Planning Based on Reasoning About Information Changes

Andrzej Skowron[1] and Piotr Synak[2]

[1] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
[2] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland

Abstract. We consider the problem of reasoning about information changes in the context of complex concepts approximated hierarchically and actions that can be triggered to change properties of investigated objects. A given object can be in an unwanted state, where some concept is not satisfied to the required degree. We would like to find a plan (sequence of actions to be executed) of object's transformation to a new state which is acceptable. Presented approach is based on reasoning about changes.

Keywords: rough sets, approximate reasoning schemes, hierarchical reasoning, concept approximation, reasoning about changes.

1 Introduction

We consider the problem of reasoning about information changes in the context of complex concepts approximated hierarchically and actions that can be triggered to change properties of investigated objects. The problem of complex concepts approximation has been already intensively studied in the literature [19,1,14,15,16,2,20,4,10,11,3]. Hierarchical reasoning seems to be crucial especially in the case when there is a big gap between description of objects and the concept. For example, it is very difficult to directly reason about the concept "safe situation on the road" just from the low-level sensor measurements. Thus, some methods of hierarchical reasoning using the domain knowledge, e.g., in the form of ontology of concepts, must be adapted to obtain satisfactory approximation of complex concepts (see, e.g., [1,19]). In addition, we consider the case when some set of actions is additionally available. An action can be triggered for a given object to change its properties. We assume to use actions if we detect that the investigated object satisfies a given concept to unsatisfactory degree or it satisfies some unwanted concept. As an example, let us consider an action as some medicine that can be applied to a patient having some disease. We want to apply such medicines that the concept of having disease is not satisfied any longer. Execution of some action corresponds to an application of some transition relation. A sequence of actions defines some kind of a plan.

Let us present some examples of the key problems. We can consider the problem of rules induction. Let x be an investigated object satisfying concept C_1

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 165–173, 2006.

and R be a transition relation that moves/changes x from C_1 to, say, C_2. From the approximation of concept C_1, e.g, by means of AR schemes [13,7,18], and approximation of relation R we obtain approximation of C_1R. On the other hand, from C_1R we can extract approximation of C_2. Let us note, that from approximations of C_1 and R we obtain approximate rules for C_1, $\neg C_1$ and R, $\neg R$, respectively. Thus, finally we also get rules for C_2 and $\neg C_2$.

Next problem is related to the induction of rules of changes. Let x satisfy a given concept C to a degree at least δ. If we apply to x some transition relation R we obtain x' satisfying C to a degree at least $\delta + \Delta\delta$. A basic question is how can we induce rules predicting changes of inclusion degree. We can generalise this problem to the case when we consider not only one particular concept but k-class classification, i.e, $\{C_1, \ldots, C_k\}$, where for a given object we obtain a vector of inclusion degrees $(\delta_1, \ldots, \delta_k)$. Then, how can we induce decision rules describing changes of a vector of inclusion degrees?

2 Hierarchical Reasoning on Complex Concepts

2.1 From Structured Objects to Complex Concepts

One of the fundamental concepts in reasoning is the notion of an object. Objects are some real entities that can be described by some physical observations or measurements. An object can be though identified with some information about it, i.e., with some vector of measurements. From this interpretation it follows that one vector of measurements can describe several objects. From the point of view of this information only, such objects are indiscernible although in fact they can be different. This way of understanding objects is used in the rough set theory [5,6,12], where for a given information system $\mathbb{A} = (U, A)$, the information about an object $x \in U$ is given by means of some attributes from A, i.e., an object x can be identified with the so-called signature of x: $Inf(x) = \{a(x) : a \in A\}$.

In a more complex case, we can consider some structure of an object. Structured or complex objects can consist of some parts which can be constrained by some relations of different nature, e.g., spatial ones. The parts can be built from yet simpler parts and therefore the structure can be hierarchical with many different levels. The relation object–part corresponds in most cases to some spatial or spatio-temporal relation [10]. These problems are considered in rough-mereological approach [8,9] representing some patterns relevant for concept approximation.

For each part of a structured object we can consider some concepts describing its properties. Thus, concepts form also a hierarchical structure and for one structured object we can have several ontologies of concepts [17]. The concepts from the lowest level of such hierarchy describe properties of simple parts. The high-level or complex concepts describe properties of complex objects.

2.2 Structured Reasoning Schemes

Properties of structured (complex) objects can be approximated by means of approximate reasoning schemes (AR schemes) [13,7,18]. Such schemes usually have a tree structure with the root labelled by the satisfiability degree of some feature by a complex object and leaves labelled by the satisfiability degrees of some other features by primitive objects (i.e., the most simple parts of a complex object). An AR scheme can have many levels. Then, from properties of basic parts and relations among them we conclude the properties of more complex parts, and after some levels, the properties of the complex target object.

Any AR scheme is constructed from labelled approximate rules, called productions. Productions can be extracted from data using domain knowledge. We define productions as parameterised implications with premises and conclusions built from patterns sufficiently included in the approximated concept.

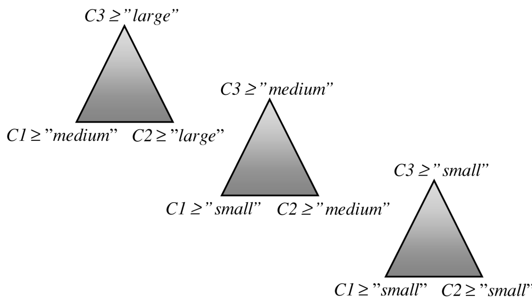

Fig. 1. An example of production as a collection of three production rules

In Figure 1, we present an example of production for some concepts *C1*, *C2*, and *C3* approximated by three linearly ordered layers *small*, *medium*, and *large*. This production is a collection of three simpler rules, called production rules, with the following interpretation: (1) if inclusion degree to a concept *C1* is at least *medium* and to a concept *C2* at least *large* then the inclusion degree to a concept *C3* is at least *large*; (2) if the inclusion degree to a concept *C1* is at least *small* and to a concept *C2* at least *medium* then the inclusion degree to a concept *C3* is at least *medium*; (3) if the inclusion degree to a concept *C1* is at least *small* and to a concept *C2* at least *small* then the inclusion degree to a concept *C3* is at least *small*.

The concept from the highest level of production is called the target concept of production, whilst the concepts from the lowest level of production are called the source concepts of production. For example, in the case of production from Figure 1, *C3* is the target concept and *C1*, *C2* are the source concepts.

One can construct an AR scheme by composing single production rules chosen from different productions from a family of productions for various target concepts. In Figure 2, we have two productions. The target concept of the first production is *C5* and the target concept of the second production is the concept *C3*. We select one production rule from the first production and one production rule from the second production. These production rules are composed and then a simple AR scheme is obtained that can be treated as a new two-level production rule. Notice that the target pattern of lower production rule in this

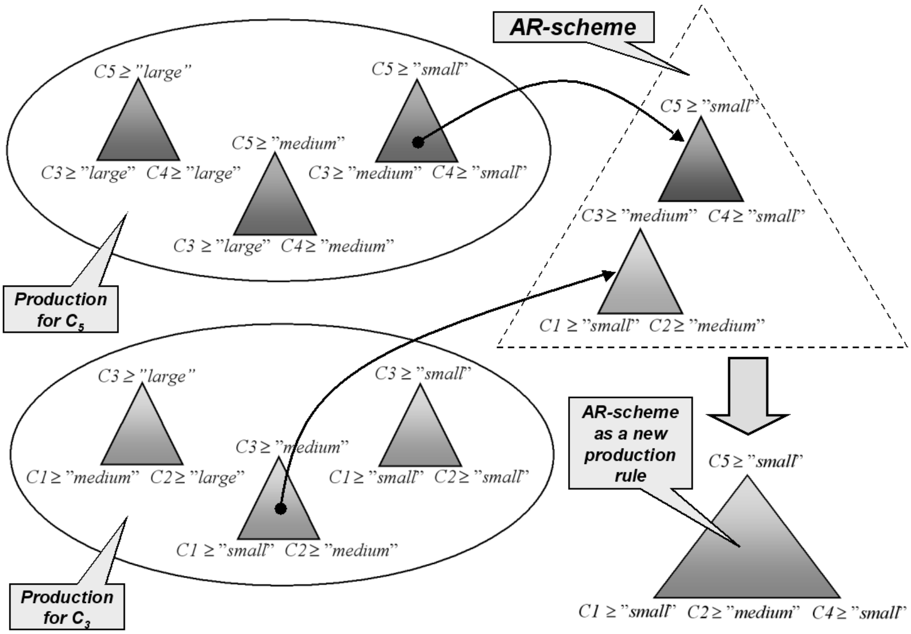

Fig. 2. Synthesis of approximate reasoning scheme

AR scheme is the same as one of the source patterns from the higher production rule. In this case, the common pattern is described as follows: inclusion degree (of some pattern) to a concept *C3* is at least *medium*.

In this way, we can compose AR schemes into hierarchical and multi-level structures using productions constructed for various concepts.

3 Reasoning Based on Changes

3.1 General Scheme of Reasoning

In this section let us present some general scheme of reasoning about complex concepts that are satisfied to an unsatisfactory degree. Such a case can be a result of some changes of the situation in time and may be required to undertake appropriate actions. Let U be a universe of objects and C be a given concept. For example, we can consider a set of patients as U and a concept of having given disease as C. Let us also denote by $\neg C$ the complementary concept to C – in our example the concept of not having given disease. Now, we can consider some set $X \subseteq U$ of objects included into $\neg C$ to a satisfactory degree, as well as $Y \subseteq U$ – the set of objects well included into C.

A given situation can dynamically be changing in time what we can refer to by states of an object. We can observe that in some states the concept C is satisfied whilst in some other states is $\neg C$. It means that there is additionally some transition relation $R \subseteq U \times U$ responsible for the process of transformation

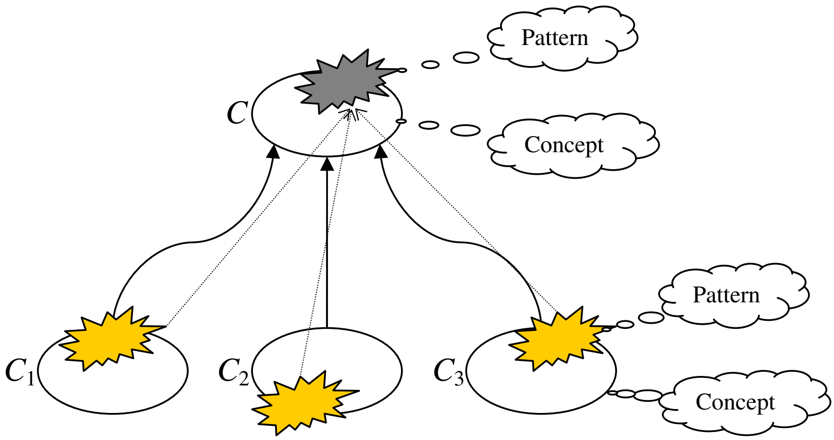

Fig. 3. Approximation of complex concept C by using patterns constructed from patterns approximating low-level concepts C_1, C_2, C_3

of objects from the set X to the set Y. Thus, we can say that $Y = XR = \{y \in U : \exists x \in X \;\; xRy\}$.

Let us recall that the reasoning about concepts related to X, here $\neg C$, is performed in a hierarchical manner by means of some patterns (see Fig. 3), and classifiers constructed using language of those patterns. In a similar way, one would have to construct a hierarchical classifier for approximation of a concept related to relation R, namely a concept satisfied by relation R to a satisfactory degree. Such a classifier, for a given pair of objects (x, y), where $x \in X, y \in Y$, must take into account: (1) properties of x by means of relevant patterns constructed for X, (2) properties of y by means of relevant patterns constructed for Y, (3) properties of the pair (x, y) by means of relevant patterns constructed for R (note that those patterns can be defined in a language much different from those in the other two cases, e.g., we can consider a closeness between x and y).

Let us emphasise that in general the situation can be much more complex. It can be impossible to approximate such a relation R that directly moves us from set X to Y. We can rather expect to be able to approximate a relation that moves us into "the right direction", i.e., to the state where the change of satisfaction degree of some concept is desired. It means that being in the state where satisfaction degree of concept $\neg C$ is high we have transition that moves us to the state where this degree is lower, i.e, change of degree is negative. We iteratively use several such transition to move in a direction of low satisfiability of $\neg C$ and high satisfiability of C.

The considered problem is related to the following situation: the reasoning about an investigated object leads us to the conclusion that the object does not satisfy a given concept to a satisfactory degree. We would like to impose such a changes that the concept is satisfied. What comes handy is a set of available actions that we can perform to change some properties of an object. The actions are given in the form of rules where premise describes objects to which given action can be applied, and conclusion specifies what will be changed after the action is triggered. In our example we can consider a patient having some disease (thus, satisfying $\neg C$). We would like to undertake some actions to treat the patient so it satisfies C into satisfactory degree. An action could correspond in this case to application of some medicine. A set of action is then a plan of therapy.

We can easily see that there can be induced several transition relations (several paths) that some of their compositions lead a given object from set X to set Y. Let us emphasise, that in each step from the pattern matched by the object and the pattern approximating transition relation we can decode the pattern matched by the transformed object. In this way, we obtain an input pattern for the next step. In each step, an object is transformed to a new state in which the satisfaction degree of a considered concept is better. However, it can appear that one or more steps of one path leads to a worse state. This can be necessary due to necessity of avoiding locally optimal states. Some heuristic search engines, including Genetic Algorithms and Simulated Annealing methods, can be utilised to generate optimal paths. Each path obtained should be additionally verified whether it is feasible. In particular, each step of a path should be verified if there are available actions making possible to realise this step. There should be also considered costs of performing actions realising all steps of a given path. Thus, the cost of a path and the quality of destination (by means of satisfaction degree of considered concept) of state should be evaluated while choosing the optimal path.

3.2 Some Detailed Issues

In this section, let us explain some details related to the possible realisations of the presented ideas. We assume that the investigated objects define some information system $\mathbb{A} = (U, A)$. The considered concept and its complement is denoted by C and $\neg C$, respectively.

Training data and training process. Each object from the training information system contains information about inclusion degrees to concepts C and $\neg C$. There are induced several hierarchical approximate reasoning schemes $\{AR_i\} = \{AR_i^C\} \cup \{AR_i^{\neg C}\}$. The input nodes of the schemes are corresponding to the low-level concepts approximated by means of patterns $\{r_i\}$. The set of patterns used in a given AR scheme depends on the low-level concept used in this scheme, however, any object from U can be tested against any pattern.

Actions and transition relation approximation. One of our assumptions is that we can have an influence on the properties of objects by means of some actions $\{ac_i\}$ we can undertake. Each action can have a cost associated with its execution. In the simplest case, an action can be precisely defined in terms of descriptors over the set of attributes A. Thus, each action can have a form of implication where the premise describes the properties of objects for which the action can be triggered whilst the conclusion defines the changes of object's properties. An example of such action is: "$a_1 = 5$ and $a_5 < 7 \Rightarrow a_1 > 10$ and $\Delta a_8 < 5$", where $a_i \in A$.

In a more complex case, we don't know precise definitions of actions but have some training data describing objects before and after action's execution (for example, we have characteristics of patients before and after application of some medicine). Thus, we also need to induce an AR scheme AR_0 approximating the concept that a given action ac is triggered. AR_0 is then a kind of approximation of transition relation between states of an object where the transition is forced by action ac. The low-level (input) concepts of obtained AR scheme AR_0 are approximated by patterns R_l^{ac} and R_r^{ac} describing properties of objects before and after execution of ac, respectively. Let us also emphasise that some of the low-level concepts can describe properties of pair of objects (x, x'). Those concepts are approximated by yet another set of patterns R_{lr}^{ac}.

In consequence, for a given object x matching patterns R_l^{ac} we can use scheme AR_0 to decode patterns matched by x after we apply action ac. In this way we have some approximation of an action in the language of patterns over the set of attributes A.

Reasoning process. Let x be an investigated object which has been evaluated by induced AR schemes $\{AR_i\}$ as satisfying $\neg C$. It means that it could be recognised by some schemes from $\{AR_i^{\neg C}\}$ (let us denote them by $AR^{\neg C}$) but also by schemes $AR^C \subseteq \{AR_i^C\}$ (in such a case conflict resolving strategies should be involved). The main problem is to find a sequence of actions that should be undertaken in order to transform object x to x' such that x' satisfies C to a satisfactory degree.

One possible way of reasoning is as follows. By examining schemes $AR^{\neg C}$ and $\{AR_i^C\} \setminus AR^C$, as well as the conflict resolving strategy, we can select (1) key schemes that recognised and evaluated x as matching $\neg C$, (2) schemes that could strongly "vote" for C but some of the input concepts were not matched by x good enough. Then, we can decide the way we want to change x. In the first case, we may force x not to match some patterns previously matched, so we can eliminate some schemes from $AR^{\neg C}$. In the second case, we may force x to match some patterns previously not matched, so we can add some "strong" schemes to AR^C.

In either cases, we have some patterns matched by x and some target patterns we would like to be matched by transformed x. Thus, we can try to iteratively combine AR schemes approximating available actions (or combine just actions in the simpler case), starting from patterns matched by x and going forward. Alternatively, we can go backward starting from the target patterns.

Let us denote a very important fact, that approximation of actions can be performed on different levels of generalisation. This is possible because the AR schemes used for approximation are hierarchical structures. Thus, by considering patterns from different levels of AR schemes we can obtain approximation of actions in the language of those patterns, and we can talk about actions as well as meta-actions.

4 Conclusions and Directions for Further Research

In the paper we discussed some problems related to reasoning about information changes in the context of complex concepts approximated hierarchically. Main issue discussed was finding a plan (sequence of actions) of which execution moves an object from some unwanted state to a satisfactory one.

There are several issues that still have to be investigated. One of them is the problem of finding execution plan in the case of classification problem where several decision classes are defined. In such a case we consider a vector of concepts and a vector of inclusion degrees. High number of different combinations of inclusion degree changes (exponential w.r.t. the number of classes) makes the training process not feasible. Some additional techniques, e.g., granulation of the space of inclusion degrees, should be adopted.

Acknowledgements

The research has been supported by the grant 8 T11C 025 19 from the Ministry of Scientific Research and Information Technology of the Republic of Poland and by the Research Center at the Polish-Japanese Institute of Information Technology, Warsaw, Poland.

References

1. Jan G. Bazan, Hoa S. Nguyen, Son H. Nguyen, and Andrzej Skowron. Rough set methods in approximation of hierarchical concepts. In Shusaku Tsumoto, Roman W. Slowinski, Jan Komorowski, and Jerzy W. Grzymala-Busse, editors, *Fourth International Conference on Rough Sets and Current Trends in Computing RSCTC*, volume 3066 of *Lecture Notes in Artificial Intelligence*, pages 346–355, Uppsala, Sweden, June 1-5 2004. Springer-Verlag.
2. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice.* Elsevier, Morgan Kaufmann, Cambridge, USA, 2004.
3. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. pages 26–45, 1987.
4. Sinh Hoa Nguyen, Jan Bazan, Hung Son Nguyen, and Andrzej Skowron. Layered learning for concept synthesis. *LNCS Transactions on Rough Sets*, 1 (LNCS 3100):193–214, 2004.
5. Zdzisław Pawlak. Information systems - theoretical foundations. *Information Systems*, 6:205–218, 1981.
6. Zdzisław Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, volume 9 of *D: System Theory, Knowledge Engineering and Problem Solving.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
7. Lech Polkowski and Andrzej Skowron. Rough mereological approach to knowledge-based distributed AI. In J. K. Lee, J. Liebowitz, and J. M. Chae, editors, *Third World Congress on Expert Systems*, pages 774–781, Soeul, Korea, February 5-9 1996. Cognizant Communication Corporation.
8. Lech Polkowski and Andrzej Skowron. Rough mereology: A new paradigm for approximate reasoning. *International Journal of Approximate Reasoning*, 15(4):333–365, 1996.

9. Lech Polkowski and Andrzej Skowron. Rough mereology in information systems. A case study: Qualitative spatial reasoning. In Lech Polkowski, Tsau Young Lin, and Shusaku Tsumoto, editors, *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, volume 56 of *Studies in Fuzziness and Soft Computing*, chapter 3, pages 89–135. Springer-Verlag/Physica-Verlag, Heidelberg, Germany, 2000.
10. John F. Roddick, Kathleen Hornsby, and Myra Spiliopoulou. YABTSSTDMR - yet another bibliography of temporal, spatial and spatio-temporal data mining research. In K. P. Unnikrishnan and R. Uthurusamy, editors, *SIGKDD Temporal Data Mining Workshop*, pages 167–175, San Francisco, CA, 2001. ACM Press.
11. Yoav Shoham. *Reasoning about change: time and causation from the standpoint of artificial intelligence*. MIT Press, Cambridge, MA, USA, 1988.
12. Andrzej Skowron and Cecylia Rauszer. The discernibility matrices and functions in information systems. In Roman Słowiński, editor, *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, volume 11 of *D: System Theory, Knowledge Engineering and Problem Solving*, chapter 3, pages 331–362. Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.
13. Andrzej Skowron and Jarosław Stepaniuk. Information granules and rough-neural computing. In Sankar K. Pal, Lech Polkowski, and Andrzej Skowron, editors, *Rough-Neural Computing: Techniques for Computing with Words*, Cognitive Technologies, pages 43–84. Springer Verlag, Heidelberg, Germany, 2004.
14. Andrzej Skowron and Piotr Synak. Complex patterns in spatio-temporal reasoning. In Ludwik Czaja, editor, *Concurrency Specification And Programming CSP*, volume 2, pages 487–499, Czarna, Poland, September 25-27 2003.
15. Andrzej Skowron and Piotr Synak. Complex patterns. *Fundamenta Informaticae*, 60(1-4):351–366, 2004.
16. Andrzej Skowron, Piotr Synak, and James Peters. Spatio-temporal approximate reasoning over hierarchical information maps. In Hans-Dieter Burkhard, editor, *Concurrency Specification And Programming CSP*, volume 2, pages 358–377, Caputh, Germany, September 24-26 2004.
17. Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.
18. Jarosław Stepaniuk. Approximation spaces, reducts and representatives. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets in Knowledge Discovery 2. Applications, Case Studies and Software Systems*, volume 19 of *Studies in Fuzziness and Soft Computing*, chapter 6, pages 109–126. Physica-Verlag, Heidelberg, Germany, 1998.
19. Piotr Synak, Jan G. Bazan, Andrzej Skowron, and James F. Peters. Spatio-temporal approximate reasoning over complex objects. *Fundamenta Informaticae*, 67(1-3):249–269, 2005.
20. W. Van Wezel, R. Jorna, and A. Meystel. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*. John Wiley & Sons, 2006.