Rough Sets and Approximation Schemes

V. W. Marek[1] and M. Truszczynski[1]

Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046, USA

Abstract. Approximate reasoning is used in a variety of reasoning tasks in Logic-based Artificial Intelligence. In this abstract we compare a number of such reasoning schemes and show how they relate and differ from the approach of Pawlak's Rough Sets.

1 Introduction

Humans reason more often than not with incomplete information. The effect is that the conclusions must often be revised, and treated as approximate. Frequently we face the following situation: some features of objects of interest are firmly established (based on observations and on domain properties), some other are known to be false. But there remains a "grey area" of features of objects of interest that are not determined by the current knowledge. In this note we discuss several schemes that have been proposed in the literature for handling approximate reasoning when available knowledge may be incomplete. They include rough sets [Paw82], approximation for propositional satisfiability [KS96], approximation semantics for logic programs including brave and skeptical answer-set semantics, Kripke-Kleene semantics and well-founded semantics [Kun87,Fit85], the semantics of repairs in databases [ABC03], knowledge compilation of propositional theories [KS96], and least- and largest- pair of fixpoints for the operator associated with a Horn program [Ll87]. For some of these, we will be able to show that they fit into the rough set paradigm.

2 Approximations and three-valued reasoning schemes

We discuss here a variety of approximating schemes. They all have a common feature – they use a three-valued approach to sets of objects.

2.1 Approximations and the ordering $\preceq_{kn}$

Given a set (universe) U, an approximation over U is any pair of subsets of U, X_1, X_2 such that $X_1 \subseteq X_2$. An approximation $\langle X_1, X_2\rangle$ provides bounds on every set X such that $X_1 \subseteq X \subseteq X_2$. The Kleene (or *knowledge*) ordering of approximations [Kl67] is defined as follows:

$$\langle X_1, X_2 \rangle \preceq_{kn} \langle Y_1, Y_2 \rangle \text{ if } X_1 \subseteq Y_1 \text{ and } Y_2 \subseteq X_2.$$

Let $\mathcal{A}_U$ be the set of all approximations in U. The structure $\langle \mathcal{A}_U, \preceq_{kn} \rangle$ is a chain-complete poset. Unless $|X| \leq 1$, this poset is not a lattice. It is a complete lower-semilattice, and the least upper bound exists for any pair of approximations that have an upper bound. The maximal elements of $\langle \mathcal{A}_U \preceq_{kn} \rangle$ are of the form $\langle X, X \rangle$ for $X \subseteq U$. They are called *exact approximations*.

2.2 Rough sets

Rough sets are special class of approximations. Let O be a finite set of objects (universe). Every equivalence relation r in U determines its concept of rough set as follows. For every $X \subseteq O$, *Pawlak's approximation* (or the *rough set* associated with X) is defined as an approximation $\langle \underline{X}, \overline{X} \rangle$ where: $\underline{X}$ is the union of all cosets of r contained in X, and $\overline{X}$ is the union of all cosets of r that have a nonempty intersection with X. The pair $\langle \underline{X}, \overline{X} \rangle$ is an approximation in O. It is characterized [MT99] as the $\preceq_{kn}$-largest approximation $\langle L, U \rangle$ so that:

1. $\langle L, U \rangle$ approximates X
2. The sets L and U are unions of cosets of r.

As each equivalence relation in O determines its own class of rough sets, the question arises how these classes are related. The collection of equivalence relations on a set O (not necessarily finite) determines a complete, but non-distributive lattice, with the refinement ordering $\sqsubseteq$. Specifically, $r_1 \sqsubseteq r_2$ if every coset of r_1 is the union of cosets of r_2. Let $r_1 \sqsubseteq r_2$ be two equivalence relations in O. One can show that for every subset X of O the Pawlak rough sets determined by r_1 and r_2, say $\langle \underline{X}_1, \overline{X} \rangle_1$ and $\langle \underline{X}_1, \overline{X} \rangle_1$, respectively, are related as follows:

$$\langle \underline{X}_1, \overline{X}_1 \rangle \preceq_{kn} \langle \underline{X}_2, \overline{X}_2 \rangle.$$

In other words, the ordering $\sqsubseteq$ in the lattice of equivalence relations on O induces the ordering $\preceq_{kn}$ in the corresponding Pawlak approximations.

2.3 Propositional satisfiability

We consider a fixed set of propositional variables At. A valuation of At is any mapping of At into $\{0, 1\}$. We can identify valuations with the subsets of At as follows. We identify a valuation v with the set $M \subseteq At$ so that $v = \chi_M$, that is, $M = \{p : v(p) = 1\}$. We write v_M for the valuation v that corresponds to M.

Now, let T be a consistent set of formulas of the propositional language $\mathcal{L}_{At}$. Then T determines an approximation $\langle X_1, X_2 \rangle$ in set At as follows: $X_1 = \{p : T \vdash p\}$, and $X_2 = \{p : T \nvdash \neg p\}$. Then $X_1 \subseteq M \subseteq X_2$ for every M such that $v_M \models T$. Let us denote this "canonical" approximation of models of T by $\langle \underline{T}, \overline{T} \rangle$. Then, we have the following property of theories $T_1 \subseteq T_2$ that are consistent and closed under consequence:

$$\langle \underline{T}_1, \overline{T}_1 \rangle \preceq_{kn} \langle \underline{T}_2, \overline{T}_2 \rangle.$$

In other words, the canonical approximation of the theory T_2 is $\preceq_{kn}$ bigger than that of T_1. The maximal approximations (i.e. Pawlak's rough sets in this case) are the complete consistent theories.

2.4 Knowledge compilation

Many tasks in knowledge representation and reasoning reduce to the problem of deciding, given a propositional CNF theory T and a propositional clause φ, whether $T \models \varphi$. This task is coNP-complete. As a way to address this computational difficulty [KS96] proposed an approach in which T is compiled off-line, possibly in exponential time, into some other representation, under which the query answering would be efficient. While there is an initial expense of the compilation, if the query answering task is frequent that cost will eventually be recuperated.

An approximation to a theory T is a pair of theories (T', T') such that

$$T' \models T \models T''.$$

If (T', T'') is an approximation to T, then $T \models \varphi$ if $T'' \models \varphi$, and $T \not\models \varphi$ if $T' \not\models \varphi$. In other words,

$$\{\varphi \colon T'' \models \varphi\} \subseteq \{\varphi \colon T \models \varphi\} \subseteq \{\varphi \colon T' \models \varphi\}.$$

Desirable approximations are "tight", that is, $\{\varphi \colon T' \models \varphi\} \setminus \{\varphi \colon T'' \models \varphi\}$ is small, and support efficient reasoning. Concerning the latter point, if U is a Horn theory and φ is a clause, then $U \models \varphi$ can be decided in polynomial time. Therefore, we define *approximations* to be pairs (T', T''), where T' and T'' are Horn theories such that $T' \models T''$.

A key problem is: given a CNF theory T, find the most precise Horn approximation to T. This problem has been studied in [KS96]. It turns out that there is a unique (up to logical equivalence) Horn least upper bound. However, there is no greatest Horn upper bound. The set of Horn lower approximations has, however, maximal elements.

2.5 Approximating semantics for logic programs

Logic Programming studies semantics of *logic programs*, i.e. sets of *program clauses*. In the simplest case those are expressions of the form $p \leftarrow q_1, \ldots, q_m, \neg r_1, \ldots, \neg r_n$. The meaning of such clause is, informally, this: "if $q_1, \ldots, q_m$ have been derived, and none of $r_1, \ldots, r_n$ has, or ever will be, then derive p" (various different meanings are also associated with program clauses). It is currently commonly assumed that the correct semantics of a logic program (i.e. set of program clauses as above) is provided by means of fixpoints of the Gelfond-Lifschitz operator GL_P. Those fixpoints are called *stable models* of P [GL88], and more recently also *answer sets* for P. The operator GL_P is antimonotone, thus existence of fixpoints of GL_P is not guaranteed. However the operator GL_P^2 is monotone, and thus possesses a least and largest fixpoints.

A number of approximation schemes for stable semantics of logic programs has been proposed. The earliest proposal is the so-called Kripke-Kleene approximation ([Kun87,Fit85]). In this approach, one defines a *three-valued* van-Emden-Kowalski operator $\mathcal{T}_P$. That operator is monotone in the ordering $\preceq_{kn}$, and thus possesses a least $\preceq_{kn}$ fixpoint. That fixpoint (which can be treated as an approximation) approximates all stable models of the logic program P. A stronger approximation scheme has been proposed in [VRS91], and is called a *well-founded model* of the program. Essentially, that model is defined by means of the least and largest fixpoint of GL_P^2. Like the Kripke-Kleene fixpoint, the well-founded approximations provides an approximation to all stable models of the program. Yet another approximation scheme which turns out to be stricter than the well-founded semantics is the *ultimate approximation* of [DMT04].

Of course, one can assign to a logic program P the $\preceq_{kn}$-largest approximation for the family of all stable models of P. Let us denote by KK_P the Kripke-Kleene approximation, WF_P the well-founded approximation, U_P, the ultimate approximation and A_P the most precise approximation of all stable models of P. Then, assuming P possesses a stable model, we have

$$KK_P \preceq_{kn} WF_P \preceq_{kn} U_P \preceq_{kn} A_P$$

and examples can be given where all the relationships are strict. The complexity of computing each of these approximations is also different, in general. Nevertheless, these constructions assign, on analogy to rough sets, approximations to programs. Thus, in case of Logic Programming approximations there exist a classification of approximations to the family of all stable models of the program.

We note the the Kripke-Kleene approximation KK_P approximates not only all stable models of P but also all supported models of P. In the case when P is a Horn program the fixpoint KK_P is given by the pair (S_l, S_u), where S_l is the least and S_u is the greatest supported model of P (which are guaranteed to exist).

2.6 Approximating possible-world structures

The language of modal logic with the semantics of autoepistemic expansions and extensions [DMT03] provides a way to describe approximations to possible-world structures. Let us consider a theory T in a language of propositional modal logic. The theory T is meant to describe a possible world structure providing the account of what is known and what is not known given T.

Since T may be incomplete, there may be several possible-world structures one could associate with T (autoepistemic logic provides a specific characterization of such structures; other nonmonotonic modal logics could be used, too [MT93]). To reason about the epistemic content of T one has two choices: to compute all possible-world structures for T according to the semantics of the autoepistemic logic, or compute an approximation to the epistemic content of T common to all these structures. The former is computationally complex, being a Σ_P^2-task. Hence, the latter is often the method of choice.

At least three different approximations can be associated with T, Kripke-Kleene approximation, the well-founded approximation and the ultimate approximation, listed here according to the precision, with which they approximate possible-world structures of T [DMT03,DMT04]. It is worth noting that the computational complexity of each of these approximations is lower that the complexity of computing even a (single) possible-world structure for T.

2.7 Minimal models reasoning and repairs in databases

Approximations play an important role in the theory and practice of databases. In this paper, we regard a database as a finite structure of some language $\mathcal{L}$ of first-order logic that does not contain function symbols. Typically, legal databases are subject to *integrity constraints*, properties that at any time the database is supposed to have. In general, integrity constraints can be represented as arbitrary formulas of $\mathcal{L}$.

Databases are frequently modified over their lifetime. Updates create the possibility of entering erroneous data, especially that in most cases databases are modified by different users at different locations. Consequently, it does happen that databases do not satisfy the integrity constraints. Once such a situation occurs, the database needs to be *repaired* [ABC03].

Let D be a database and let IC be a set of integrity constraints. A pair $R = (R^+, R^-)$ is a *repair* of D with respect to IC if $(D \cup R^+) \setminus R^- \models IC$ (the repair condition), and for every (Q^+, Q^-) such that $Q^+ \subseteq R^+$, $Q^- \subseteq R^-$, and $(D \cup Q^+) \setminus Q^- \models IC$, we have $Q^+ = R^+$ and $Q^- = R^-$ (the minimality condition). We write $R(D)$ for the database $(D \cup R^+) \setminus R^-$ resulting from D by applying a repair R. We write $Rep(D, IC)$ to denote all repairs of D with respect to IC. The minimality condition implies that if (R^+, R^-) is a repair, then $R^+ \cap D = \emptyset$ and $R^- \subseteq D$.

Repairing a database D that violates its integrity constraints IC consists of computing a repair $R \in Rep(D, IC)$ and applying it to D, that is computing $R(D)$. There are two problems, though. First, computing repairs is computationally complex (even in some simple settings deciding whether repairs exist is Σ_P^2-complete). Second, it often is the case that multiple repairs exist, which results in the need for some principled selection strategy.

These problems can be circumvented to some degree by modifying the semantics of the database. Namely, a database D with integrity constraints IC could be viewed as an *approximation* to an actual database D', not available explicitly but obtainable from D by means of a repair with respect to IC. The approximation to D' represented by (D, IC) is the pair of sets (D_l, D_u), where

$$D_l = \bigcap \{R(D)\colon R \in Rep(D, IC)\} \text{ and } D_l = \bigcup \{R(D)\colon R \in Rep(D, IC)\}.$$

In other words, expressions (D, IC) define approximations, and query answering algorithms have to be adjusted to provide best possible answers to queries to D' based on the knowledge of D_l and D_u only.

3 Further work, and conclusions

We discussed a number of approximation schemes as they appear in logic, logic programming, artificial intelligence, and databases. Doubtless there are other approaches to approximate reasoning that can be cast as approximations, and in particular rough sets. One wonders if there is a classification of approximations that allows to capture a common structure laying behind these, formally different, approaches. In other words, are there general classification principles for approximations? Are there categories of approximations that allow to classify approximations qualitatively?

Another fundamental issue is the use of languages that describe approximations. Pawlak [Paw91] noticed that, in its most abstract form, rough sets are associated with equivalence relations; each equivalence relation induces its own rough set notion. Such abstract approach leads to Universal Algebra considerations that have roots in [JT51] and have been actively pursued by Orłowska and collaborators [DO01,OS01,SI98]. One can find even more abstract versions within the Category Theory. But usually, the applications of rough sets and other approximation schemes cannot choose its own language. For instance, more often than not (and this was the original motivation of Pawlak) the underlying equivalence relation is given to the application (for instance as the equivalence induced by an information system [MP76]). Then, and the literature of rough sets is full of such considerations, one searches for the coarser equivalence relations that are generated by various attribute reduction techniques. To make the point, these equivalence relations are not arbitrary, but determined by the choice of the language used for data description. This linguistic aspect of rough sets and approximations in general, needs more attention of rough set community.

Acknowledgments

This research was supported by the National Science Foundation under Grant IIS-0325063 and by the Kentucky Science and Engineering Foundation under Grant 1036-RDE-008.

References

[ABC03] Arenas, M., Bertossi, L.E., and Chomicki, J. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming* 3(4-5): 393-424. 2003.

[DP92] Davey, B.A., and Priestley, H.A., *Introduction to Lattices and Order*, Cambridge University Press, 1992.

[DMT03] Denecker, M., Marek, V., and Truszczyński, M.: Uniform semantic treatment of default and autoepistemic logics. Artificial Intelligence Journal **143** (2003) 79–122

[DMT04] Denecker, M., Marek, V., and Truszczyński, M.: Ultimate approximation and its application in nonmonotonic knowledge representation systems. Information and Computation **192** (2004) 84–121

[DO01] Düntsch, I. and Orłowska, E. Beyond Modalities: Sufficiency and Mixed Algebras. Chapter 16 of [OS01], 2001.

[Fit85] Fitting, M.C. A Kripke-Kleene semantics for logic programs. ,newblock *Journal of Logic Programming*, 2(4):295–312, 1985.

[GL88] Gelfond, M. and Lifschitz, V. The stable model semantics for logic programming. In *Proceedings. of the International Joint Conference and Symposium on Logic Programming*. MIT Press, 1070–1080, 1988.

[Jo91] Jonsson, B. A Survey of Boolean Algebras with Oprators. In: *Algebras and Order*, pages 239–284. Kluwer, 1991.

[JT51] Jonsson, B. and Tarski, A. Boolean Algebras with Operators. *American Journal of Mathematics* 73:891–939, 1951.

[Kl67] Kleene, S.C. *Introduction to Metamathematics*. North-Holland, 1967. Fifth reprint.

[Kun87] Kunen, K.. Negation in logic programming. *Journal of Logic Programming*, 4(4):289–308, 1987.

[Ll87] Lloyd, J.W. *Foundations of Logic Programming*, Springer 1987.

[MP76] Marek, W. and Pawlak, Z. Information storage and retrieval systems, mathematical foundations, *Theoretical Computer Science* 1(4):331–354, 1976.

[MP84] Marek, W. and Pawlak, Z. Rough sets and information systems, *Fundamenta Informaticae* 7(1):105–115, 1984.

[MT93] Marek, V.W., and Truszczyński, M.: Nonmonotonic Logic; Context-Dependent Reasoning. Springer, Berlin (1993)

[MT99] Marek, V.W., Truszczynski, M. Contributions to the Theory of Rough Sets. *Fundamenta Informaticae* 39(4): 389-409. 1999.

[OS01] Orłowska, E. and Szałas, A. *Relational Methods for Computer Science Applications. Selected Papers from 4th International Seminar on Relational Methods in Logic, Algebra and Computer Science (RelMiCS'98)*, Studies in Fuzziness and Soft Computing 65, Physica-Verlag/Springer, 2001.

[Paw82] Pawlak, Z.. Rough Sets, *International Journal of Computer and Information Sciences* 11:341–356, 1982.

[Paw91] Pawlak, Z., *Rough Sets – theoretical aspects of reasoning about data*, Kluwer, 1991.

[SI98] SanJuan, E. and Iturrioz, L. Duality and informational representability of some information algebras. pages 233-247, in: *Rough Sets in Knowledge Discovery, Methodology and Applications*, L. Polkowski and A. Skowron, (eds). Physica-Verlag, 1998.

[KS96] Selman, B. and Kautz, H. Knowledge Compilation and Theory Approximation *Journal of the ACM*, 43(2):193-224, 1996

[VRS91] Van Gelder, A., Ross, K.A., and Schlipf, J.S. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.