Rough Forgetting

Patrick Doherty[1,2] and Andrzej Szałas[1,3](✉)

[1] Department of Computer and Information Science, Linköping University,
581 83 Linköping, Sweden
{patrick.doherty,andrzej.szalas}@liu.se

[2] School of Intelligent Systems and Engineering, Jinan University (Zhuhai Campus),
Zhuhai, China

[3] Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
andrzej.szalas@mimuw.edu.pl

Abstract. Recent work in the area of Knowledge Representation and Reasoning has focused on modification and optimization of knowledge bases (KB) through the use of *forgetting operators* of the form $forget(KB, \bar{R})$, where $\bar{R}$ is a set of relations in the language signature used to specify the KB. The result of this operation is a new KB where the relations in $\bar{R}$ are removed from the KB in a principled manner resulting in a more efficient representation of the KB for different purposes. The forgetting operator is also reflected semantically in terms of the relation between the original models of the KB and the models for the revised KB after forgetting. In this paper, we first develop a rough reasoning framework where our KB's consist of rough formulas with a semantics based on a generalization of Kleene algebras. Using intuitions from the classical case, we then define a forgetting operator that can be applied to rough KBs removing rough relations. A constructive basis for generating a new KB as the result of applying the forgetting operator to a rough KB is specified using second-order quantifier elimination techniques. We show the application of this technique with some practical examples.

1 Introduction and Motivations

In Artificial Intelligence, the field of Knowledge Representation and Reasoning (KRR) deals with the use of logical languages to represent knowledge or beliefs and the use of inference in some logic to derive additional knowledge or belief implicit in a base theory represented as a set of logical formulas. The explicit base theory is often called a *Knowledge Base* (KB). Consequences A of the KB are derived through a consequence relation, KB $\models A$. A signature Σ (vocabulary) is

The first author has been supported by the ELLIIT Network Organization for Information and Communication Technology, Sweden; the Swedish Foundation for Strategic Research SSF (SymbiKBot Project); and a guest professor grant from Jinan University (Zhuhai Campus). The second author has been supported by grant 2017/27/B/ST6/02018 of the National Science Centre Poland.

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 3–18, 2020.
https://doi.org/10.1007/978-3-030-52705-1_1

associated with the logical language and is used for specifying the legal relations, functions, constants, etc., used in the syntax of formulas.

In recent years, there has been much interest in the topic of *forgetting* operations in KRR [12]. Intuitions for such operators are based loosely on the fact that humans often forget what they know or believe for reasons of efficiency in reasoning. Mapping this loose intuition over to KRR results in some very powerful and useful techniques for dealing with redundant information in KB's, optimizing query retrieval in relation to KB's [10,17], progressing databases [18], forgetting with description logics [6,31] and rule based languages [26,27], dealing with missing information and dataset reduction [14], forgetting sets of literals in first-order logic [28], in addition to other techniques. In general, one major type of forgetting aims at removing information from a KB in a controlled manner where the syntactic elimination has a principled semantic correlation characterized in model theory.

Given a KB and a signature Σ, a common type of forgetting, $\mathit{forget}(KB, \Sigma')$, can be formulated where $\Sigma' \subsetneq \Sigma$ and KB' is the result of forgetting the components in Σ' in KB. One interesting question is the relation between the models and consequences of KB and the consequences of KB' after the forgetting operation is applied to KB. Initial intuitions for this type of forgetting can be traced all the way back to Boole [2] and his use of variable elimination. Assume a propositional language with signature $\Sigma = \{p, q, r\}$. Given a propositional formula A and a signature $\Sigma' = \{p\}$, the result of forgetting p in A, $\mathit{forget}(A, \Sigma')$, is $A_p^+ \vee A_p^-$, where A_p^+ is the result of replacing all occurrences of p in A with 'true' and A_p^- is the result of replacing all occurrences of p in A with 'false' and simplifying the result.

In KRR application areas such as robotics, the knowledge or beliefs robots have about different aspects of the world, is often incomplete and/or uncertain. Consequently, one wants to find a concise way to model this. Rough set theory [7,8,22,23] has been used to model different types of incompleteness using indiscernibility and approximations. The general idea is to begin with a universe of individuals and define an indiscernibility relation over these individuals. In the classical case, this generates an equivalence relation over individuals. A rough set is defined by specifying a lower and upper approximation, each consisting of a number of equivalence classes generated by the indiscernibility relation. All individuals in equivalence classes included in the lower approximation are in the rough set, all equivalence classes in the upper approximation intersect with the rough set, and the individuals in the remaining equivalence classes lie outside the set. This brings to mind a division of individuals into a tripartite division reminiscent of three-valued logics. Later in the paper, this intuition will be formalized more precisely. In the context of KRR, there has been interest in generalizations of logical languages and inference to include rough logical languages and inference using rough theories [8]. This generalization will be used as a vehicle for specifying rough forgetting operators applied to rough relations in such logics.

Another application area for rough sets and logics is with big data applications. According to [14],

> "most of the attribute values relating to the timing of big data [...] are missing due to noise and incompleteness. Furthermore, the number of missing links between data points in social networks is approximately 80% to 90% and the number of missing attribute values within patient reports transcribed from doctor diagnoses are more than 90%."

Rough sets are discussed in [14] as one of remedies to deal with missing data. In this context, the combination of rough sets with the use of forgetting operations might prove to be very useful. In cases where important information is missing, it might be useful to forget the relation or find a relation's explicit definition and – using the definition – complete parts of the missing content. In fact, the second-order quantifier elimination techniques which we describe in this paper and use as a tool for forgetting, provide us with definitions of eliminated (forgotten) relations as a *side effect.*

This paper is primarily about developing a first-order logical framework for rough theories that can be used to construct rough KB's, with a formal semantics based on rough relational structures. Given such a logic, we then define a forgetting operator that can be applied to rough theories and we provide the semantics for such an operator. The forgetting operator is based on second-order quantifier elimination techniques developed for rough theories. In previous work, we have shown how second-order quantifier elimination techniques can be automated for well-behaved fragments of second-order logic. We expand on these results in the context of rough theories.

The paper is structured as follows. In Sect. 2 we discuss the rough reasoning framework used throughout the paper. In Sect. 3, we recall definitions for forgetting used with classical logic and then generalize these and introduce rough forgetting. Next, in Sect. 4, we provide second-order quantifier elimination theorems with proofs which can serve as foundations for algorithmic techniques for rough forgetting. Section 5 provides a number of examples showing how the proposed techniques work in practice. Finally, Sect. 6 concludes the paper.

2 Rough Reasoning Framework

Rough sets [21,22] have been defined in many ways (see, e.g., [4,5,7,8,16,23,25,29,30] and numerous references there). Three- and many-valued approaches have been intensively studied in the context of rough sets [3,4,15,16]. In the current paper we will follow the presentation of [16].

Definition 1 (Approximation space). *Let $\mathcal{U}$ be a set of objects and E be an equivalence relation on $\mathcal{U}$, Then $\mathcal{A} = \langle \mathcal{U}, E \rangle$ is called an* approximation space*. By the* lower approximation *(s^+) and* upper approximation *($s^\oplus$) of a set $s \subseteq \mathcal{U}$ we mean:*

$$s^+ \stackrel{\text{def}}{=} \{x \in \mathcal{U} \mid \forall y\big(E(x,y) \to y \in s\big)\}; \quad s^\oplus \stackrel{\text{def}}{=} \{x \in \mathcal{U} \mid \exists y\big(E(x,y) \land y \in s\big)\}. \quad (1)$$

A set $s \subseteq \mathcal{U}$ is definable *in $\mathcal{A}$ iff s is a union of equivalence classes of E.* ∎

In rough sets, E represents an indiscernibility relation. Approximations are interpreted as follows, where $s \subseteq \mathcal{U}$ is a set:

- the lower approximation s^{+} represents objects certainly belonging to s;
- the upper approximation $s^{\oplus}$ represents objects possibly belonging to s.

Definition 2 (Rough sets). *For an approximation space $\mathcal{A}$, the ordered pair $\langle s_l, s_u \rangle$, where $s_l \subseteq s_u$ and s_l, s_u are definable sets, is called a* rough set *(wrt $\mathcal{A}$).*[1] ■

Remark 1. In the literature, the equivalence relation used to define rough approximations has been argued to be too strong for many application areas [8,24,25]. In fact, seriality of E (i.e., the property that $\forall x \exists y (E(x,y))$ has been proposed as the weakest well-behaved requirement on E. This ensures that the lower approximation is included in the upper approximation of a rough set [11,29].

Note also that, according to [19, Section 19.3], every reflexive similarity relation can be refined to an equivalence relation in a natural way. So reflexivity can be used as a basic requirement on indiscernibility relations.[2] ■

As shown in [16], there is a close correspondence between rough sets and Kleene algebras defined below.

Definition 3 (Kleene algebra). *An algebra $\mathcal{K} = \langle K, \cup, \cap, -, \bot, \top \rangle$ is called a* Kleene algebra *if the following hold.*

1. *$\mathcal{K}$ is a De Morgan algebra, i.e., $\langle K, \cup, \cap, \bot, \top \rangle$ is a distributive lattice with the greatest element $\top$ and the least element $\bot$, and for all $s, t \in K$,*
   (a) $-(s \cap t) = -s \cup -t$ (De Morgan property),
   (b) $- - s = s$ (involution).
2. *$s \cap -s \leq t \cup -t$, for all $s, t \in K$ (Kleene property).* ■

Note that in Definition 3 we refer to “greatest” and “least” elements. As usual in lattice theory, we mean the ordering:

$$s \leq t \overset{\text{def}}{\equiv} (s = s \cap t), \ \text{(equivalently: } t = s \cup t). \tag{2}$$

For rough sets, a subclass of Kleene algebras, rough Kleene algebras, will have the role of Boolean algebras for classical sets.

Definition 4 (Rough Kleene algebra). *Let $\mathcal{U}$ be a set of objects. A Kleene algebra $\mathcal{K} = \langle K, \cup, \cap, -, \bot, \top \rangle$ is called a* rough Kleene algebra *over $\mathcal{U}$ iff:*

[1] The set s_l serves as the *lower approximation* and s_u – as the upper approximation of a set.

[2] Note that reflexivity implies seriality.

- *K consists of pairs of sets $\langle s_l, s_u\rangle$ such that $s_l \subseteq s_u \subseteq \mathcal{U}$;*
- $\bot \stackrel{\text{def}}{=} \langle\emptyset,\emptyset\rangle, \top \stackrel{\text{def}}{=} \langle\mathcal{U},\mathcal{U}\rangle$*;*
- $-\langle s_l, s_u\rangle \stackrel{\text{def}}{=} \langle -s_u, -s_l\rangle$*.*

By a generalized rough set *we mean any element of $\mathcal{K}$.* ■

As the logical counterpart of rough Kleene algebras we will use the three valued logic of Kleene, K_3, with truth values T (true), F (false) and U (unknown), ordered by:

$$\mathrm{F} < \mathrm{U} < \mathrm{T}, \tag{3}$$

with connectives $\vee, \wedge, \neg$. The semantics of connectives is defined by:

$$\tau_1 \vee \tau_2 \stackrel{\text{def}}{=} \max\{\tau_1, \tau_2\}; \quad \tau_1 \wedge \tau_2 \stackrel{\text{def}}{=} \min\{\tau_1, \tau_2\}; \tag{4}$$

$$\neg\mathrm{F} \stackrel{\text{def}}{=} \mathrm{T}; \quad \neg\mathrm{U} \stackrel{\text{def}}{=} \mathrm{U}; \quad \neg\mathrm{T} \stackrel{\text{def}}{=} \mathrm{F}, \tag{5}$$

where $\tau_1, \tau_2 \in \{\mathrm{F}, \mathrm{U}, \mathrm{T}\}$ and max, min are the maximum and minimum wrt (3).

Let us now define the syntax of rough formulas used in this paper. In addition to connectives $\neg, \wedge, \vee$ and quantifiers $\forall, \exists$ of Kleene logic K_3, we add two connectives: $\in$ and $\subseteq$. Their intended meaning is rough set membership and rough set inclusion, respectively.

Definition 5 (Syntax of rough formulas). *Let V be a set of first-order variables (representing domain elements), C be a set of constants and R be a set of relation symbols. Then:*

- Kleene formulas*, KF, are defined by the grammar:*

$$\begin{aligned}\langle KF\rangle ::= \langle R\rangle \mid \neg\langle KF\rangle \mid \langle KF\rangle \vee \langle KF\rangle \mid \langle KF\rangle \wedge \langle KF\rangle \mid \\ \exists\langle V\rangle\langle KF\rangle \mid \forall\langle V\rangle\langle KF\rangle;\end{aligned}$$

- rough formulas*, RF, are defined by the grammar, where $\overline{C \cup V}$ denotes tuples consisting of constants and/or variables:*

$$\begin{aligned}\langle RF\rangle ::= \langle KF\rangle \mid \langle\overline{C \cup V}\rangle \in \langle KF\rangle \mid \langle KF\rangle \subseteq \langle KF\rangle \mid \\ \neg\langle RF\rangle \mid \langle RF\rangle \vee \langle RF\rangle \mid \langle RF\rangle \wedge \langle RF\rangle \mid \\ \exists\langle V\rangle\langle RF\rangle \mid \forall\langle V\rangle\langle RF\rangle.\end{aligned}$$

An occurrence of a variable is called bound *in a formula if it appears inside the scope of a quantifier. It is called* free *when it is not bound.* ■

Rough theories (rough knowledge bases) are defined below.

Definition 6 (Rough theories, rough knowledge bases). *Finite sets of rough formulas are called* rough theories *(or* rough knowledge bases*). A finite set of formulas T is understood as a single formula being the conjunction of formulas in T: $\bigwedge_{A \in T} A$.* ■

Remark 2. In the rest of the paper we will often use the traditional syntax for relations. For example, rather than writing $\forall x \exists y \big((x, y) \in r \big)$, we will write $\forall x \exists y \big(r(x, y) \big)$. ■

Definition 7 (Rough literals and facts). *By a* rough literal *we mean an expression of the form $\pm r(\bar{e})$, where $\pm$ is the empty symbol or $\neg$, r is a relation symbol and $\bar{e}$ is a tuple of constants and/or variables. By a* rough fact *we mean a rough literal not containing variables.* ■

The following important property, justifying the use of K_3 in the context of rough forgetting, is an immediate consequence of Theorems 8, 11, 15, proved in [16]. Below:

- $A_{\mathcal{K}}$ is the class of Kleene algebras;
- RS is the class of rough Kleene algebras;
- $A \models_{t,f} B$ iff for every assignment $w : RF \longrightarrow \{\mathrm{F}, \mathrm{U}, \mathrm{T}\}$,
  - $w(A) = \mathrm{T}$ implies $w(B) = \mathrm{T}$, and
  - $w(B) = \mathrm{F}$ implies $w(A) = \mathrm{F}$.

Corollary 1. *For any rough formulas $A, B \in RF$:*

$$A \models_{A_{\mathcal{K}}} B \text{ iff } A \models_{t,f} B \text{ iff } A \models_{RS} B, \tag{6}$$

where $\models_{A_{\mathcal{K}}}$ and $A \models_{RS}$ are semantic consequence relations for $A_{\mathcal{K}}$ and RS, respectively. ■

To define the semantics of rough formulas, we first need a generalization of relational structures to their rough version.

Definition 8 (Rough relational structures). *Let $\mathcal{U}$ be a set of objects, $\mathcal{K}$ be a rough Kleene algebra over $\mathcal{U}$ and $n \geq 1$ be a natural number. By an n-argument* rough relation *over $\mathcal{U}$ we mean any generalized rough set consisting of tuples of the Cartesian product $\mathcal{U}^n$. By a* rough relational structure *we mean $\langle \mathcal{U}, r_1, \ldots, r_k \rangle$ where for $1 \leq i \leq k$, r_i is an n_i-argument rough relation over $\mathcal{U}$. One-argument rough relations are called* rough concepts *and two-argument ones are called* rough roles. ■

The semantics of rough formulas is defined below, where $A(x \leftarrow a)$ denotes the formula obtained from A by substituting all free occurrences of variable x in A by constant a.

Definition 9 (Semantics of rough formulas). *Let $\mathcal{U}$ be a set, $\mathcal{K} = \langle K, \cup, \cap, -, \bot, \top \rangle$ be a rough Kleene algebra over $\mathcal{U}$ and $\mathcal{R} = \langle \mathcal{U}, r_1, \ldots, r_k \rangle$ be a rough relational structure,*

1. *The* value of a rough formula*, $vs_{\mathcal{R}} : KF \longrightarrow K$, is inductively defined by:*
   - *for a relation symbol r, $vs_{\mathcal{R}}(r) \stackrel{\text{def}}{=} r^{\mathcal{R}}$ where $r^{\mathcal{R}}$ is the relation r in $\mathcal{R}$;*[3]
   - $vs_{\mathcal{R}}(\neg A) \stackrel{\text{def}}{=} -vs_{\mathcal{R}}(A)$;
   - $vs_{\mathcal{R}}(A \vee B) \stackrel{\text{def}}{=} vs_{\mathcal{R}}(A) \cup vs_{\mathcal{R}}(B)$;
   - $vs_{\mathcal{R}}(A \wedge B) \stackrel{\text{def}}{=} vs_{\mathcal{R}}(A) \cap vs_{\mathcal{R}}(B)$;
   - $v_{\mathcal{R}}\big(\exists x(A(x))\big) \stackrel{\text{def}}{=} \bigcup\limits_{a \in \mathcal{U}} vs_{\mathcal{R}}\big(A(x \leftarrow a)\big)$;
   - $v_{\mathcal{R}}\big(\forall x(A(x))\big) \stackrel{\text{def}}{=} \bigcap\limits_{a \in \mathcal{U}} vs_{\mathcal{R}}\big(A(x \leftarrow a)\big)$.
2. *The* truth value *of a rough formula, $v_{\mathcal{R}} : RF \longrightarrow \{F, U, T\}$, is defined inductively:*
   - *for a Kleene formula A with k free variables, $\bar{a} \in \mathcal{U}^k$, and $vs_{\mathcal{R}}(A) = \langle r_l, r_u \rangle$,*
     $$v_{\mathcal{R}}\big(\bar{a} \in A\big) \stackrel{\text{def}}{=} \begin{cases} T \text{ when } a \in r_l; \\ U \text{ when } a \in r_u \setminus r_l; \\ F \text{ when } a \in U \setminus r_u. \end{cases}$$
   - $v_{\mathcal{R}}(A \subseteq B) \stackrel{\text{def}}{=} \begin{cases} T \text{ when for all } \bar{a} \in \mathcal{U}^k, v_{\mathcal{R}}(\bar{a} \in A) \leq v_{\mathcal{R}}(\bar{a} \in B); \\ F \text{ otherwise,} \end{cases}$
     where A, B are Kleene formulas with k free variables, and $\leq$ is the reflexive closure of (3);
   - $v_{\mathcal{R}}(\neg A) \stackrel{\text{def}}{=} \neg v_{\mathcal{R}}(A)$*, for* $\circ \in \{\vee, \wedge\}, v_{\mathcal{R}}(A \circ B) \stackrel{\text{def}}{=} v_{\mathcal{R}}(A) \circ v_{\mathcal{R}}(B)$*,*
     where the semantics of $\neg, \vee, \wedge$ on truth values is defined by (4)–(5);
   - $v_{\mathcal{R}}\big(\exists x(A(x))\big) \stackrel{\text{def}}{=} \max\limits_{a \in \mathcal{U}} \big\{v_{\mathcal{R}}\big(A(x \leftarrow a)\big)\big\}$*, where* max *is the maximum wrt (3);*
   - $v_{\mathcal{R}}\big(\forall x(A(x))\big) \stackrel{\text{def}}{=} \min\limits_{a \in \mathcal{U}} \big\{v_{\mathcal{R}}\big(A(x \leftarrow a)\big)\big\}$*, where* min *is the minimum wrt (3).*

We write $\mathcal{R} \models A$ to indicate that $v_{\mathcal{R}}(A) = T$. We say that formulas A and B are equivalent*, iff for every R, $v_R(A) = v_R(B)$.* ■

3 Forgetting and Rough Forgetting

In the rest of the paper, we assume that knowledge bases are given in the form of finitely axiomatizable theories. As indicated in Definition 6, each theory consisting of a finite set of axioms is understood as a single formula, being the conjunction of the axioms.

[3] To simplify notation, we use the same notation for relation symbols and corresponding rough relations. Similarly, objects in $\mathcal{U}$ are identified with constants denoting them.

3.1 Forgetting

The following definition, theorem and example have been formulated in [18].

Definition 10 (Forgetting). *Let r be a relation symbol and $\mathcal{M}_1$, $\mathcal{M}_2$ be relational structures. Then $\mathcal{M}_1 \sim_r \mathcal{M}_2$ denotes the fact that $\mathcal{M}_1$ differs from $\mathcal{M}_2$ at most in the interpretation of r.*

Let T be a theory. A theory T' is a result of forgetting r in T *iff for any relational structure $\mathcal{M}'$, $\mathcal{M}' \models T'$ iff there is a relational structure $\mathcal{M}$ such that $\mathcal{M} \models T$ and $\mathcal{M} \sim_r \mathcal{M}'$. By $\mathit{forget}(T; r)$ we denote the result of forgetting r in T.* ■

In the rest of the paper $T(r \leftarrow X)$ denotes the formula resulting from $T(r)$ by replacing every occurrence of r in T by X.

Theorem 1. *Let r be a relation symbol and X be a second-order variable with the same number of arguments as r. Then $\mathit{forget}(T; r) \equiv \exists X\big(T(r \leftarrow X)\big)$.* ■

Example 1. Let $T \equiv \big((\mathit{student}(\mathit{joe}) \vee \mathit{student}(\mathit{john})) \wedge \mathit{teacher}(\mathit{john})\big)$. Note that:

$$\begin{aligned}\big((\mathit{student}(\mathit{joe}) \vee \mathit{student}(\mathit{john})) \wedge \mathit{teacher}(\mathit{john})\big)\big(\mathit{student} \leftarrow X\big)\big) = \\ (X(\mathit{joe}) \vee X(\mathit{john})) \wedge \mathit{teacher}(\mathit{john}).\end{aligned} \tag{7}$$

Using Theorem 1 and (7) we have:

$$\mathit{forget}(T; \mathit{student}) = \exists X\big((X(\mathit{joe}) \vee X(\mathit{john})) \wedge \mathit{teacher}(\mathit{john})\big). \tag{8}$$

It can be easily shown that the formula $\exists X\big((X(\mathit{joe}) \vee X(\mathit{john})) \wedge \mathit{teacher}(\mathit{john})\big)$, thus $\mathit{forget}(T; \mathit{student})$ too, is equivalent to $\mathit{teacher}(\mathit{john})$. ■

Theorem 1 shows that the problem of computing $\mathit{forget}(T; r)$ can be reduced to second-order quantifier elimination. For this purpose, in the current paper we will adapt the techniques of [1,20] to rough theories.[4]

3.2 Rough Forgetting

Rough forgetting is defined by analogy with Definition 10.

Definition 11 (Rough forgetting). *Let r be a relation symbol and $\mathcal{R}_1$, $\mathcal{R}_2$ be rough relational structures. Then $\mathcal{R}_1 \approx_r \mathcal{R}_2$ denotes the fact that $\mathcal{R}_1$ differs from $\mathcal{R}_2$ at most in the interpretation of r.*

Let $\mathcal{T}$ be a rough theory. A theory $\mathcal{T}'$ is a result of rough forgetting r in $\mathcal{T}$ *iff for any rough relational structure $\mathcal{R}'$, $\mathcal{R}' \models \mathcal{T}'$ iff there is a rough relational structure $\mathcal{R}$ such that $\mathcal{R} \models \mathcal{T}$ and $\mathcal{R} \approx_r \mathcal{R}'$. By $\mathit{rforget}(\mathcal{T}; r)$ we denote the formula being the result of rough forgetting of r in $\mathcal{T}$.* ■

[4] For a broad discussion of related second-order quantifier elimination techniques see [13].

As in the case of classical forgetting, we have the following theorem analogous to Theorem 1, where we use a second-order quantifier, whose semantics is defined by:[5]

$$v_{\mathcal{R}}\big(\exists X\big(\mathcal{T}(X)\big) \overset{\text{def}}{=} \max_{s \in K}\{v_{\mathcal{R}}\big(\mathcal{T}(X \leftarrow s)\big)\}, \tag{9}$$

where max, min are the maximum and minimum wrt (3).

Theorem 2. *Let r be a rough relation symbol and X be a second-order variable with the same number of arguments as r. Then for every rough relational structure $\mathcal{R}$:*

$$v_{\mathcal{R}}\big(\mathit{rforget}(\mathcal{T}; r)\big) = v_{\mathcal{R}}\big(\exists X\big(\mathcal{T}(r \leftarrow X)\big)\big). \qquad \blacksquare$$

Comparing to classical forgetting, in rough forgetting we deal with rough relations rather than with the classical relations. Thus, $\exists X$ in Theorem 2 is a second-order quantification over rough sets rather than over the classical ones.

4 Eliminating Second-Order Quantifiers from Rough Formulas

To formalize second-order quantifier elimination methods and related concepts, we need a notation $A(X \leftarrow B[\bar{z}])$ defined as follows. Let A, B be rough formulas such that A contains an n-argument second-order variable X and $\bar{z}$ is a tuple of n first-order variables with free occurrences in formula B. Then:

$$A(X \leftarrow B[\bar{z}])$$

denotes the result of substituting all occurrences of the second-order variable X by $B(\bar{z})$, where $\bar{z}$ in B is respectively substituted by actual parameters of X (possibly different in different occurrences of X). For example,

$$\big(\underbrace{X(a) \vee X(b)}_{A(X)}\big)\big(X \leftarrow \underbrace{r(z, y)}_{B(z,y)}[z]\big) \text{ is } (r(a, y) \vee r(b, y)).$$

The quantifier elimination techniques we develop are based on a monotonicity property, defined as follows.

Definition 12 (Monotonicity). *Let X be a second-order variable representing n-argument relations and let $\bar{z}$ be a tuple consisting of n (first-order) variables. We say that a rough formula $A(X)$ is* monotone *in X iff for every rough relational structure $\mathcal{R}$ and rough formulas B, C not containing X and with $\bar{z}$ being all variables with free occurrences, one of the following properties holds:*

$$v_{\mathcal{R}}(B(\bar{z})) \leq v_{\mathcal{R}}(C(\bar{z})) \text{ implies } v_{\mathcal{R}}\big(A(X \leftarrow B[\bar{z}])\big) \leq v_{\mathcal{R}}\big(A(X \leftarrow C[\bar{z}])\big); \tag{10}$$

$$v_{\mathcal{R}}(B(\bar{z})) \leq v_{\mathcal{R}}(C(\bar{z})) \text{ implies } v_{\mathcal{R}}\big(A(X \leftarrow C[\bar{z}])\big) \leq v_{\mathcal{R}}\big(A(X \leftarrow B[\bar{z}])\big). \tag{11}$$

[5] Recall that K is the universe of a rough Kleene algebra $\mathcal{K}$, fixed earlier.

Properties (10) *and* (11) *are called* up-monotonicity *and* down-monotonicity of A*, respectively.* ■

The following theorem adapts Ackermann's Lemma [1,9,13] to rough theories.

Theorem 3. *Let X be an n-argument second-order variable. Let $\bar{z}$ be an n-tuple of variables, $A(\bar{z})$ be a rough formula containing no occurrences of X, with variables $\bar{z}$ occurring free, and let $B(X)$ be a rough formula with X as a free variable.*

1. If $B(X)$ is down-monotone in X then for every rough relational structure $\mathcal{R}$,

$$v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z})) \land B(X)\big)\Big) = v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big). \tag{12}$$

2. If $B(X)$ is up-monotone in X then for every rough relational structure $\mathcal{R}$,

$$v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(X(\bar{z}) \subseteq A(\bar{z})) \land B(X)\big)\Big) = v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big). \tag{13}$$

Proof. Let us prove (12).[6] Let $\mathcal{R}$ be an arbitrary rough relational structure. We have to prove three equivalences $v_{\mathcal{R}}(lhs) = \tau$ iff $v_{\mathcal{R}}(rhs) = \tau$ for $\tau \in \{\mathrm{T}, \mathrm{U}, \mathrm{F}\}$, where *lhs* and *rhs* are respectively the letfthand and the righthand side of Equation (12):

1. ($\rightarrow$) Assume that $v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z})) \land B(X)\big)\Big) = \mathrm{T}$. In this case, there is X such that $v_{\mathcal{R}}\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z}))\big) = \mathrm{T}$ and $v_{\mathcal{R}}\big(B(X)\big) = \mathrm{T}$. Thus, by Definition 9, for every $\bar{z}$, $v_{\mathcal{R}}\big(A(\bar{z})\big) \leq v_{\mathcal{R}}\big(X(\bar{z})\big)$. By down-monotonicity of $B(X)$ in X we conclude that $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big) = \mathrm{T}$.

   ($\leftarrow$) Assume that $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big) = \mathrm{T}$. To show that there is X satisfying $v_{\mathcal{R}}\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z})) \land B(X)\big) = \mathrm{T}$ it suffices to set $\forall\bar{z}\big(X(\bar{z}) \overset{\text{def}}{=} A(\bar{z})\big)$.
2. ($\rightarrow$) Assume that $v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z})) \land B(X)\big)\Big) = \mathrm{U}$. In this case,[7] there is X such that $v_{\mathcal{R}}\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z}))\big) = \mathrm{T}$ and $B(X) = \mathrm{U}$. Thus, by Definition 9, for every $\bar{z}$, $v_{\mathcal{R}}\big(A(\bar{z})\big) \leq v_{\mathcal{R}}\big(X(\bar{z})\big)$. By down-monotonicity of $B(X)$ in X we conclude that $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big) \geq \mathrm{U}$. Suppose that:

$$v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big) = \mathrm{T}. \tag{14}$$

   However, by 1.($\leftarrow$), (14) implies $v_{\mathcal{R}}\Bigg(\exists X\big(\forall\bar{z}(A(\bar{z}) \subseteq X(\bar{z})) \land B(X)\big)\Bigg) = \mathrm{T}$, contradicting the assumption. Therefore, $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A(\bar{z})[\bar{z}]\big)\Big) = \mathrm{U}$.

   ($\leftarrow$) Here, like in the previous point, it suffices to set $\forall\bar{z}\big(X(\bar{z}) \overset{\text{def}}{=} A(\bar{z})\big)$.

[6] The proof of (13) is analogous, so we skip it here.

[7] Note that $\subseteq$ is two-valued, i.e., its truth value can only be T or F.

3. ($\rightarrow$) Assume that $v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(A(\bar{z})\subseteq X(\bar{z}))\land B(X)\big)\Big)=\mathrm{F}$. By points 1.($\leftarrow$) and 2.($\leftarrow$), the value $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big)$ can neither be T nor U (since, as before, this would contradict the assumption). Therefore we can only conclude that $v_{\mathcal{R}}\Big(B\big(X{\leftarrow}A[\bar{z}]\big)\Big)=\mathrm{F}$.

4. ($\leftarrow$) Here, like in the previous points, it suffices to set $\forall\bar{z}\big(X(\bar{z})\stackrel{\mathrm{def}}{=}A(\bar{z})\big)$. ■

The following theorem adapts the fixpoint theorem proved in [20] to rough theories, where $\textsc{Lfp}\,X\big[A(X)\big]$ and $\textsc{Gfp}\,X\big[A(X)\big]$ stand for the least and the greatest fixpoint of $A(X)$ wrt X. Note that we deal with complete lattices and will always make sure that $A(X)$ is up-monotone in X, such fixpoints exist by Knaster and Tarski fixpoint theorem.

Theorem 4. *Let X be an n-argument second-order variable. Let $\bar{z}$ be an n-tuple of variables, $A(X,\bar{z})$ be a rough formula in which variables X and $\bar{z}$ are free. Let $A(X,\bar{z})$ be up-monotone in X and let $B(X)$ be a rough formula with X being a free variable.*

1. If $B(X)$ is down-monotone in X then for every rough relational structure $\mathcal{R}$,

$$v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(A(X,\bar{z})\subseteq X(\bar{z}))\land B(X)\big)\Big)= \\ v_{\mathcal{R}}\Big(B\big(X{\leftarrow}\textsc{Lfp}\,X\big[A(X,\bar{z})[\bar{z}]\big]\big)\Big). \quad (15)$$

2. If $B(X)$ is up-monotone in X then for every rough relational structure $\mathcal{R}$,

$$v_{\mathcal{R}}\Big(\exists X\big(\forall\bar{z}(X(\bar{z})\subseteq A(X,\bar{z}))\land B(X)\big)\Big)= \\ v_{\mathcal{R}}\Big(B\big(X{\leftarrow}\textsc{Gfp}\,X\big[A(X,\bar{z})[\bar{z}]\big]\big)\Big). \quad (16)$$

Proof. (Sketch) The proof is similar to the proof of Theorem 3. In the case of (15) it suffices to notice that the least X satisfying the lefthand side of the equality is defined by the least fixpoint of $A(X)$. In the case of (16) the suitable X is defined by the greatest fixpoint of $A(X)$. ■

Remark 3. Theorems 3 and 4 provide us with definitions of the least and the greatest rough relations interpreting eliminated relation symbols:

– if the lefthand side of (12) is true then the least relation X satisfying the formula $\forall\bar{z}(A(\bar{z})\subseteq X(\bar{z}))\land B(X)$ is defined by $\forall\bar{z}\big(X(\bar{z})\stackrel{\mathrm{def}}{=}A(\bar{z})\big)$;
– if the lefthand side of (13) is true then the greatest relation X satisfying the formula $\forall\bar{z}(X(\bar{z})\subseteq A(\bar{z}))\land B(X)$ is defined by $\forall\bar{z}\big(X(\bar{z})\stackrel{\mathrm{def}}{=}A(\bar{z})\big)$;
– if the lefthand side of (15) is true then the least relation X satisfying the formula $\forall\bar{z}(A(X,\bar{z})\subseteq X(\bar{z}))\land B(X)$ is defined by $\forall\bar{z}\big(X(\bar{z})\stackrel{\mathrm{def}}{=}\textsc{Lfp}\,X\big[A(X,\bar{z})\big]\big)$;

- if the lefthand side of (16) is true then the greatest relation X satisfying the formula $\forall\bar{z}(A(X,\bar{z}) \subseteq X(\bar{z})) \wedge B(X)$ is defined by $\forall\bar{z}\big(X(\bar{z}) \stackrel{\mathrm{def}}{=} \textsc{Gfp}\, X\big[A(X,\bar{z})\big]\big)$.

These definitions can be used for computing lower and upper approximations of the eliminated relations. ■

The following lemma shows monotonicity properties of connectives, useful in second-order quantifier elimination. It directly follows from Definition 9.

Lemma 1.

1. *$\bar{z} \in X$ is up-monotone in X;*
2. *$X \subseteq Y$ is down-monotone in X and up-monotone in Y;*
3. *for $\circ \in \{\vee, \wedge\}$, $X \circ Y$ is up-monotone in X and in Y;*
4. *$\neg X$ is down-monotone in X;*
5. *for $Q \in \{\forall, \exists\}$, $Qx(X(x))$ is up-monotone in X.* ■

5 Applications and Examples

5.1 The Scenario

Below we will use the following notation:

- x, y are variables denoting places and $p_1, \ldots, p_n$ are constants denoting places;
- *ice*(x) stands for "x being covered by ice", *rain*(x) – for "rain in x", *freezing*(x) – for "temperature in x being close to 0^oC", *safe*(x) – for "x being safe" and *base*(x) indicating that "there is a base in place x";
- *connected*(x, y) stands for "places x, y being (directly) connected", *slippery*(x, y) – for "connection from x to y being slippery", and *sconnected*(x, y) – for "x, y being safely connected" (perhaps indirectly, via a chain of connections *connected*()).

Let us consider a scenario formalized by the following theory T:

$$\forall x \forall y\big(\big(\mathit{ice}(x) \vee \mathit{ice}(y)\big) \subseteq \mathit{slippery}(x,y)\big) \wedge \tag{17}$$

$$\forall x \forall y\big((x = y) \vee \mathit{connected}(y,x)) \subseteq \mathit{connected}(x,y)\big) \wedge \tag{18}$$

$$\begin{aligned}&\forall x \forall y\big(\big((\mathit{connected}(x,y) \wedge \neg \mathit{slippery}(x,y)) \vee \\ &\qquad \exists z(\mathit{sconnected}(x,z) \wedge \mathit{sconnected}(z,y))\big) \subseteq \mathit{sconnected}(x,y)\big) \wedge\end{aligned} \tag{19}$$

$$\forall x\big(\mathit{base}(x) \subseteq \mathit{safe}(x)\big) \wedge \tag{20}$$

$$\forall x\big(\mathit{safe}(x) \subseteq \big(\mathit{base}(x) \vee \exists y(\mathit{sconnected}(x,y) \wedge \mathit{base}(y))\big)\big) \wedge \tag{21}$$

$$\forall x\big((\mathit{rain}(x) \wedge \mathit{freezing}(x)) \subseteq \mathit{ice}(x)\big). \tag{22}$$

Note that the relations used in (17)–(22) are rough relations which can be specified as a part of the considered theory. For example, given that there are n places, and:

- $\mathit{connected}()$'s lower approximation is $\{\langle p_1, p_{14}\rangle, \ldots, \langle p_{20}, p_n\rangle\}$ and its upper approximation is the complement of $\{\langle p_7, p_9\rangle, \ldots, \langle p_{20}, p_{30}\rangle, \ldots\}$;
- $\mathit{rain}()$'s lower approximation is $\{p_3, \ldots, p_{48}\}$ and its upper approximation is the complement of $\{p_1, p_2\}$;
- $\mathit{ice}()$'s lower and upper approximation is $\{p_1, \ldots, p_{17}\}$;
- $\mathit{freezing}()$'s lower and upper approximation is $\{p_1, \ldots, p_n\}$,

one can add to the theory the following conjunction of rough facts:

$$\begin{aligned} & \mathit{connected}(p_1, p_{14}) \land \ldots \land \mathit{connected}(p_{20}, p_n) \land \qquad (23)\\ & \qquad \neg \mathit{connected}(p_7, p_9) \land \ldots \land \neg \mathit{connected}(p_{20}, p_{30}) \land \ \ldots \land \\ & \mathit{rain}(p_3) \land \ldots \land \mathit{rain}(p_{48}) \land \neg \mathit{rain}(p_1) \land \neg \mathit{rain}(p_2) \land \qquad (24)\\ & \mathit{ice}(p_1) \land \ldots \land \mathit{ice}(p_{17}) \land \neg \mathit{ice}(p_{18}) \land \ldots \land \neg \mathit{ice}(p_n) \land \qquad (25)\\ & \mathit{freezing}(p_1) \land \ldots \land \mathit{freezing}(p_n). \qquad (26) \end{aligned}$$

Remark 4. It is important to note that the conjunction of rough facts, as specified by (23)–(26), does not affect the applicability of the second-order quantifier elimination techniques provided by Theorems 3 and 4. ■

5.2 Forgetting Rough Concepts

In the first example, let us forget $\mathit{ice}()$ in the scenario theory above. That is, we consider $\mathit{rforget}(T; \mathit{ice}())$ and, according to Theorem 2, we eliminate $\exists X$ from formula:

$$\begin{aligned} \exists X \Big(& \underbrace{\forall x \big((\mathit{rain}(x) \land \mathit{freezing}(x)) \subseteq X(x)\big)}_{\text{corresponding to (22)}} \land \\ & \underbrace{\forall x \forall y \big((X(x) \lor X(y)) \subseteq \mathit{slippery}(x, y)\big)}_{\text{corresponding to (17)}} \land B \Big), \end{aligned} \tag{27}$$

where $B \overset{\mathrm{def}}{=} \big((18) \land (19) \land (20) \land (21)\big)$. According to Lemma 1, the part of (27) corresponding to (17) is down-monotone in X thus, using equality (12) of Theorem 3, we obtain the following formula equivalent to (27):

$$\forall x \forall y \Big(\big((\mathit{rain}(x) \land \mathit{freezing}(x)) \lor (\mathit{rain}(y) \land \mathit{freezing}(y))\big) \subseteq \mathit{slippery}(x, y)\Big) \land B.$$

In the second example, let us forget $\mathit{base}()$ in the scenario theory above. We consider $\mathit{rforget}(T; \mathit{base}())$ and apply Theorem 2 to eliminate $\exists X$ from:

$$\begin{aligned} \exists X \Big(& \underbrace{\forall x \big(X(x) \subseteq \mathit{safe}(x)\big)}_{\text{corresponding to (20)}} \land \\ & \underbrace{\forall x \big(\mathit{safe}(x) \subseteq \big(X(x) \lor \exists y (\mathit{sconnected}(x, y) \land X(y))\big)\big)}_{\text{corresponding to (21)}} \land C \Big), \end{aligned} \tag{28}$$

where $C \overset{\mathrm{def}}{=} \big((17) \wedge (18) \wedge (19) \wedge (22)\big)$. According to Lemma 1, the part of (28) corresponding to (21) is up-monotone in X thus, using equality (13) of Theorem 3, the equivalent of (28) is $\forall x\big(\mathit{safe}(x) \subseteq \big(\mathit{safe}(x) \vee \exists y(\mathit{sconnected}(x,y) \wedge \mathit{safe}(y))\big)\big)$.

Observe that the resulting formula is equivalent to T, so $\mathit{rforget}(T; \mathit{base}())$ is equivalent to C. Indeed, when $\mathit{base}()$ is forgotten, the theory no longer provides useful information about $\mathit{safe}()$, too.

5.3 Forgetting Many-Argument Relations

Forgetting rough relations with more than one argument is very similar to forgetting rough concepts. To illustrate the use of Theorem 4, let us forget $\mathit{connected}()$. That is, consider $\mathit{rforget}(T; \mathit{connected}())$ and, according to Theorem 2, we eliminate $\exists X$ from:

$$\begin{aligned}\exists X \Big(& \underbrace{\forall x \forall y \big((x = y \vee X(y,x)) \subseteq X(x,y)\big)}_{\text{corresponding to (18)}} \wedge \\ & \underbrace{\forall x \forall y \big(\big((X(x,y) \wedge \neg \mathit{slippery}(x,y)) \vee}_{\text{corresponding to (19), line 1}} \\ & \quad \underbrace{\exists z(\mathit{sconnected}(x,z) \wedge \mathit{sconnected}(z,y))\big) \subseteq \mathit{sconnected}(x,y)\big)}_{\text{(19), line 2}} \wedge D \Big),\end{aligned} \tag{29}$$

where $D \overset{\mathrm{def}}{=} \big((17) \wedge (20) \wedge (21) \wedge (22)\big)$.
According to Lemma 1, the part of (29) corresponding to (19) is down-monotone in X thus, using equality (15) of Theorem 4, we obtain the following equivalent of (29):

$$\begin{aligned}\forall x \forall y \big(\big((& \mathrm{LFP}\, X(x,y) \big[x = y \vee X(y,x)\big](x,y) \wedge \neg \mathit{slippery}(x,y)) \vee \\ & \exists z(\mathit{sconnected}(x,z) \wedge \mathit{sconnected}(z,y))\big) \subseteq \mathit{sconnected}(x,y)\big) \wedge D.\end{aligned} \tag{30}$$

Note that $\mathrm{LFP}\,\big[\ldots\big]$ in (30) is equivalent to $x = y$, so (30) can further be simplified to:

$$\begin{aligned}\forall x \forall y \big(\big((& x = y \wedge \neg \mathit{slippery}(x,y)) \vee \\ & \exists z(\mathit{sconnected}(x,z) \wedge \mathit{sconnected}(z,y))\big) \subseteq \mathit{sconnected}(x,y)\big) \wedge D.\end{aligned}$$

6 Conclusions

In this paper, we provided basic foundations for the specification and application of a forgetting operator for rough theories. To do this, we defined a logical language for rough theories consisting of rough formulas and a semantics for such formulas containing rough relations, in terms of rough Kleene algebras. Using intuitions from work with forgetting operators in classical logic, we then specified a rough forgetting operator in the context of rough relational theories. We then showed how the constructive generation of the result of applying a forgetting operator to a rough theory could be achieved by using second-order quantifier elimination techniques. These foundations open up opportunities for the use of these rough logics for KRR applications and the study of additional types of forgetting operators in this context, in particular of forgetting in rule languages that use a Kleene logic-based semantics. Also, algorithmic techniques based on insights using second-order quantifier elimination techniques, are worth investigating as a basis for forgetting operators used with rough relational theories.

References

1. Ackermann, W.: Untersuchungen über das eliminationsproblem der mathematischen logik. Mathematische Annalen **110**, 390–413 (1935)
2. Boole, G.: An Investigation of The Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities. Macmillan, London, UK (1854)
3. Ciucci, D.: Orthopairs: a simple and widely used way to model uncertainty. Fundam. Inform. **108**(3–4), 287–304 (2011)
4. Ciucci, D., Dubois, D.: Three-valued logics, uncertainty management and rough sets. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XVII. LNCS, vol. 8375, pp. 1–32. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54756-0_1
5. D'eer, L., Cornelis, C.: A comprehensive study of fuzzy covering-based rough set models: definitions, properties and interrelationships. Fuzzy Sets Syst. **336**, 1–26 (2018)
6. Del-Pinto, W., Schmidt, R.: ABox abduction via forgetting in ALC. In: The 33rd AAAI Conference on Artificial Intelligence, pp. 2768–2775. AAAI Press (2019)
7. Demri, S., Orłowska, E.: Incomplete Information: Structure, Inference, Complexity. EATCS Monographs. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04997-6
8. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques. A Rough Set Approach. Studies in Fuzziness and Soft Computing, vol. 202. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-33519-6
9. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited. J. Autom. Reason. **18**(3), 297–336 (1997)
10. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: 17th IJCAI, pp. 145–151 (2001)
11. Doherty, P., Szałas, A.: On the correspondence between approximations and similarity. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 143–152. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25929-9_16
12. Eiter, T., Kern-Isberner, G.: A brief survey on forgetting from a knowledge representation and reasoning perspective. KI **33**(1), 9–33 (2019)
13. Gabbay, D., Schmidt, R., Szałas, A.: Second-Order Quantifier Elimination. Foundations, Computational Aspects and Applications, Studies in Logic, vol. 12. College Pub. (2008)
14. Hariri, R., Fredericks, E., Bowers, K.: Uncertainty in big data analytics: survey, opportunities, and challenges. J. Big Data **6**, 44 (2019)
15. Konikowska, B., Avron, A.: Reasoning about covering-based rough sets using three truth values. J. Appl. Log. IfCoLoG J. Log. Appl. **6**(2), 361–382 (2019)

16. Kumar, A., Banerjee, M.: Kleene algebras and logic: Boolean and rough set representations, 3-valued, rough set and Perp semantics. Studia Logica **105**(3), 439–469 (2017)
17. Lin, F.: On strongest necessary and weakest sufficient conditions. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proceedings of the 7th International Conference, KR2000, pp. 167–175. Morgan Kaufmann (2000)
18. Lin, F., Reiter, R.: Forget it! In: Proceedings of the AAAI Fall Symposium on Relevance, pp. 154–159 (1994)
19. Nguyen, L.A., Szałas, A.: Logic-based roughification. In: Skowron, A., Suraj, Z. (eds.) Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam. Intelligent Systems Reference Library, vol. 42, pp. 517–543. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-30344-9_19
20. Nonnengart, A., Szałas, A.: A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In: Orłowska, E. (ed.) Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa. Studies in Fuzziness & Soft Computing, vol. 24, pp. 307–328. Springer, Heidelberg (1998)
21. Pawlak, Z.: Information systems - theoretical foundations. Inf. Syst. **6**, 205–218 (1981)
22. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
23. Polkowski, L.: Rough Sets. Mathematical Foundations, Advances in Intelligent and SoftRough Sets. Mathematical Foundations, Advances in Intelligent and Soft, vol. 15. Physica-Verlag, Heidelberg (2002). https://doi.org/10.1007/978-3-7908-1776-8
24. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae **27**, 245–253 (1996)
25. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. Knowl. Data Eng. **12**(2), 331–336 (2000)
26. Wang, Y., Wang, K., Zhang, M.: Forgetting for answer set programs revisited. In: Rossi, F. (ed.) Proceedings of the IJCAI 2013, pp. 1162–1168. IJCAI/AAAI (2013)
27. Wang, Z., Wang, K., Zhang, X.: Forgetting and unfolding for existential rules. In: McIlraith, S., Weinberger, K. (eds.) Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 2013–2020. AAAI Press (2018)
28. Wernhard, C.: Literal projection for first-order logic. In: Hölldobler, S., Lutz, C., Wansing, H. (eds.) JELIA 2008. LNCS (LNAI), vol. 5293, pp. 389–402. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87803-2_32
29. Yao, Y.Y., Wong, S.K.M., Lin, T.Y.: A review of rough set models. In: Lin, T.Y., Cercone, N. (eds.) Rough Sets and Data Mining, pp. 47–75. Springer, Boston (1997). https://doi.org/10.1007/978-1-4613-1461-5_3
30. Zhang, Q., Xie, Q., Wang, G.: A survey on rough set theory and its applications. CAAI Trans. Intell. Technol. **1**(4), 323–333 (2016)
31. Zhao, Y., Schmidt, R.A.: FAME(Q): an automated tool for forgetting in description logics with qualified number restrictions. In: Fontaine, P. (ed.) CADE 2019. LNCS (LNAI), vol. 11716, pp. 568–579. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_34