Chapter 19
Logic-Based Roughification

Linh Anh Nguyen and Andrzej Szałas

Abstract. The current chapter is devoted to *roughification*. In the most general setting, we intend the term *roughification* to refer to methods/techniques of constructing equivalence/similarity relations adequate for Pawlak-like approximations. Such techniques are fundamental in rough set theory. We propose and investigate novel roughification techniques. We show that using the proposed techniques one can often discern objects indiscernible by original similarity relations, what results in improving approximations. We also discuss applications of the proposed techniques in granulating relational databases and concept learning. The last application is particularly interesting, as it shows an approach to concept learning which is more general than approaches based solely on information and decision systems.

19.1 Introduction

Rough sets are typically used to model vague concepts and relationships. They are defined in terms of lower and upper approximations of crisp sets/relations, where approximations are in place when objects may be indiscernible due to incomplete, imprecise, and approximate data or knowledge. Indiscernibility of objects is modeled by similarity relations, originally assumed to be equivalence relations [19]. In general, the lower approximation of a set consists of objects whose similarity neighborhoods are contained in the set, while the upper approximation consists of objects whose similarity neighborhoods intersect the set. Similarity neighborhoods, often

Linh Anh Nguyen
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
e-mail: nguyen@mimuw.edu.pl

Andrzej Szałas
Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland and Dept. of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
e-mail: andrzej.szalas@mimuw.edu.pl

A. Skowron and Z. Suraj (Eds.): Rough Sets and Intelligent Systems, ISRL 42, pp. 517–543.
springerlink.com

being equivalence classes, are then substantial ingredients of approximate modeling and reasoning.[1]

The current chapter is devoted to *roughification*. In the most general setting, we intend that the term *roughification* refers to methods/techniques of constructing equivalence/similarity relations adequate for approximations. For example,

- in [19] as well as many later works (see, *e.g.*, [6, 20, 21] and references there), equivalence relations are constructed from information tables and decision tables, *for example*, by reducing the number of attributes
- in [27, 28],[2] equivalence relations are constructed by rough discretization and applied in clustering and classification
- in the light of [7], approximations can be constructed on the basis of logical theories, by projecting them into weaker languages.

We propose and investigate novel roughification techniques allowing one to construct suitable equivalence relations on the basis of background knowledge. We assume that knowledge is expressed by means of logical theories in the form of relational databases (*relational roughification*) and description logic theories (*terminological roughification*). The main idea depends on placing objects in the same equivalence class when they are indiscernible by a given logical theory. We show that using the proposed techniques one can often discern objects indiscernible by original similarity relations, so improve approximations. We also discuss applications of the proposed techniques in granulating relational databases and concept learning. The last application is particularly interesting, as it shows an approach to concept learning which is more general than approaches based solely on information and decision systems.

The first technique we propose is relational roughification, allowing one to obtain congruences on the basis of knowledge contained in relational databases. This technique, in fact, allows us to granulate arbitrary relational structures. We start with a simplified case, when such knowledge consists solely of similarity relations on objects, and show a natural technique (*similarity-based roughification*) allowing one to construct equivalence relations. This technique leads to better approximations than those offered by original similarities. As a general methodological outcome, we show that indiscernibility can actually be modeled by equivalence relations even if one initially uses weaker similarity relations, perhaps more intuitive in a given application domain. This places those other approaches back in the rough set context originally proposed and developed by Pawlak.

A more advanced version of roughification introduced in this chapter is based on bisimulations in the context of description logics. Namely, indiscernibility related to a given concept can be approximated using the largest auto-bisimulation with respect to the sublanguage consisting of concept names, role names and constructors the concept depends on. Such bisimulations are equivalence relations. We give

[1] For works, where similarity relations are not assumed to be equivalence classes, see [6, 8, 9, 12, 24, 29, 30] and references there.

[2] Where the term *roughification* has most probably been introduced in the context of discretization.

a logical account of this approach, investigate its theoretical properties and use it to study the problems of concept learning and concept approximation in information systems based on description logics.

Let us emphasize that all solutions we propose are tractable in the sense that data complexity of computing constructed new equivalence relations is in PTIME in the size of the underlying domain assuming that the underlying knowledge is given by means of relational/deductive databases or knowledge databases expressed by means of description logics.

The chapter is structured as follows. We start with basic definitions and preliminaries (Section 19.2). Then, in Section 19.3 we continue with similarity-based roughification and, in Section 19.4, with relational roughification. Section 19.5 is devoted to terminological roughification. Concluding remarks are contained in Section 19.6.

19.2 Preliminaries

Let Δ be a finite set, further called a *domain*. Elements of Δ are called *objects*. By a *relational structure* we understand a tuple $\langle \Delta, \{r_i\}_{i \in I} \rangle$, where for each $i \in I$, r_i is a relation on Δ.

For the sake of simplicity, in the chapter we consider one-sorted domains only. That is, we assume that objects are of the same type. The results we provide can be generalized in a straightforward manner to many-sorted structures. This, however, is not necessary for techniques we present.

A *signature* for relational structures consists of a finite set of *individual names* (i.e. *object names*), a finite set of *predicates* (i.e. relation names), and an arity mapping that associates each of the predicates with a natural number called the arity of the predicate.[3]

A relational structure over a signature Σ can be redefined to be a pair $I = \langle \Delta^I, \cdot^I \rangle$ consisting of a non-empty set Δ^I, called the *domain* of I, and a function $\cdot^I$, called the *interpretation function* of I, which maps each individual name a of Σ to an element a^I of Δ^I and maps each n-argument predicate p of Σ to an n-argument relation p^I on Δ^I.

By a *congruence* on $\langle \Delta, \{R_i\}_{i \in I} \rangle$ we understand any equivalence relation $\approx$ on Δ which *preserves* all relations $\{R_i\}_{i \in I}$, that is, such that for any $i \in I$, if R_i is an n-argument relation and $x_1 \approx x'_1, \ldots, x_n \approx x'_n$, then $R_i(x_1, \ldots, x_n) \equiv R_i(x'_1, \ldots, x'_n)$.

Let further $\sigma \subseteq \Delta \times \Delta$ be a binary relation on Δ, representing similarity on elements of Δ. It models indiscernibility on Δ in the sense that objects $x, x' \in \Delta$ are *indiscernible* whenever $\sigma(x, x')$ holds. The pair $\langle \Delta, \sigma \rangle$ is called a *similarity space*.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ and $A \subseteq \Delta$, Pawlak-like *approximations* of A w.r.t. $\mathcal{S}$ are defined as follows:

[3] For first-order logic, one would add to a signature also a finite set of function names and information about their arities but we concentrate on relations only.

- the *lower approximation* of A w.r.t. $\mathcal{S}$, denoted by $A_{\mathcal{S}^+}$ is defined by

$$A_{\mathcal{S}^+} \stackrel{\text{def}}{=} \{x \mid \forall y[\sigma(x,y) \to A(y)]\} \tag{19.1}$$

- the *upper approximation* of A w.r.t. $\mathcal{S}$, denoted by $\mathcal{A}_{\mathcal{S}^\oplus}$ is defined by

$$A_{\mathcal{S}^\oplus} \stackrel{\text{def}}{=} \{x \mid \exists y[\sigma(x,y) \land A(y)]\}. \tag{19.2}$$

When $\mathcal{S}$ is known from context, we sometimes write $\underline{A}$ (respectively, $\overline{A}$) to denote the lower (respectively the upper) approximation of A w.r.t. $\mathcal{S}$, that is, $\underline{A} \stackrel{\text{def}}{=} A_{\mathcal{S}^+}$ and $\overline{A} \stackrel{\text{def}}{=} A_{\mathcal{S}^\oplus}$.

An information system in the rough set theory [19, 21, 20], called an *RS information system*, is usually defined to be a pair $\langle \Delta, Attrs \rangle$ of non-empty finite sets Δ and $Attrs$, where Δ is the *universe* of *objects*, and $Attrs$ is a set of *attributes*, that is, functions $A : \Delta \to V_A$, where V_A is the set of values of attribute A, called the *domain* of A.

19.3 Similarity-Based Roughification

Similarity-based roughification can be viewed as relational roughification introduced in Section 19.4. Namely, a relational structure can contain solely a similarity relation. However, similarities play a special role in defining relational roughifications. Also, intended applications make the technique interesting on its own. Therefore we discuss it separately.

19.3.1 Definitions

Observe that even if two objects x, x' are indiscernible w.r.t. a given similarity relation σ, that is, $\sigma(x,x')$ holds, it still might be the case that they can be discerned if there is an object x'' such that $\sigma(x,x'')$ and $\neg\sigma(x',x'')$. The same holds when there is an object x'' such that $\sigma(x'',x)$ and $\neg\sigma(x'',x')$. The first types of roughification reflect this phenomenon.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, by a *forward similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\triangleright} = \langle \Delta, \rho_{\mathcal{S}}^{\triangleright} \rangle$, where:

$$\rho_{\mathcal{S}}^{\triangleright}(x,x') \stackrel{\text{def}}{\equiv} \forall x'' \left[\sigma(x,x'') \equiv \sigma(x',x'')\right]. \tag{19.3}$$

By a *backward similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\triangleleft} = \langle \Delta, \rho_{\mathcal{S}}^{\triangleleft} \rangle$, where:

$$\rho_{\mathcal{S}}^{\lhd}(x,x') \stackrel{\text{def}}{\equiv} \forall x''\big[\sigma(x'',x) \equiv \sigma(x'',x')\big]. \tag{19.4}$$

By a *similarity-based roughification* induced by $\mathcal{S}$ we understand relational structure $\mathcal{R}_{\mathcal{S}}^{\bowtie} = \langle \Delta, \rho_{\mathcal{S}}^{\bowtie} \rangle$, where

$$\rho_{\mathcal{S}}^{\bowtie} \stackrel{\text{def}}{=} \rho_{\mathcal{S}}^{\rhd} \cap \rho_{\mathcal{S}}^{\lhd}. \tag{19.5}$$

19.3.2 Properties

We have the following proposition.

Proposition 19.1. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space. Then:*

- $\rho_{\mathcal{S}}^{\rhd}$ *and* $\rho_{\mathcal{S}}^{\lhd}$ *are equivalence relations on* Δ
- $\rho_{\mathcal{S}}^{\bowtie}$ *is a congruence on* $\mathcal{S}$.

Proof. The first claim is obvious by definitions (19.3) and (19.4).

To prove the second claim, note that $\rho_{\mathcal{S}}^{\bowtie}$ is the intersection of two equivalence relations, so it is an equivalence relation, too. To prove that it preserves σ, assume:

$$\rho_{\mathcal{S}}^{\bowtie}(x_1,x_1') \text{ and } \rho_{\mathcal{S}}^{\bowtie}(x_2,x_2'). \tag{19.6}$$

We have to show that $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2')$. By (19.3)–(19.5) and (19.6), in particular we have:

$$\forall x''[\sigma(x_1,x'') \equiv \sigma(x_1',x'')] \text{ and } \forall y''[\sigma(y'',x_2) \equiv \sigma(y'',x_2')]. \tag{19.7}$$

Taking $x'' = x_2$ and $y'' = x_1'$ we have $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2)$ and $\sigma(x_1',x_2) \equiv \sigma(x_1',x_2')$, so also $\sigma(x_1,x_2) \equiv \sigma(x_1',x_2')$. □

We also have the following proposition.

Proposition 19.2. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ, we have that $\rho_{\mathcal{S}}^{\rhd} \subseteq \sigma$, $\rho_{\mathcal{S}}^{\lhd} \subseteq \sigma$ and $\rho_{\mathcal{S}}^{\bowtie} \subseteq \sigma$.*

Proof. Assume that $\rho_{\mathcal{S}}^{\rhd}(x,x')$. By (19.3), for all x'', $\sigma(x,x'') \equiv \sigma(x',x'')$. In particular, for $x'' = x'$ we have $\sigma(x,x') \equiv \sigma(x',x')$. By reflexivity of σ, we have that $\sigma(x',x')$ holds, so we also have that $\sigma(x,x')$ holds.

Analogously, using (19.4) we prove $\rho_{\mathcal{S}}^{\lhd} \subseteq \sigma$. Of course, $\rho_{\mathcal{S}}^{\bowtie} \subseteq \rho_{\mathcal{S}}^{\rhd}$, which proves the last inclusion. □

Observe that reflexivity of σ corresponds to the property that for any set A, $A_{\mathcal{S}^+} \subseteq A$ (see, *e.g.*, [10]). On the other hand, the weakest requirement placed on approximations, $A_{\mathcal{S}^+} \subseteq A_{\mathcal{S}^\oplus}$, is equivalent to the seriality of σ, that is, the property $\forall x \exists y[\sigma(x,y)]$. The following example shows that seriality is not sufficient to prove Proposition 19.2.

Example 19.1. Let $\mathcal{S} = \langle \{a,b,c\}, \sigma \rangle$, where $\sigma = \{(a,c),(b,c),(c,c)\}$. Obviously, σ is serial. On the other hand, $\rho_{\mathcal{S}}^{\rhd}(a,b)$ holds, while $\sigma(a,b)$ does not. □

Using Proposition 19.2 one can show that each $\mathcal{R}_\mathcal{S}^\triangleright$, $\mathcal{R}_\mathcal{S}^\triangleleft$ and $\mathcal{R}_\mathcal{S}^\bowtie$ approximates sets better than $\mathcal{S}$, as formulated in the following proposition.

Proposition 19.3. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ and any $A \subseteq \Delta$,*

$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}_\mathcal{S}^\triangleright)^+} \subseteq A_{(\mathcal{R}_\mathcal{S}^\bowtie)^+} \subseteq A \subseteq A_{(\mathcal{R}_\mathcal{S}^\bowtie)^\oplus} \subseteq A_{(\mathcal{R}_\mathcal{S}^\triangleright)^\oplus} \subseteq A_{\mathcal{S}^\oplus}$$
$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}_\mathcal{S}^\triangleleft)^+} \subseteq A_{(\mathcal{R}_\mathcal{S}^\bowtie)^+} \subseteq A \subseteq A_{(\mathcal{R}_\mathcal{S}^\bowtie)^\oplus} \subseteq A_{(\mathcal{R}_\mathcal{S}^\triangleleft)^\oplus} \subseteq A_{\mathcal{S}^\oplus}. \qquad \square$$

19.3.3 Selected Applications

Proposition 19.3 shows that the similarity-based roughification may discern objects better than the original similarity relation. This allows us to sharpen perceptual capabilities, improving its accuracy. The following example illustrates this idea.

Example 19.2. Let a set of objects, say $\Delta = \{o_1, o_2, o_3\}$, be given. Assume that the accuracy of a sensor platform does not allow us to discern certain objects on the basis of their features. Such a situation is typically modeled by a similarity space $\langle \Delta, \sigma \rangle$ where, for example,

$$\sigma(o_1,o_1), \sigma(o_2,o_2), \sigma(o_3,o_3),$$
$$\sigma(o_1,o_2), \sigma(o_2,o_1), \sigma(o_2,o_3), \sigma(o_3,o_2),$$

that is, o_1 is indiscernible from itself and o_2, etc. On the other hand, one can discern o_1 and o_2 by comparing them with o_3. Such a comparison provides different results, allowing one to detect what object is being perceived. □

Similarity-based roughification can also be useful in decision rules mining. The obtained rules can be judged, among others, w.r.t. their classification accuracy. One faces here the overfitting/underfitting problem. Overfitting results in too many specialized rules, while underfitting causes poor classification results. The following example illustrates how can one tune decision rules using similarities resulting in better or worse approximations (by using Proposition 19.3).

Example 19.3. In the machine learning process one often obtains rules like:

$$\text{IF } \mathit{bad\_condition}(x) \text{ THEN } \mathit{maintenance}(x), \tag{19.8}$$

where objects are classified to be in "bad condition" on the basis of chosen attributes, say rust and moisture level. Particular examples in the training sets may be very specific. For example, an object o with rust level 0.743 and moisture level 0.92 may be marked as being in bad condition. One could then derive the following rule:

$$\text{IF } \mathit{rust}(x, 0.743) \text{ AND } \mathit{moisture}(x, 0.92) \text{ THEN } \mathit{maintenance}(x),$$

which definitely is too specific. One would also like to deduce that all *similar* (w.r.t. rust and moisture level) objects require maintenance, too:

$$\text{IF } \sigma(x,o) \text{ THEN } maintenance(x),$$

where σ is a similarity relation of a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with Δ consisting of value pairs $\langle rust, moisture \rangle$.

Now, if a given σ results in underfitting, one can use its roughification instead (or any suitable relation σ' such that $\rho_{\mathcal{S}}^{\bowtie} \subseteq \sigma' \subseteq \sigma$). Then, by moving σ' between the boundaries $\rho_{\mathcal{S}}^{\bowtie}$ and σ one can tune rules when new objects appear and are being classified. □

19.4 Relational Roughification

Relational roughification extends similarity-based roughification. Given a relational database, one can observe that object can be additionally discern by relations included in the database.

19.4.1 Definitions

Assume that a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ is given and $\mathcal{R}_{\mathcal{S}}^{\bowtie} = \langle \Delta, \rho_{\mathcal{S}}^{\bowtie} \rangle$ is the similarity-based roughification induced by $\mathcal{S}$.

Assume now that additional knowledge is provided by a relational or a deductive database. Even if two objects are indiscernible by $\rho_{\mathcal{S}}^{\bowtie}$, they may still be discernible by relations included in the database. For example, it might be the case that $\rho_{\mathcal{S}}^{\bowtie}(o, o')$ holds, while for a relation R in the database, it could be $R(\bar{a}, o, \bar{b})$ and $\neg R(\bar{a}, o', \bar{b})$. In such a case we can discern o and o' using R. We then have the following definition.

Given a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, by a *relational roughification* induced by $\mathcal{S}$ and an m-argument relation R we understand relational structure $\mathcal{R}_{\mathcal{S}}^{R} = \langle \Delta, \rho_{\mathcal{S}}^{R} \rangle$, where:[4]

$$\rho_{\mathcal{S}}^{R} \stackrel{\text{def}}{\equiv} \rho_{\mathcal{S}}^{\bowtie} - \{(x,x'),(x',x) \mid \exists x_1 \ldots \exists x_{m-1} [R(x_1,\ldots,x,\ldots,x_{m-1}) \wedge \neg R(x_1,\ldots,x',\ldots,x_{m-1})]\}. \tag{19.9}$$

Let us emphasize that in (19.9) we do not fix the position of x. For example, if R is a two-argument relation then (19.9) is to be understood as:

$$\rho_{\mathcal{S}}^{R} \stackrel{\text{def}}{\equiv} \rho_{\mathcal{S}}^{\bowtie} - \big(\{(x,x'),(x',x) \mid \exists x_1 [R(x_1,x) \wedge \neg R(x_1,x')]\} \cup \{(x,x'),(x',x) \mid \exists x_1 [R(x,x_1) \wedge \neg R(x',x_1)]\}\big). \tag{19.10}$$

[4] Recall that $\rho_{\mathcal{S}}^{\bowtie}$ is the similarity-based roughification induced by $\mathcal{S}$ and defined by (19.5).

Observe that one can also consider tuples of relations rather than single relations. Namely, let $\{R_i\}_{i\in I}$ be a (finite) tuple of relations. Then:

$$\rho_{\mathcal{S}}^{\{R_i\}_{i\in I}} \stackrel{\text{def}}{\equiv} \bigcap_{i\in I} \rho_{\mathcal{S}}^{R_i}. \tag{19.11}$$

19.4.2 Properties

Let us first prove that the construction provided in the previous section indeed results in an equivalence relation.

Proposition 19.4. *Let $\mathcal{S} = \langle\Delta,\sigma\rangle$ be a similarity space and R be a relation, $R \subseteq \Delta\times\ldots\times\Delta$. Then $\rho_{\mathcal{S}}^R$ is an equivalence relation on Δ.*

Proof. By Proposition 19.1, $\rho_{\mathcal{S}}^{\bowtie}$ is an equivalence relation.

Suppose $\rho_{\mathcal{S}}^R$ is not an equivalence relation. This could be caused by removing in (19.9) a pair (x,x') from $\rho_{\mathcal{S}}^{\bowtie}$. Let us now show that this cannot violate reflexivity, symmetry nor transitivity.

First note that reflexivity is preserved since there cannot exist $x_1,\ldots,x_{m-1}$ such that $R(x_1,\ldots,x,\ldots,x_{m-1})$ and, at the same time, $\neg R(x_1,\ldots,x,\ldots,x_{m-1})$.

Suppose now that $(x,x') \in \rho_{\mathcal{S}}^R$ and $(x',x) \notin \rho_{\mathcal{S}}^R$. This cannot happen since pairs (x,x') and $(x'x)$ are either not removed from $\rho_{\mathcal{S}}^{\bowtie}$ or are removed both. Therefore, symmetry is preserved.

Suppose that $(x,x'),(x',x'') \in \rho_{\mathcal{S}}^R$ and $(x,x'') \notin \rho_{\mathcal{S}}^R$. Since $\rho_{\mathcal{S}}^R \subseteq \rho_{\mathcal{S}}^{\bowtie}$, we have that $(x,x'),(x',x'') \in \rho_{\mathcal{S}}^{\bowtie}$, so also $(x,x'') \in \rho_{\mathcal{S}}^{\bowtie}$. Thus the assumption that $(x,x'') \notin \rho_{\mathcal{S}}^R$ implies that (x,x'') has been removed in (19.9), meaning that there are $x_1,\ldots,x_{m-1}$ such that

$$\begin{array}{ll} \text{either} & R(x_1,\ldots,x,\ldots,x_{m-1}) \wedge \neg R(x_1,\ldots,x'',\ldots,x_{m-1}) \\ \text{or} & \neg R(x_1,\ldots,x'',\ldots,x_{m-1}) \wedge R(x_1,\ldots,x'',\ldots,x_{m-1}). \end{array}$$

Consider the first case.[5] Since $R(x_1,\ldots,x,\ldots,x_{m-1})$ holds and $(x,x') \in \rho_{\mathcal{S}}^R$, we conclude that $R(x_1,\ldots,x',\ldots,x_{m-1})$ holds (otherwise the pair (x,x') was removed in (19.9)). Now from the fact that $R(x_1,\ldots,x',\ldots,x_{m-1})$ holds and assumption that $(x',x'') \in \rho_{\mathcal{S}}^R$, we also have that $R(x_1,\ldots,x'',\ldots,x_{m-1})$ and a contradiction is reached. □

The intersection of any collection of equivalence relations is also an equivalence relation. We then have the following corollary.

Corollary 19.1. *Let $\mathcal{S} = \langle\Delta,\sigma\rangle$ be a similarity space and $\{R_i\}_{i\in I}$ be relations such that for all $i \in I$, $R_i \subseteq \Delta\times\ldots\times\Delta$. Then $\rho_{\mathcal{S}}^{\{R_i\}_{i\in I}}$ is an equivalence relation on Δ.* □

[5] The second case can be proved analogously.

Granular computing has been considered an important issue in rough set theory and applications [15, 23, 25, 26, 17, 22]. The following proposition shows that the constructed equivalence relation can serve as a basis for granulating relations.

Proposition 19.5. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and R be a relation, $R \subseteq \Delta \times \ldots \times \Delta$. Then $\rho^R_{\mathcal{S}}$ is a congruence on $\langle \Delta, R \rangle$.*

Proof. By Proposition 19.4, $\rho^R_{\mathcal{S}}$ is an equivalence relation. To show that it preserves R, assume that:

$$\rho^R_{\mathcal{S}}(x_1, x'_1), \ldots, \rho^R_{\mathcal{S}}(x_m, x'_m). \tag{19.12}$$

We have to show that

$$R(x_1, \ldots, x_m) \equiv R(x'_1, \ldots, x'_m). \tag{19.13}$$

To prove (19.13), we proceed by induction on $0 \le k \le m$:

$$R(x_1, \ldots, x_k, x_{k+1} \ldots, x_m) \equiv R(x'_1, \ldots, x'_k, x_{k+1} \ldots, x_m). \tag{19.14}$$

1. If $k = 0$ then (19.14) is obvious.
2. Assume that the theorem holds for $0 \le k < m$. We shall show that it also holds for $(k+1)$:

$$\begin{array}{ll} R(x_1, \ldots, x_k, x_{k+1}, x_{k+2}, \ldots, x_m) \equiv & \text{(by inductive assumption (19.14))} \\ R(x'_1, \ldots, x'_k, x_{k+1}, x_{k+2}, \ldots, x_m) \equiv & \text{(by definition (19.9), assumption (19.12))} \\ R(x'_1, \ldots, x'_k, x'_{k+1}, x_{k+2}, \ldots, x_m). & \end{array}$$

By analogy to Proposition 19.5 one can prove the following proposition providing a technique for granulating relational databases (see also Section 19.4.3).

Proposition 19.6. *Let $\mathcal{S} = \langle \Delta, \sigma \rangle$ be a similarity space and $\langle \Delta, \{R_i\}_{i \in I} \rangle$ be a relational structure. Then $\rho^{\{R_i\}_{i \in I}}_{\mathcal{S}}$ is a congruence on $\langle \Delta, R \rangle$.* □

By (19.9), we have that $\rho^R_{\mathcal{S}} \subseteq \rho^{\bowtie}_{\mathcal{S}}$. By Proposition 19.2 we then have the following proposition.

Proposition 19.7. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ and relation R on $\Delta \times \ldots \times \Delta$, we have that $\rho^R_{\mathcal{S}} \subseteq \sigma$.* □

As a consequence we have the following proposition.

Proposition 19.8. *For any similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$ with reflexive σ, any relation R on $\Delta \times \ldots \times \Delta$ and any $A \subseteq \Delta$,*

$$A_{\mathcal{S}^+} \subseteq A_{(\mathcal{R}^R_{\mathcal{S}})^+} \subseteq A \subseteq A_{(\mathcal{R}^R_{\mathcal{S}})^\oplus} \subseteq A_{\mathcal{S}^\oplus}.$$ □

Remark 19.1. Note that relational roughification starts with some initial similarity relation and then improves its accuracy. If such a relation is not given, one can start with the total similarity relation $\sigma \stackrel{\text{def}}{=} \Delta \times \Delta$. If $\langle \Delta, \{R_i\}_{i \in I} \rangle$ is a relational structure then the resulting equivalence classes of $\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}}$ consist of objects indiscernible by relations $\{R_i\}_{i \in I}$. However, when $\sigma = \Delta \times \Delta$, similarity-based roughification provides no improvement, as in this case we have $\rho^{\bowtie} = \sigma$. □

19.4.3 Granulating Relational Databases

A relational database is a relational structure of the form $\langle \Delta, \{R_i\}_{i \in I} \rangle$ with finite Δ and I. Relational roughification allows us to granulate such databases in the sense that rather than using objects, we can use equivalence classes. Since an equivalence class may be represented by an arbitrary object it contains, such a granulation allows us to reduce the size of the database as well as consider classes of similar objects rather than singletons.

More precisely, given a relational database $DB = \langle \Delta, \{R_i\}_{i \in I} \rangle$ and a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, by a *granulation of DB w.r.t.* $\mathcal{S}$ we understand

$$DB/\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}} \stackrel{\text{def}}{=} \left\langle \Delta/\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}}, \{\mathbb{R}_i\}_{i \in I} \right\rangle, \tag{19.15}$$

where:

- $\Delta/\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}} \stackrel{\text{def}}{=} \{\|x\| \mid x \in \Delta\}$ is the set of equivalence classes of $\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}}$
- for $i \in I$, $\mathbb{R}_i(\|x_1\|, \ldots, \|x_m\|) \stackrel{\text{def}}{\equiv} R_i(x_1, \ldots, x_m)$.

By Proposition 19.6, $\mathbb{R}_i$ ($i \in I$) are well-defined.

Given a relational database $DB = \langle \Delta, \{R_i\}_{i \in I} \rangle$ and a similarity space $\mathcal{S} = \langle \Delta, \sigma \rangle$, rather than storing all tuples of relations in DB, it suffices to store tuples with representants of equivalence classes only. In addition, one needs to store $\rho_{\mathcal{S}}^{\{R_i\}_{i \in I}}$ in the database, but the reduction od database size can be considerable.

19.5 Terminological Roughification

In this section we study roughification for information systems specified using the formalism of description logics (DLs). Such logics describe the domain of interest by means of individuals, concepts and roles [3, 4, 16]. A concept stands for a set of individuals, while a role stands for a binary relation between individuals. DLs are fragments of classical first-order logic and variants of modal logics. Indiscernibility in DLs is related to bisimulation.

In Sections 19.3 and 19.4 we had a particular similarity relation as a starting point for the construction of the final equivalence relation (but see Remark 19.1). Here we do not need such a starting relation. But whenever it is given, we can place it among roles.

19.5.1 Description Logics and Information Systems

A *DL-signature* is a set $\Sigma = \Sigma_I \cup \Sigma_C \cup \Sigma_R$, where Σ_I is a finite set of individual names, Σ_C is a finite set of *concept names*, and Σ_R is a finite set of *role names*. Concept names are unary predicates, while role names are binary predicates. We denote concept names by letters like A and B, role names by letters like r and s, and individual names by letters like a and b.

We will consider some (additional) *DL-features* denoted by I (*inverse*), O (*nominal*), Q (*quantified number restriction*), U (*universal role*), Self (*local reflexivity of a role*). A *set of DL-features* is a set consisting of some of these names.

Let Σ be a DL-signature and Φ be a set of DL-features. Let $\mathcal{L}$ stand for $\mathcal{ALC}_{reg}$, which is the name of a description logic corresponding to propositional dynamic logic (PDL). The DL language $\mathcal{L}_{\Sigma,\Phi}$ allows *roles* and *concepts* defined recursively as follows:

- if $r \in \Sigma_R$ then r is role of $\mathcal{L}_{\Sigma,\Phi}$
- if $A \in \Sigma_C$ then A is concept of $\mathcal{L}_{\Sigma,\Phi}$
- if R and S are roles of $\mathcal{L}_{\Sigma,\Phi}$ and C is a concept of $\mathcal{L}_{\Sigma,\Phi}$ then
  - ε, $R \circ S$, $R \sqcup S$, R^* and $C?$ are roles of $\mathcal{L}_{\Sigma,\Phi}$
  - $\top$, $\bot$, $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$ and $\exists R.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $I \in \Phi$ then R^- is a role of $\mathcal{L}_{\Sigma,\Phi}$
  - if $O \in \Phi$ and $a \in \Sigma_I$ then $\{a\}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$
  - if $Q \in \Phi$, $r \in \Sigma_R$ and n is a natural number then $\geq n\, r.C$ and $\leq n\, r.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $\{Q, I\} \subseteq \Phi$, $r \in \Sigma_R$ and n is a natural number then $\geq n\, r^-.C$ and $\leq n\, r^-.C$ are concepts of $\mathcal{L}_{\Sigma,\Phi}$
  - if $U \in \Phi$ then U is a role of $\mathcal{L}_{\Sigma,\Phi}$
  - if Self $\in \Phi$ and $r \in \Sigma_R$ then $\exists r.\mathsf{Self}$ is a concept of $\mathcal{L}_{\Sigma,\Phi}$.

An *interpretation* in $\mathcal{L}_{\Sigma,\Phi}$ is a relational structure $\mathcal{I} = \left\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \right\rangle$ over Σ. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex roles and complex concepts as shown in Figure 19.1, where $\#\Gamma$ stands for the cardinality of the set Γ.

An (acyclic) *knowledge base* in $\mathcal{L}_{\Sigma,\Phi}$ is a pair $KB = \langle \mathcal{T}, \mathcal{A} \rangle$, where:

- $\mathcal{A}$ is a finite set, called the *ABox* of *KB*, consisting of *individual assertions* of the form $A(a)$ or $r(a,b)$, where $A \in \Sigma_C$, $r \in \Sigma_R$ and $a, b \in \Sigma_I$
- $\mathcal{T}$ is a finite list $(\varphi_1, \ldots, \varphi_n)$, called the *TBox* (terminological box) of *KB*, where each φ_i is a definition of one of the following forms:
  - $A = C$, where C is a concept of $\mathcal{L}_{\Sigma,\Phi}$ and $A \in \Sigma_C$ is a concept name not occurring in C, $\mathcal{A}$ and $\varphi_1, \ldots, \varphi_{i-1}$

$$
\begin{array}{rclrcl}
(R \circ S)^I & = & R^I \circ S^I & \top^I & = & \Delta^I \\
(R \sqcup S)^I & = & R^I \cup S^I & \bot^I & = & \emptyset \\
(R^*)^I & = & (R^I)^* & (\neg C)^I & = & \Delta^I \setminus C^I \\
(C?)^I & = & \{\langle x,x \rangle \mid C^I(x)\} & (C \sqcap D)^I & = & C^I \cap D^I \\
\varepsilon^I & = & \{\langle x,x \rangle \mid x \in \Delta^I\} & (C \sqcup D)^I & = & C^I \cup D^I \\
U^I & = & \Delta^I \times \Delta^I & \{a\}^I & = & \{a^I\} \\
(R^-)^I & = & (R^I)^{-1} & (\exists r.\mathsf{Self})^I & = & \{x \in \Delta^I \mid r^I(x,x)\}
\end{array}
$$

$$
\begin{array}{rcl}
(\forall R.C)^I & = & \{x \in \Delta^I \mid \forall y [R^I(x,y) \text{ implies } C^I(y)]\} \\
(\exists R.C)^I & = & \{x \in \Delta^I \mid \exists y [R^I(x,y) \text{ and } C^I(y)] \\
(\geq n R.C)^I & = & \{x \in \Delta^I \mid \#\{y \mid R^I(x,y) \text{ and } C^I(y)\} \geq n\} \\
(\leq n R.C)^I & = & \{x \in \Delta^I \mid \#\{y \mid R^I(x,y) \text{ and } C^I(y)\} \leq n\}
\end{array}
$$

Fig. 19.1. Interpretation of complex roles and complex concepts

- $r = R$, where R is a role of $\mathcal{L}_{\Sigma,\Phi}$ and $r \in \Sigma_R$ is a role name not occurring in R, $\mathcal{A}$ and $\varphi_1, \ldots, \varphi_{i-1}$.

The concept (respectively, role) names occurring in $\mathcal{A}$ are said to be *primitive* concepts (respectively, roles), while the concept (respectively, role) names occurring in the left hand side of '=' in the definitions from $\mathcal{T}$ are called *defined* concepts (respectively, roles).

An interpretation I in $\mathcal{L}_{\Sigma,\Phi}$ is a *model* of $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ if

- for every assertion $A(a) \in \mathcal{A}$, we have $a^I \in A^I$
- for every assertion $r(a,b) \in \mathcal{A}$, we have $\langle a^I, b^I \rangle \in r^I$
- for every definition $(A = C) \in \mathcal{T}$, we have $A^I = C^I$
- for every definition $(r = R) \in \mathcal{T}$, we have $r^I = R^I$.

Example 19.4. Let

$$
\begin{aligned}
\Sigma_I &= \{Alice, Bob, Claudia, Dave, Eva, Frank, George\} \\
\Sigma_C &= \{Human, Female, Male, Adult, Man, Woman, \\
&\qquad Parent, ParentWMC, DecendantOfAlice\} \\
\Sigma_R &= \{has\_child, has\_descendant, has\_parent, has\_ancestor\} \\
\mathcal{A} &= \{Female(Alice), Female(Claudia), Female(Eva), Adult(Alice), \\
&\qquad Adult(Bob), Adult(Claudia), Adult(Dave), Adult(George), \\
&\qquad has\_child(Alice, Dave), has\_child(Bob, Dave), \\
&\qquad has\_child(Claudia, Eva), has\_child(Dave, Eva), \\
&\qquad has\_child(Claudia, Frank), has\_child(Dave, Frank)\}
\end{aligned}
$$

$$\begin{aligned}\mathcal{T} = (&Human = \top,\\ &Male = \neg Female,\\ &Woman = Human \sqcap Female \sqcap Adult,\\ &Man = Human \sqcap Male \sqcap Adult,\\ &Parent = \exists has\_child.\top,\\ &ParentWMC = (\geq 5\, has\_child.\top),\\ &has\_descendant = has\_child \circ has\_child^*,\\ &has\_parent = has\_child^-,\\ &has\_ancestor = has\_parent \circ has\_parent^*,\\ &DecendantOfAlice = \exists has\_ancestor.\{Alice\}).\end{aligned}$$

Then $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ is a knowledge base in $\mathcal{L}_{\Sigma,\Phi}$, with $\Phi = \{I, O, Q\}$. The definition $Human = \top$ states that the domain of any model of KB consists of human beings. Note that, *Female* and *Adult* are primitive concepts, and *has_child* is a primitive role of KB. □

A knowledge base as defined above is similar to stratified logic programs [1]. Hence, we define the *standard model* of a knowledge base $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$ to be the interpretation $\mathcal{I}$ such that:

- $\Delta^\mathcal{I} = \Sigma_I$ (i.e. the domain of $\mathcal{I}$ consists of all the individual names of Σ)
- if A is a primitive concept of KB then $A^\mathcal{I} = \{a \mid A(a) \in \mathcal{A}\}$
- if r is a primitive role of KB then $r^\mathcal{I} = \{\langle a, b \rangle \mid r(a, b) \in \mathcal{A}\}$
- if $A \in \Sigma_C$ but A does not occur in KB then $A^\mathcal{I} = \emptyset$
- if $r \in \Sigma_R$ but r does not occur in KB then $r^\mathcal{I} = \emptyset$
- if $A = C$ is a definition from $\mathcal{T}$ then $A^\mathcal{I} = C^\mathcal{I}$
- if $r = R$ is a definition from $\mathcal{T}$ then $r^\mathcal{I} = R^\mathcal{I}$.

An *information system specified by a knowledge base* in $\mathcal{L}_{\Sigma,\Phi}$ is defined to be the standard model of the knowledge base in $\mathcal{L}_{\Sigma,\Phi}$. Note that such an information system is finite.

Example 19.5. Consider the knowledge base KB given in Example 19.4. The information system specified by KB is the interpretation $\mathcal{I}$ with:

$$\begin{aligned}\Delta^\mathcal{I} &= \{Alice, Bob, Claudia, Dave, Eva, Frank, George\}\\ x^\mathcal{I} &= x, \text{ for } x \in \{Alice, \dots, George\}\\ Human^\mathcal{I} &= \Delta^\mathcal{I}\\ Female^\mathcal{I} &= \{Alice, Claudia, Eva\}\end{aligned}$$

$$\begin{aligned}
Male^I &= \{Bob, Dave, Frank, George\}\\
Adult^I &= \{Alice, Bob, Claudia, Dave, George\}\\
Woman^I &= \{Alice, Claudia\}\\
Man^I &= \{Bob, Dave, George\}\\
has_child^I &= \{\langle Alice, Dave\rangle, \langle Bob, Dave\rangle,\\
&\quad \langle Claudia, Eva\rangle, \langle Dave, Eva\rangle,\\
&\quad \langle Claudia, Frank\rangle, \langle Dave, Frank\rangle\}\\
has_parent^I &= (has_child^I)^{-1}\\
has_descendant^I &= has_child^I \cup (has_child^I \circ has_child^I)\\
has_ancestor^I &= (has_descendant^I)^{-1}\\
Parent^I &= \{Alice, Bob, Claudia, Dave\}\\
ParentWMC^I &= \emptyset\\
DecendantOfAlice^I &= \{Dave, Eva, Frank\}. \qquad \square
\end{aligned}$$

Observe that any RS information system with discrete (or Boolean) attributes can be represented as an information system in $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma_R = \emptyset$ and $\Phi = \emptyset$. Namely,

- if an attribute A of an RS information system is Boolean, that is, $V_A = \{\mathsf{true}, \mathsf{false}\}$, then it can be treated as a concept name, standing for the set $\{x \in \Delta \mid A(x) = \mathsf{true}\}$
- if A is a discrete attribute, with $V_A = \{v_1, \ldots, v_k\}$, then it can be replaced by concept names $A_{v_1}, \ldots, A_{v_k}$, where each A_{v_i} is interpreted as the set $\{x \in \Delta \mid A(x) = v_i\}$.[6]

Example 19.6. Let

$$\begin{aligned}
Attrs &= \{Brand, Color, OpenOnSunday\}\\
V_{Brand} &= \{grocery, RTV\}\\
V_{Color} &= \{red, green, blue\}\\
\Delta &= \{shop_1, shop_2, shop_3, shop_4, shop_5\}
\end{aligned}$$

and let attribute values of the objects be the following:

	Brand	*Color*	*OpenOnSunday*
$shop_1$	*RTV*	*red*	true
$shop_2$	*RTV*	*green*	true
$shop_3$	*RTV*	*blue*	true
$shop_4$	*grocery*	*red*	false
$shop_5$	*grocery*	*green*	false

[6] For example, if *Color* is an attribute with possible values *red*, *green* and *blue*, then we can replace it by concept names *Red*, *Green*, *Blue*, and instead of writing, *for example*, $Color(x) = red$, we can write $Red(x)$.

Then the RS information system $\langle \Delta, Attrs \rangle$ can be represented by the information system $\mathcal{I}$ in $\mathcal{L}_{\Sigma,\Phi}$ specified as follows:

$$\begin{aligned}
\Phi &= \emptyset \\
\Sigma_R &= \emptyset \\
\Sigma_I &= \{shop_1, shop_2, shop_3, shop_4, shop_5\} \\
\Sigma_C &= \{RTV, Grocery, Red, Green, Blue, OpenOnSunday\} \\
\Delta^{\mathcal{I}} &= \Sigma_I \\
RTV^{\mathcal{I}} &= \{shop_1, shop_2, shop_3\} \\
Grocery^{\mathcal{I}} &= \{shop_4, shop_5\} \\
Red^{\mathcal{I}} &= \{shop_1, shop_4\} \\
Green^{\mathcal{I}} &= \{shop_2, shop_5\} \\
Blue^{\mathcal{I}} &= \{shop_3\} \\
OpenOnSunday^{\mathcal{I}} &= \{shop_1, shop_2, shop_3\}.
\end{aligned}$$

□

19.5.2 Bisimulation and Indiscernibility

In [5] Divroodi and Nguyen studied bisimulations for a number of DLs. In this subsection we generalize their notions and results to model indiscernibility of objects and study the problem of learning concepts. Let:

- Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$
- Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$
- $\mathcal{I}$ and $\mathcal{I}'$ be interpretations in $\mathcal{L}_{\Sigma,\Phi}$.

A binary relation $Z \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}'}$ is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-*bisimulation* between $\mathcal{I}$ and $\mathcal{I}'$ if the following conditions hold for every $a \in \Sigma_I^\dagger$, $A \in \Sigma_C^\dagger$, $r \in \Sigma_R^\dagger$, $x, y \in \Delta^{\mathcal{I}}$, $x', y' \in \Delta^{\mathcal{I}'}$:

$$Z(a^{\mathcal{I}}, a^{\mathcal{I}'}) \tag{19.16}$$

$$Z(x,x') \Rightarrow [A^{\mathcal{I}}(x) \Leftrightarrow A^{\mathcal{I}'}(x')] \tag{19.17}$$

$$[Z(x,x') \wedge r^{\mathcal{I}}(x,y)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y,y') \wedge r^{\mathcal{I}'}(x',y')] \tag{19.18}$$

$$[Z(x,x') \wedge r^{\mathcal{I}'}(x',y')] \Rightarrow \exists y \in \Delta^{\mathcal{I}}[Z(y,y') \wedge r^{\mathcal{I}}(x,y)], \tag{19.19}$$

if $I \in \Phi^\dagger$ then

$$[Z(x,x') \wedge r^{\mathcal{I}}(y,x)] \Rightarrow \exists y' \in \Delta^{\mathcal{I}'}[Z(y,y') \wedge r^{\mathcal{I}'}(y',x')] \tag{19.20}$$

$$[Z(x,x') \wedge r^{\mathcal{I}'}(y',x')] \Rightarrow \exists y \in \Delta^{\mathcal{I}}[Z(y,y') \wedge r^{\mathcal{I}}(y,x)], \tag{19.21}$$

if $O \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow [x = a^I \Leftrightarrow x' = a^{I'}], \tag{19.22}$$

if $Q \in \Phi^\dagger$ then

$$\text{if } Z(x,x') \text{ holds then, for every } r \in \Sigma_R^\dagger, \text{ there exists a bijection } h : \{y \mid r^I(x,y)\} \to \{y' \mid r^{I'}(x',y')\} \text{ such that } h \subseteq Z, \tag{19.23}$$

if $\{Q,I\} \subseteq \Phi^\dagger$ then (additionally)

$$\text{if } Z(x,x') \text{ holds then, for every } r \in \Sigma_R^\dagger, \text{ there exists a bijection } h : \{y \mid r^I(y,x)\} \to \{y' \mid r^{I'}(y',x')\} \text{ such that } h \subseteq Z, \tag{19.24}$$

if $U \in \Phi^\dagger$ then

$$\forall x \in \Delta^I \, \exists x' \in \Delta^{I'} \, Z(x,x') \tag{19.25}$$

$$\forall x' \in \Delta^{I'} \, \exists x \in \Delta^I \, Z(x,x'), \tag{19.26}$$

if $\mathsf{Self} \in \Phi^\dagger$ then

$$Z(x,x') \Rightarrow [r^I(x,x) \Leftrightarrow r^{I'}(x',x')]. \tag{19.27}$$

A concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ is said to be *invariant for* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-bisimulation* if, for every interpretations I and I' in $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma \supseteq \Sigma^\dagger$ and $\Phi \supseteq \Phi^\dagger$, and every $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation Z between I and I', if $Z(x,x')$ holds then $x \in C^I$ iff $x' \in C^{I'}$.

The following theorem can be proved in a similar way as [5, Theorem 3.4].

Theorem 19.1. *All concepts of* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ *are invariant for* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-bisimulation.* □

An interpretation I is *finitely branching* (or *image-finite*) w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ if, for every $x \in \Delta^I$ and every $r \in \Sigma_R^\dagger$:

- the set $\{y \in \Delta^I \mid r^I(x,y)\}$ is finite
- if $I \in \Phi^\dagger$ then the set $\{y \in \Delta^I \mid r^I(y,x)\}$ is finite.

Let $x \in \Delta^I$ and $x' \in \Delta^{I'}$. We say that x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-equivalent* to x' if, for every concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, $x \in C^I$ iff $x' \in C^{I'}$.

The following theorem can be proved in a similar way as [5, Theorem 4.1].

Theorem 19.2 (The Hennessy-Milner Property). *Let* Σ *and* $\Sigma^\dagger$ *be DL-signatures such that* $\Sigma^\dagger \subseteq \Sigma$, Φ *and* $\Phi^\dagger$ *be sets of DL-features such that* $\Phi^\dagger \subseteq \Phi$. *Let* I *and* I' *be interpretations in* $\mathcal{L}_{\Sigma,\Phi}$, *finitely branching w.r.t.* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ *and such that for every* $a \in \Sigma_I^\dagger$, a^I *is* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-equivalent to* $a^{I'}$. *Assume* $U \notin \Phi^\dagger$ *or* $\Sigma_I^\dagger \neq \emptyset$. *Then* $x \in \Delta^I$ *is* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-equivalent to* $x' \in \Delta^{I'}$ *iff there exists an* $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-bisimulation* Z *between* I *and* I' *such that* $Z(x,x')$ *holds.* □

We now have the following corollary.

Corollary 19.2. *Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, let Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and let I and I' be finite interpretations in $\mathcal{L}_{\Sigma,\Phi}$. Assume that $\Sigma_I^\dagger \neq \emptyset$ and, for every $a \in \Sigma_I^\dagger$, a^I is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $a^{I'}$. Then the relation $\{\langle x,x'\rangle \in \Delta^I \times \Delta^{I'} \mid x$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to $x'\}$ is an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I'.* □

We say that I is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-bisimilar* to I' if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I'. We say that $x \in \Delta^I$ is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-bisimilar* to $x' \in \Delta^{I'}$ if there exists an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and I' such that $Z(x,x')$ holds.

Remark 19.2. By Theorem 19.1, $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimilarity formalizes indiscernibility by the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. This is an important feature with many applications (see [7, 14, 31] for a more general context and numerous applications). Here let us emphasize that such indiscernibility relation provides the best approximations of a given concept expressed in the chosen sublanguage. Note that in [7, 14, 31] the underlying indiscernibility relation has not been constructed. □

An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-bisimulation between I and itself is called an $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$*-auto-bisimulation of* I. An $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I is said to be the *largest* if it is larger than or equal to ($\supseteq$) any other $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I.

Given an interpretation I in $\mathcal{L}_{\Sigma,\Phi}$, by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ we denote the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I, and by $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ we denote the binary relation on Δ^I with the property that $x \equiv_{\Sigma^\dagger,\Phi^\dagger,I} x'$ iff x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to x'.

Theorem 19.3. *Let Σ and $\Sigma^\dagger$ be DL-signatures such that $\Sigma^\dagger \subseteq \Sigma$, Φ and $\Phi^\dagger$ be sets of DL-features such that $\Phi^\dagger \subseteq \Phi$, and I be an interpretation in $\mathcal{L}_{\Sigma,\Phi}$. Then:*

- *the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I exists and is an equivalence relation*
- *if I is finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ then the relation $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ is the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation of I (i.e. the relations $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$ and $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincide).*

□

Theorem 19.3 can be proved as [5, Proposition 5.1 and Theorem 5.2].

By *terminological roughification* we mean any technique that uses the largest $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-auto-bisimulation relations as the equivalence relation for defining approximations.

The intended application areas are, in particular, concept learning and concept approximation in description logic-based information systems. Such applications and related techniques are studied in the next two subsections.

19.5.3 Concept Learning

Before presenting a method for learning concepts we first prove a theoretical result. We say that a set Y is *divided* by a set X if $Y \setminus X \neq \emptyset$ and $Y \cap X \neq \emptyset$. Thus, Y is not divided by X if either $Y \subseteq X$ or $Y \cap X = \emptyset$. A partition $P = \{Y_1, \ldots, Y_n\}$ is *consistent* with a set X if, for every $1 \leq i \leq n$, Y_i is not divided by X.

Theorem 19.4. *Let I be an information system in $\mathcal{L}_{\Sigma,\Phi}$, and let $X \subseteq \Delta^I$, $\Sigma^\dagger \subseteq \Sigma$ and $\Phi^\dagger \subseteq \Phi$. Then:*

1. *if there exists a concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $X = C^I$ then the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X*
2. *if the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X then there exists a concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $C^I = X$.*

Proof. As I is finite, it is finitely branching w.r.t. $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. By Theorem 19.3, $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$.

Consider the first assertion and assume that $X = C^I$ for some concept C of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$. Since $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ coincides with $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$, if x and x' belong to the same equivalence class by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$, then x is $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$-equivalent to x', and hence $x \in C^I$ iff $x' \in C^I$, that is, $\{x,x'\}$ is not divided by C^I. Therefore, the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X.

Consider the second assertion and assume that the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ is consistent with X. Let the partition be $\{Y_1,\ldots,Y_m\} \cup \{Z_1,\ldots,Z_n\}$, where $X = Y_1 \cup \ldots \cup Y_m$. Since Y_i and Z_j are different equivalence classes of $\equiv_{\Sigma^\dagger,\Phi^\dagger,I}$, we have that for each pair (i,j) with $1 \leq i \leq m$ and $1 \leq j \leq n$ there exists a concept $C_{i,j}$ of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $Y_i \subseteq C^I_{i,j}$ and $Z_j \cap C^I_{i,j} = \emptyset$. For each $1 \leq i \leq m$, let $C_i = C_{i,1} \sqcap \ldots \sqcap C_{i,n}$. Thus, $Y_i \subseteq C^I_i$, and $Z_j \cap C^I_i = \emptyset$ for all $1 \leq j \leq n$. Let $C = C_1 \sqcup \ldots \sqcup C_m$. Then, for all $1 \leq i \leq m$, $Y_i \subseteq C^I$, and for all $1 \leq j \leq n$, $Z_j \cap C^I = \emptyset$. Therefore, $C^I = X$. □

Let I be an information system in $\mathcal{L}_{\Sigma,\Phi}$, which can be either explicitly given as a finite interpretation in $\mathcal{L}_{\Sigma,\Phi}$ or specified by a knowledge base $KB = \langle \mathcal{T}, \mathcal{A} \rangle$ in $\mathcal{L}_{\Sigma,\Phi}$. Let $A_d \in \Sigma_I$ be a concept name standing for the "decision attribute". In the case when I is specified by KB, assume that A_d is not defined by the TBox $\mathcal{T}$ of KB. Suppose that A_d can be expressed by a concept C in $\mathcal{L}_{\Sigma,\Phi}$ not using A_d, and I is given as a training information system. How can we learn that concept C on the basis of I? That is, how can we learn a definition of A_d on the basis of I?

On the basis of machine learning techniques one can suggest that A_d is definable in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, for some specific $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$. One can even guide the machine learning process by extending Σ, Φ and $\mathcal{T}$ with new concepts and new roles together with their definitions before suggesting $\Sigma^\dagger$ and $\Phi^\dagger$. Without such suggestions, one can take $\Sigma^\dagger = \Sigma$ or $\Phi^\dagger = \Phi$, or use some method to try different possible values of $\Sigma^\dagger$ and $\Phi^\dagger$.

In this subsection we assume that $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$ are given, and the task is to study a definition of A_d in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ on the basis of I.

Our idea for this problem is based on the following observation:

> if A_d is definable in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ then, by the first assertion of Theorem 19.4, A^I_d must be the union of some equivalence classes of Δ^I w.r.t. $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.

Our general method is as follows:

1. Starting from the partition $\{\Delta^I\}$, make subsequent granulations to reach the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.

- The granulation process can be stopped as soon as the current partition is consistent with A_d^I (or when some criteria are met).
- The task can be done in the spirit of [5, Algorithm 1] for the case $\Phi^\dagger \subseteq \{I, O, U\}$, which is based on Hopcroft's automaton minimization algorithm [13]. That algorithm of [5] runs in polynomial time and it can be extended to deal also with the other cases of $\Phi^\dagger$. Also, one can use another strategy, optimizing some measure related to "quality" of the generated partition, but not time complexity.
- In the granulation process, we denote the blocks created so far in all steps by $Y_1, \ldots, Y_n$, where the current partition $\{Y_{i_1}, \ldots, Y_{i_k}\}$ consists of only some of them. We do not use the same subscript to denote blocks of different contents (i.e., we always use new subscripts obtained by increasing n for new blocks). We take care that, for each $1 \leq i \leq n$:
  - Y_i is characterized by an appropriate concept C_i (such that $Y_i = C_i^I$)
  - we keep information about whether Y_i is divided by A_d^I
  - if $Y_i \subseteq A_d^I$ then $LargestContainer[i] := j$, where $1 \leq j \leq n$ is the subscript of the largest block Y_j such that $Y_i \subseteq Y_j \subseteq A_d^I$

2. At the end, let $j_1, \ldots, j_h$ be all the indices from $\{i_1, \ldots, i_k\}$ such that $Y_{j_t} \subseteq A_d^I$ for $1 \leq t \leq h$, and let $\{l_1, \ldots, l_p\} = \{LargestContainer[j_t] \mid 1 \leq t \leq h\}$. Let C be a simplified form of $C_{l_1} \sqcup \ldots \sqcup C_{l_p}$. Return C as the result.

Example 19.7. Consider the information system given in Example 19.5. Assume that we want to learn a definition of concept *Parent* in the sublanguage $\mathcal{L}_{\Sigma^\dagger, \Phi^\dagger}$, where $\Sigma^\dagger = \{Adult, Female, has\_child\}$ and $\Phi^\dagger = \emptyset$. The respective steps are:

1. $Y_1 := \Delta^I$, $partition := \{Y_1\}$
2. partitioning Y_1 by *Adult*:
   - $Y_2 := \{Alice, Bob, Claudia, Dave, George\}$, $C_2 := Adult$
   - $Y_3 := \{Eva, Frank\}$, $C_3 := \neg Adult$
   - $partition := \{Y_2, Y_3\}$
3. partitioning Y_2 by *Female*:
   - $Y_4 := \{Alice, Claudia\}$, $C_4 := C_2 \sqcap Female$
   - $LargestContainer[4] := 4$ (as $Y_4 \subseteq Parent^I$)
   - $Y_5 := \{Bob, Dave, George\}$, $C_5 := C_2 \sqcap \neg Female$
   - $partition := \{Y_3, Y_4, Y_5\}$
4. partitioning Y_3 by *Female*:
   - $Y_6 := \{Eva\}$, $C_6 := C_3 \sqcap Female$
   - $Y_7 := \{Frank\}$, $C_7 := C_3 \sqcap \neg Female$
   - $partition := \{Y_4, Y_5, Y_6, Y_7\}$
5. partitioning Y_4 by *has_child*:
   - $Y_8 := \{Alice\}$, $C_8 := C_4 \sqcap \exists has\_child.C_5$
   - $LargestContainer[8] := 4$

- $Y_9 := \{Claudia\},\ C_9 := C_4 \sqcap \neg\exists has\_child.C_5$
- $LargestContainer[9] := 4$
- $partition := \{Y_5, Y_6, Y_7, Y_8, Y_9\}$

6. partitioning Y_5 by *has_child*:

- $Y_{10} := \{Bob, Dave\},\ C_{10} := C_5 \sqcap \exists has\_child.\top$
- $LargestContainer[10] := 10$ (as $Y_{10} \subseteq Parent^I$)
- $Y_{11} := \{George\},\ C_{11} := C_5 \sqcap \neg\exists has\_child.\top$
- $partition := \{Y_6, Y_7, Y_8, Y_9, Y_{10}, Y_{11}\}$.

The obtained partition is consistent with $Parent^I$, with Y_8, Y_9, Y_{10} contained in $Parent^I$, and Y_6, Y_7, Y_{11} disjoint with $Parent^I$. (It is not yet the partition corresponding to $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.)

Since $LargestContainer[8] = LargestContainer[9] = 4$, the concept we take into account before simplification is $C_4 \sqcup C_{10}$, which is

$$(Adult \sqcap Female) \sqcup (Adult \sqcap \neg Female \sqcap \exists has\_child.\top).$$

This concept can be simplified to the following equivalent form

$$Adult \sqcap (Female \sqcup \exists has\_child.\top)$$

which does not match the intended definition $Parent = \exists has\_child.\top$. However, it is equivalent in I to an acceptable definition $Parent = Adult \sqcap \exists has\_child.\top$, as all women in I are parents. □

Example 19.8. Consider again the information system given in Example 19.5. Assume that we want to learn a concept definition of $X = \{Dave, Eva, Frank\}$ in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{Alice, has\_child, has\_parent, has\_descendant, has\_ancestor\}$ and $\Phi^\dagger = \{O\}$. This task can be realized as follows:

1. $Y_1 := \Delta^I,\ partition := \{Y_1\}$
2. partitioning Y_1 by *Alice* using (19.22):

- $Y_2 := \{Alice\},\ C_2 := \{Alice\}$
- $Y_3 := \{Bob, Claudia, Dave, Eva, Frank, George\},\ C_3 := \neg\{Alice\}$
- $partition := \{Y_2, Y_3\}$

3. partitioning Y_3:

- The "selectors" are:
  - $\exists has\_child.C_3$, $\exists has\_parent.C_2$, $\exists has\_parent.C_3$,
  - $\exists has\_descendant.C_3$, $\exists has\_ancestor.C_2$, $\exists has\_ancestor.C_3$.
- If we apply the entropy gain measure then the best selectors are $\exists has\_parent.C_3$, $\exists has\_ancestor.C_2$, $\exists has\_ancestor.C_3$. Each of them partitions Y_3 into the following Y_4 and Y_5, but uses different C_4 and C_5:
  - $Y_4 := \{Dave, Eva, Frank\}$
  - $Y_5 := \{Bob, Claudia, George\}$.

4. Since the current partition $\{Y_2, Y_4, Y_5\}$ is consistent with X, the returned concept is C_4, which can be one of the following:
   - $\neg\{Alice\} \sqcap \exists has\_parent.\neg\{Alice\}$
   - $\neg\{Alice\} \sqcap \exists has\_ancestor.\{Alice\}$
   - $\neg\{Alice\} \sqcap \exists has\_ancestor.\neg\{Alice\}$.
5. If we test these solutions on the information system specified by the knowledge base that extends KB with the assertion $has\_child(Bob, George)$ then the solution $\neg\{Alice\} \sqcap \exists has\_ancestor.\{Alice\}$ has the best accuracy. □

Let us now describe our method in more details.

Let the current partition of Δ^I be $\{Y_{i_1}, \ldots, Y_{i_k}\}$. Consider partitioning of a block Y_{i_j} ($1 \leq j \leq k$). We want to find a concept D of $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, called a *selector*, to partition Y_{i_j}. Such a selector should actually partition Y_{i_j} into two non-empty parts (i.e. Y_{i_j} should be divided by D^I). It can be proved that to reach the partition corresponding to the equivalence relation $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$ it suffices to consider the following kinds of selectors:

- A, where $A \in \Sigma_C^\dagger$: this is related to (19.17)
- $\exists r.C_{i_t}$, where $r \in \Sigma_R^\dagger$ and $1 \leq t \leq k$: this is related to (19.18) and (19.19)
- in the case $I \in \Phi^\dagger$:

  $\exists r^-.C_{i_t}$, where $r \in \Sigma_R^\dagger$ and $1 \leq t \leq k$: this is related to (19.20) and (19.21)
- in the case $O \in \Phi^\dagger$:

  $\{a\}$, where $a \in \Sigma_I^\dagger$: this is related to (19.22)
- in the case $Q \in \Phi^\dagger$:

  $\geq l\, r.C_{i_t}$ and $\leq m\, r.C_{i_t}$, where $r \in \Sigma_R^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}$ and $0 \leq m < \#C_{i_t}$: this is related to (19.23)
- in the case $\{Q, I\} \subseteq \Phi^\dagger$:

  $\geq l\, r^-.C_{i_t}$ and $\leq m\, r^-.C_{i_t}$, where $r \in \Sigma_R^\dagger$, $1 \leq t \leq k$, $0 < l \leq \#C_{i_t}$ and $0 \leq m < \#C_{i_t}$: this is related to (19.24)
- in the case $\mathsf{Self} \in \Phi^\dagger$:

  $\exists r.\mathsf{Self}$, where $r \in \Sigma_R^\dagger$: this is related to (19.27).

Note that the conditions (19.25) and (19.26) are always satisfied when $I' = I$ and Z is an equivalence relation.

In practice, we prefer as simple as possible definitions for the learnt concept. Therefore, it is worth to consider also the following kinds of selectors (despite that they are expressible by the above mentioned ones), where n is the largest block subscript used so far:

- $\exists r.C_i$, $\exists r.\top$ and $\forall r.C_i$, where $r \in \Sigma_R^\dagger$ and $1 \leq i \leq n$
- in the case $I \in \Phi^\dagger$: $\exists r^-.C_i$, $\exists r^-.\top$ and $\forall r^-.C_i$, where $r \in \Sigma_R^\dagger$ and $1 \leq i \leq n$

- in the case $Q \in \Phi^\dagger$: $\geq l\, r.C_i$ and $\leq m\, r.C_i$,
  where $r \in \Sigma_R^\dagger$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$
- in the case $\{Q, I\} \subseteq \Phi^\dagger$: $\geq l\, r^-.C_i$ and $\leq m\, r^-.C_i$,
  where $r \in \Sigma_R^\dagger$, $1 \leq i \leq n$, $0 < l \leq \#C_i$ and $0 \leq m < \#C_i$.

A concept C characterizing A_d in the training information system I may not match the intended meaning of A_d. In particular, all of the above mentioned kinds of selectors do not use role constructors (like $R \sqcup S$, $R \circ S$ or R^*). However, the user acquainted with the machine learning problem for A_d may extend Σ and the TBox of the knowledge base specifying I to define new complex roles and then choose an appropriate $\Sigma^\dagger$. One can explicitly consider also selectors that use complex roles. This latter approach, in our opinion, is not appropriate, as the search space will be too large.

We now describe partitioning the block Y_{i_j} using a selector D. Recall that Y_{i_j} should be divided by D^I. The partition is done as follows:

- $s := n+1,\ t := n+2,\ n := n+2,\ Y_s := C_{i_j} \sqcap D,\ Y_t := C_{i_j} \sqcap \neg D$
- If $Y_{i_j} \subseteq A_d^I$ then
  - $\mathit{LargestContainer}[s] := \mathit{LargestContainer}[i_j]$
  - $\mathit{LargestContainer}[t] := \mathit{LargestContainer}[i_j]$

  else if $Y_s \subseteq A_d^I$ then $\mathit{LargestContainer}[s] := s$
  else if $Y_t \subseteq A_d^I$ then $\mathit{LargestContainer}[t] := t$.
- The new partition of Δ^I becomes $\{Y_{i_1}, \ldots, Y_{i_k}\} \setminus \{Y_{i_j}\} \cup \{Y_s, Y_t\}$.

An important matter is: which block from the current partition should be partitioned first? which selector should be used to partition it? This affects both the "quality" of the final partition and time complexity of the process. Some guides and possible strategies are given below:

- If two selectors D and D' partition Y_{i_j} in the same way then the simpler one is "better". For example, if $D = \exists r.C_l$, $D' = \exists r.C_m$, $Y_m \subset Y_l$, and D, D' partition Y_{i_j} in the same way, then C_l is simpler than C_m and D is more preferred than D'. This technique together with the use of *LargestContainer* guarantees that one can continue granulating the partition without the risk of worsening the "quality" of the final result. (Remember, however, that different paths resulting in the same partition may give different results, with different "quality".)
- One may prefer to partition a block divided by A_d^I first. Partitioning such a block, we may use some measure to choose a selector. A possible way is to use the entropy gain measure. Among the blocks of the current partition that are divided by A_d^I, to choose a block to partition we can also use some measure. Once again, it may be the entropy gain measure, taking into account also the possible selectors.
- Note, however, that one may be able to partition a block divided by A_d^I only after a block not divided by A_d^I has been partitioned.

- Simplicity of selectors and concepts characterizing blocks should be taken into account (*e.g.*, by combining it with the entropy gain measure). Let's say the form A is simpler than $\exists r.B$ and $\{a\}$. One may put some limits on the number of nominals and the nesting depth of $\forall$ and $\exists$ in a concept characterizing a block.
- As a possible strategy, one may follow the idea of Hopcroft's automaton minimization algorithm. The hope is that reducing the total number of created blocks (in the whole granulation process) makes the concepts characterizing the blocks of the final partition simpler. Besides, apart from quality of the result, time complexity is also important.

As usual, we may also use backtracking to find different solutions. During the search, only the best choices are tried and we will keep only a bounded number of the best solutions (according to some measure). The final solution will be the one that has the best accuracy on a test information system.

Simplifying a concept C to obtain a final definition for A_d can be done as follows:

1. We first normalize C while preserving equivalence, *for example*, by using the method proposed in [18]. Such normalization uses negation normal form, which may be essential for cutoffs described below.
2. Given a test information system $\mathcal{I}'$, we then simplify the obtained concept, without preserving equivalence, by representing the concept as a tree and repeat the following operations until accuracy of the definition cannot be improved on $\mathcal{I}'$:
   - Cut off a leaf of the tree if it improves accuracy of the definition on $\mathcal{I}'$.
   - If a subconcept of the definition can be replaced by a simpler one (*e.g.*, $\top$ or $\bot$) while not decreasing the accuracy on $\mathcal{I}'$ then do that replacement.
   - After each simplification, normalize the concept (preserving equivalence).

The other problems deserving consideration are: allowing a definition C not exactly matching A_d on $\mathcal{I}$, and classifying a new object when inconsistencies occur. The first problem can be dealt with by using standard methods and some measures. Consider the second problem. Inconsistencies may occur as in the following situation: converting a training RS information system I_0 with a decision attribute *Color* and $V_{Color} = \{red, green, blue\}$ to a training information system $\mathcal{I}$ in DL with concepts *Red*, *Green*, *Blue* to be learnt, one may get concepts C_{red}, C_{green}, C_{blue} as the result of the learning process, which overlap on a real information system $\mathcal{I}''$. A decision on whether an object x of $\mathcal{I}''$ which belongs, *for example*, to both $C_{red}^{\mathcal{I}''}$ and $C_{green}^{\mathcal{I}''}$ should be classified as red or green can be made based on the accuracy of C_{red} and C_{green} on a test information system $\mathcal{I}'$.

Note that an attempt to extend concept approximation using description logics was taken in [11] by using contextual indiscernibility relations used to represent uncertain concepts. A context is defined in [11] as a set of concepts. Roughly speaking, [11] proposes to define new atomic concepts by complex concepts and then to use those new atomic concepts for machine learning, applying traditional methods not based on description logics. The method we proposed is based on bisimulations and we find it much more promising for applications.

19.5.4 Bisimulation-Based Approximation of Concepts

The next problem we want to address is to learn a concept A_d not by giving its definition C (where A_d is a concept name and C is a complex concept), but by giving a pair $(\underline{C}, \overline{C})$ of concepts, where $\underline{C}$ plays the role of a lower approximation of A_d and $\overline{C}$ plays the role of an upper approximation of A_d. This follows the lines of Pawlak's rough set theory.

The problem is specified as follows:

- **given:** a training information system I in $\mathcal{L}_{\Sigma,\Phi}$, a concept name $A_d \in \Sigma_C$, and a sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ of $\mathcal{L}_{\Sigma,\Phi}$ with $\Sigma^\dagger \subseteq \Sigma \setminus \{A_d\}$ and $\Phi^\dagger \subseteq \Phi$
- **goal:** we want to learn an approximate definition of A_d, that is, a pair $(\underline{C}, \overline{C})$ of concepts in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ such that $\underline{C}^I \subseteq A_d^I \subseteq \overline{C}^I$ and $\underline{C}^I, \overline{C}^I$ closely approximate A_d^I.

The result of such learning can be improved by a test information system.

Our method for this problem, as described below, is based on bisimulation:

- Compute the partition of Δ^I by $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$, further denoted by $\{Y_{i_1}, \ldots, Y_{i_k}\}$, together with concepts C_{i_t} characterizing Y_{i_t} (i.e. $C_{i_t}^I = Y_{i_t}$ for $1 \leq t \leq k$) as described in the previous subsection.
- Take $\underline{C} = C_{j_1} \sqcup \ldots \sqcup C_{j_h}$, where $j_1, \ldots, j_h$ are all the indices among $i_1, \ldots, i_k$ such that $Y_{j_t} \subseteq A_d^I$ for all $1 \leq t \leq h$.
- Take $\overline{C} = C_{j'_1} \sqcup \ldots \sqcup C_{j'_{h'}}$, where $j'_1, \ldots, j'_{h'}$ are all the indices among $i_1, \ldots, i_k$ such that $Y_{j'_t} \cap A_d^I \neq \emptyset$ for all $1 \leq t \leq h'$.
- Normalize $\underline{C}$ and $\overline{C}$, while preserving equivalence.

The pair $(\underline{C}, \overline{C})$, obtained as above, is a pair of concepts in $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$ that approximates A_d on I most closely (in the sense that $\underline{C}^I \subseteq A_d^I \subseteq \overline{C}^I$ and the sets $A_d^I \setminus \underline{C}^I$ and $\overline{C}^I \setminus A_d^I$ are the smallest ones).

The accuracy on I does not imply accuracy on other information systems. Following the Ockham's razor principle, we pay attention to simplicity of $(\underline{C}, \overline{C})$ in order to increase their overall accuracy. Here, we can use the following techniques:

- We use *LargestContainer* (see Subsection 19.5.3) to obtain a simpler form for $\underline{C}$.
- In the granulation process of Δ^I, we can stop as soon as the current partition is good enough according to some measure, and use it to compute $\underline{C}$ and $\overline{C}$.
- Using a test information system we can simplify $\underline{C}$ and $\overline{C}$ (without preserving equivalence) by applying different kinds of simplification as discussed in the previous subsection, taking into account the accuracies of the lower and upper approximations on the test information system and the relation between them.

Example 19.9. Consider again the information system given in Example 19.5. We want to learn a concept definition or a concept approximation for the set $X = \{Alice, Bob, Claudia\}$ in the sublanguage $\mathcal{L}_{\Sigma^\dagger,\Phi^\dagger}$, where $\Sigma^\dagger = \{Adult, has\_child\}$ and $\Phi^\dagger = \emptyset$. This task can be realized as follows:

1. $Y_1 := \Delta^I$, $partition := \{Y_1\}$
2. partitioning Y_1 by *Adult*:
   - $Y_2 := \{Alice, Bob, Claudia, Dave, George\}$, $C_2 := Adult$
   - $Y_3 := \{Eva, Frank\}$, $C_3 := \neg Adult$
   - $partition := \{Y_2, Y_3\}$
3. partitioning Y_2 by $\exists has\_child.\top$:
   - $Y_4 := \{Alice, Bob, Claudia, Dave\}$, $C_4 := C_2 \sqcap \exists has\_child.\top$
   - $Y_5 := \{George\}$, $C_5 := C_2 \sqcap \neg\exists has\_child.\top$
   - $partition := \{Y_3, Y_4, Y_5\}$
4. partitioning Y_4 by $\exists has\_child.C_2$ (we use the selector $\exists has\_child.C_2$ instead of $\exists has\_child.C_4$ because it is simpler and has the same effect):
   - $Y_6 := \{Alice, Bob\}$, $C_6 := C_4 \sqcap \exists has\_child.C_2$
   - $Y_7 := \{Claudia, Dave\}$, $C_7 := C_4 \sqcap \neg\exists has\_child.C_2$
   - $partition := \{Y_3, Y_5, Y_6, Y_7\}$
5. The current partition cannot be granulated anymore. (It corresponds to $\sim_{\Sigma^\dagger,\Phi^\dagger,I}$.)
6. Since only Y_6 from the current partition $\{Y_3, Y_5, Y_6, Y_7\}$ is a subset of X, the lower approximation of X is characterized by $C_6 = Adult \sqcap \exists has\_child.\top \sqcap \exists has\_child.Adult$, which can be simplified to $Adult \sqcap \exists has\_child.Adult$.
7. Since only Y_6 and Y_7 from the current partition $\{Y_3, Y_5, Y_6, Y_7\}$ overlap with X, the upper approximation of X is characterized by $C_6 \sqcup C_7$, which can be simplified to $C_4 = Adult \sqcap \exists has\_child.\top$. □

19.6 Conclusions

In the current chapter, we have studied roughification methods allowing one to construct indiscernibility relations on the basis of background knowledge. We have first studied indiscernibility based on similarity relations, showing that such relations can be turned into equivalence relations providing more accurate approximations. Next, we introduced roughifications based on relational databases and finally terminological roughifications, where indiscernibility coincides with indiscernibility by formulas of considered description logics. To our best knowledge, the proposed techniques and their applications are novel. It is worth emphasizing that our work is a pioneering one that uses bisimulation for machine learning in the context of description logics.

We have considered applications of the proposed techniques for improving accuracy of approximations, granulating relational databases as well as in concept learning and concept approximations. The last mentioned application areas have usually been studied in the context of information systems using only attributes (and sometimes also "external" relational structures) [21, 20]. In approaches based on RS information systems, concepts are usually characterized by formulas built from unary predicates (corresponding to attributes), using propositional connectives. On the other hand, concept learning and concept approximation in information systems based on description logics require new methods and algorithms. Most ideas for them may be inspired from the traditional ones (like the ones based on decision rules, decision trees, reducts, and local reducts). However, additional ideas are needed to generalize such approaches to the case of description logics. We have shown that bisimulation is a good starting point.

As interesting continuations of the research reported in this chapter we consider extensions of roughifications techniques by considering other logical formalisms.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1996)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook. Cambridge University Press (2002)
3. Baader, F., Nutt, W.: Basic description logics. In: Baader et al. [2], pp. 47–100
4. Borgida, A., Lenzerini, M., Rosati, R.: Description logics for databases. In: Baader et al. [2], pp. 472–494
5. Divroodi, A., Nguyen, L.: On bisimulations for description logics. CoRR abs/1104.1964 (2011) (appeared also in the proceedings of CS&P 2011, pp. 99–110)
6. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge representation techniques. A rough set approach. STUDFUZZ, vol. 202, Springer (2006)
7. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing strongest necessary and weakest sufficient conditions of first-order formulas. In: International Joint Conference on AI, IJCAI 2001, pp. 145–151 (2000)
8. Doherty, P., Łukaszewicz, W., Szałas, A.: Tolerance spaces and approximative representational structures. In: Proceedings of 26th German Conference on Artificial Intelligence. Springer (2003)
9. 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)
10. Doherty, P., Szałas, A.: A correspondence framework between three-valued logics and similarity-based approximate reasoning. Fundamenta Informaticae 75(1-4) (2007)
11. Fanizzi, N., d'Amato, C., Esposito, F., Lukasiewicz, T.: Representing uncertain concepts in rough description logics via contextual indiscernibility relations. In: Proceedings of URSW 2008. CEUR Workshop Proceedings, vol. 423 (2008)
12. Greco, S., Matarazzo, B., Słowiński, R.: Fuzzy Similarity Relation as a Basis for Rough Approximations. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 283–289. Springer, Heidelberg (1998)
13. Hopcroft, J.: An $n \log n$ algorithm for minimizing states in a finite automaton (1971), `ftp://reports.stanford.edu/pub/cstr/reports/cs/tr/71/190/CS-TR-71-190.pdf`
14. Lin, F.: On strongest necessary and weakest sufficient conditions. In: Cohn, A., Giunchiglia, F., Selman, B. (eds.) Proc. 7th International Conf. on Principles of Knowledge Representation and Reasoning, KR 2000, pp. 167–175. Morgan Kaufmann Pub., Inc. (2000)

15. Lin, T.: Granular computing on binary relations I, II. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1: Methodology and Applications. STUDFUZZ, vol. 17, pp. 107–140. Physica-Verlag (1998)
16. Nardi, D., Brachman, R.J.: An introduction to description logics. In: Baader et al. [2], pp. 5–44
17. Nguyen, H., Skowron, A., Stepaniuk, J.: Granular computing: A rough set approach. Computational Intelligence 17, 514–544 (2001)
18. Nguyen, L.: An efficient tableau prover using global caching for the description logic $\mathcal{ALC}$. Fundamenta Informaticae 93(1-3), 273–288 (2009)
19. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about. Data. Kluwer Academic Publishers, Dordrecht (1991)
20. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Inf. Sci. 177(1), 41–73 (2007)
21. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
22. Peters, J., Ramanna, S., Skowron, A., Stepaniuk, J., Suraj, Z., Borkowski, M.: Sensor fusion: A rough granular approach. In: Proc. of the Joint 9th International Fuzzy Systems Association World Congress and 20th NAFIPS International Conference, pp. 1367–1371 (2001)
23. Polkowski, L., Skowron, A.: Towards adaptive calculus of granules. In: Zadeh, L., Kacprzyk, J. (eds.) Computing with Words in Information/Intelligent Systems, vol. 1-2, pp. 201–227. Physica-Verlag (1999)
24. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
25. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems 16/1, 57–86 (2001)
26. Skowron, A., Stepaniuk, J.: Information granules and rough-neurocomputing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neuro Computing: Techniques for Computing with Words, pp. 43–84. Springer (2004)
27. Ślęzak, D.: Rough sets and few-objects-many-attributes problem: The case study of analysis of gene expression data sets. In: FBIT, pp. 437–442. IEEE Computer Society (2007)
28. Ślęzak, D., Wróblewski, J.: Roughfication of Numeric Decision Tables: The Case Study of Gene Expression Data. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 316–323. Springer, Heidelberg (2007)
29. Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations. In: Wang, P. (ed.) Advances in Machine Intelligence & Soft Computing, pp. 17–33. Bookwrights, Raleigh (1997)
30. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. on Data and Knowledge Engineering 12(2), 331–336 (2000)
31. Szałas, A.: Second-order reasoning in description logics. Journal of Applied Non-Classical Logics 16(3-4), 517–530 (2006)