of explanations requested by type. They found that the most requested categories, in decreasing order, are: *Ontological* (58%), *Mechanistic* (19%), *Operational* (12%), and *Design rationale* (11%).

In this work, the focus is on contrastive questions in the context of goal selection, where a comparison between some similar goals is made. The causal history and model of causation are used to compare the goals. The types of questions that are answered fall in a subset between the *mechanistic* and *operational* explanations.

2.2.2 Contrastive Questions

Contrastive questions imply the contrast between the two scenarios referenced in the question. In that sense, they can be seen as *why-questions*. In (BOUWEL; WEBER, 2002), the authors distinguish four types of explanatory questions:

- (*plain fact*) Why does object a have property P?
- (*Property-contrast*) Why does object a have property P, rather than property P'?
- (*Object-contrast*) Why does object a have property P, while object b has property P'?
- (*Time-contrast*) Why does object a have property P at time t, but property P' at time t'?

The authors claim that explanations of facts should be distinguished from explanations of contrasts, both in structure and in motivation.

Explanation of facts shows the causes of the observed fact, providing a non-interrupted causal chain that ends with such fact. The motivation for such explanations can be for pure curiosity or to acquire information to predict if and in what conditions a similar fact can happen.

In turn, an explanation of contrasts provides information about the features that differentiate the factual case from its alternative. The motivation behind this type of explanation can be to evidence causes that help to achieve an ideal (P-contrast) or to remove observed differences (T- and O-contrast), and also to tell us why things had a different outcome than expected.

Van Bouwel and Weber (2002) then define how to present an adequate explanation to each of the mentioned contrastive questions:

- (*Property-contrast*) The contrast of properties can be answered as: “Object a has property P, rather than P' because it does not have properties $\{D_1,...,D_n\}$.", where $\{D_1,...,D_n\}$ are **absent** properties that would have guarantied a to acquire property P'.
- (*Object-contrast*) The contrast of objects can be answered as: “Object a has property P, while object b has property P' because a has properties $\{D_1,...D_n\}$ which object b does not have".
- (*Time-contrast*) The contrast of time can be answered as: “Object a has property P at t_1 but P' at t_2 because it had properties $\{D_1,...,D_n\}$ in the relevant time interval preceding t_1, while these properties were absent in the relevant time interval preceding t_2".

It is worth emphasizing that the authors recognize that the three mentioned types of contrast are not the only ones and that the answer model is not the only adequate way to answer the posed questions either.

Contrastive explanations are complementary to ontological and causal types of explanations. The agent's observed behavior is a direct consequence of the goal selection process and can be very complex. The ability to present to the explainee a more straight answer that makes it easier to ground the new information to a previously known/expected knowledge is very enticing. It is worth mentioning that causal and ontological explanations are still necessary to bridge knowledge gaps in the explainee understanding.

2.2.3 Explanations

In (MILLER, 2019), the author presents a definition of explanation, where an explanation is both a process and a product, and argues that an explanation consists of two processes and a product. These being:

- A cognitive process: where abductive inference is used for filling the gaps to determine an explanation of a given event, which is called *explanandum*.
- A product: the explanation resulting from the cognitive process, also called *explanans*.
- A social process: where the transference of the information between an explainer and an explainee.

This definition sets the big picture, where there is an internal process of devising the causes of a given event, determining which one is the most adequate, that is, the explanation of choice, and lastly, relaying the explanation to the targeted audience. This last step is highly dependent on the explainer and explainee. For instance: the interaction is face-to-face or is conveyed through a media, the level of expertise of the explainee on the subject, the role of the explainee in relation to the subject. This process of conveying the explanation is outside the scope of this work.

Miller (2019) points out four main finds in his work:

- Explanations are contrastive – "[P]eople do not ask why event P happened, but rather why event P happened *instead of* some event Q."
- Explanations are selected, in a biased way – "Humans are adept at selecting one or two causes from a sometimes infinite number of causes to be the explanation. However, this selection is influenced by certain cognitive biases."
- Probabilities probably do not matter – "The most likely explanation is not always the best explanation for a person, and importantly, using statistical generalizations to explain why events occur is unsatisfying, unless accompanied by an underlying causal explanation for the generalization itself."

- Explanations are social – “[Explanations] are a transfer of knowledge, presented as part of a conversation or interaction, and are thus presented relative to the explainer’s beliefs about the explainee’s beliefs.”

Miller (2019) argues that “[r]esearch shows that people do not explain the causes for an event *per se*, but explain the cause of an event *relative to some other event* that did not occur". With that, an explanation is always of the form "Why P rather than Q?", where P is the fact that requires explanation and Q is the fictional contrast case (a.k.a. *foil*). Sometimes the foil can be implicit in the question. The fact and the foil must share a broadly similar history; otherwise, where to begin answering the question could not be determined. For instance, if asked "‘Would John be run over if Bob had not drunk driving?”, at first glance, the question makes sense, but if Bob was not the one to run over John, it became very hard to determine a causal relation. Theoretically, if traced far back enough in time, any two events share some causes.

In the context of goal selection, an agent can save the history of its deliberative process, guaranteeing that the explanation is deterministic. The social aspect is also disregarded, only determining the set of relevant information is tackled, not how to select and relay such information.

Next, how explanations are selected is reviewed.

2.2.4 Preferences in Explanations

When providing a causal explanation to an event, the number of causes that can be ascribed is infinite. According to Hesslow (1988), there are three reasons for that. Firstly, the dependence of the event on immediately previous occurrences. Secondly, the ability to trace a causal chain backward in time. And thirdly, the cause can be conceptualized in infinitely many different ways. Yet when we provide a causal explanation, we present only a few causes, sometimes a single one, to the event’s occurrence, even if other causes exist. This brings out the challenge of how to select an appropriate cause for any given event?

Together with his proposed model, Hesslow (1988) presents a listing of what he called the main approaches to the explanation selection problem:

- *Unexpected condition*: based on the idea that expected conditions – a condition being some property of the world, e.g., availability of oxygen, a defective component, someone being late for an appointment – are omitted in the explanations, as they can be understood without being said. The conditions that are expected are suppressed, and the explanation focuses on the unexpected conditions that took place in the causal history of the event.

- *Precipitating causes*: based on a temporal aspect of the explanation. We tend to select conditions immediately preceding the event being explained.

- *Abnormal conditions*: similar to unexpected conditions, abnormal conditions focus on conditions that, as the name implies, are abnormal. It differs from the unexpected conditions as a condition can be unexpected but considered normal (e.g., the mail was not delivered today, but I had forgotten that it is a holiday). An abnormal condition must deviate from the normality (e.g., a defective cog can cause a manufacturing defect).

- *Variability*: the selection of conditions that are more variable than the others. Hesslow defines it as a blend of the previous three.

- *Deviation from the theoretical ideal*: based on the idea that a theoretical ideal may be used as a guide to identifying deviations. A couple of examples of a theoretical ideal are: a healthy human being, a perfectly running market (for economics), Newtonian law of motion. It differs from the abnormal conditions since no assumption that the theoretical ideal is normal needs to be made.

- *Responsibility*: rooted in the idea that we identify the cause before assigning the blame for the event. As such, the conditions selected are those that deviate from what is considered good, reasonable, or appropriate.

- *Predictive value*: based on the odds that the condition predicts the event, as such, if the event is more probable to occur when a condition a occurs than when another condition b, then a is selected as a cause for the event.

- *Replaceability and necessity*: the strength of a condition as explanatory for the event is pondered in relation to its necessity. If a condition is considered replaceable, its importance decreases in relation to the others.

- *Instrumental efficacy*: based on the idea of controllability of the conditions. If a condition can be manipulated, it then is chosen as a more suitable explanatory cause than a condition that can not.

- *Interest*: the idea that we chose a cause not by a shared sense of what is normal or by some sort of rule, but based on personal preference.

On the premise that none of the above-mentioned selection criteria could be used in every case and that choosing a selection criteria would be a recursive approach to the causal selection issue, Hesslow (1988) proposes his contrastive approach, having similar ideas to what Lipton (1990) would present closely afterward. A main aspect of the approach is that the *explanandum* (the event that needs to be explained) is composed of an *object a*, another *object of comparison b*, and a *property E* that is being compared, such that *a* has *E* and *b* does not. The comparison narrows down the range of possible conditions that can explain the event *a*. Hesslow concludes his approach with criteria for selecting the cause among the conditions that explain the *explanandum* based on the *explanatory relevance* of the condition.

Miller (2019) presents in his work some examples of studies evaluating the selection criteria used by people.

2.2.5 Grice's Conversation Rules

When two parties are engaged in a cooperative activity, their conversation can be expected to follow some rules (GRICE, 1975). Grice called these the Cooperative Principles and categorized them into four groups of maxims:

- Quantity
  - Make your contribution as informative as is required.
  - Do not make your contribution more informative than is required.
- Quality
  - Do not say what you believe to be false.
  - Do not say that for which you lack adequate evidence.
- Relation
  - Be relevant.
- Manner
  - Avoid obscurity of expression.
  - Avoid ambiguity.
  - Be brief.
  - Be orderly.

Let us evaluate how each category relates to a BDI-based cognitive agent.

The "Quantity" category is of particular interest in this work, as it is the one set of maxims that can not be expected to be trivially achievable. Besides defining all possibly related conditions to the message the agent wants to convey (in the scope of this work, an explanation), the agent needs to define a subset of such conditions such that all the events being explained are covered by the minimal amount of conditions as possible. The section 3.3 presents a method for constructing subsets of explanation conditions, where each one follows the "Quantity" maxims. Which subset is the most adequate is related to the agent's domain, and several criteria can be used. This work does not try to answer how to select the best subset for the explanation. Section 2.2.4 highlights some criteria of preference that can be used in the explanation selection.

Considering the cooperative agent, the "Quality" maxims are straightforward in this work. The agent should have a clear notion of what it deems to be true, false, or undecidable. As such, it should be simple to give only information considered true.

The "Relation" category can be met using only information that has some causal relation to the message being relayed and the request. Lastly, the "Manner" is closely related to how the message is presented, if it is clear and follows a meaningful order. Different cases require different approaches to agent-person communication. It is outside of the scope of this work to delve into the intricate subject of how to present any given explanation.

3 EXPLANATIONS FOR GOAL SELECTION

In this chapter, a method for generating explanations for the goal selection process of intelligent agents is proposed. The method has two very distinctive stages: a) generating the set of conditions (in the BDI-based agent context of this work, the conditions are beliefs) that are possible partial answers to the posed question (the *explanandum*); b) generating subsets of conditions that cover all the events (in this context, events are goals achieving a certain state) being explained. The subsets generated in b) are possible *explanans*, where the most adequate one still needs to be selected. The selection of the *explanans* is not covered in this work. A review of some applicable criteria is presented in subsection 2.2.4.

3.1 Method Architecture

The proposed method takes a few assumptions about the agent that is being explained:

- First, the agent is based on the BDI model, and as such, the beliefs have a central part in the goal selection process.
- Second, the relation between the beliefs and the goals that take place within the goal selection can be deemed as a causal relationship.
- Third, the agent uses first-order logic to represent his beliefs and rules.
- Fourth, the focus of the explanations is on the most basic beliefs, that is, those beliefs that are deemed true by themselves, without the need for supporting beliefs. The reasoning for this is that the beliefs can usually be manipulated by a user without the need for technical expertise about agents. For example, knowing that the agent believes that the path is obstructed, the user can move an object so that a robot identify the path as clear, or if the agent is capable of conversation, the user can change the agent's belief using the interface of communication (e.g., text or voice).
- Lastly, the agent has a cooperative stance, and for that reason, he follows the Conversation Maxims to the best of his ability.

Predicate logic, or First-Order Logic (FOL), is a formal logical language that uses predicates and terms to describe its facts. Terms can be constants or variables. Let $\mathcal{L}$ be a finite set of all literals defined in the language, $\neg, \wedge, \vee$ be the logical negation, conjunction, and disjunction, respectively. An atom is denoted by $pred(term_1, term_2, ..., term_n)$, where $pred$ is the predicate name, and $\{term_1, term_2, ..., term_n\}$ are terms. A literal is an atom or its negation. A rule is denoted by $\langle x, \{x_1, x_2, ..., x_n\}\rangle$ (where $x, x_1, x_2, ..., x_n$ are literals), assume that the rule name is r, $HEAD(r) = x$ is the head of the rule and $BODY(r) = \{x_1, x_2, ..., x_n\}$ the body. An atom, literal, rule or formula is said to be *grounded* iff. every term in it is a constant. The following definitions were adapted from (AMGOUD; BESNARD, 2018):

Definition 1. ***(Theory).*** *Let $\mathcal{L}$ be the finite set of all literals in the language, a theory $\mathcal{T}$ is a tuple $\mathcal{T} = \langle \mathcal{B}, \mathcal{R} \rangle$ where $\mathcal{B} \subseteq \mathcal{L}$ is a set of literals, and $\mathcal{R}$ is a set of rules.*

A sentence φ is said to be a logical consequence of a set of sentences Γ (in symbols: $\Gamma \models \varphi$) if and only if there is no model $\mathcal{I}$ in which all members of Γ are true, and φ is false.

An agent in the context of this work is an artificial entity, embodied or not, that possesses beliefs, represented as literals, goals, represented as atoms, and rules, all expressed in first-order logic (FOL).

The temporal aspect of the agent execution is captured by the cycle identification. A cycle is considered to be a single iteration of the deliberative process, and an ordered number is used to identify it. The agent have an ordered sequence of tests that a goal must pass in order to be executed. After each test, the goal is said to achieve a state (e.g. for BDI models *Desire* and *Intention*).

Definition 2. ***(Agent).*** *An agent A is a tuple $A = \langle B, G, R, P \rangle$, where B is the finite set of beliefs, expressed as literals in FOL, the agent deems true, G is the finite set of goals, R is the finite set of rules, and P is the list of plans. Let $g \in G$ be a goal such that $g = \langle gid, st \rangle$, where gid is a unique literal representing a goal g, and st is the state of goal g. Let $r \in R$ be a rule such that $r = \langle h, Body \rangle$, where h is the head of the rule and can be a literal representing a belief or goal, and $Body$ is a set of beliefs that constitutes the premises of rule r. Let $p \in P$ be a plan such that $p = \langle g, Gd, Act \rangle$, where g is a goal (potentially not grounded atom), Gd is the set of guard clauses that act as a precondition for the plan execution and are expressed as literals, and Act is an ordered list of actions that, if successful, allows the agent to achieve the goal g.*

By isolating an agent's beliefs and rules, it is possible to create a theory. This theory (Definition 1) is referred to as *knowledge base* in the rest of this work.

Definition 3. ***(Agent Knowledge Base).*** *Let $A = \langle B, G, R, P \rangle$ be an agent, where B is its set of beliefs, G is the set of goals, R is the set of rules, and P is the set of plans, the agent's knowledge base is defined as $KB = \langle B, R \rangle$.*

For this method, the nature of the agent's plan library and the actions list are not relevant. The goal and the guard clause are the only required information.

Since a causal relation is drawn between the beliefs and goals, the rules and plans of the agent knowledge base can be used to define this relationship. Section 4.2 presents an example of how that can be achieved, but the agent's implementation and model may influence the process. A cause in this method is a relation between a set of beliefs and a single event, which is either a belief or a goal.

Definition 4. ***(Cause).*** *Let KB be the agent's knowledge base, C be the finite set of causes, B be the set of the agent's beliefs, GId be the set of the agent's goal identifiers, and E be the finite set of events such that $E = \{B \cup GId\}$. A cause $c \in C$ is a tuple $c = \langle e, Cond \rangle$, where $e \in E$ is the event, and $Cond \subset B$ is a set of conditions, if $KB \models Cond$ then event e is expected to happen. Cause c is said to be a* ***cause for*** *e. A belief $b \in Cond$ is said to be* ***causally related to*** *e. If $\forall cond \in Cond$ such that $KB \models cond$ then the cause c is said to be* ***activated****, in contrast, if $\exists cond \in Cond$ such that $KB \not\models cond$ then cause c is said to be* ***deactivated****.*

When two events share a piece of causal history, they are said to be *causally related*. The causal history of an event is called a *causal tree*.

Definition 5. ***(Causal tree).*** *Let C be the set of causes, $c = \langle e, Cond \rangle$ and $c' = \langle e', Cond' \rangle \in C$ be two causes, c' and c are said to be in a causal tree if $e' \in Cond$, or if there is another cause $c'' = \langle e'', Cond'' \rangle \in C$ such that $e'' \in Cond \wedge e' \in Cond''$.*

Note that the agent's *knowledge base* is considered to be acyclic, that is, there is no sequence of rules $\langle r_1, r_2, ..., r_n \rangle$, such that:

- $head(r_i) \in body(r_{i+1})$
- $head(r_n) \in body(r_1)$

Definition 6. ***(Causally related).*** *Let C be the set of causes, $c = \langle e, Cond \rangle$ and $c' = \langle e', Cond' \rangle \in C$ be two causes, they are said to be causally related iff. c and c' are in a causal tree (Definition 5). Such relations in denoted by $related(c, c') = [True, False]$.*

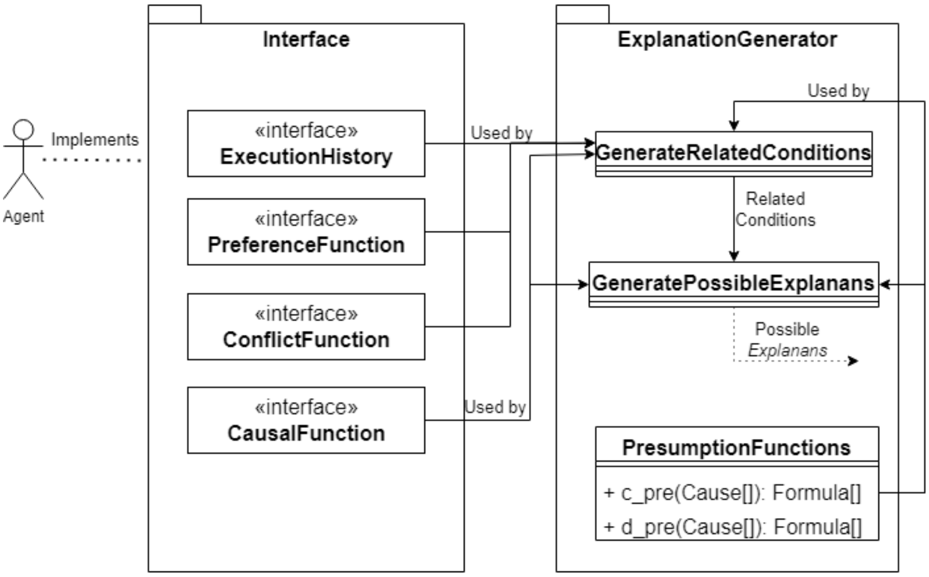
The set of *causes* describes a general relation from beliefs to another belief or goal. This relation is extracted from the agent's knowledge base. The most common sources of causal relations are rules and plans, but more complex agent models can have other elements that should be mapped to causal relations. Rules and plans can be converted to causes as follows:

Definition 7. ***(Plan to cause conversion).*** *Given a plan $p = \langle g, Gd, Act \rangle$, a cause c can be built as $c = \langle g, Gd \rangle$, assuming that none of the actions of Act fails.*

Definition 8. ***(Rule to cause conversion).*** *Let B be the set of beliefs, and GId be the set of goal identifiers. Given a standard rule $r = \langle h, Body \rangle$, a cause c can be built as $c = \langle h, Body \rangle$, iff. $h \in \{B \cup GId\}$.*

To generate the mentioned explanations, it is necessary to reproduce the agent's knowledge base when the events being explained happened. As such, a memory of the execution needs to be kept. Four components are necessary for the proposed method, as shown in Figure 2. It depicts the interface the agent needs to implement and the explanation generator. In the interface, four elements are required: i) the *execution history* is a log-like ordered list of events, the minimum data requirements of the entries are explored next; ii) the *preference function* encodes a preference relationship between two events; iii) the *causal function* is responsible for mapping events to the set of conditions that are causally related to it; finally, iv) the *conflict function* responsible for identifying if two given goals have any sort of incompatibilities between them. In turn, the explanation generator has three domain-independent elements: a) the *presumptions functions* that, given a set of beliefs, return the subset containing only the presumptions (beliefs that need no support); it is not part of the interface since it is model-independent; b) the *generate related conditions* procedure is responsible for receiving the posed question, retrieve the required information using the interface defined functions and the *presumption function*, and outputs the set of **related conditions**; lastly, c) the *generate possible explanans* procedure takes as input the set of *related conditions*, and outputs sets of **possible *explanans***, where each possible *explanans* is a subset of the related conditions that are related to every event in the *explanandum*. Procedures b) and c) are formally defined in subsections 3.2 and 3.3, respectively.

Figure 2 – Interface scheme for the explanation generation method.



The four elements in the "Interface" need to be implemented by the agent. The elements in the "ExplanationGenerator" are domain-independent. After the possible *explanans* are generated, one needs to be selected. This selection is not covered in the method.

Source: Author's own.

Each entry in the execution history needs to encapsulate, for each reasoning cycle, the set of goals that changed state and the set of beliefs that were added or removed from the agent's knowledge base. Note that the goals changes happened during the deliberation, but the beliefs changes are updated after the deliberation has finished. That way, for a cycle i, the reconstructed knowledge base needs to apply the changes only up to cycle $i - 1$.

Definition 9. ***(Execution history entry).*** *Let B be a set of beliefs, GId a set of goal identifiers, $\mathcal{S}$ be a list of possible state of a goal, I be the set of possible cycle indexes, H is the execution history, and h an entry from the execution history such that $h = \langle i, G_H, B_H \rangle$, where $i \in I$ is the cycle identifier, the set of changed goals $G_H = \{\langle gid, st \rangle$ such that $gid \in GId$ is the* *identifier of a goal and $st \in \mathcal{S}$ the new value of its state $\}$, and the set of changes in beliefs $B_H = \{\langle b, v\rangle$ such that $b \in B$ and $v = [ADD, REM]$, representing the beliefs addition or removal, respectively $\}$.*

The causal function maps each event to its conditions for every causal relation that can be ascribed to the agent's knowledge base. Note that this function returns causes even if their set of conditions is not currently satisfied by the agent's KB.

Definition 10. ***(Causal Function).*** *Let B be a set of beliefs, and GId a set of goal identifiers, $E = \{B \cup GId\}$ be the set of possible events that the agent can explain, and C the set of every cause modeled in the agent's interface. $causes : E \rightarrow 2^C$ is a function that maps an event $e \in E$ to the set of every cause related (Definition 4) to it, such that $causes(e) = \{< e_x, Cond_x > \mid < e_x, Cond_x > \in C \wedge e_x = e\}$.*

The preference function represents the precedence of a goal over another. This relationship is defined in the agent's model, by the developer. If a preference relation is not defined in the agent's model, the modeler must declare the selected goals as preferred over the non-selected ones in the function. If necessary, the cycle can be included as a function parameter so that changes in preference over time can be represented.

Definition 11. ***(Preference Function).*** *Let GId be the set of goal identifiers, I be the set of possible cycle indexes, $gid_1, gid_2 \in G$ be two goal identifiers, and $i \in I$ be a cycle index, $preferred : GId^2 \times I \rightarrow [-1, 0, 1]$ such that $preferred(gid_1, gid_2, i) = [-1, 0, 1]$, where $-1, 0, 1$ represents "gid_1 is less preferred than gid_2", "gid_1 is equally preferred than gid_2", and "gid_1 is preferred than gid_2", respectively. If the agent's model uses a fixed goal preference the cycle index i is **optional**.*

The *preference function* complexity is dependent on the agent's model. For instance, if the model has no proper preference defined, and instead randomly selects goals each cycle, the function must then make use of the optional parameter for the cycle id, and every selected goal must return that it is preferred over every non-selected goal from the given cycle, it can return *equally preferred* (denoted by 0) to every other case. The method does not impose any restrictions on the preference of goals.

The conflict function identifies if there are any conflicts between two given goals. The types of conflicts to be considered may vary with the agent's model (e.g., a model can introduce resources, and two goals can be conflicting because there is not enough of a given resource for both).

Definition 12. ***(Conflict Function).*** *Let GId be the set of goal identifiers, $gid_1, gid_2 \in GId$ be two goal identifiers, $conflict : GId^2 \rightarrow [True, False]$, where $conflict(gid_1,gid_2) = True$ iff there is a conflict between gid_1 and gid_2, $False$ otherwise.*

Given the types of questions mentioned in the previous section: ontological questions can be answered by querying an ontology of the agent about a given element or retrieving a property of the element; causal questions require a history of the agent's execution; and contrastive questions are more complex, requiring the execution history and also comparing one element with another, be it factual or not.

To generate contrastive explanations, first, they need to be formalized. The structure of contrastive explanations of Van Bouwel and Weber presented in the subsection 2.2.2, was used as a basis and adapted to the context of goal selection in intelligent agents, more specifically, BDI-based agents.

The agent is considered fully committed to cooperating with a human (expert or regular user); that is, the agent has no reason to withhold any information. One could argue that there are instances where an agent should keep some information from the requester, if the requester is not meant to access some information, or even that relaying some information can hinder the agent's goals, but those (and similar) cases are not covered in this work. The basic beliefs of the agent do not need further evidence. That is required as one could request evidence for information the agent received and has no control over. As such, Grice's maxims of Quantity and Relation are considered during the possible *explanans* generation. The Quality is straightforward because all information in the knowledge base is deemed true. The Relation is achieved by the causal relationship of beliefs and goals and its use while defining the *related conditions*. The Quantity is achieved both by the contrast of two goals, when applicable, and the *generate possible explanans* procedure. The Manner maxim is not relevant, as the convey of the explanation is beyond the scope of this method.

The following subsection presents a general formalization for explaining BDI-based intelligent agents that use beliefs and some form of causal relationships based on such beliefs, during the goal selection.

3.2 Related Conditions: Explanations for Goal Selection

This subsection presents the first procedure: *Generate Related Conditions*. As shown in section 2.2.2, Van Bouwel and Weber defined three types of contrastive questions: Property-Contrast (P-Contrast), Object-Contrast (O-Contrast), and Time-Contrast (T-Contrast). In addition to that, a new type is introduced, the Object-Time-Contrast (OT-contrast). For these four questions types, there are two different sets of elements: a) the object(s) of the question, and b) the properties of the object(s). The T-contrast and OT-contrast have a third element: the instances of time being compared.

When focusing on the goal selection phase of an agent reasoning cycle, questions about beliefs, plans, and actions are not influenced or explained by the deliberation that takes place during the goal selection. In contrast, goals are directly influenced by the deliberation. The deliberation process aims to manipulate the agent's goals by determining which ones should advance their state and associating viable plans with the intentions. A goal can be at different states during the agent's execution, according to the model that the agent is based on.

Given that only goals can be the object of the questions in this context, only the properties of goals can be used. The main property of goals is their **state**. Intentions also have their associated plan, but it is not considered a valid property for the *explanandum*. This is not an issue, as to ask *why goal g had plan p1 instead of plan p2* can be in turn expressed as *why goal gp1 (g with plan p1) instead of gp2 (g with plan p2)*, that is, instead of comparing two possible plans of a goal, two goals, each using one of the plans, are compared.

Lastly, time is seen as a modifier of the goal, which determines the frame of the history of the goal to be considered in the comparison. It is worth mentioning that in the context of goal selection, time is seen as a discrete ordered list of cycles. A cycle can be seen as an atomic time instance, where all the belief revision – the addition and removal of beliefs – already took place for the current cycle.

The formalization for each type of contrastive question is presented in the following subsections. At first, for directly subsequent states of a goal, and then for arbitrarily arranged states. Note that in some models, a goal accumulates states. One example of such models is the BBGP, where the four main states that a goal goes through during the goal processing (*Active, Pursuable, Chosen, and Executive*, in that order) require that the goal maintain the state it previously acquired, that is, if a goal has achieved the state *Active*, and then *Pursuable*, if for some reason it were to lose the *Active* state, the *Pursuable* would also be dropped. A metaphor with the towers of Hanoi, where the larger discs are the initial states, getting smaller at each subsequent state. In that sense, a goal not necessarily loses a state and goes to the next, but actually, it depends on previous stages to achieve and maintain the later ones. In these cases, when referencing a state, it refers to the most advanced state the goal has achieved.

Different patterns could be observed when analyzing how an agent can answer the questions. Such patterns represent how the reasoning mechanism deliberates about the selected goal in each step of the goal selection. The three observed patterns are: a) positive filter beliefs, composed of a set of beliefs $\{D_1, ..., D_n\}$ that can lead to the advance of the targeted goal to the next state st; b) negative filter beliefs, which comprises of a set of beliefs $\{D'_1, ..., D'_m\}$ that can impede the advance of a given goal to the next state st; and c) preference filters, which express the preference relationship between conflicting goals. A belief can be a positive or negative literal, for both positive and negative filters. What differentiates a positive from a negative filter is that positive filters beliefs support the goal progress, while negative filter beliefs act as an obstruction to such progress. Note that positive and negative filters are based on the agent's beliefs, evaluating if the conditions for the goal progress holds on the agent's knowledge base. In turn, the preference filter models a selection of a non-conflicting subset of goals, that is, given that several goals can progress at a given time, but there are conflicts between them, which subset of non-conflicting goals takes preference over the others. By definition, the preference filter requires that any necessary conditions, other than conflicting with other goals, must be met, the preference filter is mutually exclusive in relation to both positive and negative filters. That is, the conditions that trigger the positive and negative filter impede a goal of being evaluated by preference. In contrast, the process of evaluating the goal's progress for a certain state can combine positive and negative filters. Keep in mind that since positive filter beliefs assert that a should progress and negative filter beliefs assert that a should not progress, a knowledge base needs a mechanism to only accept one of the outcomes. One simple mechanism to resolve such conflicts is to give preference to negative filters, in doing so, the agent will avoid pursuing goals that may have obstructions that it does not know how to overcome. Different conflict resolution mechanisms can be used according to the intended use.

The output of the filters is the *related conditions* set.

Definition 13. ***(Related Conditions).*** *Let RC be the set of related conditions resulting from a filter. A related condition $rc \in RC$ is a presumption (Definition 17) or a preference assertion. If RC results from a preference filter, $|RC| = 1$.*

Before the *related conditions* of each question type are defined, a few required definitions need to be presented first.

The agent's goals always have some associated state. The set of possible states is dependent on the agent model. For example, the BDI model defines two states: *Desire* and *Intention*; in turn, the BBGP model defines four: *Active*, *Pursuable*, *Chosen*, and *Executive*. The possible states of a goal are ordered. Considering that order, a state can be regarded as subsequent to another if it comes afterward in the list.

Definition 14. ***(Directly Subsequent States).*** *Let $\mathcal{S} = \{st_1, st_2, ..., st_n\}$ be a finite ordered list of possible goal states. For st_j, $st_k \in \mathcal{S}$, st_k is said to be directly subsequent to st_j iff. $k = j+1$.*

Definition 15. ***(Subsequent States).*** *Let $\mathcal{S} = \{st_1, st_2, ..., st_n\}$ be a finite ordered list of possible goal states. For st_j, $st_k \in \mathcal{S}$, st_k is said to be subsequent to st_j iff. $k > j$, and is denoted by $st_k \subset st_j$[1].*

The expanded knowledge base, denoted by KB^+, is the knowledge base KB of the agent and everything that entails from it. It contains everything that the agent considers as true. It is especially useful in the following subsections when referring to knowledge that the agent is missing, that is, beliefs that are undecidable for the agent.

Definition 16. ***(Expanded Knowledge Base).*** *Let KB be the finite knowledge base of the agent, the finite expanded knowledge base $KB^+ = \{\varphi | KB \models \varphi\}$, and $\nexists \varphi \in KB^+ | KB \models \varphi \wedge \varphi \notin KB^+$.*

The explanations are built using the most fundamental beliefs of the agent, called *presumptions*, as they need no support to be considered true.

[1] The symbol '$\subset$' was chosen because the relationship between goal states mandates that for a given goal state to hold, all previous goal states must also hold. As such, the set of goals that are in a goal state st_k is a subset of goals that are in goal state st_{k-1}.

Definition 17. ***(Presumptions).*** *Let $\mathcal{L}$ be the set of every possible literal in the language, B be the set of beliefs of the agent, and R his set of rules, $\forall b \in B$ b is a presumption, and $\forall r \in R$ if $body(r) = \{\emptyset\} \wedge head(r) \in \mathcal{L}$, then $head(r)$ is a presumption.*

There are two types of presumption extraction functions, one for when the desired outcome is that a *causal rule* be activated and the other for when the desired outcome is that a *causal rule* be deactivated. The function *d_pre* maps a set of causes to the corresponding set of presumptions used in them. The following disjunctive form of the function is required for answers that need to deactivate a cause, as it represents the idea that if a single cause were removed, the event would not have happened.

Definition 18. ***(Disjunctive Presumptions).*** *Let B be the set of beliefs, E be the set of events and $e \in E$ a given event, $C = \{c_1,...,c_n\}$ be a set of causes where $\forall c = \langle e', Cond\rangle \in C,\ e' = e$, $d\_pre : C^n \rightarrow 2^B$ such that:*

$$d\_pre(C) = \begin{cases} \textit{if } |C| > 1, \bigcup\limits_{i=1}^{n} d\_pre(c_i) \\ \textit{else if } C = \langle b, \{\emptyset\}\rangle \wedge b \in B, \{b\} \\ \textit{else } C = \langle e, \{b'_1,...,b'_m\}\rangle, \bigcup\limits_{j=1}^{m} d\_pre(causes(b'_j)) \end{cases}$$

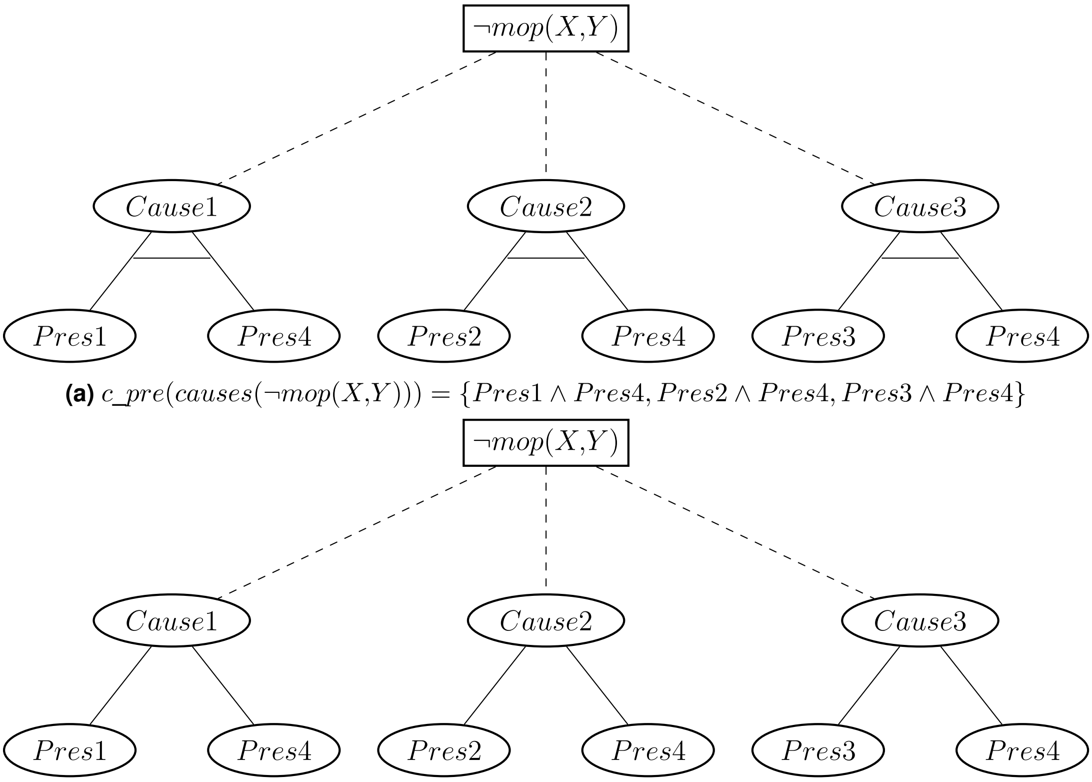
In an analogous way to d_pre, c_pre provides the set of presumptions of the causes, but in a conjunctive form. Every presumption in a cause of event e forms a single conjunctive formula. There is one formula for each different cause of e. This conjunctive form is used for answers that need to activate a cause, as it represents the idea that if all the beliefs that are missing were to become true, the event would have happened.

Definition 19. ***(Conjunctive Presumptions).*** *Let F be the set of possible formulas constructed from $\mathcal{L}$, E be the set of events and $e \in E$ a given event, $C = \{c_1,...,c_n\}$ be a set of causes where $\forall c = \langle e', Cond\rangle \in C$ $e' = e$, $c\_pre : C^n \rightarrow F^n$ such that:*

$$c\_pre(C) = \begin{cases} \textit{if } |C| > 1, \bigcup\limits_{i=1}^{n} c\_pre(c_i) \\ \textit{else if } C = \langle b, \{\emptyset\}\rangle \wedge b \in B, \{b\} \\ \textit{else } C = \langle e, \{b'_1,...,b'_m\}\rangle, \bigwedge\limits_{j=1}^{m} c\_pre(causes(b'_j)) \end{cases}$$

Figure 3 shows an example of applying the presumption functions on the causes for the goal $\neg mop(X,Y)$, which states the need for the agent to mop a region (X,Y) (the definition of the scenario, and the goal $\neg mop(X,Y)$, is presented in subsection 4.1). Rectangle denotes the event, ellipses denote causes, dashed lines denote that the causes are directly related to the event, solid lines denote that the causes are in a causal tree. If two solid lines are joined by a line, it denotes that their causes are connected by logic conjunction. In a) it is possible to see the conjunctive function, where the presumptions for each cause form a single formula. In b) it is possible to see that the causes are disjoint.

Figure 3 – Example of presumption functions on $\neg mop(X,Y)$.



(a) $c\_pre(causes(\neg mop(X,Y))) = \{Pres1 \wedge Pres4, Pres2 \wedge Pres4, Pres3 \wedge Pres4\}$

(b) $d\_pre(causes(\neg mop(X,Y))) = \{Pres1, Pres2, Pres3, Pres4\}$

Source: Author's own.

There is a preference relation between goals. This relation is especially important between conflicting goals since it is the reason for selecting one over the other. This preference is expressed as:

Definition 20. ***(Preference of goals).*** *Let GId be the set of goal identifiers, gid, $gid' \in GId$ be two goal identifiers, such that $gid \neq gid'$, where $preferred(gid, gid', \_) = 1$ (Definition 11) can be denoted as $gid > gid'$. Conversely, $gid < gid'$ expresses that $preferred(gid, gid', \_) = -1.$*

Sometimes two sets of goals need to be compared, especially when the goal being explained has a low overall preference and other compatible goals help it to become selected. The preference of a set of goals, since the explanation is post-hoc, can be inferred by the goals that progressed and the ones that did not. That is, the set of goals that progressed must be internally compatible and is preferred over any other possible subsets of goals at the time that the decision mas made. Note that both sets of goals must be internally compatible. This relation is expressed as:

Definition 21. ***(Preference over sets of goals).*** *Let G be the set of goals, $G' \subset G$ and $G'' \subset G$ be two internally non-conflicting sets of goals, $G' > G''$ expresses that G' is preferred over G''.*

Sometimes a single goal has a very high preference, where it is preferred over a set of other goals. This is important when the high preference of said goal can explain its selection and of any other compatible goals.

Definition 22. ***(Maximally preferred).*** *Let G be the set of goals, $g \in G$ be a goal, and $G' \subset G$. Goal g is said to be maximally preferred over G' when $\forall g' \in G' | g > g'$, denoted by $g \gg G'$.*

When two distinct goals are compared, their predicates commonly won't match. Consider two goals $g_1 = \langle go(x_1, y_1), st_1 \rangle$ and $g_2 = \langle go(x_2, y_2), st_2 \rangle$, whose predicates represent the goal "go to destination (X,Y)", each with a difference set of coordinates ($x_1 \neq x_2 \wedge y_1 \neq y_2$). Now consider that a predicate $clear\_path(X{,}Y)$ that returns true iff. the path from the robot's current location $(X'{,}Y')$ ($at(X'{,}Y')$) and the destination $(X{,}Y)$ is clear. Considering that g_1 and g_2 have different destinations, it is still a valid comparison to be made: "the path to g_1 is obstructed, while g_2 has a clear path", so the beliefs $clear\_path(x_1{,}y_1)$ and $clear\_path(x_2, y_2)$ in the causal tree of each goal need to be matched even thought their constants are not the same. When comparing predicates, the terms may differ, but the comparison still may be necessary. For example, suppose an embodied agent, like a robot, needs to move to different locations in a certain area. As such, when comparing distinct goals the predicate of their presumption are compared without their respective terms. To achieve that, only the name of the predicates are compared, disregarding the terms. The two following functions are defined:

Definition 23. ***(Predicate name of a literal).*** *Let $\mathcal{L}$ be the set of all literals in the language, $\varphi \in \mathcal{L}$, $\mathcal{PN}$ the set of predicate names defined in $\mathcal{L}$, and $pred \in \mathcal{PN}$, $predicate\_name : \mathcal{L} \rightarrow \mathcal{PN}$, maps a literal to the predicate that is used. $predicate\_name(\varphi) = pred$.*

Definition 24. ***(Predicates names of a formula).*** *Let F be the set of possible formulas constructed from $\mathcal{L}$, $\varphi \in F$, l be a literal in $\mathcal{L}$, $\mathcal{PN}$ the set of predicate names defined in $\mathcal{L}$, and $pred \in \mathcal{PN}$, $predicate\_name : F \rightarrow 2^{\mathcal{PN}}$, maps a formula to the predicates that it uses. $predicate\_name(\varphi) = \{pred | pred \in \mathcal{PN} \wedge \exists l \in \varphi$ such that $pred = predicate\_name(l)\}$.*

Note that the function *predicate_name* has a polymorphic definition: if the parameter is a literal, the result is a single predicate name; if the parameter is a formula, the result is a set of every predicate name used in said formula.

The following function *uses_pred* is responsible for filtering out formulas that use predicates not in a set of predicate names. The two parameters are a set of formulas to be filtered and a set of predicate names to be used as a filter.

Definition 25. ***(Uses predicate).*** *Let $\mathcal{L}$ be the set of literals, $\mathcal{PN}$ be the set of every predicate name defined in $\mathcal{L}$, $Pred \subset \mathcal{PN}$ be a set of predicate names, F be the set of possible formulas constructed from $\mathcal{L}$, and $\Phi \subset F$ a subset of formulas with size n, $uses\_pred : \mathcal{PN} \times F^n \rightarrow 2^F$, returns the formulas $\varphi \in \Phi$ whose literals are in $Pred$ (Definition 24). $uses\_pred(Pred, \Phi) = \{\varphi | \varphi \in \Phi \wedge predicate\_name(\varphi) \subset Pred\}$.*

In the following subsections, the explanations for each of the four types of contrastive questions defined in this work are presented. Each type of contrastive question represents a case that is selected in accordance with the *explanandum* posed. They all share some characteristics: i) there are three patterns that are defined for each (positive, negative, and preference filters); ii) the answer is a set RC of **related conditions**, that is either a combination of the positive and negative filters resulting presumptions **or** the first matched case of the preference filter, this latter is called a ***preference assertion***; iii) there are two sets of the patterns defined over the relation between the two states used in the *explanandum* st and st', one for when $st \subset st'$ (st is more advanced than st'), and the other for $st' \subset st$ (st' is more advanced than st); iv) the filter that needs to be evaluated is the one related to the stage that grants the most advanced state between st and st' (after the correction of the non-subsequent state), the related stage is dependent on the agent model. For instance, the BDI model has no clear distinction between the beliefs used at each state, and it is possible that all three patterns, or only a subset of them, are used for each state evaluation stage. In turn, the BBGP has a clear distinction on the beliefs for each state, where *Activated* uses a positive filter, *Pursuable* uses a negative filter, *Chosen* uses a preference filter, and *Executive* uses a positive filter.

The positive and negative filters are based on causal relations, what differentiates them is the value of the literal that represents the *event*. If the literal is positively evaluated, it is a cause supporting the event, and is managed by the positive filter. In turn, if the literal is negatively evaluated, the cause is said to be against, obstructing or impeding the event, being then managed by negative filters.

Definition 26. ***(Supporting and Impeding causes)****. Let C be the set of causes, $c = \langle e, Cond \rangle \in C$ be a cause, if e is a positive literal, then c is said to be a supporting cause for e. If e is a negative literal, then c is said to be an impeding cause for e.*

Positive and negative filters follow the same rationale in relation to the construction of its formulation:

- First, get all direct causes (Definition 10) of goal gid: $causes(gid)$.
- Second, extract the presumptions (Definitions 18 and 19): $c\_pre(...)$ or $d\_pre(...)$.
- Third, compare the presumptions with the agent's beliefs (Definition 16): $-KB^{+}$ or $\cap KB^{+}$, for "not a belief" and "is a belief", respectively.
- Lastly, when applicable, compare the resulting presumptions from both goals (function Definition 25): $Pred - Pred'$, $Pred \cap Pred'$, or $uses\_pred(...)$.

In turn, the preference filter is an ordered list of criteria, where the output is the first satisfied criteria. It makes use of the $conflict(gid,gid')$ function (Definition 12) and preference relations (Definitions 20, 21, and 22).

3.2.1 P-Contrast

Property-Contrast questions are of the form: “Why goal gid is in state st, rather than in state st'?", having a single goal, being compared in a single instance of time with a hypothetical version of itself whose state is st'. The resulting set is interpreted as “a set of absent conditions RC that would ensure that gid achieves state st'”.

Two sub-cases are defined in accordance with the states (Definition 15):

- $st' \subset st$, that is, "Why have not gid advanced its state?"
- $st \subset st'$, that is, "Why have not gid receded its state?"

For $st' \subset st$, each deliberative pattern is defined as follows:

- **Positive** – it can be inferred from the question that the goal gid did not advance its state. The explanation needs to present the set presumptions that, if present, would have ensured the goal’s progress. This set can be obtained with:
$$RC = c\_pre(causes(gid)) - KB^{+}$$
- **Negative** – it can be inferred that the goal gid did not advance its state because some presumptions impeded it. The explanations need to present a set of presumptions that, if were made absent, would have allowed the goal to progress freely. The set is obtained by:
$$RC = d\_pre(causes(\neg gid)) \cap KB^{+}$$
- **Preference** – it can be inferred that, although no presumption was missing or impeding the goal progression, a conflict of goals and a lack of preference made the goal not to be selected to advance. As such, the explanation is an assertion of preference, identifying why gid was not selected. The first case that applies:
  - $RC =$ “Goal gid has conflict with gid'”, defined as:
  $gid' \gg G_{st} \wedge conflict(gid',gid)$, where $gid' \neq gid$,
  and G_{st} is the set of goals on stage st
  - $RC =$ “Goal gid is not compatible with a preferable set of goals”, defined as: Let $G_{Sel} = \{gid_1, ..., gid_n\}$ be the set of compatible goals that were selected, and $G_g = \{gid, gid'_1, ..., gid'_n\}$ be the set of goals compatible with $gid, G_{Sel} > G_g$

For $st \subset st'$, each deliberative pattern is as follows:

- **Positive** – it can be inferred from the question that the goal gid was expected to lose its state st and recede to a previous state st'. The explanation needs to present the set of presumptions that, if present, would have ensured the goal regression. This set can be obtained with:

$$RC = d\_pre(causes(gid)) \cap KB^{+}$$

- **Negative** – it can be inferred that the goal gid did not regress its state because some expected presumptions, that would have impeded it, were absent. The explanations need to present a set of presumptions that, if in the KB, would have impeded the goal to stay with state st. The set is obtained by:

$$RC = c\_pre(causes(\neg gid)) - KB^{+}$$

- **Preference** – it can be inferred that, although no presumptions forced the goal regression, a conflict of goals was expected, but the preference made the goal to be selected to continue with the state st. As such, the explanation is an assertion of preference, identifying why gid was preferred. The first case that applies:

  - $RC =$ "gid had no incompatibilities":
    $\nexists gid' \in G_{st'} | conflict(gid, gid')$ where $G_{st'}$ is the set of goals with state st'
  - $RC =$ "gid was the most preferred goal":
    $gid \gg G_{st'}$, where $G_{st'}$ represents the goals with state st'
  - $RC =$ "gid was compatible with the most preferred goal":
    $\exists gid_m \in G$ such that $gid_m \gg G_{st'} \wedge \neg conflict(gid, gid_m)$, where $G_{st'}$ is the set of goals with state st'
  - $RC =$ "gid is compatible with the selected set of goals G_{Sel}":
    Let $G_{Sel} = \{gid, gid_1, ..., gid_n\}, \nexists G' \subset G_{st'} | G' > G_{Sel}$.

That concludes P-contrast for directly subsequent st and st', be it advancing or receding its state. Non-subsequent state and special cases are discussed next in subsection 3.2.5.

3.2.2 O-Contrast

Object-contrast are questions of the form: "Why goal gid_a is in state st, while goal gid_b is in state st'?", having two goals gid_a and gid_b, being compared in the same time frame. The resulting set is interpreted as "a set of conditions RC, absent for gid_a but present for gid_b, that if present for gid_a would ensure it to achieve state st'". It is expected that gid_a and gid_b to have shared causes in its causal tree and that they should have the same state (inferred user expectation from the nature of the question).

The two sub-cases are defined according to the states (Definition 15):

- $st' \subset st$, interpreted as "Why have not gid_a advanced its state whereas gid_b have?".
- $st \subset st'$, interpreted as "Why have not gid_a receded its state whereas gid_b have?".

For $st' \subset st$, the deliberative patterns are defined as:

- **Positive** – it is inferred that gid_b had some supporting presumptions that gid_a did not had. The explanation needs to provide the set of presumptions that gid_b have in its causal tree, that if present for gid_a, would ensure that gid_a achieves state st'. Defined as:

$$A = c\_pre(causes(gid_a)) - KB^+$$
$$B = c\_pre(causes(gid_b)) \cap KB^+$$
$$RC = uses\_pred(A, predicate\_name(A) \cap predicate\_name(B))$$

- **Negative** – it is inferred that gid_a had some impeding presumptions that gid_b did not have. The explanation needs to provide the set of presumptions that gid_a have in its causal tree and gid_b does not, that if made absent for gid_a, would ensure that gid_a receded to state st'. Defined as:

$$A = d\_pre(causes(\neg gid_a)) \cap KB^+$$
$$B = d\_pre(causes(\neg gid_b)) - KB^+$$
$$RC = uses\_pred(A, predicate\_name(A) \cap predicate\_name(B))$$

- **Preference** – it is inferred that, although no supporting presumptions were absent nor impeding presumptions were present to make gid_a not to be selected, a conflict of goals and a lack of preference combined with an unexpected conflict made gid_a not to be selected. The explanation is an assertion of preference, identifying why gid_a was not selected and gid_b was. The first case that applies:
  - $RC =$ "gid_b has no incompatibilities whereas gid_a does.":
    $\nexists gid \in G_{st} | conflict(gid_b, gid) \wedge \exists gid' \in G_{st} | conflict(gid_a, gid')$
  - $RC =$ "gid_b is preferred over gid_a, and they are incompatible":
    $gid_b > gid_a \wedge conflict(gid_b, gid_a)$
  - $RC =$ "gid_b is compatible with another goal, with higher preference":
    Let $gid \in G_{st}, gid > gid_a \wedge \neg conflict(gid_b, gid) \wedge conflict(gid, gid_a)$
  - $RC =$ "gid_a is incompatible with the selected goals":
    Let $G_a = \{gid_a, gid_1, ..., gid_n\}$ and $G_{Sel} = \{gid_b, gid'_1, ..., gid'_2\}, G_{Sel} > G_a$

For $st \subset st'$, the deliberative patterns are defined as:

- **Positive** – it is inferred that gid_a had some supporting presumptions that gid_b did not had. The explanation needs to provide the set of presumptions that gid_a have in its causal tree but not for gid_b, that if made absent for gid_a, would ensure that gid_a recedes to state st'. Defined as:

$$A = d\_pre(causes(gid_a)) \cap KB^+$$
$$B = d\_pre(causes(gid_b)) - KB^+$$
$$RC = uses\_pred(B, predicate\_name(A) \cap predicate\_name(B))$$

- **Negative** – it is inferred that gid_b had some impeding presumptions that gid_a did not had. The explanation needs to provide the set of presumptions that gid_b have in its causal tree and gid_a does not, that if present for gid_a, would ensure that gid_a receded to state st'. Defined as:

$$A = c\_pre(causes(\neg gid_a)) - KB^+$$
$$B = c\_pre(causes(\neg gid_b)) \cap KB^+$$
$$RC = uses\_pred(B, predicate\_name(A) \cap predicate\_name(B))$$

- **Preference** – it is inferred that, although no supporting presumptions were absent nor impeding presumptions were present to make gid_b not to be selected, a conflict of goals and a lack of preference combined with an unexpected conflict made gid_b not to be selected. The explanation is an assertion of preference, identifying why gid_b was not selected and gid_a was. The first case that applies:
  - $RC =$ "gid_a has no incompatibilities and gid_b does":
    $\nexists g \in G_{st} | conflict(gid_a, gid) \wedge \exists gid' \in G_{st} | conflict(gid_b, gid')$
  - $RC =$ "gid_a has a higher preference than gid_b and they are incompatible":
    $gid_a > gid_b \wedge conflict(gid_a, gid_b)$
  - $RC =$ "gid_a is compatible with another goal, with higher preference":
    Let $gid \in G_{st}$ such that $gid > gid_b \wedge \neg conflict(gid_a, gid) \wedge conflict(gid, gid_b)$
  - $RC =$ "gid_b is incompatible with the selected goals":
    Let $G_b = \{gid_b, gid_1, ..., gid_n\}$ and $G_{Sel} = \{gid_a, gid'_1, ..., gid'_2\}, G_{Sel} > G_b$

If the goals being compared are not causally related, the result from a positive or negative filter will be an empty set. In those cases, a follow-up question can be answered. This process is better explored in subsection 3.2.5. Preference filters do not require a causal relationship for the answer generation and, as such, do not result in an empty set.

3.2.3 T-Contrast

Time-contrast questions has the form: "Why is gid_a in state st at t_1, while at t_2 it is in state st'?", having a single goal gid_a being compared with itself in two distinct time frames. The resulting set is interpreted as "set of conditions RC, absent at t_1 but present at t_2, that if present at t_1, would have ensured that gid_a achieved state st'. At a first, lets assume that $t_1 < t_2$, as it is shown by the end of this subsection that t_1 and t_2 are interchangeable.

The two sub-cases are defined according to the states (Definition 15):

- $st' \subset st$, interpreted as "Why gid_a have not advanced its state at t_1 but it did at t_2?".
- $st \subset st'$, interpreted as "Why gid_a have not receded its state at t_1 but it did at t_2?".

For $st' \subset st$, the deliberative patterns are as follows:

- **Positive** – it is inferred that gid_a had supporting presumption at t_2 that it did not had at t_1. The explanations needs to provide a set of presumptions, present at t_2 but absent at t_1 that, if were made present at t_1, would ensure gid_a had achieved st' at t_1. Defined as:
$$RC = (c\_pre(causes(gid_a)) - KB_1^+) \cap (c\_pre(causes(gid_a)) \cap KB_2^+)$$
- **Negative** – it is inferred that gid_a had impeding presumption at t_1 that it did not had at t_2. The explanations needs to provide a set of presumptions, present at t_1 but absent at t_2 that, if were absent at t_1, would ensure gid_a had achieved st' at t_1. Defined as:
$$RC = (d\_pre(causes(\neg gid_a)) \cap KB_1^+) \cap (d\_pre(causes(\neg gid_a)) - KB_2^+)$$
- **Preference** – it is inferred that, although no supporting presumptions were absent nor impeding presumptions were present to make gid_a not to be selected at t_1, a conflict of goals and a lack of preference made gid_a not to be selected at t_1. The explanations need to provide an assertion of preference, identifying why gid_a was not selected at t_1, but then it was selected at t_2. The first case that applies:
  - $RC =$ "gid_a had no incompatibilities at t_2 after another goal gid ended":
    Let $gid \in G^1_{st} \wedge gid \notin G^2_{st} \wedge conflict(gid,gid_a),$ where G^1_{st} is the set of goals with state st at t_1, and G^2_{st} is the set of goals with st at t_2
  - $RC =$ "A goal gid has ended in t_2 weakening the conflicting set of goals of gid_a":
    Let $gid \in G^1_{st} \wedge gid \notin G^2_{st} \wedge conflict(gid,gid_a),$ where G^1_{st} is the set of goals with state st at $t_1,$ and G^2_{st} is the set of goals with st at t_2
  - $RC =$ "A new goal gid compatible with gid_a is available at t_2 strengthening the set of compatible goals":
    Let $gid \notin G^1_{st} \wedge gid \in G^2_{st}, \neg conflict(gid,gid_a),$ where G^1_{st} is the set

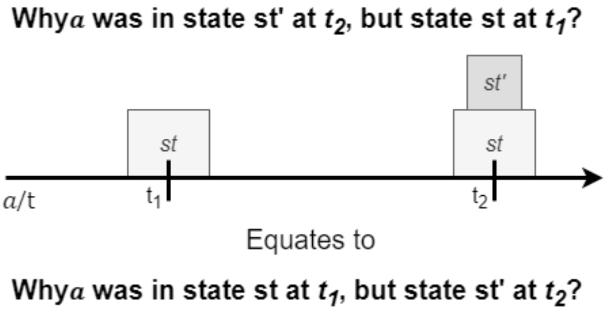
of goals with state st at $t_1,$ and G^2_{st} is the set of goals with st at t_2

When $st \subset st'$, the deliberation patterns are defined as:

- **Positive** – it is inferred that gid_a had supporting presumption at t_2 that it did not had at t_1. The explanations needs to provide a set of presumptions, present at t_2 but absent at t_1 that, if were made absent at t_2, would ensure gid_a to recede to state st' at t_2. Defined as:
$$RC = (d\_pre(causes(gid_a)) \cap KB^+_1) \cap (d\_pre(causes(gid_a)) - KB^+_2)$$
- **Negative** – it is inferred that gid_a had impeding presumption at t_1 that it did not had at t_2. The explanations needs to provide a set of presumptions, present at t_1 but absent at t_2 that, if were present at t_2, would ensure gid_a had recede to state st' at t_2. Defined as:
$$RC = (c\_pre(causes(\neg gid_a)) - KB^+_1) \cap (c\_pre(causes(\neg gid_a)) \cap KB^+_2)$$
- **Preference** – it is inferred that, although no supporting presumptions were absent nor impeding presumptions were present to make gid_a not to be selected at t_2, a lack of conflicting goals or the preference made gid_a to be selected at t_2. The explanations needs to provide a assertion of preference, identifying a why gid_a was selected at t_2 but not selected at t_1. The first case that applies:
  - $RC =$ "gid_a is incompatible with the new goal gid at t_2":
    Let $gid \notin G^1_{st'} \wedge gid \in G^2_{st'} \wedge conflict(gid,gid_a),$ where $G^1_{st'}$ is the set of goals with state st' at $t_1,$ and $G^2_{st'}$ is the set of goals with st' at t_2
  - $RC =$ "A new goal gid at t_2 strengthened the set of goals conflicting with gid_a":
    Let $gid \notin G^1_{st'} \wedge gid \in G^2_{st'} \wedge conflict(gid,gid_a),$ where $G^1_{st'}$ is the set of goals with state st' at $t_1,$ and $G^2_{st'}$ is the set of goals with st' at t_2
  - $RC =$ "A goal gid compatible with gid_a has ended at t_2 weakening the set of compatible goals":
    Let $gid \in G^1_{st'} \wedge gid \notin G^2_{st'} \wedge \neg conflict(gid,gid_a)$

To conclude T-contrast the cases where $t_1 > t_2$ need to be evaluated. In these cases, it is possible to change the order of the properties and the time so that the first t on the question is $<$ than the second one. That is possible because the question can be interpreted as: "Why gid_a advanced/receded in relation to before?". For instance, with the following question: "Why gid_a was in state st at t_a, but state st' at t_b?", by swapping the state an times results in "Why gid_a was in state st' at t_b, but state st at t_a?", as can be seen in figure 4.

Figure 4 – Inversion of times instances in questions.



Why$a$ was in state st' at t_2, but state st at t_1?

Why$a$ was in state st at t_1, but state st' at t_2?

Note how both questions translate to the same states and intervals.
Source: Author's own.

3.2.4 OT-Contrast

A more general form of the contrastive questions not presented in Bouwel and Weber's work is the Object-Time-Contrast was defined. This questions is of the form: "Why goal gid_a is in state st at t_1, while goal gid_b is in state st' at t_2?". The resulting set is interpreted as "a set of conditions RC, absent for gid_a at t_1 but present for gid_b at t_2, that is present at t_1 would have ensured that gid_a achieve state st'". Goals gid_a and gid_b are expected to be causally related, also, for the same reason as discussed in the T-contrast, let's assume that $t_1 < t_2$, and analogously with the O-contrast, the goals gid_a and gid_b are expected to be causally related.

The two sub-cases are defined according to the states (Definition 15):

- $st' \subset st$, interpreted as "Why gid_a have not advanced its state at t_1 and gid_b have advanced at t_2?".
- $st \subset st'$, interpreted as "Why gid_a have not receded its state at t_1 and gid_b have receded at t_2?".

For $st' \subset st$, the deliberative patterns are as follows:

- **Positive** – it is inferred that gid_b had supporting presumption at t_2 that gid_a did not had at t_1. The explanations needs to provide a set of presumptions, present at t_2 for gid_b but that were absent for gid_a at t_1 that, if were made present at t_1, would ensure gid_a had achieved st' at t_1. Defined as:

$$A = c\_pre(cause(gid_a)) - KB_1^+$$

$$B = c\_pre(cause(gid_b) \cap KB_2^+$$

$$RC = uses\_pred(A, predicate\_name(A) \cap predicate\_name(B))$$

- **Negative** – it is inferred that gid_a had impeding presumption at t_1 that gid_b did not had at t_2. The explanations needs to provide a set of presumptions, present at t_1 for gid_a but absent for gid_b at t_2 that, if were absent at t_1, would had ensured gid_a achieved st' at t_1. Defined as:

$$A = d\_pre(cause(\neg gid_a)) \cap KB_1^+$$

$$B = d\_pre(cause(\neg gid_b) - KB_2^+$$

$$RC = uses\_pred(A, predicate\_name(A) \cap predicate\_name(B))$$

- **Preference** – it is inferred that, although no supporting presumptions were absent nor impeding presumptions were present to make gid_a not to be selected at t_1, a conflict of goals and a lack of preference made gid_a not to be selected at t_1 that did not affected gid_b at t_2. The explanations needs to provide a assertion of preference, identifying a why gid_a was not selected at t_1 but gid_b was selected at t_2. The first case that applies:
  - $RC =$ "gid_b has no incompatibilities at t_2":
    $$\nexists gid \in G_{st}^2 | conflict(gid, gid_b)$$
  - $RC =$ "A subset of goals incompatible with gid_a are not present at t_2":
    Let $G' = \forall gid \in G_{st}^1 | conflict(gid, gid_a) \land gid \notin G_{st}^2$
  - $RC =$ "A subset of goals selected with gid_b are not present at t_1, weakening gid_a":
    Let $G' = \forall gid \in G_{Sel}^2 | \neg conflict(gid, gid_a) \land gid \notin G_{st}^1$
  - $RC =$ "gid_a and gid_b have different preference relationships".

When $st \subset st'$, the deliberative patterns are as follows:

- **Positive** – it is inferred that gid_a had supporting presumption at t_1 that gid_b did not had at t_2. The explanations needs to provide a set of presumptions, present at t_1 for gid_a but that were absent for gid_b at t_2 that, if were made absent at t_1, would ensure gid_a had recede to state st' at t_1. Defined as:

$$A = d\_pre(cause(gid_a)) \cap KB_1^+$$

$$B = d\_pre(cause(gid_b)) - KB_2^+$$

$$RC = uses\_pred(B, predicate\_name(A) \cap predicate\_name(B))$$

- **Negative** – it is inferred that gid_b had impeding presumption at t_2 that gid_a did not had at t_1. The explanations needs to provide a set of presumptions, present at t_2 for gid_b

but absent for gid_a at t_1 that, if were present at t_1, would had ensured gid_a receded to state st' at t_1. Defined as:

$$A = c\_pre(cause(\neg gid_a)) - KB_1^+$$

$$B = c\_pre(cause(\neg gid_b)) \cap KB_2^+$$

$$RC = uses\_pred(B, predicate\_name(A) \cap predicate\_name(B))$$

- **Preference** – it is inferred that, although supporting presumptions were present and impeding presumptions were absent, gid_a was expected to have not been selected, as happened to gid_b at t_2, a lack of conflicting goals or a preference made gid_a to be selected at t_1 that impeded gid_b selection at t_2. The explanations needs to provide a assertion of preference, identifying a why gid_a was selected at t_1 but gid_b was not selected at t_2. The first case that applies:
  - $RC =$ "gid_a has no incompatibilities at t_1":
  $$\nexists gid \in G^1_{st'} | conflict(gid, gid_a)$$
  - $RC =$ "A subset of goals incompatible with gid_b are not present at t_1":
  $$G' = \forall gid \in G^2_{st'} | conflict(gid, gid_b) \wedge gid \notin G^1_{st'}$$
  - $RC =$ "A subset of goals selected with gid_a are not present at t_2, weakening gid_b":
  $$G' = \forall gid \in G^1_{Sel} | \neg conflict(gid, gid_a) \wedge gid \notin G^2_{st'}$$
  - $RC =$ "gid_a and gid_b have different preference relationships".

Notice that the rationale of this type of question is that the results of both goals were expected to be the same, even in different time frames. That expectation does not follow for O-contrast.

To conclude OT-contrast, when $t_1 > t_2$, the pair goal-state can be inverted, in the same way as with T-contrast. By doing that, the requirement of the first pair happening before the second can be achieved, with an equivalent result set of conditions.

This concludes every explanation for directly subsequent state a goal can have. Next, how non-directly subsequent state interplay and possible special cases are discussed.

3.2.5 Non-Directly Subsequent States and Special Cases

In practical applications, it is expected to have some goal state beyond the sequential states for the goal selection. Two of such states are the *Cancelled* and *Completed*, both being terminal states, that is, once a goal transitions to such states, it can not change again. If the goal is still applicable, a new instance should be raised. Another example is the state *Paused*.

It differs from the previous two by not being terminal, the goal can still return to the sequential states of the goal selection.

The conditions for a goal changing to these states generally are simple, for instance, to treat exceptional circumstances (an error during the processing time), to avoid queuing issues (a goal that never progresses), or to signal the completion without loss of information (a goal that has been complete, instead of simply excluding the goal, a special state is achieved). For this reason, there is no conflicting cases or comparison applicable. To answer why a goal is in such state, it requires only to specify the causal conditional for the state transition.

When evaluating two subsequent states that do not fall on the conditions previously addressed, that is, two states that are not directly subsequent (Definition 14), such questions can be answered based on the previous constructions.

Two cases can be differentiated: a) when a goal has a state more advanced than expected, and b) when a goal is in a state previous to the expected. In the first scenario, the contrastive case shows us that it was expected that a condition blocking the goal's progress to hold. As such, by presenting the beliefs that allowed the goal to progress, it is shown that the missing blocking condition did not hold. So the answer can be provided by changing the more advanced goal state to a directly subsequent state in relation to the least advanced.

In the second scenario, where the contrast tells that a goal has not advanced as expected, some unexpected condition blocking the goal's progress holds. To answer the question, the unexpected condition needs to be provided, and, as in the first scenario, by replacing the most advanced state with a directly subsequent to the least advanced one, the set of conditions is obtained.

When the question is of the type O-contrast or OT-contrast, if the set of conditions is empty, it means that the compared goals do not share a relevant cause.

Definition 27. ***(Unrelated goals).*** *Let GId be the set of goal identifiers, $gid, gid' \in GId$ be two goal identifiers, if $\forall c \in causes(gid)\ \forall c' \in causes(gid')\ |related(c{,}c') = False$ (Definitions 10 and 6) then goals gid and gid' are said to be causally unrelated.*

In such cases, an answer saying that both goals are not causally related can be given, but often it will not be a satisfactory answer. It is possible to convert the question to a P-contrast, using the first goal and the same states of the posed O/OT-contrast. The first goal is usually the one that was unexpected for the explainee. That way, the explainee receives an answer to what would probably be his following question. For example, take the O-contrast question "Why goal gid_a has state st, while goal gid_b has state st'?", where $st' \subset st$, now let's say that the conditions returned as an empty set, instead of giving a simple answer that goals gid_a and gid_b do not share a causal history, one can provide an answer to what possibly would be a followup question. In our example, the question would be "Why goal gid_a has state st(, instead of st')?", which is translates to answering why the goal gid_a had a certain state, since posing the original question indicates that situation to be unexpected. Notice that it is still necessary to inform that gid_a and gid_b are not causally related.

3.3 Possible *Explanans*: calculating the set of possible *explanans*

The method presented in the previous section results in a set of all conditions that could be used to formulate an explanation. The explanation can have more than a single event (goal achieving a certain goal state) to be explained. As such, the conditions (presumptions or preference assertion) used in the explanation should have a causal relation with all such events. If the *related conditions* is made of a *preference assertion*, then the resulting *possible explanans* is the preference assertion. When the *related conditions* is made of presumptions, it is required to construct the sets that cover all the events, this set is the *possible explanans*. It is worth mentioning that to select a single explanation, several criteria can be used, being dependent on the context. A brief overview of these criteria is presented in subsection 2.2.4. The following method does not seek to select the answer, only to create subsets of beliefs covering all goal states to be explained. The selection should be made next, but it is outside the scope of this work.

Let RC be the set of related conditions (the presumptions or preference assertion) and E the set of events to be explained (*explanandum* – the goals with their respective state). In order to define the subsets of RC to be used as answers, a non-directed graph Gp is built, such that $Gp = (Ver, Edg)$. Lets call such graphs *explanation graphs*. The set of vertex $Ver = \{x | x \in (RC \cup E)\}$, and the set of edges Edg is:

If RC is obtained with d_pre (Definition 18) : $\forall e \in E$ and $\forall rc \in RC,$

if $\exists l \in d\_pre(causes(e)) \land predicate\_name(l) = predicate\_name(rc)$

then $(rc{,}e) \in Edg.$

If RC is obtained with c_pre (Definition 19) : $\forall e \in E$ and $\forall rc \in RC,$

if $\exists \varphi \in c\_pre(causes(e)) \land predicate\_name(\varphi) \subset predicate\_name(rc)$

then $(rc{,}e) \in Edg.$

With that, every condition is connected to every event that it explains.

A possible *explanans* is defined as:

Definition 28. ***(Possible explanans - PE)*** *Let RC be a set of related conditions and E a set of events, $PE \subset RC$ is a possible explanans iff. $\forall e \in E, \exists rc \in PE | (rc{,}e) \in Edg.$*

Remembering Grice's Cooperation Principles, there are two relevant maxims: Quantity and Relation. The Relation maxim is satisfied by the previous step, that is, let RC be the set of related conditions outputted by the *Generate Related Conditions* procedure, and $Exp \subset E$

be the set of events in the *explanandum*, $\forall rc \in RC\, \exists e \in Exp | rc$ is causally related to e. The Quantity maxim requires that only the necessary conditions, and nothing more, to be part of the explanation. A possible *explanans* guarantees that all the events in the *explanandum* are accounted for by some condition. To avoid redundant conditions, and by that satisfying the Quantity maxim, it is necessary to provide only **minimal** possible *explanans*:

Definition 29. ***(Minimal Possible explanans).*** *Let Gp be an explanation graph, and $\mathcal{PE}$ be the set of all possible explanans of Gp. $PE \in \mathcal{PE}$ is minimal iff. $\nexists PE' \in \mathcal{PE} | PE' \subset PE$ and $PE \neq PE'$.*

3.3.1 Algorithm for MPE

To calculate the Minimal Possible *Explanans* (MPE) set, Algorithm 1 is presented. The procedure presented is recursive, with five inputs: $possible\_explanans$, which is a set of sets of conditions; $event\_list$, which is the ordered list (with respect to the node degree) of all events; $index$ representing the current event of the $event\_list$ being evaluated; $covered$ which is a set of events currently covered in the branch; lastly, $partial\_cover$ is a set of conditions that is a partial cover for the events in the branch. From these five inputs, only one needs to be initialized, the $event\_list$. The other four need to be empty sets, and in the case of the $index = 0$. Every set needs to be passed by reference. The output of the procedure is the $possible\_explanans$ at its final state.

Besides the $event\_list$ being ordered, another prerequisite is imposed over the explanation graph Gp. A relation over the conditions needs to be calculated beforehand, such that every condition node has the list of its proper subsets (this list is denoted by: condition.supersetOf). For simplicity, a second list is defined, denoted by 'node.neighbors'. Both conditions and events are nodes, and $node.neighbors = \{A | (B, A) \in Edg \wedge (A, B) \in Edg\}$, where B is a different node. Note that if $node$ is a condition, its neighbors are always events, and if $node$ is an event, all its neighbors are conditions.

In short, Algorithm 1 has four statements (denoted by **$#**, where # is a number): $1 checks for the end of the search branch; $2 checks if the branch is repeating; $3 checks if the event being evaluated (by the current $index$) is already covered (this can be the case when a condition previously selected covers more than one event); and $4 branches out to every condition that can cover the evaluated event. Conditions $1 and $2 are stopping conditions, while $3 and $4 are recursive steps. This calculation can be done in $O(N_C^2 * N_E^2)$, where N_C are the condition nodes, and N_E are the event nodes.

The Algorithm 1 can have its complexity evaluated as a depth-first tree, where the maximum height is of the length of the $event\_list$, and the branching factor is at most the maximum degree of G_{nd}. The time complexity of the minimal possible explanans is $O(H^2 * Gp_{deg}^2 * Gp_{sup})$, where H is the length of the $event\_list$, Gp_{deg} is the maximum degree of Gp, and Gp_{sup} is the maximum length of condition's superset list. In the current state of the question types formalization, at most, two events (H) are evaluated, which significantly limits the search space for the MPE algorithm. The other two complexity factors (Gp_{deg}, Gp_{sup}) are deeply related to the causal model of the agent.

The spacial complexity of Algorithm 1 is $O(2H^2 + 2H + H * Gp_{deg})$, where H is the length of the $event\_list$, and Gp_{deg} is the maximum degree of Gp. The input $event\_list$ of the algorithm is constant and shared between all active recursion calls. As such, it requires $O(H)$ space. A second input is shared, the $possible\_explanans$ set, but different from the $event\_list$ it changes during execution. The worst-case scenario of such input is a combination of all possible conditions for each event. As such, its spacial complexity is $O(H * Gp_{deg})$, that is, using every possible condition to each event, for all events. The next three inputs are copied (and changed) in each procedure call. First, the $index$, which is trivially an integer, requires constant space (for each call). Next, the $covered$ list, which is an auxiliary list of all events that are currently covered by the $base\_cover$; as such, its spatial complexity is $O(H)$. Lastly, the $base\_cover$ is a list of conditions that are selected and being evaluated as possible explanans, and as such, no more than one condition for each event is ever necessary, requiring $O(H)$ space. The last step of the spatial analysis is to determine the number of procedures that can be active at any single time. Since the algorithm behaves itself as a depth-first search tree, and the maximum height of the tree is H, being the maximum number of calls in the stack. As such, the spacial requirement of the copied inputs is $O(H)$ for $index$, and $O(H * H)$ for both $covered$ and $base\_cover$.

To illustrate how each different sub-call of Algorithm 1 impacts the search space, Figure 5 is presented. In a) a fictional explanation graph is presented, with three events $a, b,$ and c, represented by squares, and six conditions $1, 2, 3, 4, 5,$ and 6, represented by circles. The edges of the graph represent which conditions can explain which events. In b) a representation of the final search tree is shown. At each level, the event that is evaluated is presented on the left as a header. Each node represents a condition selected for evaluation. In each edge, identified by the prefix $\$$, the type of sub-call used is seen. The type of call is defined by the same identifier as a comment in Algorithm 1.

A Java implementation of this algorithm is available [2]. It was built as a library, with a test case included. The user of such library needs only to define the explanation graph Gp, with the event node, condition nodes, and the relation between those two. The ordering of the events and the subset calculation are included in the recommended function. Also, an object-oriented version of the algorithm is available in Appendix A.

Next chapter a case study using both methods is presented.

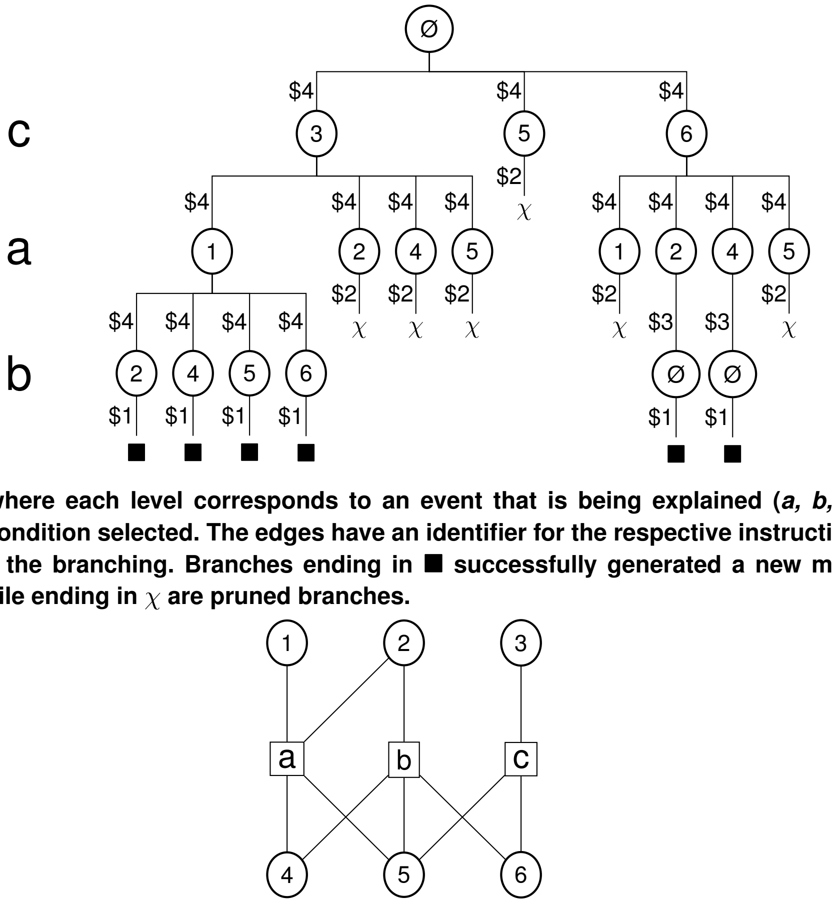
[2] MPE algorithm at: https://github.com/henriquermonteiro/MinimalPossibleExplanans

Source Code 1 – Algorithm for calculating the Minimal Possible *Explanans* from an Explanation Graph Gp.

```
procedure MPE($possible\_explanans, event\_list, index, covered, partial\_cover$)
                        ▷ If the branch has ended, the partial_cover is a possible explanans.
    if index > | event_list | then
        return { possible_explanans ∪ part_cover }                              ▷ $1
    end if

                        ▷ If partial_cover is an explored branch, returns partial_cover.
    if partial_cover ∈ possible_explanans then
        return possible_explanans                                                ▷ $2
    end if

                        ▷ If the current event is covered already, skip to next event.
    if event_list[index] ∈ covered then
        return MPE(possible_explanans, event_list, index + 1, covered, base_cover)   ▷ $3
    end if

                        ▷ Branches out to each condition that can cover the current event.
    for all condition from event_list[index].neighbors do

                        ▷ Creates a copy of covered and partial_cover for the different branches.
        copy_partial_cover = { partial_cover ∪ condition }
        copy_covered = { covered ∪ condition.neighbors }

                        ▷ Removes from copy_partial_cover the conditions that are a subset of condition
        copy_partial_cover = { copy_partial_cover − condition.supersetOf }

        MPE(possible_explanans, event_list, index + 1, copy_covered, copy_partial_cover)   ▷ $4
    end for

    return possible_explanans
end procedure
```

Source: Author's own.

Figure 5 – Search tree (a) of the MPE algorithm for an explanation graph (b).



(a) Search tree, where each level corresponds to an event that is being explained (*a, b, c*), each circle represents a condition selected. The edges have an identifier for the respective instruction in Algorithm 1 that creates the branching. Branches ending in ■ successfully generated a new minimal possible *explanans*, while ending in χ are pruned branches.

(b) Explanation graph used in a), where *a, b, c* are events to be explained and *1, 2, 3, 4, 5, 6* are conditions related to the events. Note that the degree of *a=4, b=4, c=3*. As such, the order of events for the MPE algorithm must be $c > a > b$ or $c > b > a$.

Source: Author's own.

4 BBGP AGENT CASE STUDY

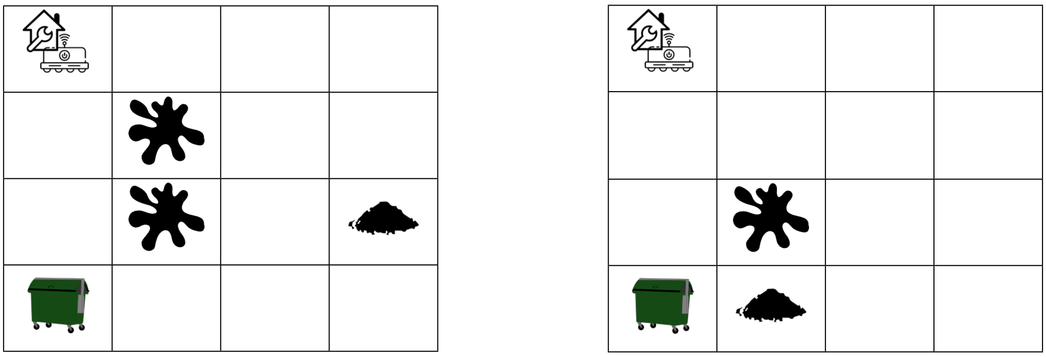
In this chapter, a case study is presented. Based on the cleaner world scenario, where a robot is responsible for cleaning an area from solid and liquid dirt. From the example situation, both steps of the method are demonstrated.

4.1 Cleaner World Scenario

In this case study, the Cleaner World scenario is used. The agent is a robot in charge of cleaning a certain area. In the robot's representation, the area is divided into cells referred to by a coordinate system $(X{,}Y)$. The robot can move along the cells, clean solid and liquid dirt by sweeping and mopping, respectively, recharging itself when necessary, replacing some broken parts, and disposing of the dirt in its internal storage. Lastly, for the robot not to disrupt the passage when there are no jobs at the moment, there is a command for it to rest. There are two special locations for the robot, the workshop and the dumpster. In the workshop, the robot can find replacement parts, recharge, and is where it is assigned to rest. The dumpster is where the dirt is disposed of. To do the mopping, the robot needs a functional mop part, and to sweep, the robot needs a functional vacuum. The mop is a part that the robot can replace by itself, given that there is a replacement available.

On a given day, the following series of events were observed. Each event is denoted by a time instance t_n, where $n \in \mathbb{N}$ such that if $n < n'$ then the event t_n occurred before $t_{n'}$. At time t_0, three cells were dirty, where (2,2) and (3,2) had liquid dirt and (3,4) solid dirt. At t_1, the robot went cleaning (2,2). Followed by cell (3,4) at t_2. The robot proceed to go back to the workshop at t_3. Next, the robot dump its reservoir in the dumpster (t_4) and returned to the workshop (t_5). At t_6 a new $solid\_dirt(4{,}2)$ was spilled. Later, the robot proceeded to clean (3,2) at t_7, clean (4,2) at t_8, and dump it's reservoir in the dumpster (t_9), and went back to the workshop at t_{10}.

Figure 6 – Depiction of the scenario in two key time instances of cleaner world scenario.



(a) World at t_0, at the beginning of execution; **(b) World at t_6, when a new perception was added.**

Source: Author's own.

Figure 6 depicts the state of the world in two time instances: a) the starting conditions before the first deliberation cycle, and b) when the new perception of $solid\_dirt(4{,}2)$ was added.

4.1.1 Knowledge Base and Plans

The agent's knowledge base and plans that created the scenario are described next.

Figure 7 – Starting beliefs.

```
workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(1,1)
have(battery, 75.0)
have(cargo, 65.0)
solid_dirt(3,4)
liquid_dirt(3,2)
liquid_dirt(2,2)
```

Source: Author's own.

Let us begin describing the agent's starting beliefs, that is, the beliefs it had at t_0 before the first deliberation cycle. Figure 7 depicts the initial beliefs. The interpretation of all belief predicates from B defined in the language are as follows:

- *workshop(X,Y)* : the workshop is located at coordinates (X,Y);
- *dumpster(X,Y)* : the dumpster is located at coordinates (X,Y);
- *at(X,Y)* : the agent is located at coordinates (X,Y);
- *have(R,Q)* : the agent has Q or more of the resource R;
- *solid_dirt(X,Y)* : there is solid dirt in coordinates (X,Y);
- *liquid_dirt(X,Y)* : there is liquid dirt in coordinates (X,Y);
- *broken(P)* : part P is broken;
- *available(I)* : item I is available;

The goals from G also have predicates representing them, whose interpretations are:

- *replace(X)* : replace component X;
- *mop(X,Y)* : mop cell at coordinates (X,Y);
- *sweep(X,Y)* : sweep cell at coordinates (X,Y);
- *recharge* : go recharge battery;
- *dispose* : go dispose the internal reservoir in the dumpster;

Figure 8 – Activation and Evaluation Rules.

Activation Rules	Evaluation Rules
$\langle replace(X), \{broken(X)\}\rangle$	$\langle \neg replace(X), \{\neg available(X)\}\rangle$
$\langle mop(X{,}Y), \{liquid\_dirt(X{,}Y)\}\rangle$	$\langle \neg mop(X{,}Y), \{\neg have(battery, 40), liquid\_dirt(X{,}Y)\}\rangle$
$\langle sweep(X{,}Y), \{solid\_dirt(X{,}Y)\}\rangle$	$\langle \neg mop(X{,}Y), \{have(cargo, 80), liquid\_dirt(X{,}Y)\}\rangle$
$\langle recharge, \{\neg have(battery, 15)\}\rangle$	$\langle \neg mop(X{,}Y)\{broken(mop), liquid\_dirt(X{,}Y)\}\rangle$
$\langle dispose, \{have(cargo, 0.1)\}\rangle$	$\langle \neg sweep(X{,}Y), \{\neg have(battery, 30), solid\_dirt(X{,}Y)\}\rangle$
$\langle recharge, \{\neg have(battery, 80)\}\rangle$	$\langle \neg sweep(X{,}Y), \{have(cargo, 90), solid\_dirt(X{,}Y)\}\rangle$
$\langle rest, \{\emptyset\}\rangle$	

Source: Author's own.

Next, the agent's rules are described. The robot's rules are depicted in Figure 8. Only the activation and evaluation rules are depicted, as the deliberation and checking stages are fixed since they are domain-independent.

For this study case, assume that the robot has a plan library that describes the steps required to achieve each goal. As per Definition 2, a plan $P = \{g, Gd, Act\}$, where g is the goal for which the plan is applicable, Gd is the set of preconditions that act as guard clauses, that is, beliefs that must hold for the safe/successful execution of the plan, and Act is the ordered list of actions that are performed during the plan execution. The contents of the plan library are disregarded for the case study.

The preference order of goals (for Definition 11) is exactly the activation rules order, in decreasing order. That is, $replace(X)$ has the highest preference, and $rest$ has the lowest one. Assume also that the plans also have a preference order that is inversely proportional to the distance from the robot's position to the position the goal takes place. In that way, the goal preference takes precedence, followed by the plan preference. In this way, two conflicting goals with the same goal preference will have their conflict resolved by how far each one is from the targeted cell.

4.2 Interface Implementation for a BBGP-like Agent

The proposed method uses an interface that requires four elements to be implemented: the *execution history* (Definition 9), the *causal function* (Definition 10), the *preference function* (Definition 11), and the *conflict function* (Definition 12). The requirements for these elements are described in Chapter 3.

For our example scenario, the agent's execution generated the log depicted in Table 1, where each row is an entry of the log. The log allows for the reconstruction of the agent knowledge base and tracks the goal state changes.

The causal function can be defined from the agent's rules and plans. This function returns a set of *causes* ($c = \langle e, Cond\rangle$ as per Definition 4) whose event e matches the input. As the plans were not defined for our study case, their respective *causes* will be omitted. The *activation* and *evaluation* rules can be easily converted simply by attributing the consequent (the goal) as e and the set of beliefs in the antecedent as $Cond$. Table 2 shows the mapping generated. The

Table 1 – Execution History.

Cycle	Changed Goals	Changed Beliefs
0	{$\emptyset$}	{ $\emptyset$ }
1	$\langle rest_1,$ Pursuable$\rangle$, $\langle mop(2{,}2)_1,$ Executive$\rangle$, $\langle mop(3{,}2)_1,$ Pursuable$\rangle$, $\langle sweep(3{,}4)_1,$ Pursuable$\rangle$	$\langle broken(mop),$ ADD$\rangle$, $\langle liquid\_dirt(2{,}2),$ REM$\rangle$, $\langle have(battery, 75),$ REM$\rangle$, $\langle have(battery, 35),$ ADD$\rangle$, $\langle have(cargo, 65.0),$ REM$\rangle$, $\langle have(cargo, 85.0),$ ADD$\rangle$, $\langle at(2{,}2),$ ADD$\rangle$, $\langle at(1{,}1),$ REM$\rangle$
2	$\langle sweep(3{,}4)_1,$ Executive$\rangle$, $\langle mop(3{,}2)_1,$ Active$\rangle$, $\langle replace(mop)_1,$ Active$\rangle$, $\langle dispose_1,$ Pursuable$\rangle$	$\langle solid\_dirt(3{,}4),$ REM$\rangle$, $\langle have(battery, 35),$ REM$\rangle$, $\langle have(battery, 5),$ ADD$\rangle$, $\langle have(cargo, 85.0),$ REM$\rangle$, $\langle have(cargo, 95.0),$ ADD$\rangle$, $\langle at(3{,}4),$ ADD$\rangle$, $\langle at(2{,}2),$ REM$\rangle$
3	$\langle recharge_1,$ Executive$\rangle$	$\langle have(battery, 5),$ REM$\rangle$,$\langle have(battery, 100),$ ADD$\rangle$, $\langle at(1{,}1),$ ADD$\rangle$, $\langle at(3{,}4),$ REM$\rangle$
4	$\langle dispose_1,$ Executive$\rangle$	$\langle have(cargo, 95.0),$ REM$\rangle$, $\langle have(cargo, 0.0),$ ADD$\rangle$, $\langle have(battery, 100),$ REM$\rangle$,$\langle have(battery, 90),$ ADD$\rangle$, $\langle at(4{,}1),$ ADD$\rangle$, $\langle at(1{,}1),$ REM$\rangle$
5	$\langle rest_1,$ Executive$\rangle$	$\langle available(mop),$ ADD$\rangle$, $\langle \neg available(mop),$ REM$\rangle$, $\langle at(1{,}1),$ ADD$\rangle$, $\langle at(4{,}1),$ REM$\rangle$
6	$\langle replace(mop)_1,$ Executive$\rangle$, $\langle rest_1,$ Pursuable$\rangle$	$\langle solid\_dirt(4{,}2),$ ADD$\rangle$, $\langle available(mop),$ REM$\rangle$, $\langle \neg available(mop),$ ADD$\rangle$, $\langle broken(mop),$ REM$\rangle$
7	$\langle mop(3{,}2)_1,$ Executive$\rangle$	$\langle liquid\_dirt(3{,}2),$ REM$\rangle$, $\langle have(battery, 90),$ REM$\rangle$, $\langle have(battery, 50),$ ADD$\rangle$, $\langle have(cargo, 0.0),$ REM$\rangle$, $\langle have(cargo, 20.0),$ ADD$\rangle$, $\langle at(3{,}2),$ ADD$\rangle$, $\langle at(1{,}1),$ REM$\rangle$
8	$\langle sweep(4{,}2)_1,$ Executive$\rangle$, $\langle dispose_2,$ Pursuable$\rangle$	$\langle solid\_dirt(4{,}2),$ REM$\rangle$, $\langle have(battery, 50),$ REM$\rangle$, $\langle have(battery, 20),$ ADD$\rangle$, $\langle have(cargo, 20.0),$ REM$\rangle$, $\langle have(cargo, 30.0),$ ADD$\rangle$, $\langle at(4{,}2),$ ADD$\rangle$, $\langle at(3{,}2),$ REM$\rangle$
9	$\langle dispose_2,$ Executive$\rangle$	$\langle have(cargo, 30.0),$ REM$\rangle$, $\langle have(cargo, 0.0),$ ADD$\rangle$, $\langle at(4{,}1),$ ADD$\rangle$, $\langle at(4{,}2),$ REM$\rangle$
10	$\langle recharge_2,$ Executive$\rangle$	$\langle have(battery, 20),$ REM$\rangle$, $\langle have(battery, 100),$ ADD$\rangle$, $\langle at(1{,}1),$ ADD$\rangle$, $\langle at(4{,}1),$ REM$\rangle$
11	$\langle rest_1,$ Executive$\rangle$	{$\emptyset$}

Source: Author's own.

BBGP model has two more stages: *deliberation* and *checking*. The *deliberation* has no rules to be converted, as it evaluates conflicts and preferences among goals, it will be encoded by the *conflict* and *preference functions*. The *checking* stage and belief rules will be discussed afterward in this subsection.

In the example, the model defines a preference, which was described with the scenario, as such, using the *execution history*, the total (decreasing) preference relation for the scenario is as follows:

$$replace(mop)_1 > mop(2{,}2)_1 > mop(3{,}2)_1 > sweep(3{,}4)_1 > sweep(4{,}2)_1 >$$

$$> recharge_1 > dispose_1 = dispose_2 > recharge_2 > rest_1$$

The *conflict function* must return $True$ if the two input goals have any conflict between them. Given the nature of the scenario and the defined goals, every goal is incompatible with each other. This is the case because the goals are localized in space, and the robot needs to

Table 2 – Mapping of the Activation and Evaluation rules to causes.

Input	Causes
$replace(X)$	$\langle replace(X),\{broken(X)\}\rangle$
$\neg replace(X)$	$\langle \neg replace(X), \{\neg available(X)\}\rangle$
$mop(X,Y)$	$\langle mop(X,Y),\{liquid\_dirt(X,Y)\}\rangle$
$\neg mop(X,Y)$	$\langle \neg mop(X,Y),\{\neg have(battery, 40), liquid\_dirt(X,Y)\}\rangle,$ $\langle \neg mop(X,Y),\{have(cargo, 80), liquid\_dirt(X,Y)\}\rangle,$ $\langle \neg mop(X,Y),\{broken(mop), liquid\_dirt(X,Y)\}\rangle$
$sweep(X,Y)$	$\langle sweep(X,Y),\{solid\_dirt(X,Y)\}\rangle$
$\neg sweep(X,Y)$	$\langle \neg sweep(X,Y),\{\neg have(battery, 30), solid\_dirt(X,Y)\}\rangle,$ $\langle \neg sweep(X,Y),\{have(cargo, 90), solid\_dirt(X,Y)\}\rangle$
$recharge$	$\langle recharge,\{have(battery,15)\}\rangle$ $\langle recharge,\{\neg have(battery,80)\}\rangle$
$dispose$	$\langle dispose,\{have(cargo,0.1)\}\rangle$
$rest$	$\langle rest,\{\emptyset\}\rangle$

Source: Author's own.

be at the specified place to perform the actions to achieve the goal. Even different goals that happen all in the same space (e.g., $rest$ and $recharge$) are considered to be in separate spaces within the workshop. The function is then trivially defined as:

Let GId be the set of goal identifiers, $gid, gid' \in GId$,

$$conflict(gid, gid') = \begin{cases} \text{if } gid \neq gid', \text{ return } True \\ \text{else return } False \end{cases}$$

4.3 Calculating the Possible *Explanans*

Based on the scenario defined in section 4.1 and the defined interface functions from section 4.2, this section presents the calculus of six different explanations using the proposed method.

Each question is answered by first identifying the information provided in the question, then rebuilding the agent knowledge base on the necessary time instance. The two explanation procedures are applied in sequence, and the set of possible *explanans* is presented at the end. Note that each *explanans* is a possible explanation, the selection of the single best explanation is outside of the scope of this work, as it is very sensitive to the domain and to whom the answer is destined.

4.3.1 *Why mop(3,2)$_1$ is active, instead of executive?*

The rationale of this question is very straightforward: why a goal is not being executed? First lets identify the information in the question:

- **Question type** – P-contrast
  - **Goal** – $mop(3,2)_1$

- **State** – Active
- **Foil State** – Pursuable (from Executive)
- **Cycle id** – 2

Notice that in the posed question, the contrasting states are *Active* and *Executive*. Chapter 3 describes only directly subsequent states comparisons, and for non-direct one, one of the states is converted to a previous one, as stated in section 3.2.5. In this example, *Executive* is converted to *Pursuable*, a previous case from Executive and directly subsequent to Active.

Since the question requires a single time frame such that the required goal was in the *Active* state, it being cycle 2, the knowledge base for that cycle can be reconstituted using the *execution history* by adding and removing the beliefs for each cycle to the initial beliefs, until the desired cycle. The relevant beliefs (KB) are:

workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(2,2)
have(battery, 35.0)
have(cargo, 85.0)
solid_dirt(3,4)
liquid_dirt(3,2)
broken(mop)

Related Conditions Procedure: First, it is necessary to evaluate the relation of the states. Be $st = Active$ the factual state, and $st' = Pursuable$ the foil state, $st' \subset st$, as such the relevant patterns are for "not advanced its state".

Next, the type of filter in use must be known. From the BBGP model used, it is known that the *Evaluation* stage uses only **negative filters**. The formula for this filter is:

$$RC = d\_pre(causes(\neg gid_a)) \cap KB^+$$

For this scenario, $KB^+ = KB$, since there are no standard rules. The formula is resolved as follows:

$$\begin{aligned} c_m : causes(mop(3{,}2)_1) = \{ & \langle mop(3{,}2)_1, \{\neg have(battery{,}40){,}liquid\_dirt(3{,}2)\}\rangle, \\ & \langle mop(3{,}2)_1, \{have(cargo{,}80){,}liquid\_dirt(3{,}2)\}\rangle, \\ & \langle mop(3{,}2)_1, \{broken(mop){,}liquid\_dirt(3{,}2)\}\rangle\} \end{aligned}$$

$$\begin{aligned} p_c : d\_pre(c_m) = \{ & \neg have(battery{,}40), liquid\_dirt(3{,}2), \\ & have(cargo{,}80), broken(mop)\} \end{aligned}$$

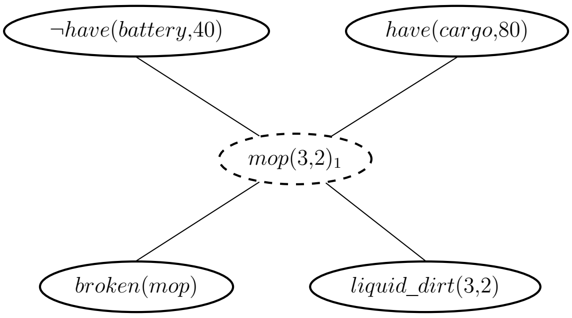
$$RC = p_c \cap KB^+ = \{\neg have(battery{,}40), liquid\_dirt(3{,}2), have(cargo{,}80), broken(mop)\}$$

That concludes the first procedure. Its output is the set RC

Possible *Explanans* Procedure: The second procedure takes the RC set in order to create the explanation graph. The set of events from the *explanadum* is $E_{Ex} = \{mop(3{,}2)_1\}$. Since $|E_{Ex}| = 1$, every condition in R is causally related to that single event. As such, every condition in R is a possible *explanans*. A depiction of the explanatory graph is shown in Figure 9.

For this case, the resulting set $RC = \{\ broken(mop),\ liquid\_dirt(3{,}2), \neg have(battery{,}40),\ have(cargo{,}80)\}$ is also the output of the second procedure. By looking into the output set, its possible to see that three of the elements are significant presumptions: $broken(mop)$, $\neg have(battery{,}40)$, and $have(cargo{,}80)$. The $liquid\_dirt(3{,}2)$ is said not to be significant as it is the activation condition of the goal. It is in the premises of the evaluation rules to avoid that the rule triggers when there is no goal to be obstructed. It serves as an example of how design decisions can improve efficiency (avoid unnecessary rules triggering) and, at the same time, make the explanation more confusing.

Figure 9 – Explanation graph of the question “Why mop(3,2) is active, instead of executive?”.



Ellipses with solid lines are the conditions (beliefs), while the dashed ellipses are the events (goals).

Source: Author’s own.

4.3.2 *Why sweep(3,4)$_1$ is executive, but mop(3,2)$_1$ is active?*

This time the rationale possibly concerns the fact that mop takes precedence over sweep. Yet the robot swept instead.

First lets identify the information in the question:

- **Question type** – O-contrast
  - **Goal$_a$** – $sweep(3{,}4)_1$
  - **Goal$_b$** – $mop(3{,}2)_1$
  - **State$_a$** – Pursuable (from Executive)
  - **State$_b$** – Active
  - **Cycle id** – 2

The question requires a single time frame where $sweep(3{,}4)_1$ was *Executive* and $mop(3{,}2)_1$ *Active* at the same time, it being cycle 2, the same as the previous question. Once again, the state *Executive* is converted to the previous state *Pursuable*. The relevant beliefs (KB) are:

workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(2,2)
have(battery, 35.0)
have(cargo, 85.0)
solid_dirt(3,4)
liquid_dirt(3,2)
broken(mop)

Related Conditions Procedure: First, it is necessary to evaluate the relation of the states. Be $st = Pursuable$ the state of the first goal, and $st' = Active$ the state of the second, $st \subset st'$, as such the relevant patterns are for "not receded its state".

Next, the type of filter in use must be known. From the BBGP model used, it is known that the *Evaluation* stage uses only **negative filters**. The formula for this filter is:

$$A = c\_pre(causes(\neg gid_a)) - KB^+$$

$$B = c\_pre(causes(\neg gid_b)) \cap KB^+$$

$$RC = uses\_pred(B, predicate(A) \cap predicate(B))$$

As $KB^+ = KB$, the formula is resolved as follows:

$$c_s : causes(\neg sweep(3{,}4)_1) = \{\langle \neg sweep(3{,}4)_1, \{\neg have(battery{,}30), solid\_dirt(3{,}4)\}\rangle,$$
$$\langle \neg sweep(3{,}4)_1, \{have(cargo{,}90), solid\_dirt(3{,}4)\}\rangle\}$$

$$p_s : c\_pre(c_s) = \{\neg have(battery{,}30) \wedge solid\_dirt(3{,}4), have(cargo{,}90) \wedge solid\_dirt(3{,}4)\}$$

$$A = p_s - KB^+ = \{\neg have(battery{,}30) \wedge solid\_dirt(3{,}4),$$
$$have(cargo{,}90) \wedge solid\_dirt(3{,}4)\}$$

$$c_m : causes(\neg mop(3{,}2)_1) = \{\langle \neg mop(3{,}2)_1, \{\neg have(battery{,}40), liquid\_dirt(3{,}2)\}\rangle,$$
$$\langle \neg mop(3{,}2)_1, \{have(cargo{,}80), liquid\_dirt(3{,}2)\}\rangle,$$
$$\langle \neg mop(3{,}2)_1, \{broken(mop), liquid\_dirt(3{,}2)\}\rangle\}$$

$$p_m : c\_pre(c_m) = \{\neg have(battery{,}40) \wedge liquid\_dirt(3{,}2),$$
$$have(cargo{,}80) \wedge liquid\_dirt(3{,}2), broken(mop) \wedge liquid\_dirt(3{,}2)\}$$

$$B = p_m \cap KB^+ = \{\neg have(battery{,}40) \wedge liquid\_dirt(3{,}2),$$
$$have(cargo{,}80) \wedge liquid\_dirt(3{,}2), broken(mop) \wedge liquid\_dirt(3{,}2)\}$$

$$RC = uses\_pred(B, predicate(A) \cap predicate(B)) = \{\neg have(battery{,}40), have(cargo{,}80)\}$$

That concludes the first procedure, the output is a set of preconditions that did not influenced $sweep(3{,}4)_1$ but impeded $mop(3{,}2)_1$, defined by the set $RC = \{\neg have(battery{,}40), have(cargo{,}80)\}$.

Possible *Explanans* Procedure: The *explanandum* defines two events to be explained $E_{Ex} = \{sweep(3{,}4)_1, mop(3{,}2)_1\}$. With the set of related conditions RC, the explanation graph $Gp =< Ver, Edg >$, such that:

$$Ver = \{sweep(3{,}4)_1, mop(3{,}2)_1, \neg have(battery{,}40), have(cargo{,}80)\}$$
$$Edg = \{(have(cargo{,}80), sweep(3{,}4)_1), (have(cargo{,}80), mop(3{,}2)_1),$$
$$(\neg have(battery{,}40), sweep(3{,}4)_1), (\neg have(battery{,}40), mop(3{,}2)_1)\}$$

A graphical depiction of Gp is shown in Figure 10. Knowing that the *MPE Algorithm* avoids redundancy in the causes, it is easy to see that there are two minimal possible *explanans* for Gp. The output of the second procedure is $\mathcal{PE} = \{\{\neg have(battery{,}40)\}, \{mop(3{,}2)_1)\}\}$.

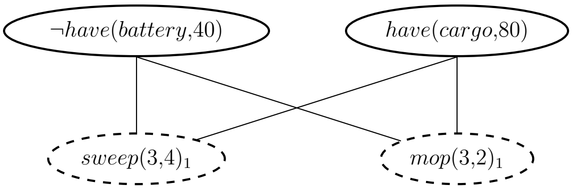
Each possible *explanans* (PE) can be interpreted as:

- $\neg have(battery{,}40)$ – the agent had battery to perform $sweep(3{,}4)_1$, but not enough to perform $mop(3{,}2)_1$.

- $mop(3,2)_1$ – the agent had enough internal storage left to perform $sweep(3,4)_1$, but not enough to perform $mop(3,2)_1$.

It is worth mentioning that $mop(3,2)_1$ had another reason not to become Pursuable, which is the $broken(mop)$. But since the mop is not related to sweeping, this piece of information was disregarded. Other types of questions can preserve that information, as shown in the previous case in 4.3.1.

Figure 10 – Explanation graph of the question "Why sweep(3,4) is executive, but mop(3,2) is active?".



Ellipses with solid lines are the conditions (beliefs), while the dashed ellipses are the events (goals).

Source: Author's own.

4.3.3 *Why sweep(4,2)$_1$ is not executive, but rest$_1$ is?*

Again the rationale is: rest is the last activity that the robot should do, and there are tasks pending. Why?

- **Question type** – O-contrast
  - **Goal$_a$** – $sweep(4,2)_1$
  - **Goal$_b$** – $rest_1$
  - **State$_a$** – Sleeping (from not Executive)
  - **State$_b$** – Active (from Executive)
  - **Cycle id** – 5

In this case, first, the *not Executive* state needs to be resolved. The state from goal $sweep(4,2)_1$ needs to be retrieved at a time frame when goal $rest_1$ is *Executive*. Such time frame corresponds to *cycle 5*. In cycle 5, the goal $sweep(4,2)_1$ is unknown, that is, it was not active. For this reason, $state_a$ is converted from *not Executive* to *Sleeping* (the fictional state that a goal has before being activated). In turn $state_b$ is converted from $Executive$ to $Active$.

The reconstructed knowledge base for cycle 5 is as follows:

workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(4,1)
have(battery, 90.0)
have(cargo, 0.0)
liquid_dirt(3,2)
broken(mop)

Related Conditions Procedure: The states relationship is $st = Sleeping$ for the first goal, and $st' = Active$ the state of the second one, $st' \subset st$, as such the relevant patterns are for "not advanced it's state".

Next, the type of filter in use must be known. From the BBGP model used, it is known that the *Activation* stage uses only **positive filters**. The formula for this filter is:

$$A = c\_pre(causes(gid_a)) - KB^+$$

$$B = c\_pre(causes(gid_b)) \cap KB^+$$

$$RC = uses\_pred(A, predicate(A) \cap predicate(B))$$

Solve the formula results in:

$$c_s : causes(sweep(4{,}2)_1) = \{\langle sweep(4{,}2)_1, \{solid\_dirt(4{,}2)\}\rangle\}$$

$$p_s : c\_pre(c_s) = \{solid\_dirt(4{,}2)\}$$

$$A = p_s - KB^+ = \{solid\_dirt(4{,}2)\}$$

$$c_b : causes(rest_1) = \{\langle rest_1, \{\emptyset\}\rangle\}$$

$$p_b : c\_pre(c_b) = \{\emptyset\}$$

$$B = p_b \cap KB^+ = \{\emptyset\}$$

$$RC = uses\_pred(A, predicate(A) \cap predicate(B)) = \{\emptyset\}$$

Notice that the resulting set is empty, as $related(sweep(4{,}2)_1, rest_1) = False$. Since both goals do not share presumptions, instead of stating that both goals are not causally related, the follow-up question *"why sweep(4,2) is not active?"* is answered instead, as recommended in section 3.2.5. The question information is:

- **Question type** – P-contrast
  - **Goal** – $sweep(4{,}2)_1$
  - **State** – Sleeping (from not Active)
  - **Foil State** – Active
  - **Cycle id** – 5

The knowledge base remains the same.

Related Conditions Procedure: The states relationship is $st = Sleeping$ for the first goal, and $st' = Active$ the state of the second one, $st' \subset st$, as such the relevant patterns are for "not advanced it's state".

The type of filter in use is the **positive filter**. The formula is:

$$RC = c\_pre(causes(gid_a)) - KB^+$$

Which solution is:

$$c_s : causes(sweep(4{,}2)_1) = \{\langle sweep(4{,}2)_1{,}\{solid\_dirt(4{,}2)\}\rangle\}$$

$$p_s : c\_pre(c_s) = \{solid\_dirt(4{,}2)\}$$

$$RC = p_s - KB^+ = \{solid\_dirt(4{,}2)\}$$

Possible *Explanans* Procedure: Since $|RC| = 1$, there is no need to construct the explanation graph. The answer is trivially given by "$solid\_dirt(4{,}2)$ is unknown".

4.3.4 *Why dispose$_2$ was executive at t$_9$, but dispose$_1$ was pursuable at t$_3$?*

This time the questions seek the difference between two similar scenarios: after the robot sweeps, it needs to dump its reservoir, but why, in one instance, did he not go to the dumpster?

The question information are:

- **Question type** – OT-contrast
  - **Goal$_a$** – $dispose_2$
  - **Goal$_b$** – $dispose_1$
  - **State$_1$** – Chosen (from Executive)
  - **State$_2$** – Pursuable
  - **Cycle id$_1$** – 9
  - **Cycle id$_2$** – 3

First make $Cycle\ id_1 < Cycle\ id_2$. For that end, every pair of elements are swapped:

- **Question type** – OT-contrast
  - **Goal$_a$** – $dispose_1$
  - **Goal$_b$** – $dispose_2$
  - **State$_1$** – Pursuable
  - **State$_2$** – Chosen (from Executive)
  - **Cycle id$_1$** – 3
  - **Cycle id$_2$** – 9

Also, $state_1$ is converted from *Executive* to *Chosen*.

Two knowledge bases need to be reconstructed: a) KB_1 for cycle 3:

workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(3,4)
have(battery, 10.0)
have(cargo, 95.0)
liquid_dirt(3,2)
broken(mop)

and b) KB_2 for cycle 9:

```
workshop(1,1)
dumpster(4,1)
at(4,2)
have(battery, 20.0)
have(cargo, 30.0)
¬ available(mop)
```

Related Conditions Procedure: The states relationship is $st = Pursuable$ for the first goal, and $st' = Chosen$ the state of the second one, $st' \subset st$, as such the relevant patterns are for “not advanced it's state”.

Next, the type of filter in use must be known. From the BBGP model used, it is known that the *Chosen* stage uses only **preference filters**. The preference filters require a series of ordered tests. The first satisfied one is the answer.

- “gid_b has no incompatibilities at t_2” is **false** as:
  $G^2_{st'} = \{rest_1, dispose_2\}$ and they are conflicting.

- “A subset of goals incompatible with gid_a are not present at t_2” is **true**:
  $G^1_{st'} = \{rest_1, recharge_2, recharge_1, dispose_1\}, G^2_{st'} = \{rest_1, dispose_2, recharge_2\},$
  which results in $recharge_1 \notin G^2_{st'} \wedge incompatible(recharge_1{,}dispose_1)$

As such, the result set of conditions $RC = \{$“A subset of goals incompatible with $dispose_1$ are not present at cycle 9”$\}$. Since RC is a *preference assertion*, the *Possible Explanans procedure* is not required.

4.3.5 *Why replace(mop)$_1$ was not executive at t_2, but it was at t_6?*

This question seeks why an action happened at a given time and not before.

The question informations are:

- **Question type** – T-contrast
  - **Goal** – $replace(mop)_1$
  - **State$_1$** – Active
  - **State$_2$** – Pursuable (from Executive)
  - **Cycle id$_1$** – 2
  - **Cycle id$_2$** – 6

The cycles are ordered. The $state_2$ is converted from *Executive* to *Evaluated*. The two knowledge bases required are: a) KB_1 for cycle 2:

```
workshop(1,1)
dumpster(4,1)
¬ available(mop)
at(2,2)
have(battery, 40.0)
have(cargo, 85.0)
solid_dirt(3,4)
liquid_dirt(3,2)
broken(mop)
```

and b) KB_2 for cycle 6:

```
workshop(1,1)
dumpster(4,1)
at(1,1)
have(battery, 90.0)
have(cargo, 0.0)
liquid_dirt(3,2)
broken(mop)
available(mop)
```

Related Conditions Procedure: The states relationship is $st = Active$ for the first goal, and $st' = Pursuable$ as the state of the second one, $st' \subset st$, as such, the relevant patterns are for “not advanced it's state”.

Next, the type of filter used for the *Pursuable* stage is **negative filters**. The formula for this filter is:

$RC = (d\_pre(causes(\neg gid_a)) \cap KB_1^+) \cap (d\_pre(causes(\neg gid_a)) - KB_2^+)$

Solving the formula:

$$c_r : causes(\neg replace(mop)_1) = \{\langle \neg replace(mop)_1, \{\neg available(mop)\}\rangle\}$$

$$p_r : d\_pre(c_r) = \{\neg available(mop)\}$$

$$RC^1 : p_r \cap KB^1 = \{\neg available(mop)\}$$

$$RC^2 : p_r - KB^2 = \{\neg available(mop)\}$$

$$RC = RC^1 \cap RC^2 = \{\neg available(mop)\}$$

Possible *Explanans* Procedure: As $|RC| = 1$, the output is RC.

The next chapter discusses the case study results and the limitations of the method.

5 DISCUSSION

The method presented allows the construction of contrastive answers for general purpose questions about an agent's goal selection process. The posed questions need to fall in one of the four types described in the formalization: P-contrast, O-contrast, T-contrast, or OT-contrast.

Using Miller's definition of explanation, the method is entirely within the *cognitive process* of determining the explanation. Yet it is not sufficient to complete it. One important step required to obtain the *product* of the cognitive process is the selection of the *explanans*, that is, given the possible answers, which one will be presented. Other than that, the *social process* is outside the scope, both in relaying the answer to the explainee as to receiving the question. This problem requires Human-Computer-Interaction and possibly Natural-Language-Processing, depending on the intended use.

Next, it is discussed how the method addresses Grice's Cooperation Principles, then some of the method's limitations are discussed. Next, the reasoning for the restriction on the goal state property always needing to be distinct is discussed. Lastly, the related work is presented.

5.1 Grice's Cooperation Principle as Requirements

Grice's Cooperation Principles, or Grice's Maxims, is a group of four categories of requirements for a conversation between cooperating parties. The maxims were proposed as an analysis of how people communicate when cooperating and the effects of deviating from an expected pattern. Its summary is presented in subsection 2.2.5.

For this work, the maxims were used as a guide for how the agent should present its explanations, as the explanation process is a kind of conversation. For this reason, for each of the maxims, how well the method can address them is discussed. Note that the method is evaluated in relation to explanations, but the maxims are for general conversation.

- **Manner** – first, let us begin with the category of maxims that are not addressed. Since each of the maxims is related to the relaying of the explanation, and that task is outside the scope of this work, the category is left as an open problem.
- **Relation** – the relation category is very brief, as it has a single maxim "*Be relevant*". The maxim is addressed by guaranteeing that every piece of information that the agent considers relevant to the explanation is causally related. This requirement is related to one of the method's assumptions, that the relationship of *goals* and *beliefs* is causal. From the rule-based nature of the agents, it is a fair assumption to be made, as removing a belief that is part of the premises of a rule will make that rule not active, and the opposite is also true, adding all missing beliefs of a rule will cause it to become active.

Take as an example the formulation from the O-contrast, “not receded” negative filter:

$$A = c\_pre(\boldsymbol{causes}(\neg gid_a)) - KB^+$$
$$B = c\_pre(\boldsymbol{causes}(\neg gid_b)) \cap KB^+$$
$$RC = uses\_pred(B, predicate(A) \cap predicate(B))$$

The *causes* function ensures the causal relation of the presumptions to the events. In short, by designing the method around causal rules derived from the relation of goals and beliefs (that can be encoded in the agent’s model as rules, plans, or other elements), every information considered is relevant to the explanation.

- **Quality** – the quality category is concerned with how trustworthy is the information used in the explanation. There are two maxims that, in short, say “do not lie” and “do not say what you do not know”. This requirement is dependent on the agent’s reasoning mechanism, that is, if the agent follows the open-world assumption – in which the only beliefs that are true are the ones that can be derived from the knowledge base, and everything else is *unknown* – then the requirement is met. Since it was designed that the knowledge base evaluates the causes and filter what is derivable and what is not, the agent does “believe” in the explanations it provides, even if factually they are wrong (e.g., the robot may say that $\neg available(mop)$, but the user knows he delivered one earlier, even the robot being wrong, it is not trying to deceive the user, it is just what he believes to be true). A compromise can be made by using a closed world assumption – everything that is not derivable from the knowledge base is *false* –, as the agent may make a decision on an inferred negation that ends up being wrong. Note that the agent can be factually wrong about its view of the world in both cases. Again, using the same previous example formulation:

$$A = c\_pre(causes(\neg gid_a)) - \boldsymbol{KB}^+$$
$$B = c\_pre(causes(\neg gid_b)) \cap \boldsymbol{KB}^+$$
$$RC = uses\_pred(B, predicate(A) \cap predicate(B))$$

The usage of KB^+ ties the presumptions of possible causes to the agent’s beliefs. Note that before the operations with KB, the formula gathers every cause that could be related to the event. The $\cap KB^+$ then guarantees that only factual presumptions are considered. In turn, the $-KB^+$ guarantees that only unknown presumptions are considered. They are both used to evaluate if a cause needs to be activated – all the presumptions made true –, or deactivated – some presumptions, for every active cause, made false –, respectively.

- **Quantity** – the quantity category is concerned with the amount of information provided, as in short, it requires that “be as informative as required, and no more than that”. It is subjective how informative a piece of information is. Not just in the sense that it depends on who receives the information, it is also not measurable. In a sense, this is part of the *selection problem* – how to select the most adequate answer – as knowing the “most informative” *explanans* could be a solution to the problem. How the method addresses this requirement is twofold: i) by contrasting with another goal, when applicable; and ii) by avoiding repetitive information, that is, if a condition c_1 can answer all the required events, there is no need to include a second condition. For example, using the same case as before:

$$A = c\_pre(causes(\neg gid_a)) - KB^+$$

$$B = c\_pre(causes(\neg gid_b)) \cap KB^+$$

$$RC = \boldsymbol{uses\_pred}(B, predicate(A) \cap predicate(B))$$

  The $uses\_pred$ function, in this scenario, looks for presumptions that are shared between the goals – this narrows down the number of presumptions that are considered. Next, the (Minimal) Possible *Explanans* procedure is responsible for generating subsets from this narrowed set of presumptions that are, in turn, related to every event of the *explanandum*.

In summary, three of the four maxims are addressed. They influenced the method’s design: first, in how the sets of related conditions were formulated; and secondly, by proposing the second procedure, responsible for removing redundancies.

Next, some of the limitations of the method are discussed.

5.2 Limitations

Two limitations are discussed in the following subsection: i) the dependence on the agent’s model; and ii) the developer’s responsibility in implementing the agent’s interface.

5.2.1 The impact of the belief-goal relationship

The proposed method is dependent on the agent’s beliefs and the ability to associate them with the goals. As such, the resulting explanations can only be as rich and precise as the agent’s knowledge base and causal function provided through the interface.

Many classical BDI models have a very limited relationship between goals and beliefs. Take as an example the *AgentSpeak* framework: the only **required** relation between a belief and a goal is the plan library, with the guard clauses. The motivation of a new goal is a *trigger* that can be received as a message, a result of plan execution, or a belief rule. As such, the motivation for a goal may not be clear. If it can not be expressed as beliefs – in the AgentSpeak, it is only possible when using rules – then the motivation is oversimplified.

Suppose the following snippet of a AgentSpeak code, containing every reference to $clean$ for the same cleaner world scenario:

```
solid_dirt(3,4)
+liquid_dirt(X,Y) <- !clean(X,Y)
+solid_dirt(X,Y) <- !clean(X,Y)
+!clean(X,Y) : clear_path(X,Y), liquid_dirt(X,Y), has_battery(20.0) <- !go(X,Y); !mop(X,Y)
+!clean(X,Y) : clear_path(X,Y), solid_dirt(X,Y), has_battery(20.0) <- !go(X,Y); !sweep(X,Y)
+!clean(X,Y) : not liquid_dirt(X,Y), not solid_dirt(X,Y) <- .drop_intention(clean(X,Y))
```

For this snippet, the explanation method can obtain a belief that motivates the goal of cleaning: the solid dirt at (3,4).

Now comparing to this snippet, again for the same problem, with the same behavior:

```
solid_dirt(3,4)
!clean(3,4)
+!clean(X,Y) : clear_path(X,Y), liquid_dirt(X,Y), has_battery(20.0) <- !go(X,Y); !mop(X,Y)
+!clean(X,Y) : clear_path(X,Y), solid_dirt(X,Y), has_battery(20.0) <- !go(X,Y); !sweep(X,Y)
```

It is easy to conceive a code that achieves the same behavior, yet the ability to track the motivation of the goal is lost. The robot wants to clean because it received a message to do so. It has an applicable plan, but what belief, if any, from the guard clause is related to the motivation? In this case, the explanation method can not narrow the options for the activation of the goal, and it is possible that the correct explanation may not even be part of the robot's beliefs.

The BBGP model used in the case study is a good example of a model that encourages and enforces a rich goal-belief relationship. The goal needs to progress over several stages, each requiring a set of beliefs to be present or absent. A goal has a motivating belief that, if the domain model allows, can make clear the belief that leads to a goal being pursued.

5.2.2 Adherence to Agent's Deliberation

One important point to be considered is that the method is entirely *post-hoc*. As such, the adherence – that is, how closely a model can represent another model – to the constructed explanations will be only as good as the adherence between the interface implementation (causal, conflict, and preference functions and the execution history) and the agent's deliberative process. The correctness of the provided information falls under the developer's responsibility.

As discussed before, the agent's model plays an important role in the explanation richness, as the explanation should be only as rich as the used model. Trying to extend the agent's model would only negatively impact the adherence of the explanation to the deliberation process, as new assumptions and relations not used by the agent would be added to the explanation model. That can generate explanations that induce the user to make false assumptions about the agent's behavior and mechanisms.

It is worth noting that the method was designed having in mind a cooperative agent. That is, at this point, the agent is deemed as fully cooperative, providing any information requested without any reason to withhold data or even to lie. As such, the agent keeps a history of its execution that tells the agent's intentions factually, even if some of its beliefs are wrong.

5.3 Single State Property Questions

Posed questions necessarily have different states being compared. Upon looking at the elements that compose the formalized question types, three elements are used: the *goal* as the event being compared, the *property* as the state of the goal, and the *time instance* upon which the comparison takes place. Considering these three elements like dimensions, with two possible values (*equal* or *distinct*), eight possibilities for question types could be formulated. Nonetheless, only four are provided. That is the case because the *property* element is always considered distinct for a contrastive question to be considered valid.

For three of the formalized question types (O/T/OT-contrast), the answer requires comparing goals or time instances: O-contrast compares two distinct goals in a single time instance; T-contrast compares a single goal in two distinct time instances, and OT-contrast compares two distinct goals in two distinct time instances. The P-contrast compares a goal with a hypothetical version of itself, but in different states, in a single time instance. In all cases, the goal states being compared are always different.

When the goal properties are the same, the intention behind the questions lacks the requirement for a comparison to be made:

- **P-contrast** – If all the properties are the same, the question is malformed: "why goal a (at time t_1) is 'Pursuable' instead of 'not Pursuable'?". The question is a contradiction, as a can not have state $St \land \neg St$ at the same time.

- **O-contrast** – When asked why two distinct goals have the same states at the same time, it is likely that both were expected to be mutually exclusive – if one goal is being pursued, the other one can not be –, which they were not. As such, the explanation should provide a reason for the mutual exclusion criteria not holding (e.g., another non-conflicting parallel activation path), or if such criteria can not be identified, simply the factual reason why both have the observed state. In both cases, no comparison is made.

- **T-contrast** – When asking why the same goal at two distinct times have the same states, the question entails that the goal was expected to have changed. As such, the question is possibly asking why the goal did not change. A better way of posing this question is by identifying if it was expected that goal had advanced or receded and then posing a question with distinct *properties*. Excluding this case, no intention can be defined for the question posed. As such, an adequate answer can not be constructed.

- **OT-contrast** – Lastly, for two different goals, at different times, with the same states, the question becomes too arbitrary, as both goals are not happening concomitantly, not even a mutual exclusion criterion can be assumed. The intention behind such a question is also not clear.

This shows that the *property* needs to be distinct in the question.

5.4 Related Work

Research in contrastive explanations grounded in social and cognitive sciences is still in its early stages (MILLER; HOWE; SONENBERG, 2017; STEPIN *et al.*, 2021).

In (KAPTEIN *et al.*, 2019), the authors present their evaluation of a virtual assistant system, to assess the use of emotions in the explanation and its effects on the adoption of the given suggestions. Although grounded on social sciences, this work does not encompass the agent's goal selection and focuses on the human-agent interaction. In a similar approach, Stange and Kopp (2020) uses emotions and BDI concepts to frame the generated explanations. It also is focused on the human-agent interaction. In (HARBERS; BOSCH; MEYER, 2010) the authors evaluate explanations types and when actions or goals are better explanations for the posed question. It also has a grounding in social sciences. All these approaches have social science's grounding but do not provide contrastive explanations.

Fan (2018) proposes a method using argumentation to explain plans. It does not seem to have social or cognitive grounding, and the method does not address goal selection.

Chakraborti *et al.* (2019) evaluates in his work the human-agent interaction aspect of the explanation as well. His work is grounded in social sciences and is capable of providing some contrastive answers, but it focuses on plan explanation. Somewhat similar to Chakraborti's work is (CRUZ; IGARASHI, 2021), where the explanation is used for debugging a reinforcement learning rule of an autonomous agent in a game, allowing the user to compare and visualize the behavior that will ensue from the changed rule.

Sklar and Azhar (2018) proposes a dialogue protocol, which is based on social sciences, but does not focus on explanation generation. It also does not encompass contrastive explanations. Another dialogue protocol was proposed by Dennis and Oren (2021), but his approach does not seem to have any grounding in social or cognitive sciences, nor is it capable of providing contrastive explanations.

Some works in computational argumentation present a somewhat similar approach to the one presented in this work, in such a way that the present problem could be reduced to use the explanation methods of these works. In (FAN; TONI, 2015) is presented an explanation semantic, where a targeted argument and its defendants compose an explanation. The difference between both approaches is in the focus of the explanations: Fan and Toni's (2015) approach is based on the argument's claim, while the approach proposed in this work is based on the agent's beliefs that are used on the premises of arguments. The beliefs are used because they are the elements that will change the agent's behavior. García *et al.* (2013) uses dialectical trees to generate its explanations, but again it is based on the claims of the arguments.

On the scope of intelligent agents, several works deal with different aspects of agency explainability: using goal hierarchy to explain behavior (HARBERS; BOSCH; MEYER, 2010); explaining the outcome of an inter-agent dialogue (RAYMOND; GUNES; PROROK, 2020); debugging the agent's behavior (WINIKOFF, 2017); and how to model and use emotions on explanations (KAPTEIN *et al.*, 2017).

The works mentioned above do not explain what motivates and enable a goal. In (MADUMAL *et al.*, 2020), the authors present a method for explaining the actions an agent performs by using structural causal models that allow a counterfactual comparison. But again, the work does not explain the agent's motivation and enabling beliefs.

In (MORVELI-ESPINOZA; POSSEBOM; TACLA, 2019) and (MORVELI-ESPINOZA; TACLA; JASINSKI, 2020) is presented an explanation method for goal selection. It explains the agent's goal selection but lacks social sciences grounding and does not account for contrastive explanations.

Table 3 presents an overview of the related works. It shows the works with some type of contrastive explanations and which works have social or cognitive sciences grounding.

To the best of our knowledge, no work has been published addressing contrastive explanations for goal selection.

Related Work	Contrastive Explanation	Social/Cognitive Grounding
(KAPTEIN *et al.*, 2019)		X
(STANGE; KOPP, 2020)		X
(HARBERS; BOSCH; MEYER, 2010)		X
(FAN, 2018)		
(CHAKRABORTI *et al.*, 2019)	Partial	X
(CRUZ; IGARASHI, 2021)	X	
(SKLAR; AZHAR, 2018)		X
(DENNIS; OREN, 2021)		
(RAYMOND; GUNES; PROROK, 2020)		
(WINIKOFF, 2017)		
(KAPTEIN *et al.*, 2017)		X
(MADUMAL *et al.*, 2020)		X
(MORVELI-ESPINOZA; POSSEBOM; TACLA, 2019)		X
(MORVELI-ESPINOZA; TACLA; JASINSKI, 2020)		X

Table 3 – List of related work.

Source: Author's own.

6 CONCLUSIONS AND FUTURE WORKS

This chapter concludes the dissertation and presents some future directions.

6.1 Conclusion

First, let us recall the research question posed in this work:

Grounded on social and cognitive sciences works, what information is required to construct contrastive explanations for BDI-based agent's goal selection process and how to generate such contrastive explanations?

Two social sciences works were used as a foundation for the method proposed:

- Bouwel and Weber's (2002) work provided the underlying structure of what is a good explanation for P/O/T-contrast questions, which was adapted to the BDI agent model and extended with the proposal of OT-contrast questions. This extended structure of contrastive questions provides a definition of the information that is required for answering contrastive questions about the goal selection, in this work, the set of presumptions that differentiate the two cases being compared.
- Grice's (1975) work provided some requirements that shaped the assumptions of the method – being for cooperative agents – and influenced the designed formulations of the related conditions. Besides, the second procedure, *Possible Explanans*, was included to meet the Quantity maxim. The formulations for the four question types in the first procedure addresses the Quality (say only what is believed to be true) and Relation (say only what is related to the question) maxims by only generating sets of presumptions that are part of a causal tree to the events, and by grounding the presumptions to the agent's knowledge base (at the time of the decision making). The MPE algorithm (second procedure) addresses the Quantity maxim (say only what is needed) by removing redundant presumptions from the final possible *explanans*.

A method was proposed in chapter 3 for generating contrastive possible *explanans* for contrastive questions about a BDI-based goal selection process. A possible *explanans* is the set of information to be relayed to the questioner to present a relevant and simple answer. Given the BDI-agent focus, answers are always based on presumptions – beliefs without premises or facts – as they are a more volatile element in the agent's knowledge base and can, in some cases, be easily manipulated without the need for expertise in agent modeling. Allowing a lay user not only to understand a behavior but having the chance of influencing it. Chapter 4 presented a case study based on the cleaner world scenario. This shows how to implement the agent's interface, how the possible *explanans* are calculated in the method, and demonstrates how to interpret some of the resulting *possible explanans*.

The method does not cover two important steps required for providing a final user-ready explanation: a) it does not tackle the *selection problem* of deciding which single *explanans* will be used as an answer; and b) the interface to the user – how to receive and interpret the question and how to present the explanation – is outside the scope of this work.

Next, some future directions are presented.

6.2 Future Work

Given the deep connection with the work of Bouwel and Weber (2002), other contrastive question types can be used with the proposed method. For that, the new question type needs to be formally defined. It is worth mentioning that the formalization grows (potentially) exponentially with the number of types. Let's take as an example the Spatial-contrast (S-contrast), mentioned in Bouwel and Weber's work, it can be combined with every other type of question, effectively adding a new dimension to the question. It would be possible: the plain S-contrast, Object-Spatial-contrast, Time-Spacial-contrast, and Object-Time-Spatial-contrast. If a combination does not result in a malformed question, then the number of resulting types is in the order of 2^{n-1}, n being the number of question types.

Another possible improvement is to include special questions that can better explain internal behaviors of the agent that are beyond the standard BDI model. A good example would be a special question type to explain trust deliberation in an agent. Since trust is another type of deliberation that the agent must perform, a dedicated question type can potentially give simpler answers.

For practical applications, an important improvement to the explanation method is to support stochastic plans. This would allow the generation of probabilistic explanations, that is, explanations of the type: "It is probable that event e happened because of conditions $\{c_1, ..., c_n\}$", by including a representation of the odds of a plan being chosen and succeeding.

The *selection problem* is left open in the method, and integrating it into the method would allow a complete explanation generation, requiring only to relay it to the user. Many criteria were proposed for this problem, and to implement them, it is necessary to include some information about the causes. For instance, knowing what presumptions can be controlled by the user would allow the use of the *instrumental efficacy* criteria. Also, knowing what presumptions are unexpected conditions for the user allows the usage of the *unexpected* or *abnormal conditions* criteria.

Some of such selection criteria require extra information about the causes. It requires causes to be compared or classified based on some characteristics. To name a few: controllability, unexpectedness, and responsibility. This extra information can be helpful (in some cases required) to tackle the *selection problem*.

The method can only compare sequential goal states. If the finite state machine (FSM) describing the goals state progression is not a directed acyclic graph (DAG), the method cannot properly perform the comparisons. The special cases discussed (*Cancelled*, *Completed*, and *Paused*) are not based on beliefs and, as such, are not comparable. The method could be extended to accommodate for a more general goal state state-machine.

This work did not tackle how a lay user could interact with the agent. One challenge arising from such interaction is “how a lay user can question the agent without knowing about the goal states and deliberation cycles?”. It is possible that only by using the observed behavior of the agent, that is, the actions it did or did not perform, the question could be mapped to the particular goal states required to satisfy the user’s query. A deeper analysis of such interactions is still required.

Lastly, the explanation method can deal with contrastive questions and provide contrastive answers. Although counterfactual answers are a type of contrastive answers, the method still lacks a way of generating the hypothesized scenario (beliefs, goals, preferences, etc.). With an efficient method for generating counterfactual scenarios, that is, generating such scenarios without the need of executing an agent with the required changes, the formalized question types should be capable of providing adequate answers. Integrating such a method into the proposed method can significantly improve the quality and usefulness of the explanations, as it will increase the range of questions that the agent can answer.

BIBLIOGRAPHY

AMGOUD, L.; BESNARD, P. A formal characterization of the outcomes of rule-based argumentation systems. **Knowledge and Information Systems**, Springer Science and Business Media LLC, v. 61, n. 1, p. 543–588, jun. 2018. Disponível em: https://doi.org/10.1007/s10115-018-1227-5.

BERMÚDEZ, J. L. **COGNITIVE SCIENCE An Introduction to the Science of the Mind Second Edition**. 2nd. ed. [*S.l.*]: Cambridge University Press, 2014. ISBN 9781107653351.

BOUWEL, J. V.; WEBER, E. Remote causes, bad explanations? **Journal for the Theory of Social Behaviour**, Wiley, v. 32, n. 4, p. 437–449, dez. 2002. Disponível em: https://doi.org/10.1111/1468-5914.00197.

CASTELFRANCHI, C.; PAGLIERI, F. The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. **Synthese**, v. 155, n. 2, p. 237–263, feb 2007. ISSN 0039-7857. Disponível em: http://link.springer.com/10.1007/s11229-006-9156-3.

CHAKRABORTI, T. *et al.* Plan explanations as model reconciliation: An empirical study. *In*: **Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction**. [*S.l.*]: IEEE Press, 2019. (HRI '19), p. 258–266. ISBN 9781538685556.

CRUZ, C. A.; IGARASHI, T. Interactive explanations: Diagnosis and repair of reinforcement learning based agent behaviors. *In*: **2021 IEEE Conference on Games (CoG)**. IEEE, 2021. Disponível em: https://doi.org/10.1109/cog52621.2021.9618999.

DANNENHAUER, D. *et al.* Learning from exploration: Towards an explainable goal reasoning agent. **Proceedings of IJCAI-18 Workshop on Adaptive Learning Agents**, 2018.

DENNIS, L. A.; OREN, N. Explaining bdi agent behaviour through dialogue. *In*: **Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems**. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2021. (AAMAS '21), p. 429–437. ISBN 9781450383073.

FAN, X. On generating explainable plans with assumption-based argumentation. *In*: **Lecture Notes in Computer Science**. Springer International Publishing, 2018. p. 344–361. Disponível em: https://doi.org/10.1007/978-3-030-03098-8_21.

FAN, X.; TONI, F. On computing explanations in argumentation. *In*: **Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence**. [*S.l.*]: AAAI Press, 2015. (AAAI'15), p. 1496–1492. ISBN 0262511290.

FLOYD, M. W. *et al.* A Goal Reasoning Agent for Controlling UAVs in Beyond-Visual-Range Air Combat. **IJCAI-17**, p. 4714–4721, 2017.

GARCÍA, A. J. *et al.* Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. **Expert Systems with Applications**, Elsevier BV, v. 40, n. 8, p. 3233–3247, jun. 2013. Disponível em: https://doi.org/10.1016/j.eswa.2012.12.036.

GEORGEFF, M. *et al.* The belief-desire-intention model of agency. *In*: **Intelligent Agents V: Agents Theories, Architectures, and Languages**. Springer Berlin Heidelberg, 1999. p. 1–10. Disponível em: https://doi.org/10.1007/3-540-49057-4_1.

GRICE, H. P. Logic and conversation. *In*: ____. **Syntax and semantics 3: Speech Acts**. Leiden, The Netherlands: Brill, 1975. p. 41 – 58. ISBN 9789004368811. Disponível em: https://brill.com/view/book/edcoll/9789004368811/BP000003.xml.

HARBERS, M.; BOSCH, K. van den; MEYER, J.-J. Design and evaluation of explainable BDI agents. *In*: **2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology**. IEEE, 2010. Disponível em: https://doi.org/10.1109/wi-iat.2010.115.

HAYNES, S. R.; COHEN, M. A.; RITTER, F. E. Designs for explaining intelligent agents. **International Journal of Human-Computer Studies**, v. 67, n. 1, p. 90–110, 2009. ISSN 1071-5819. Disponível em: https://www.sciencedirect.com/science/article/pii/S1071581908001274.

HESSLOW, G. The problem of causal selection. *In*: HILTON, D. J. (Ed.). **Contemporary Science and Natural Explanation: Commonsense Conceptions of Causality**. [*S.l.*]: New York University Press, 1988.

JOSEPHSON, J. **Abductive inference : computation, philosophy, technology**. Cambridge New York: Cambridge University Press, 1994. ISBN 0521434610.

KAPTEIN, F. *et al.* The role of emotion in self-explanations by cognitive agents. *In*: **2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)**. IEEE, 2017. Disponível em: https://doi.org/10.1109/aciiw.2017.8272595.

KAPTEIN, F. *et al.* Evaluating cognitive and affective intelligent agent explanations in a long-term health-support application for children with type 1 diabetes. *In*: **2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)**. IEEE, 2019. Disponível em: https://doi.org/10.1109/acii.2019.8925526.

LIPTON, P. Contrastive explanation. **Royal Institute of Philosophy Supplement**, Cambridge University Press, v. 27, p. 247–266, 1990.

LIU, W. *et al.* Data-driven sequential goal selection model for multi-agent simulation. *In*: **Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology - VRST '14**. ACM Press, 2014. Disponível em: https://doi.org/10.1145/2671015.2671024.

MADUMAL, P. *et al.* Explainable reinforcement learning through a causal lens. **Proceedings of the AAAI Conference on Artificial Intelligence**, Association for the Advancement of Artificial Intelligence (AAAI), v. 34, n. 03, p. 2493–2500, abr. 2020. Disponível em: https://doi.org/10.1609/aaai.v34i03.5631.

MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. **Artificial Intelligence**, v. 267, p. 1–38, 2019. ISSN 0004-3702. Disponível em: https://www.sciencedirect.com/science/article/pii/S0004370218305988.

MILLER, T. Contrastive explanation: a structural-model approach. **The Knowledge Engineering Review**, Cambridge University Press (CUP), v. 36, 2021. Disponível em: https://doi.org/10.1017/s0269888921000102.

MILLER, T.; HOWE, P.; SONENBERG, L. Explainable AI: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. **arXiv**, 2017. ISSN 23318422.

MOLINEAUX, M.; DANNENHAUER, D.; AHA, D. W. Towards Explainable NPCs: A Relational Exploration Learning Agent. **AAAI-18 Knowledge Extraction from Games Workshop**, p. 565–569, 2018. Disponível em: http://www.dustindannenhauer.com/papers/keg18.pdf.

MORVELI-ESPINOZA, M. *et al.* Argumentation-based intention formation process. **DYNA**, Universidad Nacional de Colombia, v. 86, n. 208, p. 82–91, jan. 2019. Disponível em: https://doi.org/10.15446/dyna.v86n208.66597.

MORVELI-ESPINOZA, M.; POSSEBOM, A. T.; TACLA, C. A. Argumentation-based agents that explain their decisions. *In*: **2019 8th Brazilian Conference on Intelligent Systems (BRACIS)**. IEEE, 2019. Disponível em: https://doi.org/10.1109/bracis.2019.00088.

MORVELI-ESPINOZA, M.; TACLA, C. A.; JASINSKI, H. M. R. An argumentation-based approach for explaining goals selection in intelligent agents. *In*: **Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part II**. Berlin, Heidelberg: Springer-Verlag, 2020. p. 47–62. ISBN 978-3-030-61379-2. Disponível em: https://doi.org/10.1007/978-3-030-61380-8_4.

RAO, A.; GEORGEFF, M. BDI Agents: From Theory to Practice. **Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)**, 1995.

RAYMOND, A.; GUNES, H.; PROROK, A. Culture-based explainable human-agent deconfliction. *In*: **Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems**. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2020. (AAMAS '20), p. 1107–1115. ISBN 9781450375184.

ROBERTS, M. *et al.* Coordinating robot teams for disaster relief. **Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015**, p. 366–371, 2015.

SKLAR, E. I.; AZHAR, M. Q. Explanation through argumentation. *In*: **Proceedings of the 6th International Conference on Human-Agent Interaction**. ACM, 2018. Disponível em: https://doi.org/10.1145/3284432.3284470.

STANGE, S.; KOPP, S. Effects of a social robot's self-explanations on how humans understand and evaluate its behavior. *In*: **2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)**. [*S.l.*: *s.n.*], 2020. p. 619–627.

STEPIN, I. *et al.* A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 11974–12001, 2021. Disponível em: https://doi.org/10.1109/access.2021.3051315.

WINIKOFF, M. Debugging agent programs with why? questions. *In*: **Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems**. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017. (AAMAS '17), p. 251–259.

WOOLDRIDGE, M. **An Introduction to Multiagent Systems**. 2. ed. Chichester, UK: Wiley, 2009. ISBN 978-0-470-51946-2.

APPENDIX A – Extended Minimum *Possible Explanans* Algorithm

Source Code 2 – Object-Oriented Minimal Possible *Explanans* Algorithm.

```
procedure MPE($possible\_explanans, event\_list, index, covered, base\_cover$)
                         ▷ If all events were checked, add base_cover to possible_explanans and return.
    if event_list.size() <= index then
        possible_explanans.add(base_cover)
        return possible_explanans                                                        ▷ $1
    end if

                                                        ▷ If base_cover is in possible_explanans, returns.
    if possible_explanans.contains(base_cover) then
        return possible_explanans                                                        ▷ $2
    end if

                                                     ▷ If the current_event is covered, skip to next event.
    current_event = event_list.get(index)
    if covered.contains(current_event) then
        return MPE(possible_explanans, event_list, index + 1, covered, base_cover)       ▷ $3
    end if

                                               ▷ Branch to each condition that can cover the current_event.
    for all condition from current_event.conditions do
                                    ▷ Creates a copy of covered and base_cover for the different branches.
        copy_base_cover = base_cover.clone()
        copy_base_cover.add(condition)
        copy_covered = covered.clone()
        copy_covered.addAll(condition.events)

                      ▷ Remove all conditions that are a proper subset of the current condition from base_cover.
        supersetSwap = false
        for all subset from condition.supersetOf do
            if base_cover.contains(subset) then
                copy_base_cover.remove(subset)
                supersetSwap = true
            end if
        end for

        if supersetSwap then
            copy_base_cover.add(condition)
            copy_covered.addAll(condition.events)
        end if
        MPE(possible_explanans, event_list, index + 1, copy_covered, copy_base_cover)    ▷ $4
    end for

    return possible_explanans
end procedure
```

Source: Author's own.