A study of the rough set approach for image understanding

by Peter Golibrzuch

Supervisor: Prof. Dr. Ralf Möller
Advisor: Atila Kaya, M. Sc.

Submitted in partial fulfillment of the requirements for the degree
Master of Science in Information and Media Technologies

Hamburg, January 2007

Declaration

I declare that:
this work has been prepared by me,
all literal or content based quotations are clearly pointed out,
and no other sources or aids than the declared ones have been used.

Hamburg, January 2007
Peter Golibrzuch

Acknowledgments

I would like to thank Prof. Dr. Ralf Möller for giving me the opportunity to perform this work at his department and for providing me with an interesting and challenging topic of research. I also thank Atila Kaya for his patience, inspiration and guidance during this work and for the great discussion and review sessions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advances of information technology, electrical engineering and computer science over the last decades has enabled what is often referred to as the *digital revolution*. The common understanding of this term is the digital representation of media artifacts like text, images, audio and video which had been stored on analog storage like paper, vinyl or magnetic tapes in the past. The word revolution in the term digital revolution regards the new abilities that have arisen with digital representation of media artifacts. Nowadays, media can be created, edited, transformed and, with the advent of the world wide web, published by almost everyone which had been restricted to experts in that field in the past. The revolution can also be interpreted as the enormous increase of data that comes with the new abilities of working with digital media. The increase of data led to first research on management of data, extraction of knowledge from and structured access to it. Out of this long lasting research, successful applications like search engines for the web with highly complex retrieval algorithms or pattern recognition for images have evolved, all of them on different levels and with different goals.

Most of the ideas used in research on management of digital media have their origin in the field of *artificial intelligence* which has been started out since the late 1940s [34]. The main goal of artificial intelligence is, based on the knowledge about how humans perceive, think and act, build intelligent agents that do so as well. This makes clear that artificial intelligence has strong links to other fields of science that also concern the study of mankind such as biology, psychology or philosophy.

1.1 Intelligent agents

The term artificial intelligence covers a wide field of approaches and ideas, so that it is necessary to define how the term is understood in this work. This work makes use of the idea of an intelligent agent that solves specific tasks in a human-like way within a certain domain. An intelligent agent therefore needs to be able to *perceive* its environment by means of a set of sensors. In this scenario, various devices can be seen as sensors such as cameras, microphones, temperature or humidity measuring units or also a keyboard. However, sensors can also be seen as processing units for chunks of data like text or images that enable the agent to work with that data. Data coming from the sensors can be used by an agent in two ways: it can be used for *learning* or acquiring knowledge from its environment or for acting within its environment based on its existing knowledge and data coming from sensors.

In the learning case, data from sensors is used for extracting knowledge usually in the form of patterns. The agent's knowledge also needs to be stored in a form that facilitates the agent to exploit the knowledge later. This is called *knowledge representation*. Knowledge representation can take many forms of which prominent examples are rule sets or ontologies. When learning, an agent extends its existing knowledge by extracting new patterns and storing them in its representation form of choice. The existing knowledge can be based on previous learning processes or can be initially given by a (human) expert.

In the acting case, data from sensors along with existing knowledge is used by the agent to make decision or act. This process is also called *reasoning* which is the last part of the definition of an intelligent agent in this work. Reasoning can therefore be seen as a service which is offered by the agent and has sensor data of a certain type as input information and yields a result within a certain range. What is happening in between can be regarded as a black box, in which the agents applies its "magic", namely its knowledge and reasoning methodology, on the information.

Summing up, an intelligent agent needs to have the ability to perceive its environment, be able to acquire and store knowledge and needs to employ some sort of reasoning service for acting or making decisions.

1.2 Image Understanding

As stated above, intelligent agents are generally designed for specific tasks within certain domains. In BOEMIE [7], a European IST research program, such a task is defined as *image understanding*. In image understanding an intelligent agent has the ability to perceive its environment by processing images or image data, acquire additional knowledge from annotated images, store knowledge in ontologies and recognize concepts from image data and stored knowledge. In a simplified way, such an agent is given a an image (a set of pixels) showing a couple of persons running next to each other on a red underground with white lines and the agent would deliver a concept named *Men's 100m* as a result or classification. The process makes use of low level features directly detectable from image data as well as high level reasoning necessary for delivering abstract concepts. Details on the far more complex process will be given in chapter 2. Yet, the simplified example allows for understanding idea behind such an agent.

The process of image understanding is seen as a further step in the research on management of digital media offering new solutions to challenges coming from the digital revolution. The main target of agents in image understanding is the classification of images by means of concepts describing the content of the image in a semantic way. After a learning phase, where experts train such agents, the classification would run self-controlled, allowing for classifying large amounts of images. Furthermore, extending the process of image understanding to different types of media, multi media artifacts such as pages on the world wide web consisting of text and images and eventually also audio and video data could be classified using the reasoning abilities of type specific agents by interchanging knowledge between them. As an example, an image could be classified far more detailed based on additional knowledge extracted from the surrounding text.

The process of image understanding will be the field of application for the ideas presented in this work. However, with a higher level of abstraction those ideas can also be used within the general context of multimedia interpretation. The next chapter will give a more detailed description of the constituents of the image understanding process and will introduce notions for further chapters. The following chapters will introduce the rough set approach in a general way and as a specific part of the image understanding process and present possible applications within the process. At the end of this work approximate reasoning with rough mereology will be covered as a possible extension based on rough set theory and be compared to the most prominent examples of approximate reasoning based on fuzzy set theory and probabilities.

Chapter 2

The Process of Image Understanding

The previous chapter gave a short introduction into the concept of an intelligent agent and how image understanding can be seen as a task for an intelligent agent. This chapter will give a detailed description of the image understanding process and will identify areas for optimization.

2.1 Image Understanding

Recalling the simplified example used in the introduction, one can consider the intelligent agent accepting image data as input and delivering concepts describing the picture as the corresponding output. For splitting up the process happening in between the agent will be divided into two main modules, a *low level image analysis* and a *high level reasoning service*. The input for the low level image analysis are images or image data and the corresponding output is a set of *mid-level concept (MLC) instances* [1] identified in the picture. Mid-level concepts are for example *lane*, *crossbar* or *pole* which can be detected directly from image data and low level features such as color, shape and texture. The set of MLC instances is then used as input for the high level reasoning service which delivers a *high level concept (HLC) instance* [2] as output. High level concepts are for example *Men's 100m*, *Pole Vault Jump* or *Winner's Ceremony* which can be derived based on the output of the low level image analysis.

[1]The term *mid-level concept* is a BOEMIE specific term describing concepts directly detectable from image data.

[2]The term *high-level concept* is also a BOEMIE specific term describing abstract concepts which can be only be delivered by making use of high level reasoning.

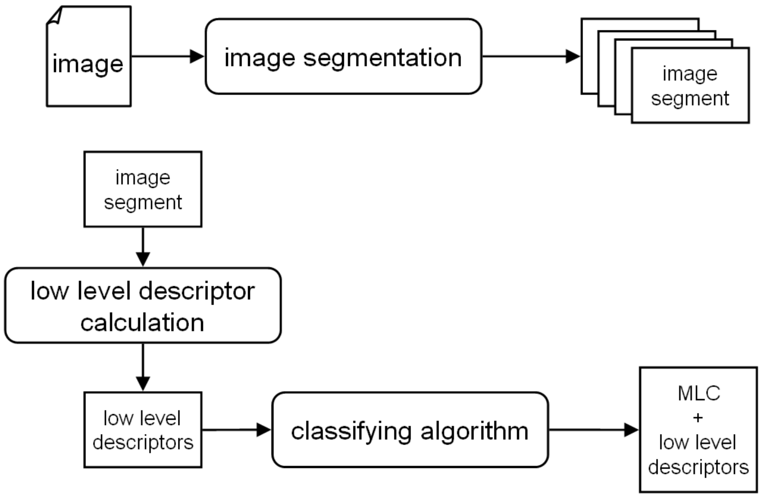


Figure 2.1: Region based approach for MLC detection

2.2 Low level image analysis

In the low level image analysis, two separate approaches are taken whose results are combined for a final set of MLC instances as output data. Additionally, the low level image analysis is able to deliver a set of spatial relations concerning the MLC instances such as $near(mlc_1, mlc_2)$ or $above(mlc_1, mlc_2)$ which is also part of the output data.

In the *region-based* approach the image is first segmented into different regions using an image segmentation technique. For each region a set of *low level descriptors* are being calculated. Such low level descriptors can directly be calculated from the image pixel values using dedicated algorithms. Examples of low level descriptors are MPEG-7 color, shape and texture descriptors as well as histograms or line orientation [8]. The low level descriptors for each image region are used as input for a classifying algorithm which assigns a MLC instance to the corresponding region. The classifying algorithm therefore needs to have knowledge available in order to decide which MLC corresponds to the set of low level descriptors given as input. This knowledge is usually learned using a supervised learning method in a previous step. The result of the region based approach is a MLC instance for each segmented region and the corresponding low-level descriptors. Additionally, for each MLC instance a *mask* is calculated that defines the set of pixels in the image that make up the MLC instance. The mask can be seen as the physical representation of the MLC instance in the image. Figure 2.2 illustrates the detection of MLC instances using the region-based approach.

In the *holistic approach* MLC instances are not derived from low level descriptors but from primitives which are simple geometric objects like lines,

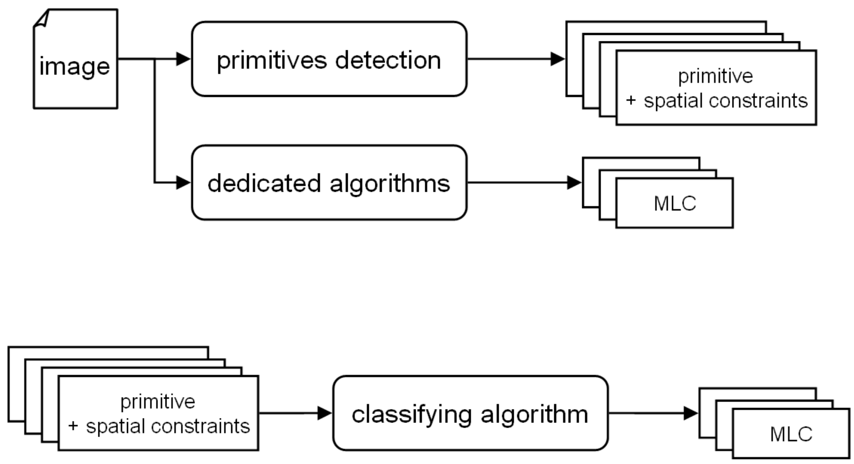


Figure 2.2: Holistic approach for MLC detection

arcs and ellipses and spatial relations between them. Additionally, dedicated algorithms are used for detecting specific objects such as parts of the human body which can be directly classified as MLC instances. In some cases low level descriptors can also be used for additional hints for MLC detection. Similar to the region based approach, the primitives are used as an input for a classifying algorithm that applies previously acquired knowledge for assigning MLC instances to a set of primitives. Using the example of a *Men's 100m* image , a *lane* as a MLC could be detected by the holistic approach through its definition as *lane = {2 parallel lines and uniform color in between}* and the according set of *line* primitives and color information from low level descriptors. The result of the holistic approach is a set of MLC instances including the masks of their corresponding pixels. Figure 2.2 illustrates the detection of MLC instances using the holistic approach.

2.2.1 Calculation of Confidence Measures

As described above, MLC detection is done separately by two different approaches. However, for the low level image analysis module to deliver a single result, the results of the region-based and holistic approach need to be unified. Additionally, the results delivered by each of the two approaches is also based on some degree of *uncertainty*. The classifying algorithms used in the region-based and holistic approach assign MLCs by comparing low level descriptors and primitives of the image with knowledge previously acquired. In the general case, low level descriptors and primitives from an image to be analyzed are not perfectly covered by existing knowledge. This results in the fact that classifying algorithms have to find a way to assign an MLC instance to a region or a set of primitives also in those cases, where there are for example two possible MLC instances for one region. Classifying algorithms use confidence measures in order to make decisions in "unclear" cases. Those confidence measures are generally calculated based on how well available information from the image fits to the existing knowledge. Depending on the used algorithm and the underlying mathematical foundations the semantics of those measures can have a probabilistic, usually frequency-based, notion or the notion of partial containment in case of fuzzy set theory as the underlying model.

In [8] a method is proposed to unify the results delivered by the region-based and holistic approaches including the calculation of a single confidence measure. It is assumed that the same MLC instance MLC_i has been detected in the image by each of the both approaches. Each approach also delivers its own confidence measure for the detected MLC instance which are denoted as $c_{H,i}$ and $c_{R,i}$ for the holistic and region-based approach, respectively. Additionally, each of the approaches also delivers the mask of the according MLC instance. The mask is defined as the set of pixels that the approach has assigned to be part of the MLC instance. Accordingly, the masks are denoted as $M_{H,i}$ and $M_{R,i}$. The calculation of a unified confidence measure can then be formally defined as

$$c_i = f(c_{H,i}, c_{R,i}, M_{H,i}, M_{R,i})$$

with c_i being the final confidence measure for MLC_i coming from both approaches.

In [8] an example function f is proposed which is defined as

$$c_i = \frac{max(c_{H,i}, c_{R,i}) \cdot |M_{H,i} \cap M_{R,i}|}{|M_{H,i} \cup M_{R,i}|}$$

where $|A|$ denotes the cardinality of the set A.

Example 2.2.1. Assume that the number of pixels covered by the intersection of the masks originating from the region-based and holistic approach is $|M_{H,i} \cap M_{R,i}| = 2045$ and accordingly those pixels covered by the union of the masks $|M_{H,i} \cup M_{R,i}| = 2304$. Furthermore assume that the confidence measure of the holistic approach has been calculated as $c_{H,i} = 0.8$ and accordingly $c_{R,i} = 0.7$. The final confidence measure is then calculated as

$$c_i = \frac{max(0.8, 0.7) \cdot 2045}{2304} = 0.71$$

It needs to be noted that the above defined function calculates the final confidence measure c_i based on different underling concepts. First, using the max-connective, the best confidence measure of either the region-based or holistic approach is chosen although both values are based on different detection concepts. This value is then multiplied with a coefficient expressing the similarity of two pixel regions based on intersection and union. The calculation of the final confidence measure c_i therefore brings up the question of the underlying semantics of this value.

The way the function is defined above, the semantics of c_i strongly depends on the concepts of confidence calculation of the holistic and region-based approach. Assuming an underlying concept of probabilities derived from data in both approaches, c_i could be interpreted as a probabilistic value based on frequency count rather than a degree of belief. It is expected that different functions will be defined for the calculation of c_i during the BOEMIE research project. An alternative could for example be the realization of f by means of a fuzzy rule base which would deliver results of fuzzy membership rather than probabilities. The rule base could be optimized for specific membership functions depending on different MLCs and the different MLC detection approaches.

The point of the semantics of c_i is stressed here as MLCs and their according confidence measure will be the input for the higher level reasoning module. In the next section, the high level reasoning module will be covered in more detail but in the first step without considering confidence measures. Confidence measures as part of the interface between low level image analysis and high level reasoning module will be covered in chapter 4 after settling the basic notions.

2.3 High level reasoning module

Recalling the example from the previous section where a high level concept like *Men's 100m* has been extracted from image data, the section above described the process in the low level image analysis module which delivers a set of MLC instances and spatial relations as a result. However, it is not possible for the low level image analysis to deliver HLC instances directly as HLCs are not directly detectable from image data. The set of MLC instances as an interface between low level image analysis and high level reasoning module can therefore be seen as a *layer of abstraction*.

This section will describe the process of yielding a HLC instance from a set of MLC instances using the high level reasoning module. In [8] the high level reasoning module is based on reasoning with description logics (DL) [2] which provides the required expressiveness to define high level concepts. Description logics build the logical foundation for a set of reasoning services offered by a reasoning system. The reasoning service proposed in the process of image understanding is called *abduction* [15]. Abduction is based on the idea of finding an explanation that concludes available background knowledge and observations to be explained. While in deduction consequences are derived from what is known, in abduction known facts are explained. In abduction the reasoning process starts with a set of given facts for which the most likely explanations are derived.

The process of abduction can be written as

$$\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$$

where Σ is the available background knowledge, Δ the explanation to be found and $\Gamma = \Gamma_1 \cup \Gamma_2$ the observation made and to be explained.

In the process of image understanding the constituents of the formula above are defined as follows:

- $\Gamma = \Gamma_1 \cup \Gamma_2$ is the input for the high level reasoning module, namely the set of MLC instances and their spatial relations. The set of MLC instances coming from low level image analysis is defined as Γ_1 which is referred to as *bona fide* assertions meaning that it is believed to be true. The spatial relations between MLC instances are defined as Γ_2 which is referred to as *fiat* assertions which need to be explained. As the fiat assertions are not believed to be true but need to be explained, they can be seen as the reasoning task. In the example of *Men's 100m*, Γ_1 would be the detected MLC instances such as *lanes* and *athletes* and Γ_2 the spatial configuration also detected by low level analysis, stating that *athletes* are *above lanes* and all *athletes* are *next to* each other. In order to reason with MLC instances and their relations, Γ is represented as an Abox that holds instances of MLCs and their relations.

- Σ is the background knowledge used by the high level reasoning module. In reasoning based on DL, knowledge is represented by means of ontologies which consist of a Tbox [3] and an Abox [4]. Details on syntax and semantics of description logics as well as representation of knowledge in form of Tbox and Abox can be found in [2]. In this work it is assumed that background knowledge is represented by a TBox which defines a set of high level concepts by means of subsumption, existential restriction and role names which is a simplification of the expressiveness of DL languages and systems. Furthermore it is assumed that the Abox belonging to background knowledge is empty. The definition of HLCs make use of MLCs (at a lower level in the concept hierarchy) so that the finite set of MLCs, which are possible to be detected by low level image analysis, form the basis for HLC definition. The same holds for spatial relations delivered by low level image analysis which can be used as role names in the concept definitions.

  Additionally, background knowledge is enriched by a set of *DL-safe rules*, further referred to as *reasoning rules* which express knowledge

[3]A Tbox is a terminological component describing concepts and roles.

[4]An Abox is the assertional component specifying individuals and their relationships.

Figure 2.3: Example image showing a pole vault event

that is not expressible with DL used in the process of image understanding. The safeness property of these rules lies in the preserving of decidability of problems to be solved by applying knowledge consisting of the TBox and the set of rules.

Background knowledge as defined above needs to be set up by an expert or knowledge engineer in a previous step. The design of how knowledge is represented by means of a Tbox and according DL-safe rules is crucial for the success of the reasoning service offered by the reasoning module. The design process therefore needs to take into account the properties of the domain which is to be represented as well as the data used for the reasoning process, namely the set of detectable MLCs and relations. Finally, it needs to be possible to define the reasoning task as a formal decision problem.

- Δ is the explanation to be found by the abduction process. Hence, Δ is empty at the beginning of the process. Δ therefore needs to be filled with instances that, together with background knowledge and Γ_1 as the observations believed to be true, logically entail Γ_2, namely the spatial configuration of the detected MLC instances.

In order to clarify the way how the high level reasoning module explains a set of MLC instances and their relations by means of an HLC, in the following an example will be given.

Example 2.3.1. Figure 2.3 shows an example image to which the image understanding process is applied. It is assumed that the low level image analysis delivers the following MLC instances and their corresponding relations:

$$\begin{array}{rcl} pole_1 & : & Pole \\ athlete_1 & : & Athlete \\ crossbar_1 & : & Crossbar \\ (athlete_1, bar_1) & : & above\_facing \\ (athlete_1, pole_1) & : & near \end{array}$$

This result of the image analysis is the observation $\Gamma = \Gamma_1 \cup \Gamma_2$ where $\Gamma_1 = \{pole_1, athlete_1, crossbar_1\}$ represents the presence of the MLC instances in the picture and believed to be true. The spatial configuration is given as $\Gamma_2 = \{(athlete_1, bar_1), (athlete_1, pole_1)\}$ and is to be explained.

One part of the background knowledge Σ defined by a knowledge engineer in a previous step consists of the following TBox:

$$\begin{array}{rcl} Man & \sqsubseteq & Person \\ Woman & \sqsubseteq & Person \\ Man & \sqsubseteq & \neg Woman \\ Athlete & \equiv & Person \sqcap \exists hasProfession.Sport \\ Foam\_Mat & \sqsubseteq & SportEquipment \\ Pole & \sqsubseteq & SportEquipment \\ Crossbar & \sqsubseteq & SportEquipment \\ Jumping\_Event & \sqsubseteq & Event \\ Pole\_Vault & \sqsubseteq & Jumping\_Event \sqcap \\ & & \exists hasPart.Pole \sqcap \\ & & \exists hasPart.Athlete \sqcap \\ & & \exists hasPart.Crossbar \sqcap \\ & & \exists hasPart.Foam\_Mat \end{array}$$

Additionally the following set of rules have been defined extending the knowledge represented in the above presented TBox:

$$\begin{array}{rcl} Pole\_Vault(X) & \rightarrow & Jumping\_Event(X), \\ & & hasPart.Pole(X, A), Pole(A) \\ & & hasPart.Athlete(X, B), Athlete(B) \\ & & hasPart.Crossbar(X, C), Crossbar(C) \\ & & hasPart.Foam\_Mat(X, D), Foam\_Mat(D) \\ & & above\_facing(B, C) \\ & & near(B, A) \end{array}$$

Considering the formula of abduction, Σ, Γ_1 and Γ_2 have been set up as defined above, while Δ is empty in the beginning as it is considered the explanation to be found. Therefore $\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$ does not hold for the above defined models. In order find a possible explanation it is necessary to apply the rule defined as part of Σ. This results in a binding of MLC instances delivered by the low level image analysis and those defined in the rule. MLCs that have not been found by low level image analysis but are defined in the rule are added. This means the existence of some missing instances is assumed by abduction. For example, variable D is instantiated with a MLC instance $foam\_mat_1$ which obviously has not been found by the low level image analysis as it is not part of the picture. However, in other cases this information can be used for finding more MLC instances in the picture, for example by triggering the holistic approach again focused on $foam\_mat$ objects. After applying the rule, Δ is filled with the instances delivered by the rule which now makes $\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$ true. The final result of the abduction is an ABox as follows:

$$\begin{array}{rcl} pole_1 & : & Pole \\ athlete_1 & : & Athlete \\ crossbar_1 & : & Crossbar \\ (athlete_1, bar_1) & : & above\_facing \\ (athlete_1, pole_1) & : & near \\ pv_1 & : & Pole\_Vault \\ (pv_1, athlete_1) & : & hasPart \\ (pv_1, crossbar_1) & : & hasPart \\ (pv_1, pole_1) & : & hasPart \end{array}$$

2.4 Discussion

The example above presented the main idea of using a high level reasoning module in the process of image understanding. In this simplified example, some complexity of the high level reasoning module is hidden. It is obvious that in a domain such as *track and field* there are numerous high level concepts. Those high level concepts do not necessarily only represent the various disciplines but can also be concepts like *winning ceremony*, *athlete interview* or *starting signal*. So the number of high level concepts represented in the knowledge base of the high level reasoning module can become enormous.

In order to find an explanation for a set of MLC instances with the process described above, it is necessary to apply each of the reasoning rules and start the abduction process for finding out whether there is a successful result. A possible optimization would be the introduction of a more target oriented approach that offers a pre-processing module that delivers possible explanations based on a specific set of MLC instances. The rule applying and abduction would then be only necessary for a smaller number of HLCs, which increases the performance as for some HLCs the process is not initiated. This pre-process for delivering possible explanations needs to make use of additional knowledge from which a set of possible HLC can be derived based on the MLC instances and relations delivered by low level image analysis. This additional knowledge can be seen as an extension of the background knowledge, however knowledge representation and how knowledge is acquired may differ. The following chapters will describe how far such a preprocess can be realized with rough set theory including its specific forms of knowledge representation and learning of knowledge.

Another challenge arising from the abduction process is that of finding the best explanation for a set of MLC instances, referred to as *HLC selection process*. With a large number of HLCs it might be possible that a specific set of MLC instances can be explained by two or more HLCs. The high level reasoning module however has to select one of the HLCs which is assumed to be the best (assumably correct) explanation. Finding the best explanation can be considered as a multiple step process which is based on several selection criteria. This includes for example the measurement based on how many MLC instances and according relations used in the definition of possible HLCs have been detected by low level image analysis. This can lead to a selection of an HLC which best matches those MLC instances and relations detected by low level image analysis.

Another criteria is based on the similarity of MLC instances detected in the image under consideration and those MLC instances of images that have been unambiguously classified being part of a specific HLC. Therefore it is assumed that images which have already been successfully explained are kept as reference or standard objects. The measurement of similarity of MLC instances is based on the similarity of low level descriptors or primitives in the low level image analysis. Those MLC instances that are part of a standard HLC are compared to those MLC instances detected in the image to be explained. If a detected set of MLC instances matches well with a set of MLC instances of a standard HLC, this information can be included within the HLC selection process for finding the best explanation.

What is however not yet exploited are confidence measures already mentioned when presenting low level image analysis. In the HLC selection process MLC instances which have been detected and assigned high confidences measures should be weighted higher compared to those MLC instances with lower confidence measures. This would also lead to MLC instances that might have been detected incorrectly not being of equal importance in the HLC selection process as it is assumed that incorrect MLC instances generally have low confidence scores. Considering the confidence measures in the HLC selection process can be seen as an additional criteria to those already mentioned above. How the rough set approach can add additional criteria for the HLC selection process, also including confidence measures, will also be covered in the following chapters.

Summing up, there are two main points for which a possible application of the rough set approach will be studied in the next chapters. The fist one is that of assisting the high level reasoning module in offering a preselection of HLCs for possible explanations based on a set of MLC instances and its relations. The second one will consider how far confidence measures as well as additional knowledge extractable by the rough set approach can be exploited for the HLC selection process.

Therefore, the next chapter will first give an introduction into the rough set approach and its basic notions. Afterwards, possible applications of the approach within the process of image understanding will be presented.

Chapter 3

Rough Sets

This chapter gives a brief introduction into rough set theory and its according notions used in solutions and approaches in further chapters. The basics of rough set theory as given in [27] will be briefly recalled and illustrated by examples. The purpose of this chapter is to present the concepts behind the application of rough set theory to extract knowledge from data and reason on the data.

3.1 Classical Set Theory and Rough Sets

In classical set theory an element is uniquely defined as either belonging to a set or not. Considering the set $X = \{x, y, z\}$, one can definitely say that x belongs to the set X ($x \in X$) whereas a does not ($a \notin X$). Classical set theory can therefore also be considered as precise and its sets can be defined as *crisp*.

For introducing the idea of a rough set compared to crisp sets from classical set theory, the most prominent example in rough set theory will be used. With medicine being one of the major fields for the application of rough set theory, the example makes use of a set of patients for which medical information has been recorded. This medical information includes a set of symptoms such as blood pressure, body temperature or blood values for each patient and whether the patient has had a certain disease a few days after recording the symptoms. The idea behind recording this information is the goal of simplifying the diagnosis of specific diseases. Based on the knowledge about the symptoms a patient shows, a certain disease could be diagnosed and medicated at early stages.

Such medical data usually shows the following characteristics: Some patients having the same symptoms all have had the same disease while others showing different symptoms did not become ill. However there is also a set of patients which showed exactly the same symptoms, but only some of them became ill while others didn't. If the concept of getting a

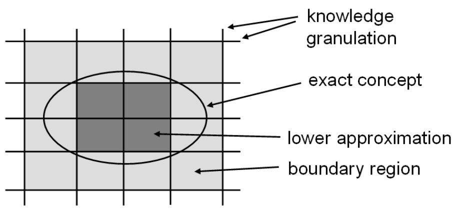


Figure 3.1: Vagueness in the sense of rough set theory

disease is seen with regard to the information about the symptoms, it is said to be a *vague* concept as for some patients it is not possible to define getting a disease based on the patients symptoms. Rough set theory handles this vagueness by means of boundaries. All patients for which the symptoms definitely have led to a disease are said to be part of the *lower approximation* while all patients for which the symptoms definitely did not lead to a disease are part of the *outside region*. Those patients showing the same symptoms but for which only some have become ill are said to be part of the *boundary region*. Figure 3.1 shows the ideas of boundaries in a graphical way. In rough set theory the concept of vagueness is assumed to be based on *lack of knowledge* which in the used example means that there might be additional symptoms for which all patients can be classified for getting a disease or not, based on the information about the symptoms. In the case of a vague concept, the *granulation of knowledge* induced by the symptoms is too large to define the concept of getting a disease in a crisp manner.

The idea of vague concepts is only the starting point of several techniques for handling data using rough set theory. The main focus in this work will be put on the extraction of knowledge from data with rough set theory which is also referred to as *approximate reasoning from data*. Therefore this chapter will set up the basic notions and mathematical foundations of rough set theory needed for a further application within the context of image understanding.

3.2 Information Systems

Data is intuitively often presented in tabular form. A table containing data usually consists of rows or data sets that describe certain objects or events. Each row is assigned a label or ID to identify the object's data. Data values of each row are organized into columns where each column describes a certain *attribute*. Values of data can come from observations or experiments based on sensor data or can also be given by a domain expert.

Formally an *information system* is defined as a pair $\mathcal{A} = (U, A)$, where U is a finite non-empty set of objects called *universe of discourse* and A is

U	Wind	Tide	Swell	Weather
x_1	offshore	high	1 m	sunny
x_2	onshore	low	1 m	cloudy
x_3	onshore	high	2 m	cloudy
x_4	offshore	high	0 m	sunny
x_5	onshore	low	1 m	cloudy
x_6	offshore	high	2 m	sunny
x_7	offshore	low	0 m	sunny

Table 3.1: An example information system

a non-finite set of *attributes* such that $a : U \to V_a$ for each $a \in A$. The set V_a is called the *value set* of a.

Example 3.2.1. Table 3.1 shows a simple example of an information system. The universe of discourse consists of seven objects and can be written as follows: $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$. Accordingly the set of attributes $A = \{Wind, Tide, Swell, Weather\}$ consists of four elements.

3.3 Indiscernibility

In information systems, elements of the universe of discourse can be distinguished only by their attribute values. Often several objects have the same attribute values and can therefore not be distinguished on the basis of the knowledge available in the according information system. This fact is covered by the notion of *indiscernibility.*

For the definition of indiscernibility the notion of an *equivalence relation* is needed. An equivalence relation is defined as a binary relation $R \subseteq X \times X$ which is reflexive (xRx, an object is related to itself), symmetric (if xRy implies yRx) and transitive (if xRy and yRz implies xRz).

In an information system $\mathcal{A} = (U, A)$ with $B \subseteq A$, an equivalence relation $IND_{\mathcal{A}}(B)$ is defined as:

$$IND_{\mathcal{A}}(B) = \{(x, x') \in U^2 \mid \forall a \in B \ a(x) = a(x')\}$$

An equivalence relation creates *equivalence classes* to which each object of the universe uniquely belongs to. The equivalence class of an object $x \in U$, denoted $[x]_B$, consists of all objects $y \in U$ such that xRy with $R = IND_{\mathcal{A}}(B)$.

In other words, for the attribute set A and its subsets there exists an equivalence relation that divides the universe of discourse into certain *partitions.* The notion of equivalence relation, equivalence classes and partitions defined above will be illustrated in the following example:

U	Wind	Tide	Swell	Weather	Condition
x_1	offshore	high	1 m	sunny	good
x_2	onshore	low	1 m	cloudy	good
x_3	onshore	high	2 m	cloudy	good
x_4	offshore	high	0 m	sunny	bad
x_5	onshore	low	1 m	cloudy	bad
x_6	offshore	high	2 m	sunny	good
x_7	offshore	low	0 m	sunny	bad

Table 3.2: An example decision system describing surfing conditions

Example 3.3.1. In the information system of Table 3.1 non-empty subsets of the attribute set A are for example $\{Wind\}$, $\{Swell\}$ and $\{Wind, Tide\}$. Each of those subsets create partitions of the universe by applying the indiscernibility relation defined above.

$$IND(\{Wind\}) = \{\{x_1, x_4, x_6, x_7\}, \{x_2, x_3, x_5\}\}$$

$$IND(\{Swell\}) = \{\{x_1, x_2, x_5\}, \{x_3, x_6\}, \{x_4, x_7\}\}$$

$$IND(\{Wind, Tide\}) = \{\{x_1, x_4, x_6\}, \{x_2, x_5\}, \{x_3\}, \{x_7\}\}$$

For $\{Wind\}$ there are two equivalence classes and for $\{Swell\}$ and $\{Wind, Tide\}$, three and four, respectively. It is said that by employing the knowledge of $\{Wind\}$, objects x_1, x_4, x_6 and x_7 are *indiscernible*, or they are *indiscernible* with respect to the attribute $Wind$. Accordingly objects x_3 and x_7 are *discernible* with respect to the attributes *Wind* and *Tide*.

3.4 Decision Systems

While information systems give a structured overview of data, one is usually interested to draw conclusions from that data. In order to draw conclusions from data, a classification of data is needed that allows for referencing objects to a certain classification value. Classification of data in information systems is enabled by adding an extra attribute that describes the conclusion that can be drawn from data. Such information systems are called *decision systems* and are formally defined as $\mathcal{A} = (U, A \cup \{d\})$ with $d \notin A$ called *decision attribute*. Accordingly attributes $a \in A$ are called *conditional attributes*. Furthermore the cardinality of the value set V_d of the decision attribute d is called the *rank of d*.

Example 3.4.1. Table 3.2 shows a decision system created by adding the decision attribute *Condition* to Table 3.1. The rank of the decision attribute *Condition* is two with $V_d = \{good, bad\}$. Looking at the data of Table 3.2 one can find out, that the each of the objects x_1, x_3, x_4, x_6 and x_7 can uniquely be assigned with a value of the decision attribute by only considering the conditional attributes of A. Objects x_2 and x_5 however are indiscernible with respect to A and have both different values for the decision attribute *Condition*.

3.5 Set Approximation

Referring to Example 3.4.1 the concept *Condition* = *good* (*Condition* = *bad*, respectively) can not be exactly defined, or in other words, it can not be defined in a *crisp* manner by only employing the knowledge of the attribute set A. Similar to the example from the introduction, some objects of the universe can be identified as certainly belonging to the concept *Condition* = *good*, while others as certainly belonging to the concept *Condition* = *bad*. In Example 3.4.1 objects x_1, x_3 and x_6 certainly belong to the concept *Condition* = *good* and therefore make up the *lower approximation* of the concept *Condition* = *good*. Objects x_4 and x_7 certainly do not belong to the concept *Condition* = *good* and they make up the outside region of the concept *Condition* = *good*. Objects x_2 and x_5 however can not be uniquely classified based on their attribute values and therefore belong to the *boundary region*.

Formally these notions are defined as follows: Let $\mathcal{A} = (U, A)$, be an information system and let $B \subseteq A$ and $X \subseteq U$. The *B-lower* approximation of X then is constructed by

$$\underline{B}X = \{x | [x]_B \subseteq X\}.$$

Accordingly the *B-upper* approximation of X is constructed by

$$\overline{B}X = \{x | [x]_B \cap X \neq \emptyset\}.$$

The set

$$BN_B(X) = \overline{B}X - \underline{B}X$$

is called the *B-boundary region* of X. A concept that can be defined with an empty boundary region is called *crisp*, concepts with non-empty boundary regions are called *rough* sets respectively.

Example 3.5.1. Taking the decision system from Table 3.2 and the attribute set A the concept *Condition* = *good* can be approximated as follows:

$$\begin{array}{lcl} \underline{A}X & = & \{\{x_1, x_3, x_6\}\} \\ \overline{A}X & = & \{\{x_1, x_3, x_6\}, \{x_2, x_5\}\} \\ BN_A(X) & = & \{\{x_2, x_5\}\} \end{array}$$

With $B_1 = \{\mathit{Swell}\}$ the concept *Condition* = *good* is approximated by

$$\begin{array}{lcl}\underline{B_1}X & = & \{\{x_3, x_6\}\} \\ \overline{B_1}X & = & \{\{x_3, x_6\}, \{x_1, x_2, x_5\}\} \\ BN_{B_1}(X) & = & \{\{x_1, x_2, x_5\}\}\end{array}$$

With $B_2 = \{Wind\}$ the concept $Condition = good$ is approximated by

$$\begin{array}{lcl}\underline{B_2}X & = & \emptyset \\ \overline{B_2}X & = & \{\{x_1, x_4, x_6, x_7\}, \{x_2, x_3, x_5\}\} = \text{U} \\ BN_{B_2}(X) & = & \{\{x_1, x_4, x_6, x_7\}, \{x_2, x_3, x_5\}\} = \text{U}\end{array}$$

In the case of the decision system of example 3.4.1 the concept $Condition = good$ can not be defined exactly even using the complete attribute set A. However there might be an attribute $a_i \notin A$ that assigns object x_2 and x_5 each into a different equivalence class (and therefore further partitioning the universe) which would allow for crisp definition of the concept. If such an a_i exists and is not available in the according decision system, a concept is said to be not exactly definable due to *lack of knowledge*.

The approximation of the concept $Condition = good$ using $B_2 = \{Wind\}$ has the special property that the lower boundary is the empty set and the upper boundary is made up of all element of the universe. Such an approximation is called *totally B-undefinable*. In fact, there are four classes of rough sets:

- X is *roughly B-definable*, iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) \neq U$
- X is *internally B-undefinable*, iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) \neq U$
- X is *externally B-undefinable*, iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) = U$
- X is *totally B-undefinable*, iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) = U$

In order to define how well a concept can be approximated, a measure called *accuracy of approximation* is introduced where $|X|$ denotes the cardinality of the set X:

$$\alpha_B = \frac{|\underline{B}(X)|}{|\overline{B}(X)|}$$

Obviously an approximation with $\alpha_B = 1$ is called crisp and $\alpha_B < 1$ rough, respectively, as defined above.

Example 3.5.2. Taking the attribute sets from the example above, one obtains the following results of approximations of the concept $X : Condition = good$: Using the attribute set A, the concept X is roughly A-definable with an accuracy of $\alpha_A = 3/5$. Accordingly the concept X is roughly B_1-definable with an accuracy of $\alpha_{B_1} = 2/5$. Using the attribute set B_2, the concept X is totally B_2-undefinable with $\alpha_{B_2} = 0$.

3.6 Rough Membership Function

As stated in the introduction, in classical set theory an element either belongs to a set or not. If one considers the property of belonging to a set as a function (membership function), in classical set theory this function would have only one of the values 0 or 1.

In rough set theory the membership function has different properties. In the decision system of Table 3.2, each object of the universe either belongs to the concept *Condition* = *good* or not. Now let us consider an object x_i which has the same attribute values as objects x_2 and x_5 but for which the classification, namely the *Condition* is not known. With the knowledge available about object x_i and objects of the known universe described in the decision system, the *Condition* for object x_i can not be derived exactly. The problem in this case is that object x_i is an element of an equivalence class that is part of the boundary region. In order to derive measures to which degree a single object belongs to a concept, the rough membership function is introduced:

$$\mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

Example 3.6.1. In the following table, several objects and their according rough membership values with regard to concept X : *Condition* = *good* are given:

U	Wind	Tide	Swell	Weather	$\mu_X^A(x)$
x_i	onshore	low	1 m	cloudy	1/2
x_{i+1}	offshore	high	1 m	sunny	1
x_{i+2}	offshore	low	0 m	sunny	0

The values of the rough membership function can also be interpreted as a probability measure. In the example above x_i has a probability of 0.5 for belonging to concept X. Since the probability is based on some background knowledge, this can in fact be seen as a conditional probability $P(x \in X|u)$ where u is the background knowledge being the attribute values of object x as well as knowledge gained about the universe and its partition. It should be noted that the rough membership values and according probability measures are directly derived from data. This leads to the typical property of frequency count approaches where a different distribution of the universe of discourse delivers differing results and measures.

For the interested reader Appendix C gives a comparison of the rough membership and fuzzy membership functions.

3.7 Rough Sets and Knowledge Discovery from Databases

Since the introduction of rough set theory by Z. Pawlak in the early 1980s, further research has led to various applications and extensions of rough set theory in several fields like medicine, economic sciences, artificial intelligence and computer science. However, all of these applications are based on the general use of rough set theory already introduced in the very first publications referred to as *knowledge discovery from databases*. The following section will present the standard rough set theory technique for extracting knowledge from (potentially huge amounts of) tabular data and its representation in the form of rules [1].

In the examples used throughout this chapter, one might be interested in extracting short and general rules like *if (Swell = 2 m) then (Condition = good)*. Obviously this holds true for the universe of discourse in the according decision system. The generation of such rules may aim at finding a minimal decision algorithm expressing knowledge of the related decision system or at finding rules that classify yet unseen objects into decision classes based on the objects' attribute values and the knowledge given by a decision system.

3.7.1 Decision Rules

In a decision system, each row corresponds to a *decision rule* based on attribute value pairs for condition and decision attributes. Instead of deriving a rule like

if (Swell = 2 m) then (Condition = good)

in the first step it is more obvious to derive rules for each object, e.g. for object x_3 in Table 3.2:

if (Wind = onshore) and (Tide = high) and (Swell = 2 m)
and (Weather = cloudy) then (Condition = good)

However, when trying to derive rules for object x_2 and x_5, one finds out that although having the same attribute value pairs, each object leads to a different decision. Therefore rules derived from objects x_2 and x_5 are called *conflicting* or *inconsistent*. Accordingly rules derived from other objects are called called *non-conflicting* or *consistent*. This notion is extended to a decision system, being either *consistent* or *inconsistent* considering the set of rules derived from it.

It can therefore be stated that a decision system with an attribute set A is *inconsistent*, if a concept approximation results in a non-empty boundary

[1]Note that the term *rules* used in the context of rough set theory denotes a different concept from that of *reasoning rules* used in the high level reasoning module.

region, meaning $(\exists X \,|\, BN_A(X) = \overline{A}X - \underline{A}X \neq \emptyset$ or $\alpha_A(X) < 1)$. Otherwise, if all concept approximations result in empty boundary regions, meaning $(\forall X \mid BN_A(X) = \overline{A}X - \underline{A}X = \emptyset$ or $\alpha_A = 1)$, the underling decision system is said to be *consistent*.

Generally, especially in the case of huge sets of data, it is assumed that the according decision system is inconsistent. However, in case of several decision concepts, for some of the decision concepts, the same decision system might be consistent.

3.7.2 Reducts

In order to derive rules that are shorter than those having all attribute value pairs of conditional attributes, redundant and unimportant information needs to be dropped. This idea is referred to as *reduct calculation*. Having a closer look at Table 3.2, one can find out that attributes $Wind$ and $Weather$ obviously have a *functional dependency*, being $(Wind = offshore) \Leftrightarrow (Weather = sunny)$ and $(Wind = onshore) \Leftrightarrow (Weather = cloudy)$. This can also be interpreted in such a way that attributes $Wind$ and $Weather$ represent the same knowledge and that in fact only one of both is needed for preserving the partition of the universe.

Example 3.7.1. Using the attribute sets $A = \{Wind, Tide, Swell, Weather\}$, $B_1 = \{Tide, Swell, Weather\}$ and $B_2 = \{Wind, Tide, Swell\}$ results in the following equal partitions on Table 3.2

$$IND(A) = IND(B_1) = IND(B_2) = \{\{x_1\}, \{x_2, x_5\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\}$$

where $IND(A)$ denotes the equivalence relation induced by the attribute set A.

In the notion of rough sets, B_1 and B_2 are *reducts* of A. For deriving reducts from decision systems, the following notions are introduced, with $B \subseteq A$ and $a \in B$.

- An attribute a is *dispensable* in B if $IND(B) = IND(B - \{a\})$, otherwise a is *indispensable* in B.
- The set B is *independent* if all its attributes are indispensable.
- The set B is a reduct of A if B is independent and $IND(B) = IND(A)$.
- The core of B is the set of all indispensable attribute of B: $Core(B) = \bigcap Red(B)$ with $Red(B)$ the set of all reducts of B.

In the example above the core of the attribute set A is $Core(A) = \{Tide, Swell\}$. In other words, the attributes $Tide$ and $Swell$ have the most significance for preserving the partition of the universe. Dropping one these attributes would result in a much broader partition.

Decision Relative Reducts

In a decision system like Table 3.2, it is not only important to drop redundant information only concerning the conditional attributes but also information that is redundant concerning the decision concepts. For example, the attribute set $B_3 = \{Tide, Swell\}$ is sufficient for describing the decision concept $Condition = good$ in the same way as the full attribute set A or reducts of A.

Therefore the notion of *dependencies of attributes* similar to functional dependencies in databases is introduced. Let D and C be subsets of A. D *depends in degree* k *on* C, denoted $C \Rightarrow_k D$ with

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|}$$

where

$$POS_C(D) = \bigcup_{X \in U/IND(D)} \underline{C}X$$

For $k = 1$, D *depends totally on* C, for $k < 1$, D *depends partially on* C. $POS_C(D)$ is called the *positive region* and represents all elements in the lower boundary of concept D by employing knowledge of C.

Example 3.7.2. For the attribute sets $A = \{Wind, Tide, Swell, Weather\}$, $B_1 = \{Tide, Swell, Weather\}$ and $B_3 = \{Tide, Swell\}$, the decision attribute set $D = \{Condition\}$ and concept $X : Condition = good$, the following results are obtained:

$$POS_A(D) = \{\{x_1\}, \{x_3\}, \{x_6\}\}$$

$$POS_{B_1}(D) = \{\{x_1\}, \{x_3\}, \{x_6\}\}$$

$$POS_{B_3}(D) = \{\{x_1\}, \{x_3, x_6\}\}$$

$$k_A = \gamma(A, D) = k_{B_1} = \gamma(B_1, D) = k_{B_3} = \gamma(B_3, D) = 3/7$$

Although the attribute sets A, B_1 and B_3 generate different partitions of the universe, they result in the same number of elements in the lower boundary for concept X and therefore D depends on A (B_1 and B_3 respectively) in degree 3/7. Therefore the attribute set $B_3 = \{Tide, Swell\}$ is sufficient for preserving the partition of the universe needed for the concept $X : Condition = good$.

According to the definitions of reducts above, notions for the definitions of decision relative reducts are introduced, with $C, D \subseteq A$ and $a \in C$

- An attribute a is *D-dispensable* in C if $POS_C(D)) = POS_{C-\{a\}}(D)$, otherwise a is *D-indispensable* in C.

- The set C is *D-independent* if all it attributes are D-indispensable.
- The set C is a *D-reduct* of A if C is D-independent and $POS_C(D)) = POS_A(D)$.
- The *D-core* of C is the set of all D-indispensable attribute of C: $Core_D(C) = \bigcap Red(C)_D$ with $Red_D(B)$ the set of all D-reducts of C.

Details on the calculation of reducts and decision relative reducts by means of discernibility matrices are given in [38].

Value Reducts

With reducts and decision relative reducts at hand, rules can be generated that are shorter than rules based on attribute-value pairs of the complete attribute set. However, an even shorter rule like *if (Swell = 2 m) then (Condition = good)*, which is obviously true in the decision system in consideration, can not be derived by using reducts or decision relative reducts.

The idea of *value reducts* is based on the general idea of reducts, namely preserving the partition (in case of reducts) or the positive region (in case of decision relative reducts). However, instead of working with complete attributes, only single attribute values are dropped to further simplify generated rules. This again goes back to discernibility, meaning to find out which attribute values are necessary in order to discern objects.

Example 3.7.3. The partitions of the universe of discourse of Table 3.2 generated by the attribute set $\{Swell\}$ and $\{Condition\}$ are presented below.

$$IND(\{Swell\}) = \{\{x_1, x_2, x_5\}, \{x_3, x_6\}, \{x_4, x_7\}\}$$

$$IND(\{Condition\}) = \{\{x_1, x_2, x_3, x_6\}, \{x_4, x_5, x_7\}\}$$

It can be seen from the partitions and according attribute values from the table, that the equivalence class of objects with *Swell = 2 m* is a subset of the equivalence class of concept X : *Condition = good*, $\{x_3, x_6\} \subset \{x_1, x_2, x_3, x_6\}$. Accordingly the equivalence class of objects with *Swell = 0 m* are a subset of the equivalence class of concept Y : *Condition = bad*, $\{x_4, x_7\} \subset \{x_4, x_5, x_7\}$. Hence, attribute values *2 m* and *0 m* are already sufficient for discerning the according objects from other objects relative to the partition employed by the attribute set $\{Condition\}$.

Formal definitions and further details on value reducts can be found in [27]. For the calculation of value reducts and decision relative value reducts by means of discernibility matrices, refer to [38].

3.7.3 Rule Selection

As presented so far, there are several ways to reduce the redundant information while preserving the discernibility relative to the decision in consideration. Instead of generating rules made up of attribute value pairs of the complete attribute set A, several shorter rules can be calculated. However, for various reducts, also different rules are generated that are based on the same information about a certain object. For example, from the decision system of Table 3.2 19 rules can be derived as given in Table B.1, although Table 3.2 contains 7 objects only.

Minimal decision algorithm

One of the application areas of rules which are derived by applying rough set theory to data is the implementation of control algorithms[2] or control protocols. The basic idea of this approach is that at certain points of time, observations are being recorded that include a number of attribute values and a decision. These observations are represented as a decision system to which rough set theory techniques can be applied. An example of this approach can be found in [27, 21] where the decisions of a stoker (a human operator) of a rotary clinker kiln have been used for the implementation of a control algorithm for automatic operation.

In such cases it is assumed that all possible situations have been recorded and have been represented in the decision system. Hence, the decision system completely represents the world of the domain in consideration, also called *closed world assumption*. In the example decision system of Table 3.2 this would mean that all possible situations are being represented. E.g. the case *Swell = 3 m* does not exist. Though this does not seem to make sense in this example, there are other domains where such an assumption can be applied [27, 21].

In the following a minimal decision algorithm based on the decision system of Table 3.2 is presented. Note that there is more than one solution for such a minimal algorithm. Besides the attribute set $\{Tide, Swell\}$, further algorithms based on the attribute sets $\{Wind, Swell\}$ and $\{Weather, Swell\}$ can be derived.

if (Tide = low) and (Swell = 1m) then (Condition = good or bad)
if (Swell = 2m) then (Condition = good)
if (Swell = 0m) then (Condition = bad)
if (Tide = high) and (Swell = 1m) then (Condition = good)

Note that the first rule is inconsistent, as no decision for concept *Condition = good* or *Condition = bad* can me made. This stems from the fact that

[2] Note that in rough set theory, algorithms (control or decision algorithms) are understood as a fine set of instructions in form of decision rules.

the underlying decision system is inconsistent and the concepts *Condition = good* or *Condition = bad* can only be approximated by a rough set.

Classification of unseen objects

While minimal algorithms aim at representing the knowledge of the underlying decision system in a minimal way, they are not well suited for classifying yet unseen objects. Assume for example the following object x_i which is to be classified:

$$x_i : (Wind = onshore), (Tide = high), (Swell = 3m), (Weather = sunny)$$

In the decision algorithm presented above, no rule would *fire* as none of the conditions are fulfilled with object x_i . However, in the set of decision rules of Table B.1, rule 16 would fire and therefore object x_i would be classified as *Condition = good*.

The set of decision rules of Table B.1 is based on the set of all reducts. The complexity of calculating the set of all reducts is exponential [38]. Therefore several heuristic algorithms including genetic and covering algorithms [3] and LEM2 algorithm [12] have been proposed which calculate good results for reducts and according rule sets in an appropriate time.

For most practical applications for which rough set theory is used it is more important to find reducts that are able to classify unseen objects rather than finding minimal sets or the complete set of decision rules. One approach which aims to deliver appropriate reducts for practical applications, uses *approximate reducts* [40] where reducts are calculated based on a certain threshold e.g. preserving the indiscernibility relation up to a certain level. Those reducts are concise and do not preserve the positive region of the underlying decision system, however the rules derived from those reducts are much smaller and better suited for classifying unseen objects. The error introduced by the approximation of reducts needs to be fine-tuned for the properties of data from which rules are generated.

A variant of approximate reducts are *dynamic reducts* [6] which also have been proven successful for classifying unseen objects. With dynamic reducts, the original decision table is first randomly split into several smaller tables. For those subsets of the universe, reducts are being calculated. The most stable reducts, e.g., those calculated from several subtables, are then used for deriving according decision rules. Dynamic reducts are generally not consistent with the original decision system, however they are better suited for classifying unseen cases.

3.8 Summary

This chapter introduced the basic ideas of rough set theory. The starting point was a vague concept and the inability of classical set theory to handle such vague concepts. Next, information and decision systems have been introduced as typical representation forms of data and its classification. In order to describe decision concepts based on attribute value pairs of objects, the indiscernibility relation is needed for partitioning the universe of discourse into equivalence classes. Based on these partitions it is able to approximate decision concepts by means of an upper and lower boundary. Furthermore, the partition allows for the definition of a rough membership function by which the membership of each object to a certain decision concept can be calculated.

With knowledge discovery from databases being the major application of rough set theory, the approach for deriving decision rules from decision systems has been described. The approach includes the calculation of decision relative reducts and value reducts from which rules can directly be derived. Finally, the different cases of minimal rule sets and rule sets optimized for classification of unseen objects have been discussed. The basic notions introduced in this chapter will build the theoretical foundation for approaches presented in the following chapters.

Chapter 4

Rough Sets in Image Understanding

In the previous chapter the foundations and basic notions of the rough set approach have been presented as well as knowledge discovery from databases as the main field of application. In this chapter, possible applications of rough set theory within the image understanding process will be studied. Therefore, the two major points, which will be focused on, are recalled: In order to provide for a more goal oriented search of HLCs in the image understanding process, a preselection of HLCs based on the set of MLC instances and relations extracted from low level image analysis will be added to the high level reasoning module. Additionally, the use of information coming from the rough set approach will be considered for assistance of finding the best explanation. Within this context the possibility of using confidence measures assigned to MLC instances and according relations will be studied as well. Both cases can be seen as an extension of the background knowledge of the high level reasoning module.

4.1 Learning from Images

Typically applications exploit rough set theory for extracting knowledge in form of rules from generally huge data sets. The rule sets extracted from that data express the extracted knowledge of the complete data set. In the process of image understanding presented so far, knowledge is represented in various forms. In low level image analysis, classifiers are defined which are able to detect MLC instances based on low level descriptors or geometric primitives while in the high level reasoning module knowledge is expressed as ontologies and reasoning rules.

In case of the high level reasoning module, ontologies and rules are expected to be defined in the design phase by a domain expert or knowledge engineer. For the low level image analysis some classifiers or dedicated algorithms may also be defined in a design phase while some classifiers are expected to be extracted from data as well.

As presented in the previous chapter, rough set theory is purely data grounded, so that its application requires data sets from which the knowledge is to be extracted. The extraction of knowledge is also called *learning* and in case of rough set theory where data sets are the knowledge foundation, it is called *learning from observation*. Each object or row in an information table is considered to be an observation that has been made and from which can be learned. As for each observation that has been made, the according decision or result is also given, it is said to be a *supervised learning* approach. In the case of rough set theory, each observation is accompanied with an according decision value. The knowledge extracted from data in form of rules can then be used as an approximation of a certain concept that is based on specific decision attribute values.

In machine learning theory, this approach is also called *inductive learning*. The definition of inductive learning [34] is given as

Given a collection of examples of f,
return a function h that approximates f.

In the case of rough set theory, the data basis is the collection of examples of one ore more concepts f and the extracted rule set can be considered as the function h that approximates those concepts. As large data sets are generally assumed to be inconsistent in the sense of rough set theory, this vagueness is also reflected in the extracted rule set. Generally there will be rules that have the same conditional attribute values but different decision attribute values.

Before going into the details of the application of rough set theory within the image understanding process, some prerequisites for the approach are defined next.

In order to use the rough set approach in the process of image understanding, it is assumed that there is a finite set of pictures that can be used as *training data* from which rules are extracted. A single training object is considered as a set of MLC instances and according relations and an assigned HLC instance which describes the image by means of a high level concept. The set of MLC instances and according relations are assumed to be delivered by low level image analysis while the assigned HLC instance needs needs to be specified by a domain expert. For this purpose the domain expert is provided with appropriate tools to annotate the images that constitute training data. Training data is presented in tabular form where each MLC instance and each relation is used as a conditional attribute and *HLC* as the decision attribute. Table 4.1[1] shows a subset of training data in tabular form. With the increasing number of MLC instances and especially

[1]The meaning of the attributes is as follows: MLC instances: grnd - Ground Area, athl -

U	grnd	athl	pole	cb	jav	t(a,j)	n(a,p)	n(a,c)	HLC
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_i	1	1	0	0	1	1	0	0	Javelin
x_{i+1}	0	0	0	0	1	0	0	0	Javelin
x_{i+2}	0	1	1	1	0	0	1	1	Pole_Vault
x_{i+3}	1	1	0	0	0	0	0	0	High_Jump
x_{i+4}	1	1	1	0	0	0	1	0	High_Vault
x_{i+5}	0	1	0	1	0	0	0	1	High_Jump
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 4.1: An example training data decision system

relations, the information system will contain a large number of attributes which increases the complexity of reduct calculation. The tabular representation of the results of the low level image analysis forms the basis for the application of rough set theory within the process of image understanding.

The formal definition of the decision system derived from low level image analysis results can therefore be given as

$$\mathcal{A} = (U, A \cup \{d\})$$

where U is the finite set of training data, A is the attribute set defined as $A = MLC \cup Rel$ with MLC being the set of detectable MLC instances and Rel being the set of possible relations between MLC instances[2]. The value set V_A of all attribute in A is defined as $V_A = \{0, 1\}$ where 1 denotes that the according MLC instance or relation has been detected in the image, 0 otherwise. The value set V_d of the decision attribute d is defined as $V_d = HLC$ where HLC is the set of all possible high level concepts within the domain of consideration.

4.2 Preselection of possible high level concepts

With the preliminaries set up in the previous section, this section will describe how knowledge extracted by using rough set theory can be used for implementing a preselection of possible high level concepts as successful explanations.

For illustrating the approach Table 4.2 will be used throughout the example which is a simplified version of Table 4.1. The focus is on the decision between two HLCs *Paul_ Vault* and *High_Jump*, however the approach can

Athlete, pole - Pole, cb - Crossbar, jav - Javelin; Relations: t(a,j) - touches(athlete,javelin), n(a,p) - near(athlete,pole), n(a,c) - near(athlete,crossbar)

[2]Here training data is only used for the rough set approach and classifiers of low level image analysis have already been trained in a previous step.

Class	athlete	pole	crossbar	position	HLC	support
c_1	1	0	1	b	High_Jump	59
c_2	1	1	0	u	Pole_Vault	9
c_3	1	1	1	f	Pole_Vault	49
c_4	1	0	1	u	Pole_Vault	7
c_5	1	0	0	u	High_Jump	18
c_6	1	0	0	u	Pole_Vault	7
c_7	1	0	1	f	Pole_Vault	17
c_8	1	0	1	u	High_Jump	12

Table 4.2: A refined training data decision system

be used for an arbitrary number of HLCs as decision values. The attribute set A of the decision system is defined as

$$A = \{athlete, pole, crossbar, position\}$$

where *athlete*, *pole* and *crossbar* are MLC instances and *position* is derived from a set of spatial relations. The value set of the MLC attributes is $\{0, 1\}$ whereas the value set of *position* is $\{f, b, u\}$ with f denoting $facing$, b denoting $backwards$ and u denoting *undefined* expressing the position of the athlete above the crossbar. Defining a value set other than $\{0, 1\}$ for *position* is only because of readability reasons and could also be expressed by two relations, e.g. $above\_facing(athlete, crossbar)$ and $above\_backwards(athlete, crossbar)$ where $above\_facing = 0$ and $above\_backwards = 0$ is equivalent to $position = u$.

The data presented in Table 4.2 can be considered training data from which knowledge is to be extracted. In order to extract representative knowledge, training data has been selected in such a way that it reflects the typical appearances of still images of high jump or pole vault events. Therefore a set of each 89 images has been selected for which the low level image analysis delivers 3 different classes for high jump and 5 different classes for pole vault. Classes are understood in the sense that from all images in the same class the instances of the same MLCs and relations have been detected. In Appendix A typical pictures for each class are presented. In Table 4.2 the *support* for each class is also presented denoting the number of images in each class. Note that generally for all learning methods the quality of training data is crucial for the quality of the extracted knowledge.

Applying an exhaustive algorithm for rule generation to the set of training data yields to the set of rules presented in Table 4.3. The exhaustive algorithm calculates all possible decision relative reducts and value reducts and generates rules based on these reducts. The mathematical foundations of this calculation has been presented in chapter 3.

Considering the generated rules it becomes obvious that the existence of an athlete in the pictures can be ignored when the training pictures of *pole*

No.	Rule condition	Decision	support
r_1	if position = b	then HLC = High_Jump	50
r_2	if position = f	then HLC = Pole_Vault	66
r_3	if crossbar = 1 and position = u	then HLC = High_Jump	21
r_4	if crossbar = 1 and position = u	then HLC = Pole_Vault	7
r_5	if pole = 0 and position = u	then HLC = High_Jump	39
r_6	if pole = 0 and position = u	then HLC = Pole_Vault	14
r_7	if pole = 1	then HLC = Pole_Vault	58
r_8	if pole = 0 and crossbar = 0	then HLC = High_Jump	18
r_9	if pole = 0 and crossbar = 0	then HLC = Pole_Vault	7

Table 4.3: Rules extracted from training data

vault and *high jump* are used. However it should be noted that in a real scenario with a large number of HLCs the existence of an athlete in a picture might not be ignored by attribute reduction using rough set algorithms.

In Table 4.3 each rule is also assigned a support value. The support value denotes the number of images (or objects in general) from which the according rule has been derived from.

As an example, the rule set of Table 4.3 will now be used to propose an HLC for which the abduction process in the high level module should be started. This can be seen as using knowledge extracted from training images for a preselection of possible explanation based on the set of MLC instances and relations coming from low level image analysis. Note that in this simple example the benefit of choosing one out of two possible explanations might seem quite small, with a large number of high level concepts however the goal directed search becomes more clear.

Assume that low level image analysis delivers the following set of MLC instances and relations[3]:

$$x_1 = \{athlete, pole, crossbar, position(f)\}$$

Looking at the rule set it becomes obvious that for object x_1 rules 2 and 7 will fire which leads to the proposal of *Pole_ Vault* as a possible explanation for which the abduction process should be started.

For another object x_2 delivered by low level image analysis

$$x_2 = \{athlete, pole, crossbar, position(u)\}$$

the rules 3, 4 and 7 will fire, however from those rules no proposal can be made as the firing rules are *conflicting* meaning that rule 3 has *High_Jump* as decision value while rules 4 and 7 have *Pole_ Vault* as decision value. Note

[3]Note that for the sake of simplicity *athlete* stands for *athlete* = 1 and MLC instances not detected are not presented.

that object x_2 is an object that has not occurred in the training set from which rules have been extracted. Objects to be classified which have not occurred in the training set are also referred to as *unseen objects* or *unseen cases* which however are assumed to be possible objects for classification. The problem of conflicting rules however does not only occur with unseen objects but also with objects that have been part of the training set, such as

$$x_3 = \{athlete, crossbar, position(u)\}$$

for which rules 3 and 4 would fire which also leads to a conflict. As a result, a unique decision can not be obtained in case of conflicting rules. Rule conflicts can occur in cases of unseen objects or might stem from the fact that the underlying decision system has been inconsistent.

4.2.1 Conflict resolution

In the classical application of rough set theory for knowledge discovery from databases the goal is to find a unique decision from the knowledge provided even in such cases where there are conflicting firing rules. In the process of image understanding however, it is rather wanted to have multiple decisions with different priorities on the decision values (HLCs) in order to start the high level reasoning with the best explanation from the point of training data. The priority can also be used as an additional selection criteria for the selection of the best explanation in the HLC selection process.

In the classical rough set approach *conflict resolution* meaning selecting a single decision within a number of conflicting rule decisions is based on a numerical value which can also be used for a priority of the decision values. The calculations of the numerical value has various degrees of freedom and can be based on several quality measures of a rule set, which will be defined next.

Numerical measures

Formally a decision rule is defined as

$$\alpha \rightarrow \beta$$

read as *if* α *then* β where the pattern α is called the *antecedent* of the rule and β is called the *consequent* of the rule, respectively.

The *support* of a decision rule $\alpha \rightarrow \beta$ is the number of objects from which the rule has been calculated denoted $support(\alpha, \beta)$. In Table 4.3 the support for each rule is given.

Based on the support the *strength* of a rule can be calculated as

$$strength(\alpha, \beta) = \frac{support(\alpha, \beta)}{|U|}$$

No.	Strength	Certainty	Coverage
r_1	0.28	1	0.56
r_2	0.37	1	0.74
r_3	0.12	0.75	0.24
r_4	0.04	0.25	0.08
r_5	0.22	0.74	0.44
r_6	0.08	0.26	0.16
r_7	0.33	1	0.65
r_8	0.10	0.72	0.20
r_9	0.04	0.28	0.08

Table 4.4: Rules extracted from training data

where $|U|$ denotes the number of all images in the training data or objects in the universe in general. The strength of a rule states how well the rule covers or represents the data set.

The *certainty* of a rule is given as

$$certainty(\alpha, \beta) = \frac{support(\alpha, \beta)}{|\alpha|}$$

where $|\alpha|$ denotes the number of objects satisfying the pattern α as the antecedent of the rule. The certainty is a frequency based estimate of the conditional probability $P(\beta|\alpha)$. A rule $\alpha \rightarrow \beta$ with a certainty $certainty(\alpha, \beta) = 1$ is called a *certain* decision rule, while rules with $certainty(\alpha, \beta) < 1$ are called *uncertain* decision rules. In the case of conflicting rules $certainty(\alpha, \beta) < 1$ always holds. The certainty measure therefore gives an estimate about the probability of a consequent β given an antecedent α.

The *coverage* of a rule is given as

$$coverage(\alpha, \beta) = \frac{support(\alpha, \beta)}{|\beta|}$$

where $|\beta|$ denotes the number of objects having the rule consequent pattern β. Accordingly, the coverage can be interpreted as the conditional probability $P(\alpha|\beta)$. The coverage states how well a pattern α represents a decision β. In other words a rule with a high coverage reveals a "typical" pattern α for a decision β. Note however, that those "typical" patterns can also be highly uncertain.

Table 4.4 presents the rule quality measures of the rules given in Table 4.3. Note that in the special case of an equal distribution of two decision values $coverage(\alpha, \beta) = 2 \cdot strength(\alpha, \beta)$ holds.

The numerical measures *strength*, *certainty* and *coverage* can be used to calculate a uniform quality measure for a set of rules. The calculation of this measure can generally be tuned for the specific set of rules in order to obtain the best quality under consideration of the rule set's properties. In [1] several quality measures are given which have been evaluated on an empirical basis.

Voting

However, the numerical measures defined in the section above do not yet solve the problem of conflicting rules. The process of *voting* is a general technique for deriving a priority measure from a set of conflicting rules which can take into account the quality measures defined in the section above. Usually, the decision class with the highest priority measure is chosen as the final decision.

In the voting process the set of rules taking part in the classification is denoted as Rl.

For a single object x_i for which a decision is to be made, the process works as presented in the following:

1. The set of rules Rl is searched for those rules for which the object x_i satisfies the rule's antecedent. Those rules selected are said to be *firing* for object x_i and are denoted $Rl(x_i)$. If none of the rules can fire, either a fallback mechanism is initiated or the process ends with an error.

$$Rl(x_i) = \{r \in Rl | x_i \models \alpha\}$$

2. After identifying the set of firing rules, an election process among the rules is initiated. Therefore, each firing rule $r \in Rl(x_i)$ casts a number of votes in favor of its decision class. In a simple case, the number of votes to be casted usually depends on the support of the rule. However, it can also be based on the quality measures defined in the section above which can be tuned for the rule set's properties. For each decision class the number of votes is added up from which the final result is derived. For example, the number votes in favor of a certain decision class β is the sum of all votes casted by all rules predicting β and firing for x_i.

$$R_{x_i,\beta} = \{r \in Rl(x_i) | r \ predicts \ \beta\}$$
$$votes(\beta) = \sum_{r \in R_{x_i,\beta}} votes(r)$$

3. Finally a *decision score* is calculated by dividing each number of votes for a certain decision class by a normalization value. The normalization value is based on the sum of all votes casted in the step before.

HLC	decision score
High_Jump	0.87
Long_Jump	0.78
Javelin	0.45
Pole_Vault	0.10
Hurdles	0.00
⋮	0.00

Table 4.5: Decision scores for several HLCs

$$decision\_score(x_i, \beta) = \frac{votes(\beta)}{norm(x_i)}$$

In order to clarify the voting process, an example will be given with the data of Table 4.2.

Example 4.2.1. Taking object $x_2 = \{athlete, pole, crossbar, position(u)\}$ again, it should now be classified by using a conflict resolution based on voting. For the sake of simplicity, the number of votes to be casted in favor of a certain decision class is defined to be based merely on the support of the rule.

As already mentioned above the set of firing rules for object x_2 is $Rl(x_2) = \{r_3, r_4, r_7\}$ with $R_{x_2,High\_Jump} = \{r_3\}$ and $R_{x_2,Pole\_Vault} = \{r_4, r_7\}$. Accordingly, the number of votes in favor of the two decision classes are calculated as $votes(High\_Jump) = 21$ and $votes(Pole\_Vault) = 65$ with a normalization value of 86. The leads to the decision score $decision\_score(x_2, High\_Jump) = 0.24$ and $decision\_score(x_2, Pole\_Vault) = 0.76$.

This example shows how a decision score is derived based on the voting process using the support as a rule quality measure. Again it should be pointed out here that the number of votes each rule casts can also be determined by a far more complex quality measure. The calculation of the quality measure remains as one on many parameters within the use of the rough set approach that needs to be adjusted for the specific data to be applied on.

Furthermore, the example might not reveal the main intention of prioritizing the HLCs for which the abduction process is to be initiated. Table 4.5 shows the result of an arbitrary object (set of MLC instances and relations) that has been classified by the above mentioned process.

In the case of Table 4.5 the high level reasoning module would apply the abduction process first by trying to explain the delivered set of MLC instances and relations through the high level concept *High_Jump*. As part of the process the reasoning rules defined for *High_Jump* would be applied. Using the decision scores resulting from the rough set approach, a goal

MLC	conf. measure
athlete	0.89
pole	0.25
crossbar	0.95
position(b)	0.81

Table 4.6: Confidence measures for MLC instances of object x_4

directed search is introduced by a priority on the HLCs to be tried first. The rough set approach therefore introduces a method to guide the high level reasoning module based on knowledge that has been extracted from image training data.

How the high level reasoning module proceeds after a successful explanation using a certain HLC still needs to be defined. One simple and fast way would be taking the first successful explanation starting with HLCs having the highest decision score. It would also be possible to define a certain threshold t_{conf} that determines for which HLCs the abduction process is to be initiated. This would then allow for finding several successful explanations for which the HLC selection process needs to be applied. Recalling chapter 2, several selection criteria have already been identified that can be used to find the best explanation. The decision score could also be taken into account in the selection process being an additional selection criteria. For example, assume that for Table 4.5 *High_Jump* and *Long_Jump* both have yielded a successful explanation in the abduction process. Then the decision score could make the necessary difference in the selection process if other criteria applied fails to identify the best explanation. It should be stressed again that all measures originating from the rough set approach are data grounded which makes it necessary to invest in the quality of training data as well as fine-tuning possible parameters accordingly in order to obtain satisfying results.

4.3 Integrating confidence measures from low level image analysis

The approach presented for exploiting rough set theory in the process of image understanding so far ignores confidence measures that are assigned to every MLC instance by the low level image analysis. Recalling chapter 2, the final confidence measure of a MLC instance is based on two different measures resulting from the holistic and region based approaches as part of the low level analysis process. Although the underlying semantics of the final measure is not completely clear, it is interpreted as a pseudo probabilistic value stating the certainty as a degree of belief that the identified object or region in the picture is a MLC instance of a specific type.

Figure 4.1: Image for object x_4

In the following, the confidence measures from low level image analysis will be used within the rough set approach in order to reflect the certainty of specific MLC instances.

Assume that based on the image in Figure 4.1 the low level image analysis delivers the following set of MLC instances and relations:

$$x_4 = \{athlete, pole, crossbar, position(b)\}$$

Obviously the image in Figure 4.1 is a *High_Jump* event. However, in the lower part of the picture the low level image analysis detects an object which is identified as a *pole*. Using the classification approach as described above with a support-based voting process for conflict resolution, the decision scores are $ds_{hj} = 0.46$ for *High_Jump* and $ds_{pv} = 0.54$ for *Pole_Vault*, respectively. As part of the rough set module, this leads to *Pole_Vault* being a better explanation than *High_Jump* [4].

Next, the confidence measures delivered by low level analysis will be considered. Therefore it is assumed that each MLC instance and each relation are assigned with an according value as given in Table 4.6. Note, that the MLC instance *pole* that is detected by low level image analysis has been assigned a low confidence measure which might result from e.g. not being in focus or not having the appropriate size as defined in the analysis module.

In order to reflect the confidence measures in the rough set approach, an extension of the voting process is proposed next.

The first step of the voting process remains the same in such a way that the set of all rules is scanned for those rules that fire. In the support based voting approach all firing rules predicting a decision class β cast the number

[4] It is assumed that the high level reasoning module finds both HLCs *Pole_Vault* and *High_Jump* as explanations. With different ontology and rules, the high level reasoning module might have also only found one explanation.

of votes depending on the support of the rule, so that

$$R_\beta = \{r \in Rl(x) | r \ predicts \ \beta\}$$

$$votes(\beta) = \sum_{r \in R_\beta} votes(r)$$

with

$$votes(r) = support(r)$$

In order to integrate the confidence measures from low level image analysis, the number of votes each rule casts has to reflect the measure values. As an example, two firing rules having the same support, but different confidence measures of MLC instances in their antecedents, also need to cast different numbers of votes. Therefore each antecedent α is considered on the basis of its constituents $a_1, a_2, ..., a_n$

$$\alpha = a_1 \wedge a_2 \wedge ... \wedge a_n$$

where $a_i, \ i = 1, ..., n$ is a MLC instance or a relation in case of the image understanding process. For each α a confidence measure needs to be calculated which is based on the confidence measures of its constituents $a_1, a_2, ..., a_n$. A simple way to do this is to use the arithmetic mean of confidence measures of the constituents which results in a confidence measure for the rule:

$$cm_r = \frac{1}{n} \cdot \sum_{i=1}^{n} cm_{a_i}$$

However other functions or approaches can be used for calculating the resulting confidence measure of α reflecting different type of constituents. For example a weighted sum could be applied in order to weigh constituents of a certain type higher than others, e.g. considering MLC instances more important than relations.

The confidence measure of each firing rule can then be used to weigh the number of votes each rule gets to cast. In a support based voting process, the number of votes is then calculated as:

$$votes(r) = support(r) \cdot cm_r$$

This means that the higher the confidence measure of the rule, the more votes are being casted by rule, where the rule's confidence measure is computed based on the confidence measures of the constituents of the rule's antecedent. Again, the integration of confidence measures in the voting process can also be used in conjunction with more complex rule quality measures as presented before instead of only using the support of a rule.

In order to clarify the extended voting process, the following example is given.

Example 4.3.1. For an object $x_4 = \{athlete, pole, crossbar, position(b)\}$, the confidence measures of Table 4.6 and the set of rules of Table 4.3 are given. First all firing rules are identified to constitute the set of firing rules $Rl(x_4) = \{r_1, r_7\}$ with $R_{Pole\_Vault} = \{r_7\}$ and $R_{High\_Jump} = \{r_1\}$. Next, for each rule the number of votes is calculated as

$$votes(r_1) = support(r_1) \cdot cm_{\alpha(r_1)} = 45$$

with $support(r_1) = 50$ and $cm_{\alpha(r_1)} = 0.81$. As rule r_1 is the only rule firing in favor of *High_Jump* this results in $votes(High\_Jump) = 45$. Accordingly $votes(Pole\_Vault) = votes(r_7) = 14.5$ is obtained with $support(r_7) = 58$ and $cm_{\alpha(r_7)} = 0.25$. Using normalization as presented in the voting process, decision scores for *High_Jump* and *Pole_ Vault* are finally obtained as $ds_{hj} = 0.76$ and $ds_{pv} = 0.24$.

This simple example shows how the confidence measures of constituents of a firing rule can be reflected in the final decision scores. The proposed approach allows for exploiting confidence measures from low level image analysis by the high level reasoning module. Again it should be noted that the confidence measures from low level image analysis are interpreted as probability measures being degrees of belief. This is assumed for the low level image analysis and definitely holds for decision scores directly extracted from data by the rough set approach. This results in an interpretation of the measures in such a way that the higher the value the more certain low level image analysis is about a MLC instance, or HLC for the high level reasoning module respectively. However, it still remains as an open issue to define the semantics of a final certainty measure that is computed from constituent measures. In the process of image understand these constituents are two measures from low level image analysis coming from the holistic and region based approach, as well as a decision score based on training images and applying rough set theory. Nevertheless experiments can be conducted to proof the practicability of such an approach. This however is outside the scope of this work.

4.4 Summary

This chapter presented possible applications of the rough set approach within the context of image understanding. For exploiting the rough set approach, the method of image understanding needs to be extended by a learning phase which allows for extracting knowledge in form of rules from training image data. This additional knowledge which can be seen as an extension of the background knowledge of the high level reasoning module, can then be used for introducing a goal directed search within the high level reasoning module: Based on a set of MLC instances and relations, HLCs as possible successful explanations can be preselected using knowledge extracted from image training data.

Furthermore, the rough set approach introduces a decision score that is based on quality measures of rules which have fired for a certain object. This decision score can then also be used as an additional criteria for the selection of the best explanation in case several HLCs lead to a successful result in the abduction process.

As the calculation of the decision score offers several degrees of freedom, it allows for integrating confidence measures from low level image analysis as well. This enables reflecting the uncertainty of the low level image analysis within the high level reasoning module up to a certain degree.

For deriving successful results with the approach as presented in this chapter it will be necessary to make use of several tuning parameters at different stages of the approach. This includes the selection of training data as well as the selection of a specific rule generation algorithm in the learning phase. For the appropriate application of rules to classify an object, a way for the calculation of rule quality measures and the integration of confidence measures has to be chosen. All these parameters are to be tuned in an extensive "train and test" phase in order to find out the optimal parameter values and selections.

Apart from the approach of using rough set theory in the image understanding process as proposed in this work, alternative approaches are also considered as part of the BOEMIE research project. One of the approaches is based on probabilistic default reasoning with conditional constraints [20] where statistical knowledge for different classes of objects is used to derive conclusions about properties of individuals. For an application of rough set theory for probabilistic default reasoning Appendix D presents an idea how statistical knowledge can be learned from data.

Chapter 5

Approximate Reasoning

The main contribution of this work has been applications of rough set theory in the process of image understanding so far. This chapter will first present *approximate reasoning* from a general point of view. Afterwards, it will focus on different solutions for approximate reasoning including rough mereology as an extension of rough set theory. Rough mereology as an approximate reasoning approach will be presented in detail for which the basic notions of rough set theory are needed. The chapter concludes with a comparison of the different approaches to approximate reasoning presented in the following sections.

5.1 Approximate Reasoning

Coming back to the concept of an intelligent agent used in the introduction, an intelligent agent is based on the idea of acting an reasoning like humans. Considering the abduction process of the high level reasoning module as part of the overall reasoning ability of the agent, the abduction process is a purely logical approach which can deliver one or more explanations or fail. Regarding the approaches of the low level image analysis as well as the rough set approach, results are obviously not only true or false but assigned a specific confidence measure that expresses the level of uncertainty. The way humans are reasoning is generally also based on some level of uncertainty. Humans are generally not able to have complete knowledge of a given fact, however they are able to make decision or derive conclusions based on those uncertain facts.

As a first example consider a sentence like *fast cars are expensive*. This sentence states a simple fact which humans regard as a true statement within a certain domain. However, considering the constituents of the sentence, it becomes clear that they handle some sort of uncertainty. For example, the linguistic label *fast* is not defined in a crisp manner, meaning there is no definition of the exact speed range of *fast*. The same fact holds for the label *expensive*. However, humans obviously have the ability to handle this *vagueness*[1] or *graduality* of the natural language. This also includes reasoning fact such as implications in the form of "if a car is fast, then it is expensive". One approach for handling those concepts for an intelligent agent has led to the emerging of a field called *computing with words* which is mainly realized by fuzzy set approaches. Therefore this chapter will give an overview on how reasoning with *fuzzy sets* and according *fuzzy rules* can be used for approximate reasoning.

As a second example, consider a statement like *if the car doesn't start, the battery is not ok.* This sentence actually expresses a specific combination of cause and effect, namely the effect that the car doesn't start is caused by the battery not being ok. However there might also be other reasons why the car doesn't start, e.g. because a cable is broken or because there is no gas. Considering the way humans handle those statements, one can introduce the concept *degrees of belief*. E.g., if the car doesn't start, a persons believes in the battery not being ok in a high degree while a broken cable or no gas have lower degrees of belief. Also, the person might not have all possible reasons in mind for the fact that the car is not starting, however humans are still able to reason in such situations and can act based on their degrees of belief in a specific situation. Another characteristic in this way of reasoning is the fact that degrees of belief change, whenever new knowledge has been revealed. E.g., if the person has just fueled the car the degrees of belief for the battery not being ok or the cable being broken will be higher than without the knowledge of a fueled car. These facts can be taken into account by exploiting *probability theory* and *bayesian reasoning*, which will become the second focus of this chapter.

Compared to fuzzy set or probability approaches to approximate reasoning, *rough mereology* as the third approach presented in this chapter, is a novel methodology. Rough mereology is based on the ideas and notions of rough set theory as presented in this work so far and is combined with *mereology* being the theory of being a part. The very basic underlying concept of rough mereology is based on the extension of *being a part* in mereology to *being a part in a degree*. The approach itself however, is based on various constituents which will be covered in detail later. The level of detail will be higher than for fuzzy and probabilistic approaches. In fact, literature on rough mereology for approximate reasoning, especially for readers not grounded in the mathematical theory, is rare. Therefore this work tries to give a concise presentation of rough mereology for approximate reasoning. As an example for rough mereology consider a statement like *given a specification, deliver an object satisfying the specification to a certain degree*. The example shows that rough mereology is not based on general concepts of human reasoning as linguistic labels in fuzzy or degrees

[1] Note, that the definition of vagueness differs from the definition in rough set theory.

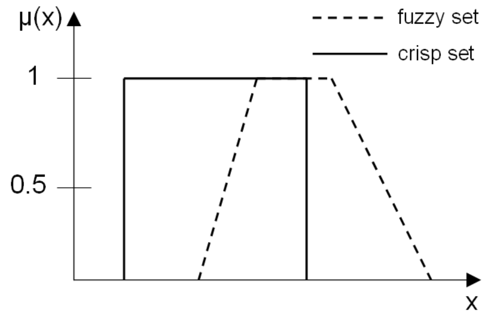


Figure 5.1: Membership functions of a fuzzy and a classical set

of belief in probabilistic approaches. Rough mereology is rather based on an artificially designed scheme that makes use of knowledge that is directly extracted from data. This relates rough mereology to the basic rough set approach and can also be seen as reasoning from data.

This chapter concludes with a detailed discussion on the different approaches for approximate reasoning presented in this work and according different fields of application.

5.2 Fuzzy Set Theory for Approximate Reasoning

As already stated in the introduction to rough set theory, in classical set theory an element either belongs to a set or not which is also referred to as *crispness*. Considering the property of belonging to a set as a *membership function* the values of such a membership are only $\{0, 1\}$. In fuzzy set theory the notion of belonging to a set is extended in such a way that an object can have a membership value of a degree between 0 and 1 where 0 and 1 correspond to the membership in classical set theory and values between are considered as partial membership. Figure 5.1 illustrates the idea of membership functions of a classical and fuzzy set. Fuzzy set theory can actually be seen as an generalization of classical set theory as every classical set can be defined as a fuzzy set with a membership function as illustrated in Figure 5.1.

With classical set theory being the basis for classical logic, fuzzy set theory serves as a foundation for fuzzy logic where truth values are not only 0 and 1 (*true* and *false*) but can also take values in the interval $[0, 1]$. For the logical connectives $\wedge$ (and), $\vee$ (or), $\neg$ (not) and $\rightarrow$ (implication) used in classical logic, according functions and relations are defined in fuzzy logic. Pairs of *t-norms* and *t-conorms* are used for logical $\wedge$, $\vee$ and $\neg$ and fuzzy relations represent implication. For a short introduction into t-norms and t-conorms refer to Appendix C. How implication is handled in fuzzy logic will be covered in the following sections along with reasoning based on fuzzy rules.

5.2.1 Fuzzy Rules

As already stated in the introduction, statements like *fast cars are expensive* can also be interpreted as rules in the form of *if a car is fast then it is expensive*. With rules being a possible way of knowledge representation, this rule can be seen as a *fuzzy rule* where *fast* and *expensive* are defined as fuzzy sets. Furthermore, for a specific car c_i the maximum speed might be known exactly, for example 195.5 km/h, or it might also be defined by a fuzzy set that describes the maximum speed as *approximately 190 km/h*. Such data (or information) that is known for a specific object is referred to as *fuzzy data*.

A fuzzy rule can then be written as

$$\text{if } \mathit{c.speed} \text{ is } fast \text{ then } \mathit{c.price} \text{ is } \mathit{expensive}$$

where *fast* and *expensive* are defined as fuzzy sets.

A general fuzzy rule is written as

$$\text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ then } y \text{ is } B$$

with the fuzzy membership functions $\mu_{A_1}(x_1)$, $\mu_{A_2}(x_2)$ and $\mu_B(y)$. The antecedent of such a rule can be the combination of several propositions by means of logical connectives and negation.

In classical logic, implication is denoted as

$$\varphi \to \psi$$

which can be used for representing rules like

$$\text{if } \varphi \text{ then } \psi$$

For fuzzy rules implication is not defined in a unique way but can be given by means of an *implication function*. For example the rule above can be realized by a fuzzy relation

$$R = I(\top(A_1, A_2), B)$$

where $\top$ is a t-norm and I is an implication function.

The resulting fuzzy membership function is then derived as

$$\mu_R(x_1, x_2, y) = I(\top(\mu_{A_1}(x_1), \mu_{A_2}(x_2)), \mu_B(y))$$

Name	$\varphi \rightarrow \psi$
Lukasiewicz	$min\{1-\varphi+\psi, 1\}$
Gödel	$\begin{cases} 1 & \text{if } \varphi \leq \psi \\ \psi & \text{else} \end{cases}$
Goguen	$\begin{cases} 1 & \text{if } \varphi = 0 \\ min\{1, \frac{\psi}{\varphi}\} & \text{else} \end{cases}$
Kleene-Dienes	$max\{1-\varphi, \psi\}$
Zadeh	$max\{1-\varphi, min\{\varphi, \psi\}\}$
Reichenbach	$1-\varphi+\varphi \cdot \psi$
Mamdani	$min\{\varphi, \psi\}$

Table 5.1: Fuzzy implication functions

5.2.2 Fuzzy implication functions

Compared to classical logic where implication is defined uniquely, in fuzzy logic there are several functions that can be used to implement implication. Table 5.1 lists some of the most popular implication functions used in fuzzy logic.

Following [9] implication functions like those of Table 5.1 can be classified into different categories:

- Fuzzy implications functions based on the classical implication where $\varphi \rightarrow \psi$ is defined by $\neg\varphi \vee \psi$:

  $$I(\varphi, \psi) = \bot(\neg\varphi, \psi)$$

  which are also called S-implications. Examples of this type of implication functions are the *Kleene-Dienes* and *Lukasiewicz* implication functions where the according t-conorms are $\bot_{max}(a, b) = max\{a, b\}$ for Kleene-Dienes and $\bot_{Luka}(a, b) = min\{a + b, 1\}$ for *Lukasiewicz* respectively.

- Fuzzy implication based on the implication in quantum logic:

  $$I(\varphi, \psi) = \bot(\neg\varphi, \top(\varphi, \psi))$$

- Fuzzy implications reflecting partial ordering on propositions:

  $$I(\varphi, \psi) = \begin{cases} 1 & \text{if } \varphi \leq \psi \\ 0 & \text{if } \varphi = 1 \wedge \psi = 0 \\ (0, 1) & \text{else} \end{cases}$$

  Most of these implication functions are R-implications as they are derived from a pair of t-norms and t-conorms by means of a residual. Examples in this category are Gödel and Goguen implications which are also R-implications.

- Fuzzy implications interpreted as a conjunction:

$$I(\varphi, \psi) = \top(\varphi, \psi)$$

  These implication functions do actually not represent implication in the classical sense because they are based on conjunction. Conjunction based fuzzy implication functions are mainly applied in fuzzy control where the Mamdani conjunction is often used for implementing fuzzy controllers.

With several options for implementing implication in fuzzy logic, also different implication semantics come along with the implication functions. Obviously, for the same truth values for φ and ψ different implication functions yield different results as $I(\varphi, \psi)$. Therefore when designing a system that is based on fuzzy rules, the choice of an appropriate implication function needs to represent the required semantics of the implication of an according rule.

5.2.3 Reasoning with fuzzy rules

In the previous section various implication functions have been presented that can be used for representing implication in case of fuzzy rules. What remains open is the way how fuzzy rules can be used for inferring knowledge based on fuzzy information.

In classical logic, *if-then*-rules can be represented as *modus ponens* which is an inference rule that allows for inferring knowledge based on a given fact. For example, a rule

$$\text{if } \varphi \text{ then } \psi$$

can be written as

$$\frac{(\varphi \rightarrow \psi), \varphi}{\psi}$$

as modus ponens.

For reasoning with fuzzy rules, the *compositional rule of inference* as a realization of the *generalized modus ponens* has been proposed which can be written as

if x is A then y is B
x is A'
y is B'

where the rule *if x is A then y is B* is a fuzzy rule and *x is A'* is fuzzy data from which *y is B'* is derived.

As an example take the rule

$$\text{if } \textit{c.speed} \text{ is } fast \text{ then } \textit{c.price} \text{ is } \textit{expensive}$$

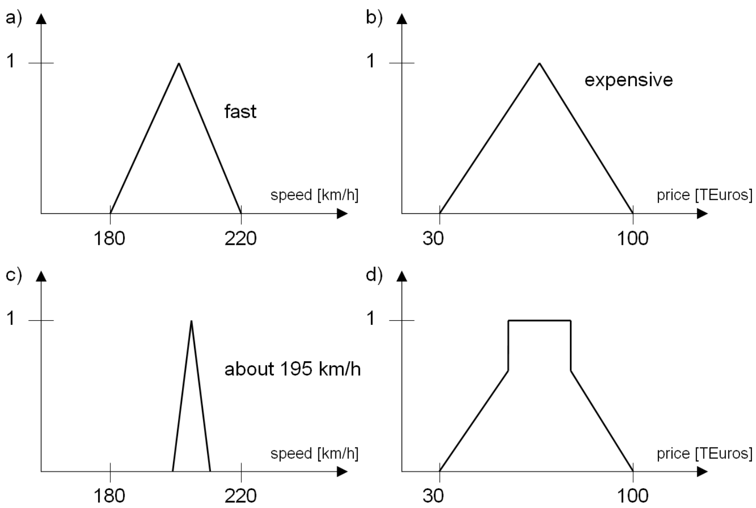


Figure 5.2: An example fuzzy rule applied to fuzzy data

with a given fact of a specific car's speed of *about 195 km/h* from which the car's price can be inferred.

In the generalized modus ponens B' is calculated from A' by

$$B' = A' \circ R$$

where R is a fuzzy relation representing the rule *if x is A then y is B.*

Figure 5.2 gives a graphical example of the generalized modus ponens with a fuzzy rule. In graphs a) and b) the fuzzy sets for the labels *fast* and *expensive* are given, depending on the speed and the price of a car. Graph c) represents the fuzzy information that is known about a specific car's speed defined as *about 195 km/h.* Graph d) shows the result as the car's price inferred from the the rule if *c.speed* is $fast$ then *c.price* is *expensive* with the fuzzy information of graph c) and using the Gödel implication function. For details on the exact calculation of fuzzy rules using the compositional rule of inference refer to [17].

5.2.4 Fuzzy rule base

So far, reasoning with a single fuzzy rule has been presented. Generally, not only a single fuzzy rule is used to express certain knowledge but several fuzzy rules are combined in a *fuzzy rule base.* A fuzzy rule base can therefore be seen as a special form of knowledge representation in form of rules.

For fuzzy rule bases, specific properties have been identified being *continuity*, *consistency* and *completeness.* In order to satisfy the continuity property of a rule base, rules with adjacent antecedents also need to have adjacent consequents. Adjacency of fuzzy sets generally means the overlapping of fuzzy sets. In most cases a rule base is considered *continuous* when there is a fuzzy partition of the universe, meaning that the universe is partitioned by fuzzy sets that overlap only with the next higher or lower fuzzy set.

The consistency of a rule base states how far the knowledge expressed by the set of rules in the according rule base is consistent. Consistency of rule bases is not only a problem of rule base design but is often unavoidable with large and complex rule bases. A simple example for two inconsistent rules are

$$r_1 : \text{if obstacle in front then go left}$$
$$r_2 : \text{if obstacle in front then go right}$$

Apart from this obvious example, rule bases can easily become inconsistent when extensively using the *or* connective in the rule antecedents. Note, that question of consistency of a rule base is not restricted to fuzzy rule bases which are focused in this section.

The *completeness* of a fuzzy rule base states how well certain input values are being covered by an according rule base. In the worst case there may be some regions in the universe of discourse that are not covered by an according rule antecedent which are called *blank spot*. The according region in the universe is called *incomplete*. On the semantics level this expresses a lack of knowledge as no membership function for the according input data has been defined. Completeness can be expressed in different grades which is computed based on α as the sum of membership values of input data. Subcompleteness ($\alpha < 1$), strict completeness ($\alpha = 1$) and overcompleteness ($\alpha > 1$) define different levels of completeness where in the case of regions with strict completeness the membership values of certain input data is equally distributed over the universe of discourse.

Aggregation of rules

As already stated above, a fuzzy rule base generally consists of a set of fuzzy rules. Each of these rules is transformed into a fuzzy relation using an implication function as discussed above. For a set of rules it is then possible to either transform all rules into a single fuzzy relation and apply it to data or first calculate the result of each rule and then aggregate the results.

The two different options are defined as

- *global inference* where the final result is based on a single relation R that is aggregated based on relations R_k of according r_k rules (also called FATI - "first aggregate, then infer") and
- *local inference* where the result of each rule is calculated and then aggregated into a final result (also called FITA - "first infer, then aggregate).

It can be shown that, using t-norms as implication functions (classical conjunction based implication), there is no difference between global and local inference, whereas for classical implication based implication functions the results of global and local inference can differ. Also if the input data for the rules are singletons, which are fuzzy sets that have a nonzero membership value for only one element of the universe of discourse, there is no difference between results obtained by global and local inference. However, the choice of a rule aggregation strategy is an additional parameter when designing an approximate reasoning system based on fuzzy set theory.

5.2.5 Semantics of Fuzzy Sets

In this work, approximate reasoning based on fuzzy set theory has been presented as a method to handle the graduality or vagueness of the human language or human thinking in general and to reason with these properties. The success of expert systems based on fuzzy rules therefore depends on the design of fuzzy membership functions, definition of rules, choices of implication functions and a rule aggregation strategy as well as rule base properties mentioned above. Finally, all these parameters need to be tuned in such a way that the reasoning results are similar to those derived by humans which was one of the basic ideas behind the concepts of fuzzy set theory [45].

A different view on fuzzy set theory to be mentioned in this work is *possibility theory* where fuzzy set theory is used as a different means for representing uncertainty. In this context, fuzzy sets are seen as *possibility distributions* of an object having a certain property. For example, given a membership function $\mu_{fast}(x)$, the higher the value $\mu_{fast}(x)$, the higher the possibility that $speed(x) = fast$. Note that the concept of possibility differs from the concept of probability, as high possibility does not imply high probability. The famous example is *Hans' breakfast* stating that it is possible to eat 5 eggs for breakfast, though not very probable. Possibility theory is also based on a duality principle where for each possibility measure and according *necessity* measure is calculated. Possibility and necessity are then used together to give information about the certainty of an event. A detailed discussion on possibility theory and its use within the context of uncertainty handling can be found in [10].

5.3 Probabilistic reasoning

The next mechanism for approximate reasoning presented in this work is based on probability theory. The theory of probabilities has been studied for several hundreds of years in various forms of which the most relevant for approximate reasoning is *bayesian networks*. As already discussed in the introduction, probabilities allow expressing degrees of belief for different events, e.g. the probability of a starting motor. In the introduction of this chapter the concept of cause and effect has also already been presented which is formalized in bayesian networks.

The following section will therefore give a brief introduction into probability theory, its concepts and according mathematical laws. Afterwards bayesian reasoning as a means for approximate reasoning with probabilities and bayesian networks will be presented along with an illustrating example.

5.3.1 Probability theory

In the introduction the notion of a degree of belief as a means for expressing an agent's certainty has already been introduced. For example, a person beliefs that the motor of its car starts with a probability of 90% if there is enough gas and the battery is ok. The person also beliefs that the motor does not start with a probability of 10% even though there is enough gas and the battery is ok. The reason for the car not starting even though there is enough gas and the battery is ok, can have several causes such as a broken cable, spark plugs not being ok or an electronical problem. Accordning to [34], those cases belonging to the probability of 10% can be classified into three main classes:

- *Laziness*: There are simply too many reasons why the car does not start even though there is enough gas and the battery is ok.
- *Theoretical ignorance*: There is no exact theory covering all reasons why the car doesn't start.
- *Practical ignorance*: Even though all reasons are known, there might still remain cases which are uncertain as there is no knowledge about them, e.g. the usage of a unique oil mixture.

Probability theory therefore offers a way to summarize the uncertainty that comes from an agent's laziness and ignorance. Still the question remains open, how an agent is able to give an exact numerical value for its degree of belief. This question has led to endless discussions on the origin of probabilities. The *frequentist* view in this debate requires probabilities coming from experiments. For example, if the car's motor started 90 times out of 100 with enough gas and the battery being ok the according probability is 0.9. Note that the rule quality measures of the rough set approach are also based on the frequentist view of experiments. The *objectivist* view sees probabilities rather as properties of objects than the degree of belief of an agent. For example, a dice has the property that each number has a probability of 1/6. Experiments of the frequentist view can then be seen as a method to obtain those properties of objects, e.g. rolling the dice 10.000 times. Obviously, the objectivist view requires that there is some knowledge about the probabilistic properties of an object, which however is not always practical. The *subjectivist* view sees probabilities as real degrees of belief as it has been presented in this work so far. In the subjectivist view probabilities reflect the agent's knowledge and experience about the world rather than deriving numerical values from experiments or physical properties of objects. If not stated otherwise, probabilities will be considered degrees of belief as defined in the subjectivist view in the following sections.

5.3.2 Laws of probability theory

In probability theory there are several axioms and rules that define how probability measures can be used in order to calculate the likelihood of an event. This section gives a brief introduction into the basic notions of probability theory according to [25].

If a probability measure is given without any other information, the measure is called *prior probability*. For example, if there is no additional information about the status of the battery being ok, a degree of belief can be given as $P(battery\ ok = true) = 0.95$ where $battery\ ok = true$ is a proposition describing the event.

Whenever other information is known, *conditional probabilities* need to be given. For example, given that the battery is ok, the probability of the motor starting is given as $P(a|b) = 0.9$ with $a : motor\ starting = true$ and $b : battery\ ok = true$. The conditional probability therefore states the probability of a given the information b. Note that the conditional probability can be the same as the prior probability. For example $P(battery\ ok = true) = P(battery\ ok = true|headache = false)$ as the status of the battery is obviously independent of the driver's headache.

For all probabilities it always holds that $0 \leq P(a) \leq 1$ and the probabilities of necessarily true and false events are $P(true) = 1$ and $P(false) = 0$.

Assuming that two events are exclusive, meaning that only one of them can occur, disjunction is calculated as

$$P(a \vee b) = P(a) + P(b)$$

If the two events are not exclusive, e.g. $a : battery\ ok = true$ and $b : enough\ gas = true$ which both can hold at the same time, disjunction is calculated as

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

If the two events are exclusive and exhaustive, meaning that only one of them can occur and they are the only possible events that can occur, e.g. $a : battery\ ok = true$ and $b : battery\ ok = false$, disjunction sums up to 1

$$P(a \vee b) = P(a) + P(\neg a) = 1$$

This is given more generally for a set n of exclusive and exhaustive events as

$$\sum_{i=1,\ldots,n} P(a_i) = 1$$

The calculation of the probability of two events occurring together, such as a : *battery ok* $=$ *true* and b : *enough gas* $=$ *true*, is done using the product rule given as

$$P(a \wedge b) = P(a) \cdot P(b|a) = P(b) \cdot P(a|b)$$

For two independent events this can be simplified as $P(a \wedge b) = P(a) \cdot P(b)$, e.g., assuming that there is enough gas independent from the status of the battery.

The results of the product rule are used to derive Bayes' theorem which builds the foundation of reasoning systems based on probabilities:

$$P(a|b) = \frac{P(b|a) \cdot P(a)}{P(b)}$$

With the rules and laws presented in this section, the next section will describe how these rules can be used for approximate reasoning in bayesian networks.

5.3.3 Bayesian networks

The previous section presented the basic laws and rules of probability theory. These laws and rules correspond to the laws and rules for reasoning based on fuzzy set theory. Therefore an analogy between the probability theory and fuzzy logics can be observed. In order to use probability theory to reason about uncertain knowledge, a way to represent this knowledge is needed. The knowledge requires to include the prior and conditional probability measures of events in a specific domain, but also needs to represent the causal connections of events. As an example, the fact the starting of a motor is effected by the battery status needs to be represented as well as the fact that the battery status does not effect the level of gas. Defining those dependencies or influences is not only easier for a domain expert, but also reduces the complexity of calculation of probabilities.

The main approach for representing dependencies of events are *bayesian networks*. Formally, a bayesian network is a set of nodes with directed arcs as connections between the nodes. Every node is connected to another node but there are no directed cycles. Each node gives information about the probability of a certain event which is influenced by its parents nodes only. Nodes that have no parents are considered conditionally independent from all events for which the prior probability is given. Nodes with parents are given with a *conditional probability table*, denoting the conditional probabilities of all possible combinations of values of the parent nodes.

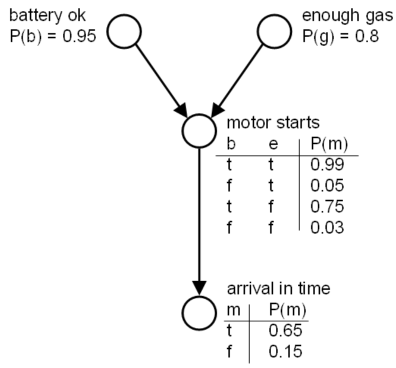


Figure 5.3: A simple bayesian network

Figure 5.3 shows a simple bayesian network representing the domain about persons who get to work by car. The network is based on an example given in [25] and has been adapted in this work. The network captures the fact that whether the motor starts is influenced by the status of the battery and whether there is enough gas. Both facts are considered to be independent from all other events so that their prior probabilities are given. Note that in a more detailed network, also causes for the battery status and the gas can be given. E.g. whether the battery has been charged or somebody has filled the tank. In the network shown in Figure 5.3 all these causes are reflected in the prior probabilities representing the laziness and ignorance mentioned earlier.

Whether the motor starts or not is directly influenced by the status of the battery and the amount of gas available. This fact is captured by two parent nodes for the node representing the starting motor. Along with this node, the according conditional probability table is given, presenting the conditional probabilities of $P(m|b \wedge e), P(m|\neg b \wedge e), P(m|b \wedge \neg e)$ and $P(m|\neg b \wedge \neg e)$ with *m: motor starts = true*, *b: battery ok = true* and *e: enough gas = true*. Note that the event *enough gas = false* does not mean that there is no gas in the tank anymore but only that the warning light is on. Therefore the degree of belief for the motor starting with the battery being ok and not enough gas is still 0.75.

The last node gives the conditional probabilities for the event representing the arrival at work in time. Again, laziness and ignorance are reflected in the probabilities. The network has been designed in such a way that only a starting motor influences the arrival in time. However facts like traffic jam or a broken tire can still mean that somebody gets late to work even though the motor starts. On the other hand, if the motor does not start, there is still ways to get to work in time, e.g. by taking the subway or calling a cab.

As an example, the network of Figure 5.3 will be used to calculate the probability of the battery being ok, a sufficient level of gas, that the motor starts and getting to work in time.

$$P(b \wedge e \wedge m \wedge a) = P(b) \cdot P(e) \cdot P(m|b \wedge e) \cdot P(a|m) = 0.95 \cdot 0.8 \cdot 0.99 \cdot 0.65 = 0.49$$

Assuming that everything is "normal", there is still only a chance of about 50% to get to work in time given the information of the network in Figure 5.3.

In fact, a bayesian network represents the complete description of a domain. In the network of Figure 5.3 this means that all possible events considering the domain of getting to work along with their probability measures are given. The probabilities of combinations of events can be calculated from the network as shown above. Formally, a bayesian network represents the *full joint probability distribution* that covers the probabilities of all possible combinations of events within a certain domain. The full joint probability distribution can also be given as a table which presents the probabilities of all combinations of events. However such tables generally become very large and causal dependencies of events can not be seen as simple as in a bayesian network.

A single entry in the full joint probability distribution can be calculated as

$$P(x_1 \wedge ... \wedge x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i))$$

where $parents(X_i)$ denotes the events of the parent nodes of x_i.

Apart from the calculation of single entries in the full joint probability distribution, bayesian networks are mainly used to compute the probabilities of events given the knowledge (or evidence) that certain events have occurred. Instead of computing the probability of $P(b \wedge e \wedge m \wedge a)$ one might also be interested in the probability of the battery being ok given the evidence that one has arrived to work in time, denoted as $P(b|a)$. This can be calculated by the formula

$$P(x|e) = \alpha P(x, e) = \alpha \sum_{y} P(x, e, y)$$

where e is the set of events that have occurred and y the set of non-evidence events being all possible combinations of events that have not occurred. α is a normalization factor that remains constant and ensures that $P(x|e) + P(\neg x|e) = 1$. This results in

$$P(b|a) = \alpha P(b, a) = \alpha \sum_{m} \sum_{e} P(b)P(e)P(m|b, e)P(a|m)$$

where the summation is calculated over *motor starts* = *true*, *motor starts* = *false*, *enough gas* = *true* and *enough gas* = *false* as the non-evidence

motor starts	enough gas	P(b)	P(e)	P(m|b,e)	P(a|m)
true	true	0.95	0.8	0.99	0.65
true	false	0.95	0.2	0.75	0.65
false	true	0.95	0.8	0.01	0.15
false	false	0.95	0.2	0.25	0.15

Table 5.2: Calculation of *P(battery ok = true | arrival in time = true)*

events. The evidence event is given as *a: arrival in time = true* and *b: battery ok = true* is the event for which the probabilities are calculated. This results in $P(b|a) = 0.986$ summing up the products $P(b)P(e)P(m|b,e)P(a|m)$ of each row of Table 5.2 and multiplying the sum with an according α[2]. This means that given the evidence that one has arrived at work in time, the probability of the battery being ok is 98.6 % which is higher than the according prior probability. Given the evidence of specific events, an agent making use of bayesian inference revises its degrees of belief by means of calculations in corresponding networks.

The approach for calculating probabilities from a bayesian network given some evidence is also called *inference by enumeration* as values are calculated by computing sums of products of conditional probabilities from a bayesian network. As given in [34], algorithms based on this approach are not suitable for practical applications because of their high computational complexity. Therefore several improvements for this approach have been proposed also including approximate approaches that do not deliver exact but sufficiently good numerical values from bayesian networks.

5.3.4 Summary

This section gave a brief introduction into probability theory and inference using bayesian networks as an approximate reasoning approach. The main idea behind probabilistic inference is an agent that expresses its degrees of belief of certain events by means of numerical values and in case of bayesian networks also defining the cause and effects of events by means of conditional probabilities. An agent can then make use of a network to calculate the probabilities of single events or a combination of events given some evidence. Besides bayesian networks there are also other approaches for approximate reasoning based on probabilities such as *evidence theory.* In evidence theory, there is a distinction between uncertainty and ignorance and the belief in a certain event is based on a probability mass function which allows for the definition of the belief in an event and the plausibility of an event.

[2] For *battery ok = true* the term $\sum_m \sum_e P(b)P(e)P(m|b,e)P(a|m)$ results in 0.58995 and for *battery ok = false* 0.00865 respectively. This results in $\alpha = 1.67$ ensuring $P(b|a) + P(\neg b|a) = 1$.

5.4 Rough Mereology

Compared to fuzzy set and probabilistic approaches to approximate reasoning, rough mereology is an approach that emerged lately and has first been published in the mid 1990s. Rough mereology is a foundation of *granular computing* [39] where information is seen in form of granules for which calculi are developed. In granular computing approaches such as rough set theory, fuzzy set theory, neural networks and rough mereology are combined in order to allow for new formalisms which are able to handle incomplete, imprecise and inexact information. Rough mereology is seen as the basis for future applications in fields such as computer aided design, adaptive control of complex systems, business re-engineering and cooperative problem solving [31].

Although the approach is mathematically formulated in numerous publications, each with slightly adapted notation depending on the context of the publication, there are only few works giving links to real world examples. Also, at the time of writing, no implemented approximate reasoning system based on rough mereology is known. [22] and [43] give examples of research projects with prototype implementations, however the concepts used therein are not based on approximate reasoning but only on rough mereology as a means to express the similarity of objects.

This work aims to give a concise overview on the numerous but mainly theoretical work on rough mereology and approximate reasoning schemes. The focus in this work will be put on the constituents of the approach and the way they work together allowing for finding approximate solutions to given problems.

5.4.1 Approximate synthesis of objects

Rough mereology as an approximate reasoning approach is based on the idea of a set of *reasoning agents* that perform approximate reasoning by assembling complex objects from simpler constructs. Note that the term *agent* here slightly differs from the definition of an intelligent agent in the introduction. In this section reasoning agents are constituents of a specific approximate reasoning scheme with properties being further on. In the following, the approach will first be presented in a non-formal way. Afterwards, the approach will be formalized and illustrated by an example.

Each agent in the rough mereological approach is equipped with an information system which contains information about the agent's objects. Each agent represents an information unit and the objects in each agent's information system are part of the agent's knowledge and are generally of a specific type. The information system is a general information system as introduced in the section about rough set theory, having a universe of discourse with the agent's objects, a set of attributes and according attribute values. Objects

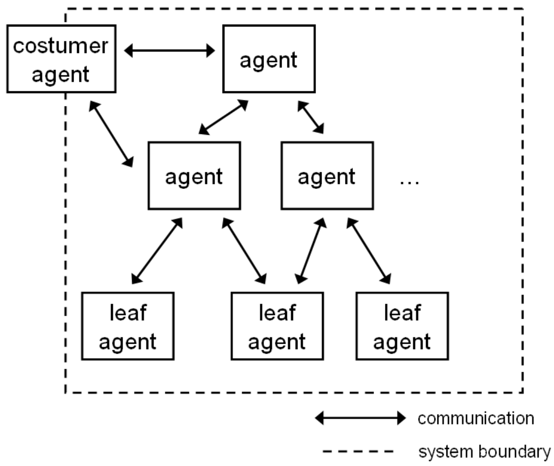


Figure 5.4: A set of reasoning agents

can for example represent artificial data or real world objects about which information in the form of attribute values is known. The agent's objects are conceptually separated in two different classes of objects - standard and non-standard objects. Standard objects, also referred to as *standards*, can be considered reference points for the reasoning approach. Standards can for example represent ideal data or ideal objects that in reality do not exist. As an example, a standard could be a non-real software product that completely fulfills certain requirements while non-standard objects represent real software products fulfilling some of the requirements only. The agent's objects are perceived by means of their attribute values as presented in the rough set approach. Two objects having the same attribute values are indiscernible and therefore considered equivalent.

Additionally, an agent is equipped with a set of operations. The set of operation enables the agent to construct objects in its information system from objects of other agents. Therefore each agent has a set of child agents that can send their (simpler) objects to their parents in order to construct more complex objects. Hence, the complete set of agents can be seen as a tree-like structure. Furthermore, there are agents in the complete set of agents that are not equipped with a set of operations. The objects of such agents are therefore not constructed from objects of other agents. Those objects are called *inventory* and the agents dealing with them are called *leaf agents*. Leaf agents therefore do not have any child agents.

In order to interact with the set of agents a *customer agent* is introduced. The costumer agent has the ability to communicate with agents being part and not being part of the set of agents. A customer agent can for example be a *human computer interface* by which humans can interact with the set

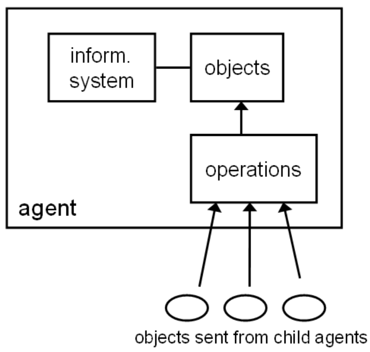


Figure 5.5: A single reasoning agent

of agents. Figures 5.4 and 5.5 show the concepts of the set of agents, objects and operations.

The next constituents of the rough mereological approach are *features* of standard objects and according *specifications*. For each standard, there is a set of features or properties which are offered by the standard. In case of software products, features of standards can for example be transactionality, platform independence or the support for a specific programming language. Features can be expressed either using a language that makes use of the attribute values of the agent's information system or by a specific language that is understandable by the agent. Such languages can for example be based on formulas of boolean propositions. As objects of non-leaf agents are constructed by assembling objects from child agents, features of standard objects also need to be linked to the features of standard objects at child agents. This linkage is also called *decomposition rule*, as specific or a set of features of a standard is decomposed into features of standards of child agents. Each decomposition rule also contains the according operation needed for assembling according objects.

Conceptually, specifications are making use of features. Specifications can be seen as artifacts expressing the requirement of a set of features. The main idea behind this concept is that the costumer agent issues specifications that have to be fulfilled by the set of agents. As an example, the costumer agent might issue a specification stating that a specific software product needs to be transactional. This specification is then accepted by the set of agents and decomposed into specifications that state requirements for child agents which e.g. represent single software components. The decomposition of specifications is based on the decomposition rules that link features of standards to features of standards of child agents. A successful decomposition of a specification results in a selection of standard objects and according operations along the set of agents that fulfill the decomposed

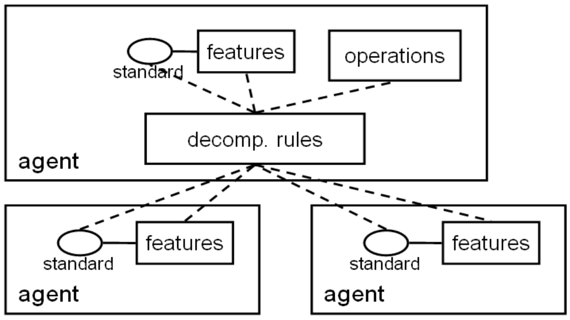


Figure 5.6: Decomposition rules linking features of standards and operations

specifications at all levels of the set of agents. Figure 5.6 illustrates the concept of decomposition rules.

Summing up, the approach as presented so far offers a reasoning service that can be stated as follows: Given a specification, the set of agents is able to deliver a result stating whether the specification can be fulfilled using standard objects as part of the agents universes. Note, that the reasoning service and according features and specifications only make use of standard objects and that non-standard objects are not considered.

In the next step, the reasoning service as stated above is extended in such a way that allows for issuing specifications with a fulfillment degree: Given a specification and an according fulfillment degree, the set of agents is able to deliver a result stating whether the specification can be fulfilled to the required degree using non-standard objects as part of the agents universes. The main idea behind this extension is based on the similarity of standard and non-standard objects. Given a specification that can be fulfilled by assembling standard objects, meaning that the specification is fulfilled to a degree 1, the same specification can also be fulfilled up to a certain degree by non-standard objects.

The measure of similarity is based on the *mereological distance* of objects. The mereological distance is calculated using a rough inclusion function that measures similarity as being a part of an object to a certain degree. The rough inclusion function is based on the attribute values of objects of an agent. Therefore the mereological distance can only be expressed for objects being part of the same universe of discourse. The mereological distance between a standard and non-standard object is then used as a fulfillment degree of according features. E.g., given a standard object representing a software component that has platform independence as a feature, this feature can also be fulfilled by a non-standard object to a certain degree that is determined by the mereological distance of both objects.

The similarity measure enables the costumer agent to issue specifications with fulfillment degrees which are first decomposed based on decomposition rules and standard objects as presented above. If the decomposition yields a successful result, the agents use the information about the mereological distances of standard and non-standard objects in their universes to deliver non-standard objects that fulfill the specification to the required degree. Note that it also might happen that high degrees of fulfillment can not be satisfied by non-standard objects. This can be considered as failure of fulfilling a specification of the costumer agent depending on whether standard objects correspond to real objects or not. Note also that even though two objects at non-leaf agents might have the required mereological distance, a final result can only be given if the specification and according fulfillment degrees are fulfilled at all levels of the decomposition.

After the presentation of the terminology and an overview of the approach, the next section formalizes the approach on a mathematical foundation which is followed by an illustrating example.

5.4.2 Formalizing the approach

In order to formalize the approximate reasoning approach based on rough mereology, the following section introduces the constituents of the reasoning scheme as defined in [31]. Some of the used terms have already been introduced in the previous section.

The approximate reasoning schema based on rough mereology is derived from the general reasoning scheme as given in [31] and based on a set of agents Ag where for each agent $ag \in Ag$ a label is defined as:

$$\begin{gathered} lab(ag) = (\mathcal{A}(ag), M(ag), L(ag), St(ag), Link(ag), O(ag), Unc\_rel(ag), \\ H(ag), Unc\_rule(ag), Dec\_rule(ag)) \end{gathered}$$

$lab(ag)$ is a set of sets that make up the reasoning abilities and tools of an agent. In the following the various parts of an agent's label will be explained in detail:

- $\mathcal{A}(ag) = (U(ag), A(ag))$ is the information system of an agent where $U(ag)$ is the universe of discourse of the agent including standard and non-standard objects and $A(ag)$ the according attribute set.
- $M(ag) = (U(ag), [0,1], \mu(ag))$ is a rough mereological model about the agent's universe of discourse where $\mu(ag)$ is a rough inclusion function over the universe of discourse. Details on rough inclusions will be given in following sections. For now, $M(ag)$ can be interpreted as the agent's ability to express the similarity of objects in $U(ag)$ based on mereological distances.

- $L(ag)$ is a language used to express the features or properties of objects in $U(ag)$. $L(ag)$ can for example be derived from attributes and according attribute values from the agent's information system $\mathcal{A}(ag)$.

- $St(ag) = \{st(ag)_1, ..., st(ag)_n\} \subset U(ag)$ is the set of standard objects of an agent ag. Accordingly the set $NonSt(ag) = \{x | x \notin St(ag)\}$ is the set of non-standard objects. Generally the set of non-standard objects can be considered the complement of $St(ag)$.

- $Link(ag)$ is a collection of strings in the form of $ag_1 ag_2 ... ag_k ag$. These strings are to be interpreted in such a way that agents $ag_1 ag_2 ... ag_k$ are children of ag and can send to ag their simpler objects. ag uses these objects for assembling more complex objects in $U(ag)$. A specific standard of ag however might only need some of the agents of the string $ag_1 ag_2 ... ag_k ag$. Hence, a decomposition of a standard object of ag into simpler objects can be considered as a more specific subset of the agents in the string. Generally, $Link(ag)$ can have more than one element which is regarded as the ability of an agent to synthesize objects using different child agents. Formally this is considered as the option of renegotiating the synthesis scheme.

- $O(ag)$ is the set of operations of an agent ag. Each operation $o \in O(ag)$ defines which standard from children of ag are needed in order to produce a standard at ag. One standard $st(ag)$ may be assembled by different operations of $O(ag)$. An operation can also be seen as a function or a mapping which has certain standards of children of ag as input and delivers $st(ag)$ as output.

- $Unc\_rel(ag)$ is the set of uncertainty relations. A specific uncertainty relation $unc\_rel_i \in Unc\_rel(ag)$ is defined as
  $(o_i, \rho_i,$
  $ag_1, st(ag_1), ag_2, st(ag_2), ..., ag_k, st(ag_k), ag, st(ag),$
  $\mu_0(ag_1), \mu_0(ag_2), ..., \mu_0(ag_k), \mu_0(ag_k))$

  where o_i is the operation which produces $st(ag)$ from $st(ag_1), st(ag_1), ..., st(ag_k)$ with $ag_1, ..., ag_k$ being children of ag.

  ρ_i is a set of tuples of non-standard objects and uncertainty bounds

  $$\rho_i((x_1, \epsilon_1), (x_2, \epsilon_2), ..., (x_k, \epsilon_k), (x, \epsilon))$$

  where $x_1 \in U(ag_1), ..., x_k \in U(ag_k)$ being non-standard objects from children of ag and $x \in U(ag)$ a non-standard object of ag. $\epsilon_1, \epsilon_2, ..., \epsilon_k \in [0, 1]$ are rough mereological distances (similarity measures, uncertainty degrees) calculated based on the rough inclusions
  $\mu_0(ag_1), \mu_0(ag_2), ..., \mu_0(ag_k), \mu_0(ag_k)$ as $\epsilon_k = \mu_0(x_k, st(ag_k))$ for the

children of ag and $\epsilon = \mu_0(x, st(ag))$ at ag. $\epsilon_k = \mu_0(x_k, st(ag_k))$ therefore expresses the similarity of the non-standard object x_k and the standard object $st(ag_k)$ at agent ag_k.

An uncertainty relation unc_rel_i relates the distances of non-standard objects to standard objects at children of ag to a standard and a non-standard object of ag using a specific operation o_i. Uncertainty relations are needed for deriving uncertainty rules which are defined next.

- $Unc\_rule(ag)$ is the set of uncertainty rules. A specific uncertainty rule $unc\_rule_j \in Unc\_rule(ag)$ is defined as
  $(o_j, f_j,$
  $\mu_0(ag_1), st(ag_1), \mu_0(ag_2), st(ag_2), ..., \mu_0(ag_k), st(ag_k), \mu_0(ag), st(ag))$

  where $ag_1, ..., ag_k$ are children of ag and f_j is a function (*rough mereological connective*) with the property:

  if objects $x_1 \in U(ag_1), ..., x_k \in U(ag_k)$ satisfy the conditions $\mu_0(x_i, st(ag_i)) \geq \epsilon(ag_i)$ for $i = 1, 2, ..., k$ then $\mu_0(x, st(ag)) \geq f_j(\epsilon(ag_1), \epsilon(ag_2), ..., \epsilon(ag_k))$

  where $x = o_j(x_1, x_2, ..., x_k)$ is the non-standard object at ag produced from non-standard objects of children of ag using operation o_j.

  Hence, a specific uncertainty rule unc_rule_j states to which degree non-standard objects at children of ag need to be similar to according standards, so that $st(ag)$ and x are sufficiently similar to the degree $\epsilon = \mu_0(x, st(ag))$.

- $H(ag)$ is the agent's strategy how to derive uncertainty rules from uncertainty relations. $H(ag)$ can for example be an algorithm as defined in [31] which will be applied in an example later in this chapter. Figure 5.7 illustrates the concept of extracting uncertainty relations.

- $Dec\_rule(ag)$ is the set of decomposition rules. A specific decomposition rule $dec\_rule_j \in Dec\_rule(ag)$ is defined as

  $$(o_j, \Phi(ag_1), st(ag_1), \Phi(ag_2), st(ag_2), ..., \Phi(ag_k), st(ag_k), \Phi(ag), st(ag)$$

  where $\Phi(ag_1) \in L(ag_1), ..., \Phi(ag_k) \in L(ag_k), \Phi(ag) \in L(ag)$. Each $\Phi(ag_k)$ therefore represents a specification (requirement, set of properties / features) which a certain standard $st(ag_k)$ at ag_k fulfills. Again, $ag, ag_1, ..., ag_k$ are children of ag. Each $\Phi(ag_k)$ is expressed in the language $L(ag_k)$ understandable for the agent ag_k for which attribute values from the agent's information system $\mathcal{A}(ag_k)$ can be used for example. At the agent ag accordingly $st(ag)$ needs to fulfill $\Phi(ag)$ where $st(ag)$ is being produced using operation o_j so that $st(ag) = o_j(st(ag_1), ..., st(ag_1))$

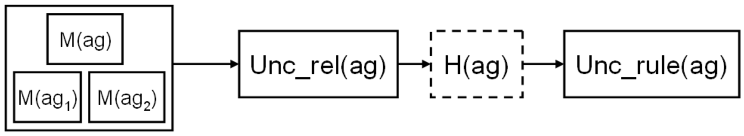


Figure 5.7: Extracting uncertainty relations and uncertainty rules from information systems

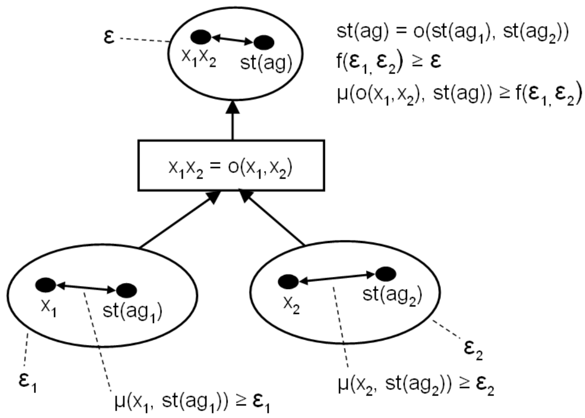


Figure 5.8: Applying uncertainty rules during object synthesis

Additionally, there are agents which contain objects that can not be decomposed any further. Those agents are called *leaf agents* denoted $Leaf(Ag)$. The union of all universes of leaf agents is called inventory $Inv = \cup\{U(ag) : ag \in Leaf(Ag)\}$ and makes up the basic ingredients on which the reasoning is based. The inventory Inv can be considered as the set of resources available for producing complex objects. The inventory can also have different abstraction levels at different leaf agents. E.g., in case of a real world production, objects at certain leaf agents could correspond to raw material while other leaf agents deliver pre-produced parts.

The label of an leaf agent can then be defined as:

$$lab\_leaf(ag) = (\mathcal{A}(ag), M(ag), L(ag), St(ag))$$

In this section, the constituents of the reasoning scheme have been introduced in a formal way. As reasoning scheme can be considered as a static skeleton on which the reasoning is organized, the next section will describe the different phases of the reasoning approach. Describing the different phases allows for a better understanding about how the constituents of the scheme are linked to each other.

5.4.3 Phases in the Approximate Reasoning Scheme

Design phase

The first phase in rough mereology for approximate reasoning is the design process of the reasoning scheme. One part of the design phase is the definition of agents and their according information systems. This includes the definition of the attribute set A as well as the objects in the agent's universe. Within the universe, standard objects need to be marked and non-standard objects can be considered as training data for the following learning phase. Along with the information system the rough mereological model of the agent is also defined which includes the rough inclusion function the agent uses for calculating closeness (uncertainty) measures between standard and non-standard objects. For expressing requirements on objects to be assembled by the agent, the agent's language $L(ag)$ is defined as well. As already stated, $L(ag)$ can for example be based on the agent's attribute set.

The design phase also includes the definition of the agent's operations for assembling its objects from simpler constructs sent by its children. The agent's children can be explicitly defined as $Link(ag)$ or $Link(ag)$ can be derived implicitly from the agent's operations and agents used therein. In order to decompose specifications, a set of decomposition rules is defined, which link features of standards of the agent to corresponding standards at the agent's children using one of the agent's operations. Another part of the design phase is the definition of a strategy $H(ag)$ for calculating uncertainty rules from uncertainty relations in the learning phase.

Summing up, the design phase results in the definition of the following constituents of the above defined scheme for each non-leaf agent

$$\mathcal{A}(ag), M(ag), L(ag), St(ag), Link(ag), O(ag), H(ag), Dec\_rule(ag)$$

and

$$\mathcal{A}(ag), M(ag), L(ag), St(ag)$$

for each leaf agent as a part of the agent's label $lab(ag)$.

Generally, the reasoning scheme is assumed to be designed by an expert or knowledge engineer, however there are approaches for deriving some of the scheme's constituents from background knowledge and data [35].

Learning Phase

The first step of the learning phase is setting up uncertainty relations. For each standard and each operation which assembles the standard, closeness measures concerning standards used for the assembly are related to closeness measures of the assembled standard. This is done by calculating rough mereological closeness measures using the rough inclusion function defined in the design phase for each standard and according non-standard objects at the agent and the agent's children.

The strategy $H(ag)$ defined in the design phase is then applied to the calculated uncertainty relations. This results in a set of uncertainty rules which define the minimum closeness measures to be satisfied at children of ag to assemble an object at ag satisfying properties or features up to a certain degree.

The learning phase therefore delivers the missing parts of the agent's label for non-leaf agents:

$$Unc\_rel(ag), Unc\_rule(ag)$$

After the calculation of uncertainty rules, uncertainty relations are not needed anymore as they only make up the input data for the strategy $H(ag)$.

The success of the learning phase is determined by a train and test scenario. As already mentioned, non-standard objects can be considered training data for the scheme as they are used for calculating uncertainty rules. For testing the scheme and its learned rules, the customer agent issues specifications accompanied with an according fulfillment degree, for which it has the knowledge to assess the objects synthesized by the scheme. E.g., issuing a requirement with a fulfillment degree of 0.8, the costumer agent can decide whether the assembled objects delivered by the scheme really fulfills the specification with a degree of at least 0.8. The success of the learning phase and implicitly also the success of the designed scheme, can the be determined by the costumer agent by setting thresholds e.g. based on frequency count, meaning that a certain percentage of the tested objects need to be successful.

If the scheme is not deemed successful, possible ways for improvement are redesigning (parts of) the scheme or improving the quality of the training data.

Reasoning phase

Finally, after a successful learning phase, the scheme is used for assembling objects based on a specification and a fulfillment degree. The procedure described in this section is of course also used during the testing of the scheme in the learning phase.

The reasoning phase is divided into two distinct stages - the decomposition of specifications which is performed top-down in the scheme and the synthesis of objects which is a bottom-up process starting from the leaf agents.

Once the customer agent has issued a specification accompanied with a corresponding fulfillment degree to an agent, the agent checks its information system whether it can fulfill the specification exactly, meaning that there is a standard in the agents universe that fulfills the issued specification completely (in a degree of 1). Theoretically the specification can be accepted by multiple agents if it is issued in a language understandable for the according agents. This can result in more than one solution of the issued specification. For the sake of simplicity it is assumed that the costumer agent can only communicate with a single agent, called *root agent*. If there is at least one standard in the root agent's universe fulfilling the issued specification exactly, the root agent checks whether for those standards and the issued specification there are any decomposition rules defined that allow for decomposing (formulating) the issued specification into a language understandable for the root agent's children. If this step terminates successfully, the set of uncertainty rules is checked whether there exist uncertainty measures satisfying the fulfillment degree of the customer and according measures for the children of the root agent. If there are uncertainty measures satisfying the fulfillment degree of the customer, meaning that there are non-standard objects having a satisfactory mereological closeness to standard objects, then the according uncertainty measures will be used for requesting corresponding objects from the root agent's children. So the root agent will send a message to its children consisting of a part of the decomposed specification derived from the selected decomposition rule as well as the uncertainty measure from the selected uncertainty rule.

The above described procedure is executed at each agent within the decomposition scheme that is determined by the agents' decomposition rules. Whenever one of the steps does not terminate successfully, e.g. there is no decomposition rule enabling the decomposition of the specification or there are no uncertainty measures with the needed fulfillment degree, the procedure halts and a failure message is propagated to the root agent which tells the costumer that no objects satisfying the requirement can be assembled. If there are alternative operations and decomposition rules for certain objects along the decomposition of a specification, these alternatives can be chosen before sending failure messages.

The complete process terminates successfully when the overall specification can be decomposed to specifications for leaf agents and each leaf agent can deliver an inventory object with the required rough mereological closeness. Figure 5.9 illustrates the process in a graphical way.

At this point, the bottom-up process for synthesizing objects starts. Each leaf agent selects non-standard objects satisfying the specification up to the required fulfillment degree and sends those objects along with the closeness (uncertainty) measure to its parent which assembles more complex constructs using the objects sent by its children. Whether a leaf agent just sends a single object or a set of objects is not defined formally. By sending sets of objects, alternative solutions with different fulfillment degrees can be assembled.

Each non-leaf agent applies the operation selected during the require-

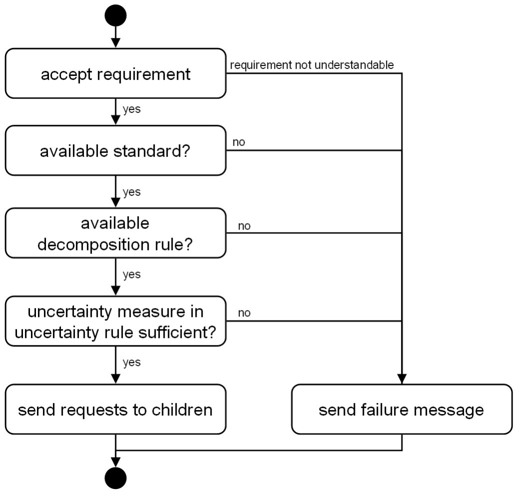


Figure 5.9: Requirement decomposition process at non-leaf agents

ment decomposition process to the objects sent by its children. Based on the uncertainty measures sent along with the objects coming from the children, the rough mereological connective f of the according uncertainty rule is applied in order to calculate the uncertainty measure of the synthesized object from the respective standard. In case that uncertainty rules have been derived by the algorithm defined in [31] where uncertainty rules represent an approximation of f, uncertainty rules can directly be used for calculating the uncertainty measure of the produced object.

The reasoning phase ends when the synthesis process arrives at the root agent which delivers the final object to the costumer agent.

5.4.4 Rough inclusions

Rough inclusion functions have already been mentioned as functions used for calculating the similarity or closeness of objects in the information system of an agent. This section will define the basic properties of rough inclusions and give a link to the underlying model of mereology.

Rough inclusions are based on the mereology of Lésniewski [18]. Detailed discussions on Lésniewski's mereology and its extension to rough mereology can be found in [30], [31], [32], [33] and [35]. In this work only the basic notions necessary for rough inclusions will be presented.

Mereology is the theory of parts and wholes and provides the foundations of part-whole reasoning. Lésniewski has been one of the first authors of mereology whose work has been extended by several authors introducing additional axioms.

The basic notion of mereology is that of a *part*. Lésniewski introduced the relation of *being a part*, denoted π and read "being a part of", defined as

P(1) : $x\pi y \wedge y\pi z \rightarrow x\pi z$
P(2) : $x\pi y \rightarrow \neg(y\pi x)$

P(1) states the transitivity property of the part relation, meaning that if x *is a part of* y and y *is a part of* z then x *is a part of* z as well. P(2) states antisymmetry meaning that if x *is a part of* y then it is not true that y *is a part of* x. From this follows that x can not be a part of itself, $x\pi x$ for no x.

Furthermore, Lésniewski introduced the relation of *being an element*, denoted *el* and read "being an element of", defined as:

El : $xely \leftrightarrow x\pi y \vee x = y$

From El, P(1) and P(2) the following properties of the element relation are obtained:
El(1) : $xelx$
El(2) : $xely \wedge yelx \rightarrow x = y$
El(3) : $xely \wedge yelz \rightarrow xelz$

These basic relations of *being a part* and *being an element* allow for the definition of requirements for rough inclusions which introduce the notion of *being a part in a degree.* A rough inclusion μ is defined on an universe U with values in the interval [0,1] satisfying the following requirements:

RI(1) : $\mu(x, x) = 1$ for each $x \in U$
RI(2) : $\mu(x, y) = 1 \leftrightarrow xely$ for each pair $x \in U$ and $y \in U$
RI(3) : $\mu(x, y) = 1 \rightarrow (\mu(z, x) = r \rightarrow \mu(z, y) = r)$ for each $z \in U$ and each $r \in [0, 1]$
RI(3) : if $\mu(x, y) = r$ and $s < r$ then $\mu(x, y) = s$

A simple example of a rough inclusion over a universe U can be defined as follows where $\subset$ is the relation π of being a part and $\subseteq$ the relation *el* of *being an element*:

$$\mu(X, Y) = \frac{|X \cap Y|}{|X|}$$

with $X \subseteq U$ and $Y \subseteq U$ and $|A|$ denoting the cardinality of A.

In rough mereology as an approximate reasoning scheme where rough inclusion values need to be calculated from attribute values of objects in their according information systems, a rough inclusion is defined as follows:

For a partition $P = \{A_1, ..., A_k\}$ of the attribute set A into non-empty sets $A_1, ..., A_k$ and a set of weights $W = \{w_1, ..., w_k\}$ where all weights sum up to 1 and $i = 1, 2, .., k$:

U	web	asyn	state	gnu
$st_1(ag_{gui}) = x_1$	1	0	0	1
$st_2(ag_{gui}) = x_2$	0	0	1	0
x_3	0	1	0	1
x_4	1	1	0	0

Table 5.3: Information system of the GUI agent ag_{gui}

$$\mu(x, y) = \sum_{i=1}^{k} w_i \cdot \frac{|IND_i(x, y)|}{|A_i|}$$

where the indiscernibility relation is defined as $IND_i(x, y) = \{a \in A_i : a(x) = a(y)\}$. The indiscernibility relation introduced in rough set theory therefore is the foundation of rough inclusion functions defined for information systems.

For the example following in the next section, a simplified version of the above defined rough inclusion function is given. For a non-partitioned attribute set A and a single weight $w = 1$ the rough inclusion is defined as

$$\mu_0(x, y) = \frac{|IND(x, y)|}{|A|}$$

with the indiscernibility relation $IND(x, y) = \{a \in A : a(x) = a(y)\}$.

Further ideas on rough inclusions based on information systems have been studied in [33] among them rough inclusions based on functions known from fuzzy set theory like the Lukasiewicz and Menger rough inclusions. In [33] it is also shown that symmetry and transitivity generally do not hold for rough inclusions.

5.4.5 Example

In order to illustrate the reasoning scheme and its according phases, this section presents a simple example making use of rough mereology as an approximate reasoning approach.

The set of agents $Ag = \{ag_{gui}, ag_{srv}, ag_{sys}\}$ consists of three agents of which ag_{gui} delivers graphical user interface (GUI) software components, ag_{srv} delivers server software components and ag_{sys} assembles them to final software systems. The set of leaf agents is defined as $Leaf(Ag) = \{ag_{gui}, ag_{srv}\}$. Accordingly the inventory is made up of the objects in the information systems of agents ag_{gui} and ag_{srv}. Tables 5.3 and 5.4 show the information systems of the according agents. The sets of standard objects at leaf agents are given as $St(ag_{gui}) = \{x_1, x_2\}$ and $St(ag_{srv}) = \{y_1, y_3\}$, respectively.

U	soa	trans	multi	high	cra
$st_1(ag_{srv}) = y_1$	1	1	1	1	1
y_2	1	0	0	0	0
$st_2(ag_{srv}) = y_3$	0	0	1	0	1
y_4	1	1	1	0	0

Table 5.4: Information system of the server agent ag_{srv}

For the system agent which assembles software components into a final system, the information table is given in Table 5.5. The label $lab(ag_{sys})$ of the system agent defines a single operation $o_o \in O(ag_{sys})$ which takes objects from the GUI and server agent and produces a final system. Formally the operation is given by $xy = o_o(x, y)$ where $xy \in U(ag_{sys})$, and $x \in U(ag_{gui})$ and $y \in U(ag_{srv})$ respectively. Accordingly the set of standards at ag_{sys} is given as $St(ag_{sys}) = \{x_1y_1, x_1y_3, x_2y_1, x_2y_3\}$. $lab(ag_{sys})$ of the system agent also defines a set of decomposition rules given in Table 5.6 and $H(ag_{sys})$ as an algorithm. Note, that the decomposition rules as given in Table 5.6 are defined for operation o_o and are not derived from the attribute values of the according object. Also, it is generally not necessary to give specifications for standards of leaf agents, as leaf agents do not decompose specifications any further.

After the scheme has been set up as described above, the learning process starts by extracting uncertainty relations from information systems and obtaining uncertainty rules by applying the strategy $H(ag_{sys})$. In order to extract uncertainty relations from information systems, for each agent rough inclusion measures are calculated. The scheme has been designed in such a way that the rough mereological models $M(ag_{gui})$, $M(ag_{srv})$ and $M(ag_{sys})$ all provide the same rough inclusion function μ_0 as given in section 5.4.4. The results of applying μ_0 to the informations systems of ag_{gui} and ag_{srv} are given in Tables 5.7 and 5.8. For readability reasons the rough inclusion measures of ag_{sys} can be found in the Appendix in Tables B.2 and B.3.

Next, for each standard at agent ag_{sys}, uncertainty relations are being extracted. Table 5.9 shows the uncertainty relations for operation o_o, $st_1(ag_{gui}) = x_1$, $st_1(ag_{srv}) = y_1$ and $st_1(ag_{sys}) = x_1y_1$. Table 5.9 illustrates the intended meaning of uncertainty relations: The rough inclusion values of standards at child agents ag_{gui} and ag_{srv} are being related to rough inclusion values to standards at parent agent ag_{sys}. As rough inclusion values are interpreted as uncertainty measures, the uncertainty of child agents is related to the uncertainty at the according parent agent.

The next step is applying the strategy $H(ag_{sys})$ which has been defined in $lab(ag_{sys})$ as the algorithm introduced in [31]. The algorithm extracts for each value of $\epsilon = \mu(xy, x_1y_1)$ at ag_{sys} those values of $\epsilon_1 = \mu(x, x_1)$ and $\epsilon_2 = \mu(y, y_1)$ at ag_{gui} and ag_{srv}, so that ϵ is preserved with minimum values

U	scale	reuse	plat	mon	lin
$st_1(ag_{sys}) = x_1y_1$	1	0	1	0	1
x_1y_2	0	1	0	0	1
$st_2(ag_{sys}) = x_1y_3$	0	1	0	0	1
x_1y_4	0	1	0	0	1
$st_3(ag_{sys}) = x_2y_1$	1	0	0	0	1
x_2y_2	1	0	0	0	1
$st_4(ag_{sys}) = x_2y_3$	0	1	0	1	0
x_2y_4	1	0	0	1	1
x_3y_1	1	0	0	0	0
x_3y_2	0	1	0	0	0
x_3y_3	0	1	0	0	0
x_3y_4	1	0	0	0	0
x_4y_1	1	0	1	0	0
x_4y_2	0	1	1	0	0
x_4y_3	0	1	0	0	0
x_4y_4	1	0	1	0	1

Table 5.5: Information system of the system agent ag_{sys}

rule	$\Phi(ag_{sys}), st(ag_{sys})$	$\Phi(ag_{gui}), st(ag_{gui})$	$\Phi(ag_{srv}), st(ag_{srv})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$scale \wedge plat, x_1y_1$	$jav \vee web, x_1$	$j2ee, y_1$
$i+1$	$te \wedge de \wedge eg, x_1y_1$	jam, x_1	fra, y_1
$i+2$	$reuse \wedge mon, x_2y_3$	$comp \wedge tom, x_2$	$ham \wedge lis, y_3$
$i+3$	$bli \wedge soo, x_2y_1$	lan, x_2	ace, y_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 5.6: Subset of decomposition rules $Dec\_rule(ag_{sys})$ of system agent for operation o_o

	x_1	x_2	x_3	x_4
x_1	1	0.25	0.5	0.5
x_2	0.25	1	0.25	0
x_3	0.5	0.25	1	0.5
x_4	0.5	0	0.5	1

Table 5.7: Rough inclusion values of the GUI agent ag_{gui}

	y_1	y_2	y_3	y_4
y_1	1	0.2	0.6	0.6
y_2	0.2	1	0.4	0.6
y_3	0.6	0.4	1	0.4
y_4	0.6	0.6	0.4	1

Table 5.8: Rough inclusion values of the server agent ag_{srv}

of ϵ_1 and ϵ_2 [3] . Details on the algorithm are given in [31].

Using Table 5.10 the algorithm will be explained by an example for $\epsilon = 1$. Table 5.10 therefore contains only relevant objects from Table 5.9. For $\epsilon = 1$, there are three objects at ag_{sys}, obviously the standard x_1y_1 itself, as well as the assembled objects x_1y_4 and x_4y_4. Comparing objects x_1y_1 and x_1y_4 one can see that $\epsilon_1 = 1$ and $\epsilon_2 = 0.6$ is sufficient for $\epsilon = 1$, so that thresholds for $\epsilon = 1$ are set to $\epsilon_1 = 1$ and $\epsilon_2 = 0.6$. Comparing objects x_1y_4 and x_4y_4 it can be concluded that $\epsilon_1 = 0.5$ may also be sufficient for $\epsilon = 1$. However, there exists another object x_3y_4 with $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.6$ which yields only $\epsilon = 0.6$. This means that $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.6$ are not sufficient for $\epsilon = 1$. So the result of the algorithm is the vector $(\epsilon_1 = 1, \epsilon_2 = 0.6, \epsilon = 1)$ which expresses the minimum uncertainty measures that have to be satisfied by objects at agent ag_{gui} and ag_{srv}, so that the produced object at ag_{sys} has an uncertainty measure of 1. The final result of uncertainty rules for standard x_1y_1 is given in Table 5.11. Note that in some cases, certain values of ϵ may not be fulfilled. E.g., applying $H(ag)$ may result in fewer values for ϵ than given in the corresponding uncertainty relations[4].

The results given in Table 5.11 can be interpreted as inference rules. As an example, row r_3 from Table 5.11 can be formulated as an inference rule as follows:

If an object x_i at ag_{gui} has a rough inclusion measure with standard x_1

[3] Note that in [32] this step is not presented which reduces the traceability of the approach presented therein.

[4] In [31], e.g., the set of values of ϵ originating from uncertainty relations is $\{1, 0.75, 0.5, 0.25, 0\}$ whereas set of values of ϵ in according uncertainty rules is $\{1, 0.5, 0.25, 0\}$. Additionally, note that in Table 5 of [31] ϵ for x_2y_1 needs to be 0.75 instead of 0.5 which can simply be seen from Table 4 in the same publication. This typing mistake however does not affect the final result.

x	$\epsilon_1 = \mu(x, x_1)$	y	$\epsilon_2 = \mu(y, y_1)$	$\epsilon = \mu(xy, x_1y_1)$
x_1	1	y_1	1	1
x_1	1	y_2	0.2	0.6
x_1	1	y_3	0.4	0.4
x_1	1	y_4	0.6	1
x_2	0.25	y_1	1	0.8
x_2	0.25	y_2	0.2	0.4
x_2	0.25	y_3	0.4	0
x_2	0.25	y_4	0.6	0.6
x_3	0.5	y_1	1	0.6
x_3	0.5	y_2	0.2	0.2
x_3	0.5	y_3	0.4	0.2
x_3	0.5	y_4	0.6	0.6
x_4	0.5	y_1	1	0.8
x_4	0.5	y_2	0.2	0.4
x_4	0.5	y_3	0.4	0.1
x_4	0.5	y_4	0.6	1

Table 5.9: Uncertainty relations for operation o_o, $st_1(ag_{gui}) = x_1$, $st_1(ag_{srv}) = y_1$ and $st_1(ag_{sys}) = x_1y_1$.

x	$\epsilon_1 = \mu(x, x_1)$	y	$\epsilon_2 = \mu(y, y_1)$	$\epsilon = \mu(xy, x_1y_1)$
x_1	1	y_1	1	1
x_1	1	y_4	0.6	1
x_4	0.5	y_4	0.6	1
x_3	0.5	y_4	0.6	0.6

Table 5.10: Applying $H(ag_{sys})$ for extracting an uncertainty rule for $\epsilon = 1$

row	ϵ_1	ϵ_2	ϵ
r_1	1	0.6	1
r_2	0.25	1	0.8
r_3	0.25	0.6	0.6
r_4	1	0.4	0.4
r_5	0.5	0.2	0.2
r_6	0.25	0.4	0

Table 5.11: Uncertainty rules for operation o_o, $st_1(ag_{gui}) = x_1$, $st_1(ag_{srv}) = y_1$ and $st_1(ag_{sys}) = x_1y_1$.

of at least 0.25 and an object y_i at ag_{srv} has a rough inclusion measure with standard y_1 of at least 0.6 then the object $x_i y_i$ assembled at ag_{sys} using operation o_o has a rough inclusion measure with standard $x_1 y_1$ of at least 0.6.

This can also be regarded in such a way that object $x_i y_i$ satisfies the properties or features of standard $x_1 y_1$ with a degree of at least 0.6.

Note that the values presented in Table 5.11 are actually input and output values of the rough mereological connective function f defined in the according uncertainty rule. The values in Table 5.11 are therefore considered as an approximation of f as $\epsilon = f(\epsilon_1, \epsilon_2)$ which is highly non-linear. In some rows of Table 5.11 the t-norm *min* or the t-conorm *max* can be used for f, however there is no generic function valid for all rows of Table 5.11 even in this simple example.

Note also that uncertainty rules need to be extracted on a per standard and per operation basis. With a single operation $o_o \in O(ag_{sys})$, similar rules for standards $x_1 y_3$, $x_2 y_1$ and $x_2 y_3$ at ag_{sys} need to be extracted during the learning phase. The example only presented the calculation for the standard $x_1 y_1$.

Reasoning phase

What has been presented in the example so far has covered the design and learning phase of rough mereology as an approximate reasoning approach. For an example of the reasoning phase, assume that the costumer agent issues a specification $\Phi = plat \wedge scale$ and a fulfillment degree $\epsilon = 0.4$. The server agent ag_{sys} accepts the specification and using the decomposition rules of Table 5.6 finds standards at ag_{gui} and ag_{srv} that can be used to assemble standard $x_1 y_1$ by applying operation o_o. Note, that the specification can also be decomposed for agents ag_{gui} and ag_{srv} in case they would not be leaf agents. The top-down process of the reasoning phase ends by selecting standards x_1 and y_1 at the corresponding agents and the operation o_o at ag_{sys}. Next, the required fulfillment degrees for objects at ag_{gui} and ag_{srv} are derived from Table 5.11.

For $\epsilon = 0.4$ it can be seen from row r_4 that the closeness measures for objects regarding x_1 at ag_{gui} need to have a minimum value of $\epsilon_1 = 1$ and $\epsilon_2 = 0.4$ for objects regarding y_1 at ag_{srv}, respectively. Using those values in Tables 5.7 and 5.8 showing the rough inclusion values of agent ag_{gui} and ag_{srv}, selected objects fulfilling the closeness measures are x_1 at ag_{gui} and y_1, y_3 and y_4 at ag_{srv}. Applying operation o_o to the selected objects, ag_{sys} assembles objects $x_1 y_1$, $x_1 y_3$ and $x_1 y_4$ and delivers them to the costumer agent. Using this approach, the delivered objects satisfy the specification $\Phi = plat \wedge scale$ with a fulfillment degree of at least $\epsilon = 0.4$.

Note that the set of assembled objects includes the standard $x_1 y_1$ at ag_{sys} as well as objects that are using standard x_1 of ag_{gui}. If the scheme had been designed in such a way that only non-standard objects are to be delivered, the given specification could not be satisfied with $\epsilon = 0.4$. Reducing ϵ to 0.2, the set of delivered objects at ag_{sys} would also include objects x_3y_2 and x_3y_4 which are assembled using non-standard objects. This example also shows that the uncertainty rules are to be considered as the minimum requirement to be fulfilled. Given a specification and $\epsilon = 0.2$, the scheme delivers all objects satisfying the specification with at least 0.2 including objects with $\epsilon = 1$. In the case more than one object is delivered to the customer agent, the customer agent may select one of the assembled objects based on external criteria.

5.4.6 Summary

The previous sections presented rough mereology as an approximate reasoning approach. The approach was first introduced in a non-formal way in order to get a first overview about the approach and to get familiar with the terminology. Afterwards, the approach was formalized and all of the constituents of the reasoning approach were defined. Furthermore, rough inclusions have been introduced as a means to express the closeness of objects. Those closeness measures are used for expressing uncertainty along the reasoning scheme. Finally an example has been given that presented the approach on a step by step basis.

As with other reasoning approaches presented in this work, the success of rough mereology depends on the design of the scheme as well as the selection of standard and non-standard objects. The underlying semantics of this approach is that of being a part in a degree. This has been extended to objects in information systems by defining rough inclusion functions which express the similarity of objects within the same universe. The similarity or closeness of objects is then used in a synthesis process which is also a part of the reasoning semantics.

5.5 Summary and Discussion

In this chapter three different approaches to approximate reasoning have been presented. Each of these approaches can be used by an intelligent agent in order to cope with uncertain or inexact information. However, each approach covers a different sort of approximate reasoning which in the cases of reasoning with fuzzy sets and bayesian reasoning has been motivated by the way humans reason under uncertainty. The motivation of rough mereology does not come from human reasoning but extending the notion of being a part to being a part in a degree and setting up a reasoning scheme with various constituents around it.

Fuzzy set theory allows for dealing with the graduality of the human language[5] which is also often referred to as vagueness. The vagueness is handled by means of fuzzy data, e.g., information to reason on, and fuzzy rules representing fuzzy knowledge about a certain domain. Compared to probability theory, the vagueness as a specific form of uncertainty does not change during the reasoning process. In fuzzy set theory vagueness is seen as a property of the domain or the world in general, independent from what an agent knows or beliefs. In bayesian reasoning however, the agent's beliefs about the world are expressed. This also results in a change of the degree of belief in a specific event whenever new evidence is known. For example, knowing that one arrived at work in time, increases the agent's belief in the battery being ok, as discussed earlier in this chapter. The dependencies of events are expressed by a network which covers some of the agents knowledge about the world. While graduality is the main aspect for fuzzy set theory, in bayesian reasoning or probability theory in general, laziness and ignorance are covered by degrees of belief. Expressing the agent's belief in a certain event, given some evidence or not, implicitly expresses the uncertainty of the belief coming from ignorance and laziness. In a bayesian network, causes for events that are not denoted by a node in the network are included in according probability measures representing the ignorance and laziness about such causes.

In rough mereology as another approximate reasoning approach, the reasoning process is divided into two stages. In the top-down process, a specification is decomposed into specifications for reasoning agents at lower levels in the scheme. The decomposition with according decomposition rules can be performed by classical logic or boolean reasoning [36]. In the bottom-up process, solutions are constructed such that they satisfy the specification up to a given degree. Compared to graduality in fuzzy sets or ignorance, laziness and degrees of belief in bayesian reasoning, in rough mereology similarities of objects are expressed by the notion of being a part in a degree. Along with the similarity measures which express the uncertainty of an agent towards an object's properties regarding a specific standard, a construction process delivers a final solution. The final solution is made up of constructions at different levels in the scheme while preserving an overall uncertainty bound during the construction process.

In all three approaches, knowledge representation is realized in a different manner. In reasoning with fuzzy sets, rules are used to represent the knowledge about the world. The antecedent of a rule consists of fuzzy sets representing facts about the domain that can be combined by logical connectives. Accordingly, the consequent of a rule represents the fuzzy result obtained from the given the antecedent of the rule. Several fuzzy rules are combined to fuzzy rule bases which represent the agent's knowledge about the world or a given domain. Once set up, a fuzzy rule base can be used to

[5] For a discussion on the interpretation of fuzzy set as possibility distribution see [10]

reason on input information given as crisp or fuzzy data. A closer look at a fuzzy rule base shows that knowledge is represented in two ways. Each rule states the knowledge about the relations of a specific rule's antecedent and consequent. Furthermore, knowledge is also represented in the definition of the fuzzy membership functions used in the antecedents and consequents of the rule. For example, representing the linguistic label *fast* by means of a fuzzy membership function over the velocity of an object, is also a form of knowledge representation.

In bayesian reasoning, knowledge is represented in two ways as well. The structure or topology of a bayesian network represents the causes and effects of events. The network therefore represents the agent's knowledge about dependencies and influences of events. It is also assumed that the nodes in the network are the most important causes and effects of the domain in consideration and that less important events are reflected in the probability measures only. Apart from the topology, the probability measures are also a form of knowledge representation. By defining the degrees of belief the knowledge or experience of an agent about events is given as a numerical measure. While prior probabilities reflect the agent's degrees of belief without any other information given, the conditional probability tables strongly depend on the topology of the network. Yet, the numerical measures express the importance of a specific event as influences on other events, especially when a node has more than one parent node.

In rough mereology several constituents of the approach participate in the knowledge representation. Considering the reasoning scheme only, each reasoning agent is a knowledge representation unit itself. For a single reasoning agent, the set of operations is the agent's knowledge how to construct objects based on objects sent by other agents. Accordingly, decomposition rules represent the agent's knowledge of decomposing specifications for its children. Furthermore, the attribute set of the agent's information system can be regarded as a way of perceiving the objects in the agent's universe, which also is a form of knowledge. Regarding all agents together, the possibility of communication between specific agents can also be seen as a kind of knowledge representation. Apart from the reasoning scheme, the objects in the agent's unverse are considered training data from which knowledge is extracted by means of uncertainty relations and uncertainty rules. In this context, the agent's rough inclusion function for computing similarity measures as well as the selection of standards and training data have a great impact on the extracted knowledge. Especially standards as predefined objects for which features and properties are known, constitute some of the domain knowledge.

Applying one of the approaches for implementing an intelligent agent or a practical reasoning system, it first needs to be assured that the chosen approach suits the domain and data to be reasoned about. An important question considering the three approaches presented in this work is whether the domain is predominantly gradual in case of fuzzy sets, whether uncertainty needs to represented by degrees of belief or whether similarity towards standards and a construction process meets the reasoning requirements. Also hybridization of formalisms is possible as presented in [13] where fuzzy sets and probabilities are used in combination to allow for linguistic probabilities. Linguistic probabilities enable the definition of probability measures which are given as linguistic labels represented by fuzzy sets. Furthermore, it is also possible to see one approach as a generalization of another. In [31] an example is given where rough mereology is used to implement a fuzzy controller where standards are defined as fuzzy sets and operations perform the tasks of implication functions.

Still, after choosing an approach (or a combination of approaches), the success of an application will strongly depend on the design of the system. This includes the definition of fuzzy rules and the according constituents, the design of a causal network and the definition of probabilities or setting up a reasoning scheme based on rough mereology. When designing the system, the data to reason on and the reasoning semantics of the domain need to be taken into account. For example, for a fuzzy reasoning system, the choices of an implication function needs to reflect the intended way of reasoning in the domain under consideration. As a further challenge the availability of required domain knowledge can be identified which is necessary for the definition of fuzzy membership functions and probability measures as well as well-defined standards and sufficiently good training data in the case of rough mereology.

Chapter 6

Conclusion and Future Work

The first part of this work covered a possible application of rough set theory in the image understanding process. The approach developed in this work offers a goal directed search of high level concepts for which it is probable that the abduction inference service succeeds. The approach presented is seen as a blueprint that describes in detail how rough set theory for knowledge discovery from databases can be exploited for the image understanding process. What remains an open issue are experiments as a proof of concept. As soon as real training data in form of annotated images and corresponding MLC instances detected by low level image analysis becomes available in future phases of the BOEMIE project, the approach presented in this work can be applied to investigate the practicability of the solution. This includes fine-tuning various parameters along the approach including several reduct calculation and rule application options as well as different kinds of rule quality measure computations. Also, the integration of the confidence measure coming from low level image analysis needs to be adapted to the according semantics of the value. Furthermore, it might also be necessary to develop strategies for preprocessing data coming from low level image analysis before applying rough set theory to it.

For conducting experiments, software applications based on open source programming libraries such as ROSETTA [24] and RSES [37] are available that already offer standard implementations of several rough set theory related algorithms for reduct calculation and rule application. While some of the functionality can be used directly to apply rough set theory to training data of the image understanding process, other parts need to be extended by the concepts presented in this work. The computation of rule quality measures adapted to the image understanding process and the integration of the confidence measures coming from low level image analysis can be identified as major parts for which an extension is necessary.

Further research also needs to be directed towards alternative solutions to rough set theory in the image understanding process. This can include different machine learning approaches such as decision tree learning, neural networks or support vector machines. Also layered learning for concept synthesis [4, 5, 23] as an extension of rough set theory might be a suitable solution. In the mentioned approach concepts are approximated by rough set theory at different levels in a hierarchy which suits ontologies as knowledge representation within the high level reasoning module.

Rough mereology in image understanding

While rough set theory has been studied in great detail over the last decades, rough mereology as a novel methodology for approximate reasoning offers various research directions. In this work, the main ideas of numerous theoretical publications about rough mereology have been presented in a concise and coherent manner. Additionally, well-known formalisms for approximate reasoning such as fuzzy set theory and bayesian reasoning have been compared to rough mereology. Therefore this work is assumed to be the basis for further works on rough mereology including practical applications. According to [31], rough mereology will allow advances in distinct application fields such as computer aided design, medicine, economics and software engineering. Yet, generic and practical applications of rough mereology for approximate reasoning need to be developed in order to show the practicability of the approach. The process of image understanding can also be identified as a possible field for the application of rough mereology. Further research needs to be conducted in this direction developing a reasoning scheme that suits the requirements of the image understanding process.

A first idea in this direction developed during this work defines a reasoning agent for each HLC and MLC. HLC agents construct HLC instances from MLC instances using according operations and MLC instances sent by MLC agents. Given a set of MLC instances from low level image analysis, MLC agents compute mereological distances from respective standards and send the detected MLC instances along with similarity measures of respective standards to their parents. HLC agents then use the sent MLC instances to construct HLC instances within a certain similarity bound to according HLC standards. The approach only uses the bottom up process as the decomposition of specifications is not needed. The final constructed HLC instance can be delivered with several respective HLC standards of the same agent and according similarity measures. The synthesized solution can be interpreted in such a way that the constructed HLC instance satisfies the properties of the respective standard HLC instances up to a certain degree. This also results in different solutions at various HLC agents constructed from the same set of MLC instances. For example, there could be several constructed pole vault and high jump HLC instances, each having different properties and features. The selection of a final solution is then based on the highest similarity measure in combination with external criteria specifying features of the constructed HLC.

The approach sketched here is at a very early stage and is not yet implemented. However it can serve as a starting point for further research in the field of approximate reasoning and image understanding. Especially the design of a reasoning scheme for image understanding remains as a challenging task. One important solution to be found in first steps is the definition of information systems of HLC and MLC agents. While for MLC agents low level features can be used as attributes and therefore the basis for similarity calculation, such information is not available for HLC instances. Furthermore it is necessary to investigate whether rough mereology should cover the complete image understanding process, including reasoning based on low level features and high level concepts or whether it can be used as an alternative or additional approach. For example, rough mereology can be used as an additional approach for low level image analysis only, where MLC instances are detected based on similarity measures calculated from low level features.

Also, formalisms that allow for integrating uncertainty handling mechanisms directly within the high level reasoning module such as probabilistic extensions to description logic [16, 11] or fuzzy descriptions logics [42] are promising approaches for the image understanding problem. Both concepts are based on approximate reasoning approaches also presented in this work. Within this context, it needs to be considered whether uncertainty in the reasoning process suits the ideas of graduality, degrees of belief or similarity of objects.

Appendix A

Example pictures

Figure A.1: Representative High Jump image for images in class c_1

Figure A.2: Representative Pole Vault image for images in class c_2

Figure A.3: Representative Pole Vault image for images in class c_3

Figure A.4: Representative Pole Vault image for images in class c_4

Figure A.5: Representative High Jump image for images in class c_5

Figure A.6: Representative Pole Vault image for images in class c_6

Figure A.7: Representative Pole Vault image for images in class c_7

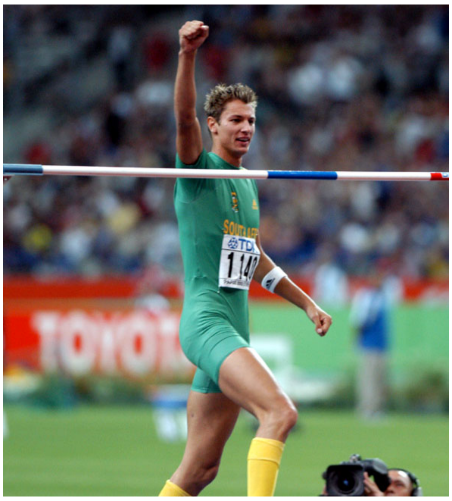


Figure A.8: Representative HIgh Jump image for images in class c_8

Appendix B

Additional Tables

No.	rule antecedent	rule consequent	support
r_1	if (Wind=onshore) and (Tide=low)	then (Condition=good)	1
r_2	if (Wind=onshore) and (Tide=low)	then (Condition=bad)	1
r_3	if (Weather=cloudy) and (Tide=low)	then (Condition=good)	1
r_4	if (Weather=cloudy) and (Tide=low)	then (Condition=bad)	1
r_5	if (Tide=low) and (Swell=1m)	then (Condition=good)	1
r_6	if (Tide=low) and (Swell=1m)	then (Condition=bad)	1
r_7	if (Wind=onshore) and (Swell=1m)	then (Condition=good)	1
r_8	if (Wind=onshore) and (Swell=1m)	then (Condition=bad)	1
r_9	if (Weather=cloudy) and (Swell=1m)	then (Condition=good)	1
r_{10}	if (Weather=cloudy) and (Swell=1m)	then (Condition=bad)	1
r_{11}	if (Swell=2m)	then (Condition=good)	2
r_{12}	if (Swell=0m)	then (Condition=bad)	2
r_{13}	if (Wind=offshore) and (Swell=1m)	then (Condition=good)	1
r_{14}	if (Weather=sunny) and (Swell=1m)	then (Condition=good)	1
r_{15}	if (Tide=high) and (Swell=1m)	then (Condition=good)	1
r_{16}	if (Wind=onshore) and (Tide=high)	then (Condition=good)	1
r_{17}	if (Weather=cloudy) and (Tide=high)	then (Condition=good)	1
r_{18}	if (Wind=offshore) and (Tide=low)	then (Condition=bad)	1
r_{19}	if (Weather=sunny) and (Tide=low)	then (Condition=bad)	1

Table B.1: Rules generated from Table 3.2

	x_1y_1	x_1y_2	x_1y_3	x_1y_4	x_2y_1	x_2y_2	x_2y_3	x_2y_4
x_1y_1	1	0.6	0.4	1	0.8	0.4	0	0.6
x_1y_2	0.6	1	0.8	0.6	0.4	0.8	0.4	0.2
x_1y_3	0.4	0.8	1	0.4	0.6	1	0.6	0.4
x_1y_4	1	0.6	0.4	1	0.8	0.4	0	0.6
x_2y_1	0.8	0.4	0.6	0.8	1	0.6	0.2	0.8
x_2y_2	0.4	0.8	1	0.4	0.6	1	0.6	0.4
x_2y_3	0	0.4	0.6	0	0.2	0.6	1	0.4
x_2y_4	0.6	0.2	0.4	0.6	0.8	0.4	0.4	1
x_3y_1	0.6	0.2	0.4	0.6	0.8	0.4	0.4	0.6
x_3y_2	0.2	0.6	0.8	0.2	0.4	0.8	0.8	0.2
x_3y_3	0.2	0.6	0.8	0.2	0.4	0.8	0.8	0.2
x_3y_4	0.6	0.2	0.4	0.6	0.8	0.4	0.4	0.6
x_4y_1	0.8	0.4	0.2	0.8	0.6	0.2	0.2	0.4
x_4y_2	0.4	0.8	0.6	0.4	0.2	0.6	0.6	0
x_4y_3	0.2	0.6	0.8	0.2	0.4	0.8	0.8	0.2
x_4y_4	1	0.6	0.4	1	0.8	0.4	0	0.6

Table B.2: Rough inclusion values of system agent, part 1

	x_3y_1	x_3y_2	x_3y_3	x_3y_4	x_4y_1	x_4y_2	x_4y_3	x_4y_4
x_1y_1	0.6	0.2	0.2	0.6	0.8	0.4	0.2	1
x_1y_2	0.2	0.6	0.6	0.2	0.4	0.8	0.6	0.6
x_1y_3	0.4	0.8	0.8	0.4	0.2	0.6	0.8	0.4
x_1y_4	0.6	0.2	0.2	0.6	0.8	0.4	0.2	1
x_2y_1	0.8	0.4	0.4	0.8	0.6	0.2	0.4	0.8
x_2y_2	0.4	0.8	0.8	0.4	0.2	0.6	0.8	0.4
x_2y_3	0.4	0.8	0.8	0.4	0.2	0.6	0.8	0
x_2y_4	0.6	0.2	0.2	0.6	0.4	0	0.2	0.6
x_3y_1	1	0.6	0.6	1	0.8	0.4	0.6	0.6
x_3y_2	0.6	1	1	0.6	0.4	0.8	1	0.2
x_3y_3	0.6	1	1	0.6	0.4	0.8	1	0.2
x_3y_4	1	0.6	0.6	1	0.8	0.4	0.6	0.6
x_4y_1	0.8	0.4	0.4	0.8	1	0.6	0.4	0.8
x_4y_2	0.4	0.8	0.8	0.4	0.6	1	0.8	0.4
x_4y_3	0.6	1	1	0.6	0.4	0.8	1	0.2
x_4y_4	0.6	0.2	0.2	0.6	0.8	0.4	0.2	1

Table B.3: Rough inclusion values of system agent, part 2

Appendix C

Rough and Fuzzy membership function

In order to understand the differences between the rough membership function and the fuzzy membership function, a short comparison of both concepts will be presented in the following. This chapter makes use of the notions introduced in chapter 3.

C.1 Fuzzy set theory

In fuzzy set theory [44] the membership of an element to a certain concept is given by its *fuzzy membership function*. As an example Figure C.1 shows the membership functions μ_{Safe} and $\mu_{Challenging}$ of the linguistic labels *safe* and *challenging* depending on the swell size in a graph. Both correspond to the definition of a fuzzy set. As an example, an object for which the swell size of 2 m is known belongs to concept *safe* with a degree of 1 and to concept *challenging* with a degree of 0.1. This brings up the question, how a value for the concept *safe and challenging* can be calculated which would result in a set theoretic intersection of the both concepts.

For calculating the intersection of two fuzzy sets $\mu \cap \mu'$it is assumed

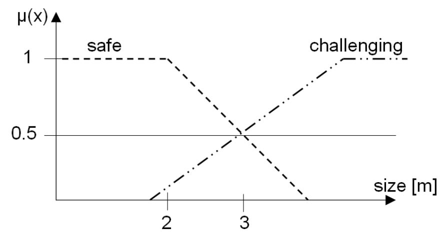


Figure C.1: Fuzzy Membership Function

that the intersection is calculated per element. This results in a function $\sqcap : [0,1]^2 \rightarrow [0,1]$ in such a way that

$$(\mu \cap \mu')(x) = \sqcap(\mu(x), \mu'(x))$$

In order to accept a function $\sqcap$ as an operator for set theoretic intersection it has to hold for axioms given as boundary condition, monotonicity, commutativity, associativity, continuity and subidempotency. Such a function $\top$ is called a *t-norm* [29]. Some prominent t-norms are defined as:

$$\begin{array}{lll} \top_{min}(a,b) & =^{def} & min\{a,b\} \\ \top_{Luka}(a,b) & =^{def} & max\{0, a+b-1\} \\ \top_{prod}(a,b) & =^{def} & a \cdot b \end{array}$$

Accordingly the definition for functions $\perp$ called *t-conorms* are used for the union of fuzzy sets:

$$\begin{array}{lll} \perp_{max}(a,b) & =^{def} & max\{a,b\} \\ \perp_{Luka}(a,b) & =^{def} & min\{a+b, 1\} \\ \perp_{prod}(a,b) & =^{def} & a+b-ab \end{array}$$

As a result intersection and union of fuzzy sets can now be calculated by applying an according t-norm or t-conorm.

$$(\mu \cap_{\top} \mu')(x) = \top(\mu(x), \mu'(x))$$

$$(\mu \cap_{\perp} \mu')(x) = \perp(\mu(x), \mu'(x))$$

De Morgan's laws

$$\overline{\mu \cap \mu'} = \overline{\mu} \cup \overline{\mu'}$$

$$\overline{\mu \cup \mu'} = \overline{\mu} \cap \overline{\mu'}$$

hold for all pairs of t-norms and t-conorms. However, they do not form a boolean algebra, because there is no complement operator for which $\mu \cap \overline{\mu} = 0$ and $\mu \cup \overline{\mu} = 1$ always holds.

It has been found [17] that the pair $\{min,\ max\}$ is the only pair of t-norms and t-conorms that preserve the property of distributivity. If the property $\mu \cap \mu = \mu$ and $\mu \cup \mu = \mu$ is needed for fuzzy sets, the pair *{min, max}* has to be used.

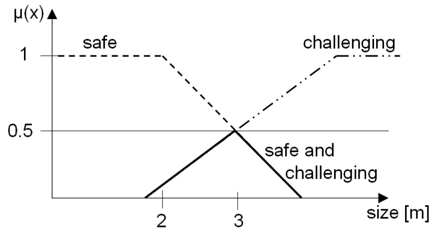
Accordingly in [44] the pair *{min, max}* has been used for calculation with fuzzy sets in the following way:

$$(\mu \cap \mu')(x) = min\{\mu(x), \mu'(x)\}$$

$$(\mu \cup \mu')(x) = max\{(\mu(x), \mu'(x)\}$$

$$\overline{\mu}(x) = 1 - \mu(x)$$

Example C.1.1. Using the pair of *{min, max}*, the linguistic label of *"safe and challenging"* is calculated by means of fuzzy set theory. The solid line in the figure denotes the application of the *min* operator.



For an object for which the swell size of 2 m is known, the membership to concept *safe and challenging* results in a degree of 0.1. Accordingly an an object for which the swell size of 3 m is known belongs to concept *safe and challenging* with a degree of 0.5.

Example C.1.2. As stated above, an object for which the swell size of 2 m is known, the membership to the single concept *safe* has a degree of 0.1. Using $\overline{\mu}(x) = 1 - \mu(x)$ to calculate the complement concept *not safe*, the membership results in a degree of 0.9. Calculating the concept *safe and not safe* results in $min\{0.1, 0.9\} = 0.1$ instead of 0 in case of a boolean algebra.

C.2 Calculating composite concepts using Rough Sets

For showing the differences between the fuzzy membership function and the rough membership function, the calculation of *composite concepts* will be considered in rough set theory . In rough sets, composite concepts can for example be found when for each object in the universe there is more than one decision attribute. Combinations of such decision concepts can then also be calculated using set theoretic operators.

Example C.2.1. In the following table an information system is presented with the attribute set $A = Swell$ and two different decision concepts X : $Safe$ and Y : $Challenging$. The third decision concept $Z = Safe \cap Challenging$ is calculated under set theoretic intersection. Horizontal lines are demarcations of equivalence classes created by $A = Swell$.

U	$Swell$	$Safe$	$Challenging$	$Safe \cap Challenging$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_i	3 m	yes	no	no
x_{i+1}	3 m	no	yes	no
x_{i+2}	3 m	yes	yes	yes
x_{i+3}	3 m	no	yes	no
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Using the rough membership function defined above, the following results for an object x_k with $Swell = 3m$ are obtained.

$$\mu_X^{\{Swell\}}(x_k) = 1/2$$
$$\mu_Y^{\{Swell\}}(x_k) = 3/4$$
$$\mu_{Z=X\cap Y}^{\{Swell\}}(x_k) = 1/4$$

The example above shows the fact, that in rough set theory the membership of an object to a composite concept is calculated using background data as already defined in chapter 3. It has been found [28] that there is no function $\sqcap$ independent from background data which allows for the calculation of the membership based on its constituents in the form of

$$(\mu \cap \mu')(x) = \sqcap(\mu(x), \mu'(x)).$$

In the same work it has been stated that the following properties of the rough membership function are valid:

$$\mu_{X\cap Y}^B(x) \leq min\{\mu_X^B(x), \mu_Y^B(x)\}$$
$$\mu_{X\cup Y}^B(x) \geq max\{\mu_X^B(x), \mu_Y^B(x)\}$$

C.3 Rough-Fuzzy hybridization

Despite the differences in rough and fuzzy set theory, there are several attempts in literature for combining both approaches. Such a combination of approaches or theories is also reffered to as *hybridization*. One way of hybridization are *rough-fuzzy sets* [41], where equivalence classes are being defined using fuzzy membership functions. In that case an object does not belong to a single equivalence class but may belong to several equivalence classes to a certain degree. Equivalence classes in the basic rough set approach may then be considered as a general case where each object belongs to an equivalence class to the degree 0 or 1.

Further approaches for combining fuzzy and rough sets interpret attribute values of conditional and decision attributes as membership values of predefined fuzzy sets [14]. Using the fuzzy sets defined in Figure C.1, an attribute a would be defined by the fuzzy sets $safe$ and *challenging*. An object for which the swell size of 2 m is known, would then result in attribute values for a in the form of membership of degree 1 to the concept $safe$ and a membership of degree 0.1 to the concept *challenging*: $a(x_i) = \{\mu_{safe}(2m), \mu_{challenging}(2m)\} = \{1, 0.1\}$

Appendix D

Learning of defaults

D.1 Learning of defaults for probabilistic default reasoning with conditional constraints

In [20] it is assumed that there is some *statistical knowledge about a set of individuals* in the form of "all penguins are birds" or "between 90% and 95% of all birds can fly". In the following an approach is presented in order to derive such knowledge (defaults) from data by means of rough set theory.

Assume there exists statistical knowledge about a set of 100 patients that classifies each patient into a certain decision class *disease* = *yes* or *disease* = *no*. The following table represents that knowledge.

disease	support
yes	30
no	70

From the knowledge represented in the table, one can derive the following statement:

- 30% of all patients have a disease

However, there might be some additional knowledge in the form of conditional attributes that describe properties of individuals. In the following we assume that we know the tempretature of the set of patients. Horizontal lines denote equivalence classes.

temperature	*disease*	support
40	yes	10
38	yes	20
38	no	70

Considering the knowledge about the patients temperature, the following statement can be derived:

- At least 10% of all patients have a disease

Alhough data actually tells us that 30% of all patients have a disease, percepting individuals by means of their properties (e.g. temperature) increases uncertainty.

Now the knowledge about patients is increased by means of a second attribute telling whether a patient feels sick.

sickness	temperature	*disease*	support
yes	40	yes	10
yes	38	yes	20
yes	38	no	20
no	38	no	50

Including the knowledge about sickness, the following statement can be derived:

- Between 10% and 50% of all patients have a disease.

In a next step, the knowledge about patients is increased by means of a third attribute telling whether a patient feels pain.

pain	sickness	temperature	*disease*	support
yes	yes	40	yes	10
yes	yes	38	yes	15
no	yes	38	yes	5
no	yes	38	no	20
no	no	38	no	50

Including the knowledge about sickness, the following statement can be derived:

- Between 25% and 50% of all patients have a disease.

By increasing the knowledge about patients, the concept of having a disease is being approximated. However percentages are not given purely based on statistics but also consider the knowlegde about individuals.

Interchanging attributes

In the examples above, *disease* has always been considered the attribute of interest. However, it is possible to make statements about every attribute. Assume that the decision attribute now is *pain* and the set of conditional attributes is $A = \{sickness, temperature, disease\}$.

pain	sickness	temperature	disease	support
yes	yes	40	yes	10
yes	yes	38	yes	15
no	yes	38	yes	5
no	yes	38	no	20
no	no	38	no	50

This allows for deriving the following statement:

- Between 10% and 30% of all patients feel pain.

Using available knowledge

Furthermore, one can also derive statements for individuals for which certain knowledge is available. For example, we would like to gain knowledge about patients with a temperature of 38. This reduces the number of patients to 90 for which the following statement can be derived:

- Between 17% and 44% of all patients with a temperature of 38 have a disease (between 15/90 and 40/90).

or for $A = \{sickness, temperature, disease\}$:

- At most 22% of all patients with a temperature of 38 feel pain (between 0/90 and 20/90).

Bibliography

[1] Aijun An, Nick Cercone (2000). *Rule Quality Measures Improve the Accuracy of Rule Induction: An Experimental Approach.* In: Foundations of Intelligent Systems: 12th International Symposium, ISMIS 2000, Charlotte, NC, USA, October 11-14, 2000. Proceedings

[2] Franz Baader, Diego Calvanese, Deborah McGuinness (2003). *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press.

[3] Jan G. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, J. Wroblewski (2000). *Rough set algorithms in classification problem.* In: L. Polkowski, S. Tsumoto, and T. Lin, editors, Rough Set Methods and Applications, Physica-Verlag, Heidelberg New York, pp. 49-88.

[4] Jan G. Bazan, Sinh Hoa Nguyen, Hung Son Nguyen, Andrzej Skowron (2004). *Rough Set Methods in Approximation of Hierarchical Concepts.* In: RSCTC 2004, LNAI 3066, pp. 346-355.

[5] Jan G. Bazan, Sinh Hoa Nguyen, Hung Son Nguyen, Andrzej Skowron (2004). *Layered Learning for Concept Synthesis.* In: Transactions on Rough Sets I, LNCS 3100, pp. 187-208, Springer Verlag.

[6] Jan. G. Bazan, Andrzej Skowron , P. Synak (1994) *Dynamic reducts as a tool for extracting laws from decision tables* In: Proceedings of RSSC'94, Lecture Notes in Artificial Intelligence 869, pp. 346-355, Springer-Verlag.

[7] Boemie (2006). *Bootstrapping Ontology Evolution with Multimedia Information Extraction.* WWW page, December 2006. http://www.boemie.org.

[8] Boemie (2006). *Methodology for Semantics Extraction from Multimedia Content.* Deliverable D2.1, Version 1.3, August 2006.

[9] Didier Dubois, Henri Prade (1991). *Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions.* In: Fuzzy Sets and Systems 40, pp. 143-202.

[10] Didier Dubois, Henri Prade (2001). *Possibility Theory, Probability Theory and Multiple-valued Logics: A Clarification.* In: Annals of Mathematics and Artificial Intelligence 32, pp. 35-66.

[11] Rosalba Giugno, Thomas Lukasiewicz (2002). *P-SHOQ (D): A probabilistic extension of SHOQ (D) for probabilistic ontologies in the semantic web.* In: Lecture Notes in Computer Science Volume 2424, Spinger Verlag.

[12] J. Grzymala-Busse (1997). *A New Version of the Rule Induction System.* LERS Fundamenta Informaticae, Vol. 31(1), pp. 27-39

[13] J. Halliwell, Q. Shen (2002). *Towards a Linguistic Probability Theory.* In: Proceedings of the 11th International Conference on Fuzzy Sets and Systems (FUZZ-IEEE '02).

[14] Richard Jensen (2005). *Combining rough and fuzzy sets for feature selection.* PhD Thesis, University of Edinburgh.

[15] John R. Josephson, Susan G. Josephson (2003). *Abductive Inference: Computation, Philosophy, Technology.* Cambridge University Press.

[16] Daphne Koller, Alon Levy, Avi Pfeffer (1997). *PClassic: a tractable probabilistic description logic.* In: Proceedings of the AAAI Fourteenth National Conference on Artificial Intelligence.

[17] Rudolf Kruse, Joan E. Gebhardt, F. Klowon (1994). *Foundations of Fuzzy Systems.* 1st edition, John Wiley & Sons, Inc., New York, NY, USA.

[18] Stanislaw Lesniewski (1991). *Foundations of the General Theory of Sets.* In: Stanislaw Lesniewski: Collected Works - Volumes I and II, Dordrecht Kluwer Academic Publishers

[19] T. Y. Lin, Q. Liu (1996). *First-Order Rough Logic I: Approximate Reasoning via Rough Sets* In: Fundamenta Infomaticae, vol. 27, Nos.2,3, pp. 137-154.

[20] Thomas Lukasiewicz (2000). *Probabilistic default reasoning with conditional constraints.* In: Proceedings of the 8th International Workshop on Non-monotonic Reasoning, Breckenridge, CO, USA, 2000

[21] Adam Mrozek (1992). *Rough Sets in Computer Implementation of Rule-Based Control of Industrial Processes.* In: Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, Dordrecht, Kluwer Academic Publishers.

[22] Hung Son Nguyen, Andrzej Skowron, Marcin S. Szczuka (2001). *Situation Identification by Unmanned Aerial Vehicle.* In: RSCTC '00: Revised Papers from the Second International Conference on Rough Sets and Current Trends in Computing, pp. 49-56, Springer-Verlag

[23] Sinh Hoa Nguyen, Hung Son Nguyen (2005). *Improving Rough Classifiers Using Concept Ontology.* In: PAKDD 2005, LNAI 3518, pp. 312-322, Springer Verlag.

[24] Aleksander Øhrn. *ROSETTA: A Rough Set Toolkit for Analysis of Data.* Website: http://rosetta.lcb.uu.se/general/, last accessed: December 31st, 2006.

[25] Simon Parsons, Anthony Hunter (1998). *A review of uncertainty handling formalisms.* In: Applications of Uncertainty Formalisms, Lecture Notes in Artificial Intelligence, Volume 1455, pp. 8-37., Springer Verlag, Berlin

[26] Simon Parsons, Miroslav Kubat (1994). *A first-order logic for reasoning under uncertainty using rough sets* In: Journal of Intelligent Manufacturing 5, pp. 211-223.

[27] Z. Pawlak (1992). *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer, Dordrecht

[28] Z. Pawlak, Andrzej Skowron (1994) *Rough membership functions.* In: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, Wiley, NewYork, pp. 251-271.

[29] L. Polkowski (2002). *Rough Sets. Mathematical Foundations.* Physica, Heidelberg.

[30] L. Polkowski, Andrzej Skowron (1996). *Rough Mereology: A new paradigm for approximate reasoning.* In: Int. Journal of Approximate Reasoning, 15(4), pp. 333-365.

[31] L. Polkowski, Andrzej Skowron (1998). *Approximate reasoning about complex objects in distributed systems: Rough mereological formalization.* In: Computational Intelligence in Software Engineering. Advances in Fuzzy Systems-Applications and Theory 16, 237-267, World Scientific, Singapore.

[32] L. Polkowski, Andrzej Skowron (1996). *Rough mereological approach to knowledge-based distributed AI.* In: Critical Technology: Proceedings of WCES-3 Third World Congress on Expert Systems, 774-781, Seoul, Korea, February 5-9, 1996.

[33] L. Polkowski (2004). *Toward Rough Set Foundations. Mereological Approach.* In: Rough Sets and Current Trends in Computing. LNAI 3066, pp.8-25, Springer-Verlag, Berlin, Heidelberg.

[34] Stuart Russell, Peter Norvig (2003). *Artificial Intelligence: A Modern Approach.* 2nd edition, Prentice-Hall, Englewood Cliffs, NJ.

[35] Andrzej Skowron (2004). *Approximate Reasoning in Distributed Environments.* In: Intelligent Technologies for Information Analysis, pp. 426-468, Springer Verlag.

[36] Andrzej Skowron (2001). *Rough sets and boolean reasoning.* In: Granular computing: an emerging paradigm, pp. 95-124, Physica-Verlag, Heidelberg, Germany.

[37] Andrzej Skowron et al. *RSES: Rough Set Exploration System.* Website: http://logic.mimuw.edu.pl/~rses/, last accessed: December 31st, 2006.

[38] Andrzej Skowron, Cecylia M. Rauszer (1992). *The discernibility matrices and functions in information systems.* In: Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, chapter: 3, pp. 331-362, Kluwer, Dordrecht

[39] Andrzej Skowron, Jaroslaw Stepaniuk (2001). *Information granules: Towards foundations of granular computing.* In: Int. J. Intell. Syst. 16(1), pp. 57-85

[40] D. Slezak (1996) *Approximate reducts in decision tables.* In: Bouchon Meunier, Delgado, Verdegay, Vila, and Yager, Proceedings of the Sixth International Conference. Information Processing Management of Uncertainty in Knowledge-Based Systems pp. 1159-1164.

[41] P. Srinivasan, M.E. Ruiz, D.H. Kraft, J. Chen (1998). *Vocabulary mining for information retrieval: rough sets and fuzzy sets.* In: Information Processing and Management, Vol. 37, No. 1, pp. 15-38.

[42] Umberto Straccia (1998). *A fuzzy description logic.* In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), pp. 594-599, Madison, USA.

[43] Adam Szmigielski (2002) *Rough Mereological Localization and Navigation.* In: TSCTC '02: Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing, pp. 629-638, Springer-Verlag

[44] Lofti A. Zadeh (1965). *Fuzzy Sets.* In: Information and Control, 8, pp. 338-353

[45] Lofti A. Zadeh (1973). *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*. In: IEEE Trans. on Systems, Man, and Cybernetics, 3: pp. 28-44