TLP **8** (3): 271–300, 2008. © 2008 Cambridge University Press
doi:10.1017/S1471068407003195 First published online 4 January 2008 Printed in the United Kingdom

Building Rules on Top of Ontologies for the Semantic Web with Inductive Logic Programming

FRANCESCA A. LISI

Dipartimento di Informatica, Università degli Studi di Bari, Via Orabona 4, 70125 Bari, Italy
(*e-mail:* lisi@di.uniba.it)

submitted 28 July 2006; revised 10 September 2007; accepted 18 October 2007

Abstract

Building rules on top of ontologies is the ultimate goal of the logical layer of the Semantic Web. To this aim, an ad-hoc markup language for this layer is currently under discussion. It is intended to follow the tradition of hybrid knowledge representation and reasoning systems, such as $\mathcal{AL}$-log that integrates the description logic $\mathcal{ALC}$ and the function-free Horn clausal language DATALOG. In this paper, we consider the problem of automating the acquisition of these rules for the Semantic Web. We propose a general framework for rule induction that adopts the methodological apparatus of Inductive Logic Programming and relies on the expressive and deductive power of $\mathcal{AL}$-log. The framework is valid whatever the scope of induction (description versus prediction) is. Yet, for illustrative purposes, we also discuss an instantiation of the framework which aims at description and turns out to be useful in Ontology Refinement.

KEYWORDS: Inductive logic programming, hybrid knowledge representation and reasoning systems, ontologies, semantic web

1 Introduction

During the last decade, increasing attention has been paid on *ontologies* and their role in Knowledge Engineering (Staab and Studer 2004). In the philosophical sense, we may refer to an ontology as a particular system of categories accounting for a certain vision of the world. As such, this system does not depend on a particular language: Aristotle's ontology is always the same, independently of the language used to describe it. On the other hand, in its most prevalent use in Artificial Intelligence, an ontology refers to an engineering artifact (more precisely, produced according to the principles of *Ontological Engineering* (Gómez-Pérez *et al.* 2004)), constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words. This set of assumptions has usually the form of a first-order logical theory, where vocabulary words appear as unary or binary predicate names, respectively called

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

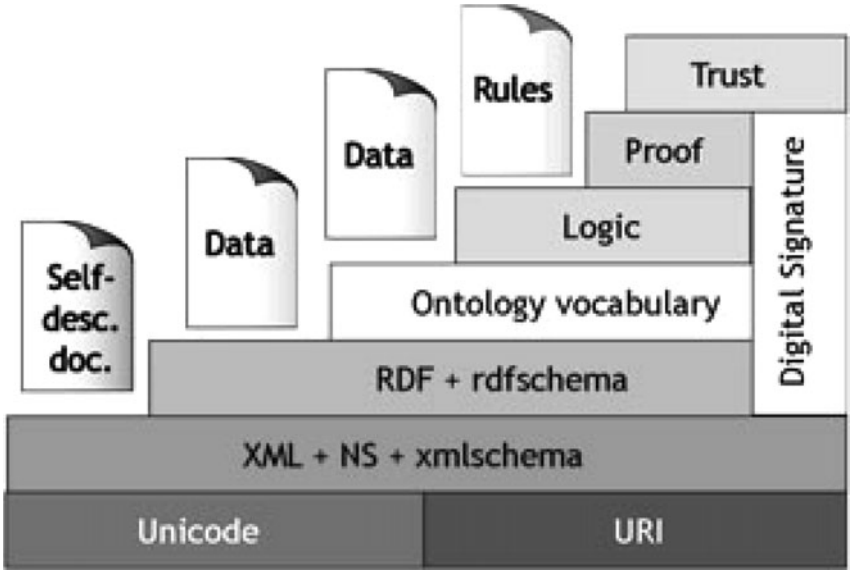


Fig. 1. Architecture of the Semantic Web.

concepts and relations. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation. The two readings of ontology described above are indeed related to each other, but in order to solve the terminological impasse, the word conceptualization is used to refer to the philosophical reading as appears in the following definition, based on (Gruber 1993): *An ontology is a formal explicit specification of a shared conceptualization for a domain of interest*. Among the other things, this definition emphasizes the fact that an ontology has to be specified in a language that comes with a formal semantics. Only by using such a formal approach, ontologies provide the machine-interpretable meaning of concepts and relations that is expected when using an ontology-based approach. Among the formalisms proposed by Ontological Engineering, the most currently used are *Description Logics* (DLs) (Baader *et al.* 2003). Note that DLs are decidable fragments of First Order Logic (FOL) that are incomparable with Horn Clausal Logic (HCL), as regards the expressive power (Borgida 1996) and the semantics (Rosati 2005).

Ontology Engineering, notably its DL-based approach, is playing a relevant role in the definition of the *Semantic Web*. The Semantic Web is the vision of the World Wide Web enriched by machine-processable information, which supports the user in his tasks (Berners-Lee *et al.* 2001). The architecture of the Semantic Web is shown in Figure 1. It consists of several layers, each of which is equipped with an ad-hoc markup language. In particular, the design of the markup language for the *ontological layer*, OWL[1], has been based on the very expressive DL $\mathcal{SHOIN}(\mathbf{D})$ (Horrocks *et al.* 2000, 2003). Whereas OWL is already undergoing the standardization process

[1] `http://www.w3.org/2004/OWL/`.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

at W3C, the debate around a unified language for *rules* is still ongoing. Proposals like SWRL[2] extend OWL with constructs inspired to Horn clauses in order to meet the primary requirement of the *logical layer*: "to build rules on top of ontologies." SWRL is intended to bridge the notorious gaps between DLs and HCL in a way that is similar in the spirit to hybridization in Knowledge Representation and Reasoning (KR&R) systems, such as $\mathcal{AL}$-log (Donini *et al.* 1998). Generally speaking, *hybrid systems* are KR&R systems, which are constituted by two or more subsystems dealing with distinct portions of a single knowledge base by performing specific reasoning procedures (Frisch and Cohn 1991). The motivation for investigating and developing such systems is to improve on two basic features of KR&R formalisms, namely *representational adequacy* and *deductive power*, by preserving the other crucial feature, i.e., *decidability*. In particular, combining DLs with HCL can easily yield to undecidability if the interface between them is not reduced (Levy and Rousset 1998). The hybrid system $\mathcal{AL}$-log integrates $\mathcal{ALC}$ (Schmidt-Schauss and Smolka 1991) and DATALOG (Ceri *et al.* 1990) by using $\mathcal{ALC}$ concept assertions essentially as type constraints on variables. It has been very recently mentioned as the blueprint for *well-founded* Semantic Web rule markup languages because its underlying form of integration (called *safe*) assures semantic and computational advantages that SWRL—though more expressive than $\mathcal{AL}$-log—currently can not assure (Rosati 2005).

Defining rules (including the ones for the Semantic Web) has been usually considered as a demanding task from the viewpoint of Knowledge Engineering. It is often supported by Machine Learning algorithms that can vary in the approaches. The approach known under the name of Inductive Logic Programming (ILP) seems to be promising for the case at hand due to the common roots with Logic Programming (Flach and Lavrač 2002). ILP has been historically concerned with rule induction from examples and background knowledge within the representation framework of HCL and with the aim of prediction (Nienhuys-Cheng and de Wolf 1997). More recently, ILP has moved toward either different FOL fragments (e.g., DLs) or new learning goals (e.g., description). In this paper, we resort to the methodological apparatus of ILP to define a *general* framework for learning rules on top of ontologies for the Semantic Web within the KR&R framework of $\mathcal{AL}$-log. The framework proposed is general in the sense, that it is valid whatever the scope of induction (description versus prediction) is. For the sake of illustration, we concentrate on an instantiation of the framework for the case of description.

The paper is organized as follows. Section 2 introduces the basic notions of $\mathcal{AL}$-log. Section 3 defines the framework for learning rules in $\mathcal{AL}$-log. Section 4 illustrates an instantiation of the framework. Section 5 concludes the paper with final remarks. Appendix A clarifies the links between OWL and DLs.

2 Basics of $\mathcal{AL}$-log

The system $\mathcal{AL}$-log (Donini *et al.* 1998) integrates two KR&R systems: Structural and relational.

[2] `http://www.w3.org/Submission/SWRL/`.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Table 1. *Syntax and semantics of $\mathcal{ALC}$*

bottom (resp. top) concept	$\bot$ (resp. $\top$)	$\emptyset$ (resp. $\Delta^{\mathcal{I}}$)
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
value restriction	$\forall R \cdot C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y\ (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
existential restriction	$\exists R \cdot C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y\ (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
equivalence axiom	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
subsumption axiom	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$\langle a,b \rangle : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

2.1 The structural subsystem

The structural part Σ is based on $\mathcal{ALC}$ (Schmidt-Schauss and Smolka 1991) and allows for the specification of knowledge in terms of classes (*concepts*), binary relations between classes (*roles*), and instances (*individuals*). Complex concepts can be defined from atomic concepts and roles by means of constructors (see Table 1). Also, Σ can state both is-a relations between concepts (*axioms*) and instance-of relations between individuals (resp. couples of individuals) and concepts (resp. roles) (*assertions*). An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for Σ consists of a domain $\Delta^{\mathcal{I}}$ and a mapping function $\cdot^{\mathcal{I}}$. In particular, individuals are mapped to elements of $\Delta^{\mathcal{I}}$, such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (*Unique Names Assumption* (UNA) (Reiter 1980)). If $\mathcal{O} \subseteq \Delta^{\mathcal{I}}$ and $\forall a \in \mathcal{O} : a^{\mathcal{I}} = a$, $\mathcal{I}$ is called *$\mathcal{O}$-interpretation*. Also, Σ represents many different interpretations, i.e., all its models (*Open World Assumption* (OWA) (Baader *et al.* 2003)).

The main reasoning task for Σ is the *consistency check*. This test is performed with a *tableau calculus* that starts with the tableau branch $S = \Sigma$ and adds assertions to S by means of *propagation rules*, such as

- $S \rightarrow_{\sqcup} S \cup \{s : D\}$ if
  (1) $s : C_1 \sqcup C_2$ is in S,
  (2) $D = C_1$ and $D = C_2$,
  (3) neither $s : C_1$ nor $s : C_2$ is in S.
- $S \rightarrow_{\forall} S \cup \{t : C\}$ if
  (1) $s : \forall R \cdot C$ is in S,
  (2) sRt is in S,
  (3) $t : C$ is not in S.
- $S \rightarrow_{\sqsubseteq} S \cup \{s : C' \sqcup D\}$ if
  (1) $C \sqsubseteq D$ is in S,
  (2) s appears in S,
  (3) C' is the NNF concept equivalent to $\neg C$
  (4) $s : \neg C \sqcup D$ is not in S.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

- $S \rightarrow_{\perp} \{s : \perp\}$ if
  (1) $s : A$ and $s : \neg A$ are in S, or
  (2) $s : \neg\top$ is in S,
  (3) $s : \perp$ is not in S,

until either a contradiction is generated or an interpretation satisfying S can be easily obtained from it.

2.2 The relational subsystem

The relational part of $\mathcal{AL}$-log allows one to define DATALOG[3] programs enriched with *constraints* of the form $s : C$, where s is either a constant or a variable, and C is an $\mathcal{ALC}$-concept. Note that the usage of concepts as typing constraints applies only to variables and constants that already appear in the clause. The symbol & separates constraints from DATALOG atoms in a clause.

Definition 1
A *constrained* DATALOG *clause* is an implication of the form

$$\alpha_0 \leftarrow \alpha_1, \ldots, \alpha_m \& \gamma_1, \ldots, \gamma_n,$$

where $m \geqslant 0$, $n \geqslant 0$, α_i are DATALOG atoms and γ_j are constraints. A *constrained* DATALOG *program* Π is a set of constrained DATALOG clauses.

An *$\mathcal{AL}$-log knowledge base* $\mathcal{B}$ is the pair $\langle \Sigma, \Pi \rangle$, where Σ is an $\mathcal{ALC}$ knowledge base and Π is a constrained DATALOG program. For a knowledge base to be acceptable, it must satisfy the following conditions:

- The set of DATALOG predicate symbols appearing in Π is disjoint from the set of concept and role symbols appearing in Σ.
- The alphabet of constants in Π coincides with the alphabet $\mathcal{O}$ of the individuals in Σ. Furthermore, every constant in Π appears also in Σ.
- For each clause in Π, each variable occurring in the constraint part occurs also in the DATALOG part.

These properties state a *safe* interaction between the structural and the relational part of an $\mathcal{AL}$-log knowledge base, thus solving the semantic mismatch between the OWA of $\mathcal{ALC}$ and the CWA of DATALOG (Rosati 2005). This interaction is also at the basis of a model-theoretic semantics for $\mathcal{AL}$-log. We call Π_D the set of DATALOG clauses obtained from the clauses of Π by deleting their constraints. We define an *interpretation* $\mathcal{J}$ for $\mathcal{B}$ as the union of an $\mathcal{O}$-interpretation $\mathcal{I}_{\mathcal{O}}$ for Σ (i.e., an interpretation compliant with the unique names assumption) and an Herbrand interpretation $\mathcal{I}_{\mathcal{H}}$ for Π_D. An interpretation $\mathcal{J}$ is a *model* of $\mathcal{B}$, if $\mathcal{I}_{\mathcal{O}}$ is a model of Σ, and for each ground instance $\alpha'_0 \leftarrow \alpha'_1, \ldots, \alpha'_m \& \gamma'_1, \ldots, \gamma'_n$ of each clause $\alpha_0 \leftarrow \alpha_1, \ldots, \alpha_m \& \gamma'_1, \ldots, \gamma'_n$ in Π, either there exists one γ'_i, $i \in \{1, \ldots, n\}$, that is not satisfied by $\mathcal{J}$, or $\alpha'_0 \leftarrow \alpha'_1, \ldots, \alpha'_m$ is satisfied by $\mathcal{J}$. The notion of *logical*

[3] For the sake of brevity, we assume the reader to be familiar with DATALOG.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

consequence paves the way to the definition of answer set for queries. *Queries* to $\mathcal{AL}$-log knowledge bases are special cases of Definition 1. An *answer* to the query Q is a ground substitution σ for the variables in Q. The answer σ is *correct* w.r.t. a $\mathcal{AL}$-log knowledge base $\mathcal{B}$ if $Q\sigma$ is a logical consequence of $\mathcal{B}$ ($\mathcal{B} \models Q\sigma$). The *answer set* of Q in $\mathcal{B}$ contains all the correct answers to Q w.r.t. $\mathcal{B}$.

Reasoning for $\mathcal{AL}$-log knowledge bases is based on *constrained SLD-resolution* (Donini *et al.* 1998), i.e., an extension of SLD-resolution to deal with constraints. In particular, the constraints of the resolvent of a query Q and a constrained DATALOG clause E are recursively simplified by replacing couples of constraints $t : C$, $t : D$ with the equivalent constraint $t : C \sqcap D$. The one-to-one mapping between constrained SLD-derivations and the SLD-derivations obtained by ignoring the constraints is exploited to extend known results for DATALOG to $\mathcal{AL}$-log. Note that in $\mathcal{AL}$-log, a derivation of the empty clause with associated constraints does not represent a refutation. It actually infers that the query is true in those models of $\mathcal{B}$ that satisfy its constraints. Therefore, in order to answer a query, it is necessary to collect enough derivations ending with a constrained empty clause, such that every model of $\mathcal{B}$ satisfies the constraints associated with the final query of at least, one derivation.

Definition 2
Let $Q^{(0)}$ be a query $\leftarrow \beta_1, \ldots, \beta_m \& \gamma_1, \ldots, \gamma_n$ to a $\mathcal{AL}$-log knowledge base $\mathcal{B}$. A *constrained SLD-refutation* for $Q^{(0)}$ in $\mathcal{B}$ is a finite set $\{d_1, \ldots, d_s\}$ of constrained SLD-derivations for $Q^{(0)}$ in $\mathcal{B}$, such that

(1) for each derivation d_i, $1 \leqslant i \leqslant s$, the last query $Q^{(n_i)}$ of d_i is a constrained empty clause;
(2) for every model $\mathcal{J}$ of $\mathcal{B}$, there exists at least one derivation d_i, $1 \leqslant i \leqslant s$, such that $\mathcal{J} \models Q^{(n_i)}$.

Constrained SLD-refutation is a complete and sound method for answering *ground* queries (Donini *et al.* 1998).

Lemma 1
Let Q be a ground query to an $\mathcal{AL}$-log knowledge base $\mathcal{B}$. It holds that $\mathcal{B} \vdash Q$, if and only if $\mathcal{B} \models Q$.

An answer σ to a query Q is a *computed answer* if there exists a constrained SLD-refutation for $Q\sigma$ in $\mathcal{B}$ ($\mathcal{B} \vdash Q\sigma$). The set of computed answers is called the *success set* of Q in $\mathcal{B}$. Furthermore, given *any* query Q, the success set of Q in $\mathcal{B}$ coincides with the answer set of Q in $\mathcal{B}$. This provides an operational means for computing correct answers to queries. Indeed, it is straightforward to see that the usual reasoning methods for DATALOG allow us to collect in a finite number of steps enough constrained SLD-derivations for Q in $\mathcal{B}$ to construct a refutation, if any. Derivations must satisfy both conditions of Definition 2. In particular, the latter requires some reasoning on the structural component of $\mathcal{B}$. This is done by applying the tableau calculus as shown in the following example.

Constrained SLD-resolution is *decidable* (Donini *et al.* 1998). Furthermore, because of the safe interaction between $\mathcal{ALC}$ and DATALOG, it supports a form of

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

closed world reasoning, i.e., it allows one to pose queries under the assumption, that part of the knowledge base is complete (Rosati 2005).

3 The general framework for learning rules in $\mathscr{AL}$-log

In our framework for learning in $\mathscr{AL}$-log, we represent inductive hypotheses as constrained DATALOG clauses and data as an $\mathscr{AL}$-log knowledge base $\mathscr{B}$. In particular, $\mathscr{B}$ is composed of a *background knowledge* $\mathscr{K}$ and a set O of *observations*. We assume $\mathscr{K} \cap O = \emptyset$.

To define the framework, we resort to the methodological apparatus of ILP, which requires the following ingredients to be chosen:

- the *language* $\mathscr{L}$ *of hypotheses*,
- a *generality order* $\succeq$ for $\mathscr{L}$ to structure the space of hypotheses,
- a *relation* to test the *coverage* of hypotheses in $\mathscr{L}$ against observations in O w.r.t. $\mathscr{K}$.

The framework is general, meaning that it is valid, whatever the scope of induction (description/prediction) is. Therefore, the DATALOG literal $q(\vec{X})$[4] in the head of hypotheses represents a concept to be either discriminated from others (*discriminant induction*) or characterized (*characteristic induction*).

This section collects and upgrades theoretical results published in (Lisi and Malerba 2003a,b; Lisi and Esposito 2004).

3.1 The language of hypotheses

To be suitable as language of hypotheses, constrained DATALOG clauses must satisfy the following restrictions.

First, we impose constrained DATALOG clauses to be linked and connected (or range-restricted) as usual in ILP (Nienhuys-Cheng and de Wolf 1997).

Definition 3
Let H be a constrained DATALOG clause. A term t in some literal $l_i \in H$ is *linked* with linking-chain of length 0, if t occurs in $head(H)$, and is linked with linking-chain of length $d+1$, if some other term in l_i is linked with linking-chain of length d. The link-depth of a term t in some $l_i \in H$ is the length of the shortest linking-chain of t. A literal $l_i \in H$ is linked, if at least, one of its terms is linked. The clause H itself is linked, if each $l_i \in H$ is linked. The clause H is *connected*, if each variable occurring in $head(H)$ also occur in $body(H)$.

Second, we impose constrained DATALOG clauses to be compliant with the bias of Object Identity (OI) (Semeraro *et al.* 1998). This bias can be considered as an extension of the UNA from the semantic level to the syntactic one of $\mathscr{AL}$-log. We would like to remind the reader that this assumption holds in $\mathscr{ALC}$. Also, it holds

[4] $\vec{X}$ is a tuple of variables.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

naturally for ground-constrained DATALOG clauses because the semantics of $\mathscr{AL}$-log adopts Herbrand models for the DATALOG part and $\mathscr{O}$-models for the constraint part. Conversely, it is not guaranteed in the case of non-ground constrained DATALOG clauses, e.g., different variables can be unified. The OI bias can be the starting point for the definition of either an equational theory or a quasi-order for constrained DATALOG clauses. The latter option relies on a restricted form of substitution, whose bindings avoid the identification of terms.

Definition 4
A substitution σ is an *OI-substitution* w.r.t. a set of terms T iff $\forall t_1, t_2 \in T: \ t_1 \neq t_2$ yields that $t_1\sigma \neq t_2\sigma$.

From now on, we assume that substitutions are OI-compliant.

3.2 The generality relation

In ILP, the key mechanism is *generalization* intended as a search process through a partially ordered space of hypotheses (Mitchell 1982). The definition of a generality relation for constrained DATALOG clauses can disregard neither the peculiarities of $\mathscr{AL}$-log nor the methodological apparatus of ILP. Therefore, we rely on the reasoning mechanisms made available by $\mathscr{AL}$-log knowledge bases and propose to adapt Buntine's generalized subsumption (Buntine 1988) to our framework as follows.

Definition 5
Let H be a constrained DATALOG clause, α a ground DATALOG atom, and $\mathscr{J}$ an interpretation. We say that H *covers* α under $\mathscr{J}$, if there is a ground substitution θ for H ($H\theta$ is ground), such that $body(H)\theta$ is true under $\mathscr{J}$ and $head(H)\theta = \alpha$.

Definition 6
Let H_1, H_2 be two constrained DATALOG clauses and $\mathscr{B}$ an $\mathscr{AL}$-log knowledge base. We say that H_1 *$\mathscr{B}$-subsumes* H_2, if for every model $\mathscr{J}$ of $\mathscr{B}$ and every ground atom α, such that H_2 covers α under $\mathscr{J}$, we have that, H_1 covers α under $\mathscr{J}$.

We can define a generality relation $\succeq_{\mathscr{B}}$ for constrained DATALOG clauses on the basis of $\mathscr{B}$-subsumption. It can be easily proven that $\succeq_{\mathscr{B}}$ is a quasi-order (i.e., it is a reflexive and transitive relation) for constrained DATALOG clauses.

Definition 7
Let H_1, H_2 be two constrained DATALOG clauses and $\mathscr{B}$ an $\mathscr{AL}$-log knowledge base. We say that H_1 is *at least, as general as* H_2 under $\mathscr{B}$-subsumption, $H_1 \succeq_{\mathscr{B}} H_2$, iff H_1 $\mathscr{B}$-subsumes H_2. Furthermore, H_1 is *more general than* H_2 under $\mathscr{B}$-subsumption, $H_1 \succ_{\mathscr{B}} H_2$, iff $H_1 \succeq_{\mathscr{B}} H_2$ and $H_2 \not\succeq_{\mathscr{B}} H_1$. Finally, H_1 is *equivalent* to H_2 under $\mathscr{B}$-subsumption, $H_1 \sim_{\mathscr{B}} H_2$, iff $H_1 \succeq_{\mathscr{B}} H_2$ and $H_2 \succeq_{\mathscr{B}} H_1$.

The next lemma shows the definition of $\mathscr{B}$-subsumption to be equivalent to another formulation, which will be more convenient in later proofs than the definition based on covering.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Definition 8

Let $\mathscr{B}$ be an $\mathscr{AL}$-log knowledge base and H be a constrained DATALOG clause. Let $X_1,\ldots,X_n$ be all the variables appearing in H, and $a_1,\ldots,a_n$ be distinct constants (individuals) not appearing in $\mathscr{B}$ or H. Then the substitution $\{X_1/a_1,\ldots,X_n/a_n\}$ is called a *Skolem substitution* for H w.r.t. $\mathscr{B}$.

Lemma 2

Let H_1, H_2 be two constrained DATALOG clauses, $\mathscr{B}$ an $\mathscr{AL}$-log knowledge base, and σ a Skolem substitution for H_2 with respect to $\{H_1\}\cup\mathscr{B}$. We say that $H_1 \succeq_{\mathscr{B}} H_2$, iff there exists a ground substitution θ for H_1, such that (i) $head(H_1)\theta = head(H_2)\sigma$ and (ii) $\mathscr{B}\cup body(H_2)\sigma \models body(H_1)\theta$.

Proof

($\Rightarrow$) Suppose $H_1 \succeq_{\mathscr{B}} H_2$. Let $\mathscr{B}'$ be the knowledge base $\mathscr{B}\cup body(H_2)\sigma$ and $\mathscr{J} = \langle \mathscr{I}_{\mathscr{O}}, \mathscr{I}_H\rangle$ be a model of $\mathscr{B}'$, where $\mathscr{I}_{\mathscr{O}}$ is the minimal $\mathscr{O}$-model of Σ and $\mathscr{I}_H$ be the least Herbrand model of the DATALOG part of $\mathscr{B}'$. The substitution σ is a ground substitution for H_2, and $body(H_2)\sigma$ is true under $\mathscr{J}$, so H_2 covers $head(H_2)\sigma$ under $\mathscr{J}$ by Definition 5. Then, H_1 must also cover $head(H_2)\sigma$ under $\mathscr{J}$. Thus, there is a ground substitution θ for H_1, such that $head(H_1)\theta = head(H_2)\sigma$, and $body(H_1)\theta$ is true under $\mathscr{J}$, i.e., $\mathscr{J}\models body(H_1)\theta$. By properties of the least Herbrand model, it holds that $\mathscr{B}\cup body(H_2)\sigma \models \mathscr{J}$, hence, $\mathscr{B}\cup body(H_2)\sigma \models body(H_1)\theta$.

($\Leftarrow$) Suppose there is a ground substitution θ for H_1, such that $head(H_1)\theta = head(H_2)\sigma$ and $\mathscr{B}\cup body(H_2)\sigma \models body(H_1)\theta$. Let α be some ground atom and $\mathscr{J}_\alpha$ some model of $\mathscr{B}$, such that H_2 covers α under $\mathscr{J}_\alpha$. To prove that $H_1 \succeq_{\mathscr{B}} H_2$, we need to prove that H_1 covers α under $\mathscr{J}_\alpha$.

Construct a substitution θ' from θ as follows: for every binding $X/c \in \sigma$, replace c in bindings in θ by X. Then, we have $H_1\theta'\sigma = H_1\theta$ and none of the Skolem constants of σ occurs in θ'. Then, $head(H_1)\theta'\sigma = head(H_1)\theta = head(H_2)\sigma$, so $head(H_1)\theta' = head(H_2)$. Since H_2 covers α under $\mathscr{J}_\alpha$, there is a ground substitution γ for H_2, such that $body(H_2)\gamma$ is true under $\mathscr{J}_\alpha$, and $head(H_2)\gamma = \alpha$. This implies that $head(H_1)\theta'\gamma = head(H_2)\gamma = \alpha$.

It remains to show that $body(H_1)\theta'\gamma$ is true under $\mathscr{J}_\alpha$. Since $\mathscr{B}\cup body(H_2)\sigma \models body(H_1)\theta'\sigma$ and $\leftarrow body(H_1)\theta'\sigma$ is a ground query, it follows from Lemma 1, that there exists a constrained SLD-refutation for $\leftarrow body(H_1)\theta'\sigma$ in $\mathscr{B}\cup body(H_2)\sigma$. By Definition 2, there exists a finite set $\{d_1,\ldots,d_m\}$ of constrained SLD-derivations, having $\leftarrow body(H_1)\theta'\sigma$ as top clause and elements of $\mathscr{B}\cup body(H_2)\sigma$ as input clauses, such that for each derivation d_i, $i\in\{1,\ldots,m\}$, the last query $Q^{(n_i)}$ of d_i is a constrained empty clause and for every model $\mathscr{J}$ of $\mathscr{B}\cup body(H_2)\sigma$, there exists at least, one derivation d_i, $i\in\{1,\ldots,m\}$, such that $\mathscr{J}\models Q^{(n_i)}$. We want to turn this constrained SLD-refutation for $\leftarrow body(H_1)\theta'\sigma$ in $\mathscr{B}\cup body(H_2)\sigma$ into a constrained SLD-refutation for $\leftarrow body(H_1)\theta'\gamma$ in $\mathscr{B}\cup body(H_2)\gamma$, thus proving that $\mathscr{B}\cup body(H_2)\gamma \models body(H_1)\theta'\gamma$. Let $X_1,\ldots,X_n$ be the variables in $body(H_2)$, such that $\{X_1/c_1,\ldots,X_n/c_n\}\subseteq\sigma$, and $\{X_1/t_1,\ldots,X_n/t_n\}\subseteq\gamma$. If we replace each Skolem constant c_j by t_j, $1\leqslant j\leqslant n$ in both the SLD-derivations and the models of $\mathscr{B}\cup body(H_2)\sigma$, we obtain a constrained SLD-refutation of $body(H_1)\theta'\gamma$ in

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

$\mathscr{B} \cup body(H_2)\gamma$. Hence, $\mathscr{B} \cup body(H_2)\gamma \models body(H_1)\theta'\gamma$. Since $\mathscr{J}_\alpha$ is a model of $\mathscr{B} \cup body(H_2)\gamma$, it is also a model of $body(H_1)\theta'\gamma$. □

The relation between $\mathscr{B}$-subsumption and constrained SLD-resolution is given below. It provides an operational means for checking $\mathscr{B}$-subsumption.

Theorem 1
Let H_1, H_2 be two constrained DATALOG clauses, $\mathscr{B}$ an $\mathscr{AL}$-log knowledge base, and σ a Skolem substitution for H_2 with respect to $\{H_1\} \cup \mathscr{B}$. We say that $H_1 \succeq_{\mathscr{B}} H_2$ iff there exists a substitution θ for H_1, such that (i) $head(H_1)\theta = head(H_2)$ and (ii) $\mathscr{B} \cup body(H_2)\sigma \vdash body(H_1)\theta\sigma$, where $body(H_1)\theta\sigma$ is ground.

Proof
By Lemma 2, we have $H_1 \succeq_{\mathscr{B}} H_2$, iff there exists a ground substitution θ' for H_1, such that $head(H_1)\theta' = head(H_2)\sigma$ and $\mathscr{B} \cup body(H_2)\sigma \models body(H_1)\theta'$. Since σ is a Skolem substitution, we can define a substitution θ, such that $H_1\theta\sigma = H_1\theta'$ and none of the Skolem constants of σ occurs in θ. Then, $head(H_1)\theta = head(H_2)$ and $\mathscr{B} \cup body(H_2)\sigma \models body(H_1)\theta\sigma$. Since $body(H_1)\theta\sigma$ is ground, by Lemma 1, we have $\mathscr{B} \cup body(H_2)\sigma \vdash body(H_1)\theta\sigma$, so the thesis follows. □

The decidability of $\mathscr{B}$-subsumption follows from the decidability of both generalized subsumption in DATALOG (Buntine 1988) and query answering in $\mathscr{AL}$-log (Donini *et al.* 1998).

3.3 Coverage relations

When defining coverage relations, we make assumptions as regards the representation of observations, because it impacts the definition of coverage. In ILP, there are two choices: we can represent an observation as either a ground definite clause or a set of ground unit clauses. The former is peculiar to the normal ILP setting (also called *learning from implications*) (Frazier and Page 1993), whereas the latter is usual in the logical setting of *learning from interpretations* (De Raedt and Džeroski 1994). The representation choice for observations and the scope of induction are orthogonal dimensions, as clearly explained in (De Raedt 1997). Therefore, we prefer the term "observation" to the term "example" for the sake of generality.

In the logical setting of *learning from entailment* extended to $\mathscr{AL}$-log, an observation $o_i \in O$ is represented as a ground-constrained DATALOG clause having a ground atom $q(\vec{a}_i)$[5] in the head.

Definition 9
Let $H \in \mathscr{L}$ be a hypothesis, $\mathscr{K}$ a background knowledge and $o_i \in O$ an observation. We say that H *covers* o_i *under entailment* w.r.t $\mathscr{K}$, iff $\mathscr{K} \cup H \models o_i$.

In order to provide an operational means for testing this coverage relation we resort to the deduction theorem for first-order logic formulas (Nienhuys-Cheng and de Wolf 1997).

[5] $\vec{a}_i$ is a tuple of constants.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Theorem 2
Let Σ be a set of formulas, and ϕ and ψ be formulas. We say that $\Sigma \cup \{\phi\} \models \psi$, iff $\Sigma \models (\phi \rightarrow \psi)$.

Theorem 3
Let $H \in \mathscr{L}$ be a hypothesis, $\mathscr{K}$ a background knowledge, and $o_i \in O$ an observation. We say that H *covers* o_i *under entailment w.r.t.* $\mathscr{K}$ iff $\mathscr{K} \cup body(o_i) \cup H \vdash q(\vec{a}_i)$.

Proof
The following chain of equivalences holds:

- H covers o_i under entailment w.r.t. $\mathscr{K} \leftrightarrow$ (by Definition 9).
- $\mathscr{K} \cup H \models q(\vec{a}_i) \leftarrow body(o_i) \leftrightarrow$ (by Theorem 2).
- $\mathscr{K} \cup H \cup body(o_i) \models q(\vec{a}_i) \leftrightarrow$ (by Lemma 1).
- $\mathscr{K} \cup body(o_i) \cup H \vdash q(\vec{a}_i)$. □

In the logical setting of *learning from interpretations* extended to $\mathscr{AL}$-log, an observation $o_i \in O$ is represented as a couple $(q(\vec{a}_i), \mathscr{A}_i)$, where $\mathscr{A}_i$ is a set containing ground DATALOG facts concerning the individual i.

Definition 10
Let $H \in \mathscr{L}$ be a hypothesis, $\mathscr{K}$ a background knowledge and $o_i \in O$ an observation. We say that H *covers* o_i *under interpretations w.r.t.* $\mathscr{K}$, iff $\mathscr{K} \cup \mathscr{A}_i \cup H \models q(\vec{a}_i)$.

Theorem 4
Let $H \in \mathscr{L}$ be a hypothesis, $\mathscr{K}$ a background knowledge, and $o_i \in O$ an observation. We say that H *covers* o_i *under interpretations w.r.t.* $\mathscr{K}$, iff $\mathscr{K} \cup \mathscr{A}_i \cup H \vdash q(\vec{a}_i)$.

Proof
Since $q(\vec{a}_i)$ is a ground query to the $\mathscr{AL}$-log knowledge base $\mathscr{B} = \mathscr{K} \cup \mathscr{A}_i \cup H$, the thesis follows from Definition 10 and Lemma 1. □

Note that both coverage tests can be reduced to query answering.

4 Instantiating the framework for Ontology Refinement

Ontology Refinement is a phase in the Ontology Learning process that aims at the adaptation of an existing ontology to a specific domain or the needs of a particular user (Maedche and Staab 2004). In this section, we consider the problem of Concept Refinement which is about refining a known concept, called *reference concept*, belonging to an existing taxonomic ontology in the light of new knowledge coming from a relational data source. A taxonomic ontology is an ontology organized around the is-a relationship between concepts (Gómez-Pérez *et al.* 2004). We assume that a *concept* $\mathscr{C}$ consists of two parts: an *intension* $int(\mathscr{C})$ and an *extension* $ext(\mathscr{C})$. The former is an expression belonging to a logical language $\mathscr{L}$, whereas the latter is a set of objects that satisfy the former. More formally, given,

- a taxonomic ontology Σ,
- a reference concept $C_{ref} \in \Sigma$,

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

- a relational data source Π,
- a logical language $\mathscr{L}$,

the *Concept Refinement* problem is to find a taxonomy $\mathscr{G}$ of concepts $\mathscr{C}_i$, such that (i) $int(\mathscr{C}_i) \in \mathscr{L}$ and (ii) $ext(\mathscr{C}_i) \subset ext(C_{ref})$. Therefore, $\mathscr{G}$ is structured according to the *subset relation* between concept extensions. Note that C_{ref} is among both the concepts defined in Σ and the symbols of $\mathscr{L}$. Furthermore, $ext(\mathscr{C}_i)$ relies on notion of satisfiability of $int(\mathscr{C}_i)$ w.r.t. $\mathscr{B} = \Sigma \cup \Pi$. We would like to emphasize that $\mathscr{B}$ includes Σ because in Ontology Refinement, as opposite to other forms of Ontology Learning, such as Ontology Extraction (or Building), it is mandatory to consider the existing ontology and its existing connections. Thus, a formalism like $\mathscr{AL}$-log suits very well the hybrid nature of $\mathscr{B}$ (see Section 4.1).

In our ILP approach, the Ontology Refinement problem at hand is reformulated as a Concept Formation problem (Lisi and Esposito 2007). *Concept Formation* indicates a ML task that refers to the acquisition of conceptual hierarchies, in which each concept has a flexible, non-logical definition and in which learning occurs incrementally and without supervision (Langley 1987). More precisely, it is to take a large number of unlabeled training instances: to find clusterings that group those instances in categories: to find an intensional definition for each category that summarized its instances; and to find a hierarchical organization for those categories (Gennari *et al.* 1989). Concept Formation stems from *Conceptual Clustering* (Michalski and Stepp 1983). The two differ substantially in the methods: The latter usually applies bottom-up batch algorithms, whereas the former prefers top-down incremental ones. Yet, the methods are similar in the scope of induction, i.e., *prediction*, as opposite to (Statistical) Clustering (Hartigan 2001) and Frequent Pattern Discovery (Mannila and Toivonen 1997), whose goal is to describe a dataset. According to the commonly accepted formulation of the task (Langley 1987; Gennari *et al.* 1989), Concept Formation can be decomposed in two sub-tasks,

(1) clustering,
(2) characterization.

The former consists of using internalized heuristics to organize the observations into categories, whereas the latter consists in determining a concept (that is, an intensional description) for each extensionally defined subset discovered by clustering. We propose a pattern-based approach for the former (see Section 4.2) and a bias-based approach for the latter (see Section 4.3). In particular, the clustering approach is pattern-based because it relies on the aforementioned commonalities between Clustering and Frequent Pattern Discovery. Descriptive tasks fit the ILP setting of *characteristic induction* (De Raedt and Dehaspe 1997). A distinguishing feature of this form of induction is the density of solution space. The setting of *learning from interpretations* has been shown to be a promising way of dealing with such spaces (Blockeel *et al.* 1999).

Definition 11
Let $\mathscr{L}$ be a hypothesis language, $\mathscr{K}$ a background knowledge, O a set of observations, and $M(\mathscr{B})$ a model constructed from $\mathscr{B} = \mathscr{K} \cup O$. The goal of *characteristic*

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

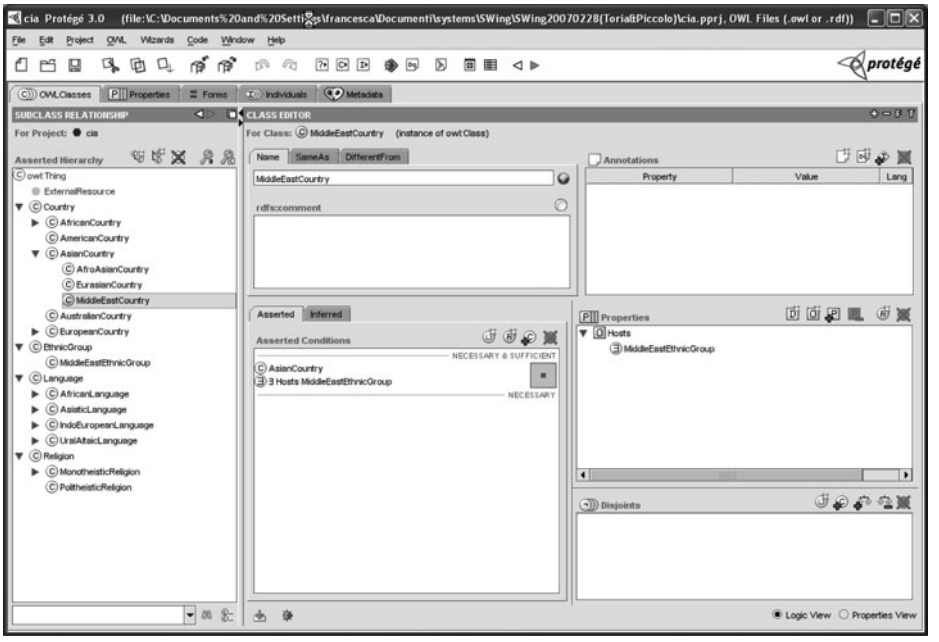


Fig. 2. The ontology $\Sigma_{\texttt{CIA}}$ used as an example throughout Section 4.1.

induction from interpretations is to find a set $\mathscr{H} \subseteq \mathscr{L}$ of hypotheses, such that (i) $\mathscr{H}$ is true in $M(\mathscr{B})$, and (ii) for each $H \in \mathscr{L}$, if H is true in $M(\mathscr{B})$, then $\mathscr{H} \models H$.

In the following subsection, we will clarify the nature of $\mathscr{K}$ and O.

4.1 Representation choice

The KR&R framework for conceptual knowledge in the Concept Refinement problem at hand is the one offered by $\mathscr{AL}$-log.

The **taxonomic ontology** Σ is a $\mathscr{ALC}$ knowledge base. From now on, we will call *input concepts* all the concepts occurring in Σ.

Example 1

Throughout this section, we will refer to the $\mathscr{ALC}$ ontology $\Sigma_{\texttt{CIA}}$ (see Figure 2) concerning countries, ethnic groups, languages, and religions of the world, and built according to Wikipedia[6] taxonomies. For instance, the expression

$$\texttt{MiddleEastCountry} \equiv \texttt{AsianCountry} \sqcap \exists \texttt{Hosts.MiddleEasternEthnicGroup}.$$

is an equivance axiom that defines the concept `MiddleEastCountry` as an Asian country, which hosts at least, one Middle Eastern ethnic group. In particular, Armenia (`'ARM'`) and Iran (`'IR'`) are two of the 15 countries that are classified as Middle Eastern.

[6] `http://www.wikipedia.org/`.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

The **relational data source** Π is a DATALOG program. The extensional part of Π is partitioned into portions $\mathscr{A}_i$, each of which refers to an individual a_i of C_{ref}. The link between $\mathscr{A}_i$ and a_i is represented with the DATALOG literal $q(a_i)$. The pair $(q(a_i), \mathscr{A}_i)$ is called *observation*. The intensional part (IDB) of Π together with the whole Σ is considered as *background knowledge* for the problem at hand.

Example 2
An $\mathscr{AL}$-log knowledge base $\mathscr{B}_{\texttt{CIA}}$ has been obtained by integrating $\Sigma_{\texttt{CIA}}$ and a DATALOG program $\Pi_{\texttt{CIA}}$ based on the online 1996 CIA World Fact Book[7]. The extensional part of $\Pi_{\texttt{CIA}}$ consists of DATALOG facts[8] grouped according to the individuals of `MiddleEastCountry`. In particular, the observation $(q(\texttt{'IR'}), \mathscr{A}_{\texttt{IR}})$ contains DATALOG facts, such as

```
language('IR','Persian',58).
religion('IR','ShiaMuslim',89).
religion('IR','SunniMuslim',10).
```

concerning the individual `'IR'`. The intensional part of $\Pi_{\texttt{CIA}}$ defines two views on `language` and `religion`:

```
speaks(CountryID, LanguageN)←language(CountryID,LanguageN,Perc)
                            & CountryID:Country, LanguageN:Language.
believes(CountryID, ReligionN)←religion(CountryID,ReligionN,Perc)
                            & CountryID:Country, ReligionN:Religion.
```

that can deduce new DATALOG facts when triggered on $\mathscr{B}_{\texttt{CIA}}$. It forms the background knowledge $\mathscr{K}_{\texttt{CIA}}$ together with the whole $\Sigma_{\texttt{CIA}}$.

The **language** $\mathscr{L}$ contains expressions, called $\mathscr{O}$-queries, relating individuals of C_{ref} to individuals of other input concepts (*task-relevant concepts*). An *$\mathscr{O}$-query* is a constrained DATALOG clause of the form

$$Q = q(X) \leftarrow \alpha_1, \ldots, \alpha_m \& X : C_{ref}, \gamma_2, \ldots, \gamma_n,$$

which is compliant with the properties mentioned in Section 3.1. The $\mathscr{O}$-query

$$Q_t = q(X) \leftarrow \& X : C_{ref}$$

is called *trivial* for $\mathscr{L}$, because it contains only the constraint for the *distinguished variable* X. Furthermore, the language $\mathscr{L}$ is *multi-grained*, i.e., it contains expressions at multiple levels of description granularity. Indeed, it is implicitly defined by a *declarative bias specification*, which consists of a finite alphabet $\mathscr{A}$ of DATALOG predicate names appearing in Π and finite alphabets Γ^l (one for each level l of description granularity) of $\mathscr{ALC}$ concept names occurring in Σ. Note that α_i's are taken from $\mathscr{A}$ and γ_j's are taken from Γ^l. We impose $\mathscr{L}$ to be finite by specifying some bounds, mainly *maxD* for the maximum depth of search and *maxG* for the maximum level of granularity.

[7] `http://www.odci.gov/cia/publications/factbook/`.
[8] `http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-rel-facts.flp`.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Example 3

We want to refine the concept `MiddleEastCountry` belonging to $\Sigma_{\texttt{CIA}}$ in the light of the new knowledge coming from $\Pi_{\texttt{CIA}}$. More precisely, we want to describe Middle East countries (individuals of the reference concept) with respect to the religions believed and the languages spoken (individuals of the task-relevant concepts) at three levels of granularity ($maxG = 3$). To this aim, we define $\mathscr{L}_{\texttt{CIA}}$ as the set of $\mathscr{O}$-queries with $C_{ref} = \texttt{MiddleEastCountry}$ that can be generated from the alphabet

$\mathscr{A} = \{\texttt{believes/2}, \texttt{speaks/2}\}$

of DATALOG binary predicate names, and the alphabets

$\Gamma^1 = \{\texttt{Language}, \texttt{Religion}\}$
$\Gamma^2 = \{\texttt{IndoEuropeanLanguage}, \ldots, \texttt{MonotheisticReligion}, \ldots\}$
$\Gamma^3 = \{\texttt{IndoIranianLanguage}, \ldots, \texttt{MuslimReligion}, \ldots\}$

of $\mathscr{ALC}$ concept names for $1 \leqslant l \leqslant 3$, up to $maxD = 5$. Note that the names in $\mathscr{A}$ are taken from $\Pi_{\texttt{CIA}}$, whereas the names in Γ^l's are taken from $\Sigma_{\texttt{CIA}}$. Examples of $\mathscr{O}$-queries in $\mathscr{L}_{\texttt{CIA}}$ are:

$Q_t = \texttt{q(X)} \leftarrow \texttt{\&\ X:MiddleEastCountry}$
$Q_1 = \texttt{q(X)} \leftarrow \texttt{speaks(X,Y)\ \&\ X:MiddleEastCountry, Y:Language}$
$Q_2 = \texttt{q(X)} \leftarrow \texttt{speaks(X,Y)\ \&\ X:MiddleEastCountry, Y:IndoEuropeanLanguage}$
$Q_3 = \texttt{q(X)} \leftarrow \texttt{believes(X,Y)\&\ X:MiddleEastCountry, Y:MuslimReligion,}$

where Q_t is the trivial $\mathscr{O}$-query for $\mathscr{L}_{\texttt{CIA}}$, $Q_1 \in \mathscr{L}^1_{\texttt{CIA}}$, $Q_2 \in \mathscr{L}^2_{\texttt{CIA}}$, and $Q_3 \in \mathscr{L}^3_{\texttt{CIA}}$.

Output concepts are the concepts automatically formed out of the input ones by taking into account the relational data source. Thus, an output concept $\mathscr{C}$ has an $\mathscr{O}$-query $Q \in \mathscr{L}$ as intension and the set $answerset(Q, \mathscr{B})$ of correct answers to Q w.r.t. $\mathscr{B}$ as extension. Note that this set contains the substitutions θ_i's for the distinguished variable of Q, such that there exists a correct answer to $body(Q)\theta_i$ w.r.t. $\mathscr{B}$. In other words, the extension is the set of individuals of C_{ref} satisfying the intension. Also, with reference to Section 3.3, note that proving that an $\mathscr{O}$-query Q covers an observation $(q(a_i), \mathscr{A}_i)$ w.r.t. $\mathscr{K}$ equals to proving that $\theta_i = \{X/a_i\}$ is a correct answer to Q w.r.t. $\mathscr{B}_i = \mathscr{K} \cup \mathscr{A}_i$.

Example 4

The output concept having Q_1 as intension has extension $answerset(Q_1, \mathscr{B}_{\texttt{CIA}}) = \{\texttt{'ARM'}, \texttt{'IR'}, \texttt{'SA'}, \texttt{'UAE'}\}$. In particular, Q_1 covers the observation $(\texttt{q('IR')}, \mathscr{A}_{\texttt{IR}})$ w.r.t. $\mathscr{K}_{\texttt{CIA}}$. This coverage test is equivalent to answering the query $\leftarrow \texttt{q('IR')}$ w.r.t. $\mathscr{K}_{\texttt{CIA}} \cup \mathscr{A}_{\texttt{IR}} \cup Q_1$.

Output concepts are organized into a taxonomy $\mathscr{G}$ rooted in C_{ref} and structured as a Directed Acyclic Graph (DAG) according to the *subset relation* between concept extensions. Note that one such ordering is in line with the set-theoretic semantics of the subsumption relation in ontology languages (see, e.g., the semantics of $\sqsubseteq$ in $\mathscr{ALC}$).

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

4.2 Pattern-based clustering

Frequent Pattern Discovery is about the discovery of regularities in a dataset (Mannila and Toivonen 1997). A frequent pattern is an intensional description, expressed in a language $\mathscr{L}$, of a subset of a given dataset **r**, whose cardinality exceeds a user-defined threshold (*minimum support*). Note that patterns can refer to multiple levels of description granularity (*multi-grained patterns*) (Han and Fu 1999). Here, **r** typically encompasses a taxonomy $\mathscr{T}$. More precisely, the problem of *frequent pattern discovery at l levels of description granularity*, $1 \leqslant l \leqslant maxG$, is to find the set $\mathscr{F}$ of all the frequent patterns expressible in a multi-grained language $\mathscr{L} = \{\mathscr{L}^l\}_{1 \leqslant l \leqslant maxG}$ and evaluated against **r** w.r.t. a set $\{minsup^l\}_{1 \leqslant l \leqslant maxG}$ of minimum support thresholds by means of the evaluation function *supp*. In this case, $P \in \mathscr{L}^l$ with support s is frequent in **r**, if (i) $s \geqslant minsup^l$ and (ii) all ancestors of P w.r.t. $\mathscr{T}$ are frequent in **r**. The blueprint of most algorithms for frequent pattern discovery is the *levelwise search* method (Mannila and Toivonen 1997), which searches the space $(\mathscr{L}, \succeq)$ of patterns organized according to a generality order $\succeq$ in a breadth-first manner, starting from the most general pattern in $\mathscr{L}$ and alternating candidate generation and candidate evaluation phases. The underlying assumption is that $\succeq$ is a quasi-order monotonic w.r.t. *supp*. Note that the method proposed in (Mannila and Toivonen 1997) is also at the basis of algorithms for the variant of the task defined in (Han and Fu 1999).

A frequent pattern highlights a regularity in **r**, therefore, it can be considered as the clue of a data cluster. Note that clusters are concepts partially specified (called *emerging concepts*): only the extension is known. We propose to detect emerging concepts by applying the method of (Lisi and Malerba 2004) for frequent pattern discovery at l, $1 \leqslant l \leqslant maxG$, levels of description granularity and k, $1 \leqslant k \leqslant maxD$, levels of search depth. It adapts (Mannila and Toivonen 1997; Han and Fu 1999) to the KR&R framework of $\mathscr{AL}$-log as follows. For $\mathscr{L}$ being a multi-grained language of $\mathscr{O}$-queries, we need to define first *supp*, then $\succeq$. The *support* of an $\mathscr{O}$-query $Q \in \mathscr{L}$ w.r.t. an $\mathscr{AL}$-log knowledge base $\mathscr{B}$ is defined as

$$supp(Q, \mathscr{B}) = |answerset(Q, \mathscr{B})| / |answerset(Q_t, \mathscr{B})|$$

and supplies the percentage of individuals of C_{ref} that satisfy Q.

Example 5
Since $|\ answerset(Q_t, \mathscr{B}_{\texttt{CIA}})\ | = |\ \texttt{MiddleEastCountry}\ | = 15$, then $supp(Q_1, \mathscr{B}_{\texttt{CIA}}) = 4/15 = 26 \cdot 6\%$.

Being a special case of constrained DATALOG clauses, $\mathscr{O}$-queries can be ordered according to the $\mathscr{B}$-subsumption relation introduced in Section 3.2. It has been proved that $\succeq_{\mathscr{B}}$ is a quasi-order that fulfills the condition of monotonicity w.r.t. *supp* (Lisi and Malerba 2004). Also, note that the underlying reasoning mechanism of $\mathscr{AL}$-log makes $\mathscr{B}$-subsumption more powerful than generalized subsumption as illustrated in the following example.

Example 6
It can be checked that $Q_1 \succeq_{\mathscr{B}} Q_2$ by choosing $\sigma = \{\texttt{X/a}, \texttt{Y/b}\}$ as a Skolem substitution for Q_2 w.r.t. $\mathscr{B}_{\texttt{CIA}} \cup \{Q_1\}$ and $\theta = \emptyset$ as a substitution for Q_1. Similarly it can be

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

proved that $Q_2 \not\succeq_{\mathscr{B}} Q_1$. Furthermore, it can be easily verified that Q_3 $\mathscr{B}$-subsumes the following $\mathscr{O}$-query in $\mathscr{L}^3_{\texttt{CIA}}$

Q_4= `q(A)` ← `believes(A,B)`, `believes(A,C)`&
`A:MiddleEastCountry`, `B:MuslimReligion`

by choosing σ=$\{$`A/a`, `B/b`, `C/c`$\}$ as a Skolem substitution for Q_4 w.r.t. $\mathscr{B}_{\texttt{CIA}} \cup \{Q_3\}$ and θ=$\{$`X/A`, `Y/B`$\}$ as a substitution for Q_3. Note that $Q_4 \not\succeq_{\mathscr{B}} Q_3$ under the OI bias. Indeed, this bias does not admit the substitution $\{$`A/X`, `B/Y`, `C/Y`$\}$ for Q_4, which would make it possible to verify conditions (i) and (ii) of the $\succeq_{\mathscr{B}}$ test.

We would like to emphasize that Σ, besides contributing to the definition of $\mathscr{L}$ (see Section 4.1), plays a key role in the $\succeq_{\mathscr{B}}$ test.

4.3 Bias-based characterization

Since several frequent patterns can have the same set of supporting individuals, turning clusters into concepts is crucial in our approach. Biases can be of help. A *bias* concerns anything which constrains the search for theories (Utgoff and Mitchell 1982). In ILP, *language bias* has to do with constraints on the clauses in the search space, whereas *search bias* has to do with the way a system searches its space of permitted clauses (Nédellec *et al.* 1996). The choice criterion for concept intensions has been obtained by combining two orthogonal biases: a language bias and a search bias (Lisi and Esposito 2006). The former allows the user to define conditions on the form of $\mathscr{O}$-queries to be accepted as concept intensions. For example, it is possible to state which is the minimum level of description granularity (parameter *minG*) and whether (all) the variables must be ontologically constrained or not. The latter allows the user to define a preference criterion based on $\mathscr{B}$-subsumption. More precisely, it is possible to state whether the *most general description (m.g.d.)* or the *most specific description (m.s.d.)* w.r.t. $\succeq_{\mathscr{B}}$ has to be preferrred. Since $\succeq_{\mathscr{B}}$ is not a total order, it can happen that two patterns P and Q, belonging to the same language $\mathscr{L}$, can not be compared w.r.t. $\succeq_{\mathscr{B}}$. In this case, the m.g.d. (resp. m.s.d) of P and Q is the union (resp. conjunction) of P and Q.

Example 7
The patterns

`q(A)` ← `speaks(A,B)`, `believes(A,C)` & `A:MiddleEastCountry`, `B:ArabicLanguage`

and

`q(A)` ← `believes(A,B)`, `speaks(A,C)` & `A:MiddleEastCountry`, `B:MuslimReligion`

have the same answer set $\{$`'ARM'`, `'IR'`$\}$ but are incomparable w.r.t. $\succeq_{\mathscr{B}}$. Their m.g.d. is the union of the two

`q(A)` ← `speaks(A,B)`, `believes(A,C)` & `A:MiddleEastCountry`, `B:ArabicLanguage`
`q(A)` ← `believes(A,B)`, `speaks(A,C)` & `A:MiddleEastCountry`, `B:MuslimReligion`

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Their m.s.d. is the conjunction of the two

q(A) ← believes(A,B), speaks(A,C), speaks(A,D), believes(A,E) &
A:MiddleEastCountry, B:MuslimReligion, C:ArabicLanguage

The extension of the subsequent concept will be {'ARM', 'IR'}.

The two biases are combined as follows. For each frequent pattern $P \in \mathscr{L}$ that fulfills the language bias specification, the procedure for building the taxonomy $\mathscr{G}$ from the set $\mathscr{F} = \{\mathscr{F}_k^l \mid 1 \leqslant l \leqslant maxG, 1 \leqslant k \leqslant maxD\}$ checks whether a concept $\mathscr{C}$ with $ext(\mathscr{C}) = answerset(P)$ already exists in $\mathscr{G}$. If one such concept is not retrieved, a new node $\mathscr{C}$ with $int(\mathscr{C}) = P$ and $ext(\mathscr{C}) = answerset(P)$ is added to $\mathscr{G}$. Note that the insertion of a node can imply the reorganization of $\mathscr{G}$ to keep it compliant with the subset relation on extents. If the node already occurs in $\mathscr{G}$, its intension is updated according to the search bias specification.

4.4 Experimental results

In order to test the approach we have extended the ILP system $\mathscr{AL}$-QuIn (Lisi 2006)[9] with a module for postprocessing frequent patterns into concepts. The goal of the experiments is to provide an empirical evidence of the orthogonality of the two biases and of the potential of their combination as choice criterion. The results reported in the following are obtained for the problem introduced in Example 3 by setting the parameters for the frequent pattern discovery phase as follows: $maxD = 5$, $maxG = 3$, $minsup^1 = 20\%$, $minsup^2 = 13\%$, and $minsup^3 = 10\%$. Thus, each experiment starts from the same set $\mathscr{F}$ of 53 frequent patterns out of 99 candidate patterns. Also, all the experiments require the descriptions to have all the variables ontologically constrained, but vary as to the user preferences for the minimum level of description granularity ($minG$) and the search bias (m.g.d./m.s.d.).

The first two experiments both require the descriptions to have all the variables ontologically constrained by concepts from the second granularity level on ($minG = 2$). When the m.g.d. criterion is adopted, the procedure of taxonomy building returns the following 12 concepts:

C-1111 $\in \mathscr{F}_1^1$
q(A) ← A:MiddleEastCountry
{ARM, BRN, IR, IRQ, IL, JOR, KWT, RL, OM, Q, SA, SYR, TR, UAE, YE}

C-5233 $\in \mathscr{F}_3^2$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:MonotheisticReligion
{ARM, BRN, IR, IRQ, IL, JOR, KWT, RL, OM, Q, SA, SYR, TR, UAE}

C-2233 $\in \mathscr{F}_3^2$
q(A) ← speaks(A,B) & A:MiddleEastCountry, B:AfroAsiaticLanguage
{IR, SA, YE}

C-3233 $\in \mathscr{F}_3^2$
q(A) ← speaks(A,B) & A:MiddleEastCountry, B:IndoEuropeanLanguage
{ARM, IR}

[9] $\mathscr{AL}$-QuIn is implemented with Prolog, but equipped with a module for preprocessing OWL ontologies in order to enable Semantic Web Mining applications.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

C-8256 $\in \mathscr{F}_5^2$
q(A) ← speaks(A,B), believes(A,C) &
A:MiddleEastCountry, B:AfroAsiaticLanguage, C:MonotheisticReligion
{IR, SA}

C-6256 $\in \mathscr{F}_5^2$
q(A) ← believes(A,B), believes(A,C) &
A:MiddleEastCountry, B:MonotheisticReligion, C:MonotheisticReligion
{BRN, IR, IRQ, IL, JOR, RL, SYR}

C-2333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,'Druze') & A:MiddleEastCountry
{IL, SYR}

C-3333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:JewishReligion
{IR, IL, SYR}

C-4333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:ChristianReligion
{ARM, IR, IRQ, IL, JOR, RL, SYR}

C-5333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:MuslimReligion
{BRN, IR, IRQ, IL, JOR, KWT, RL, OM, Q, SA, SYR, TR, UAE}

C-14356 $\in \mathscr{F}_5^3$
q(A) ← believes(A,B), believes(A,C) &
A:MiddleEastCountry, B:ChristianReligion, C:MuslimReligion
{IR, IRQ, IL, JOR, RL, SYR}

C-5356 $\in \mathscr{F}_5^3$
q(A) ← believes(A,B), believes(A,C) &
A:MiddleEastCountry, B:MuslimReligion, C:MuslimReligion
{BRN, IR, SYR}

organized in the DAG $\mathscr{G}_{\mathtt{CIA}}$ (see Figure 3). They are numbered according to the chronological order of insertion in $\mathscr{G}_{\mathtt{CIA}}$ and annotated with information of the generation step. From a qualitative point of view, concepts C-2233[10] and C-5333 well characterize Middle East countries. Armenia (ARM), as opposite to Iran (IR), does not fall in these concepts. It rather belongs to the weaker characterizations C-3233 and C-4333. This suggests that our procedure performs a "sensible" clustering. Indeed, Armenia is a well-known borderline case for the geo-political concept of Middle East, though the Armenia is usually listed among Middle Eastern ethnic groups. Modern experts tend nowadays to consider it as part of Europe, therefore, out of Middle East. But in 1996, the online CIA World Fact Book still considered Armenia as part of Asia.

When the m.s.d. criterion is adopted (see Figure 4), the intensions for the concepts C-2233, C-3233, C-8256, C-2333, and C-3333 change as follows:

[10] C-2233 is less populated than expected because $\mathscr{B}_{\mathtt{CIA}}$ does not provide facts on the languages spoken for all countries.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press



Fig. 3. Taxonomy $\mathscr{G}_{\texttt{CIA}}$ obtained with the m.g.d. criterion for $minG = 2$.

C-2233 $\in \mathscr{F}_3^2$
q(A) ← speaks(A,B) & A:MiddleEastCountry, B:ArabicLanguage
{IR, SA, YE}

C-3233 $\in \mathscr{F}_3^2$
q(A) ← speaks(A,B) & A:MiddleEastCountry, B:IndoIranianLanguage
{ARM, IR}

C-8256 $\in \mathscr{F}_5^2$
q(A) ← speaks(A,B), believes(A,C) &
A:MiddleEastCountry, B:ArabicLanguage, C:MuslimReligion
{IR, SA}

C-2333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,'Druze'), believes(A,B), believes(A,C), believes(A,D) &
A:MiddleEastCountry, B:JewishReligion,
C:ChristianReligion, D:MuslimReligion
{IL, SYR}

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

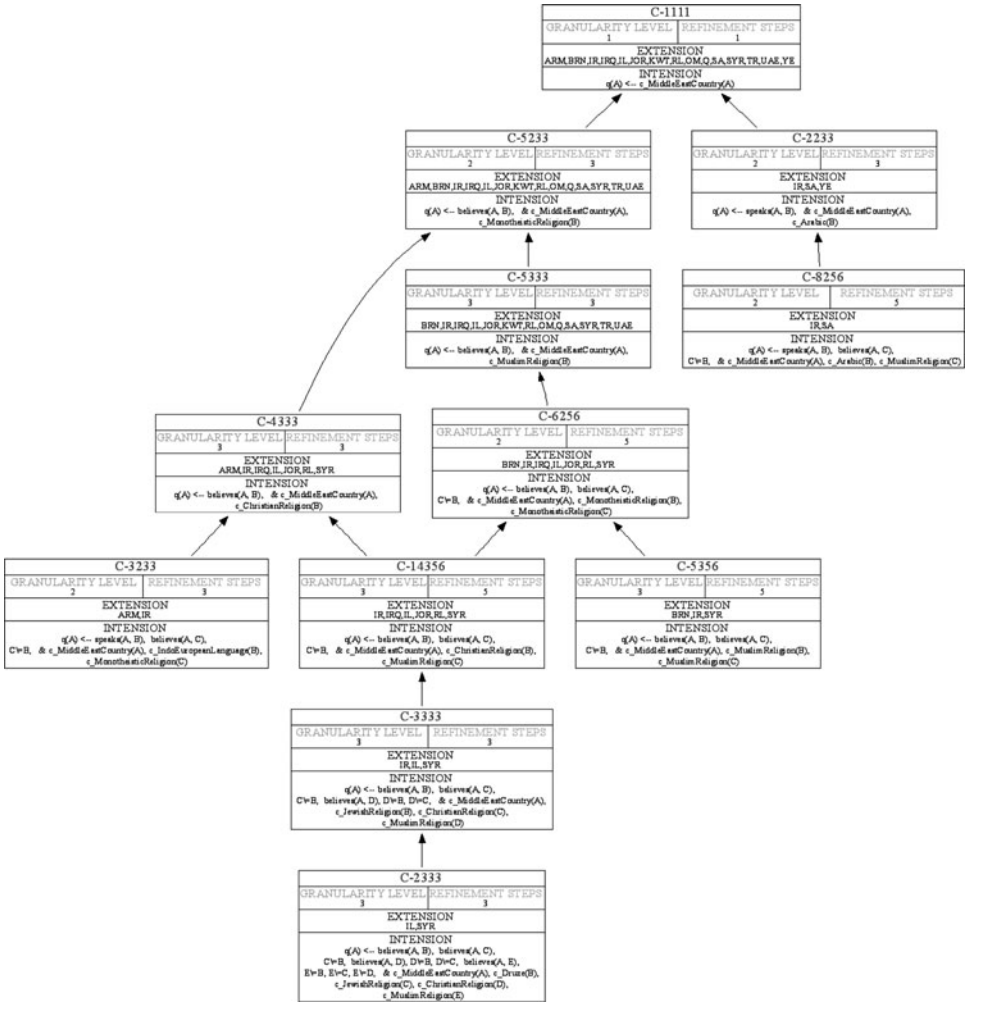


Fig. 4. Taxonomy $\mathscr{G}'_{\texttt{CIA}}$ obtained with the m.s.d. criterion for $minG = 2$.

$\texttt{C-3333} \in \mathscr{F}_3^3$

```
q(A) ← believes(A,B), believes(A,C), believes(A,D) &
        A:MiddleEastCountry, B:JewishReligion,
        C:ChristianReligion, D:MuslimReligion
```

{IR, IL, SYR}

In particular, C-2333 and C-3333 look quite overfitted to data. Yet, overfitting allows us to realize that what distinguishes Israel (IL) and Syria (SYR) from Iran is just the presence of Druze people. Note that the clusters do not change because the search bias only affects the characterization step.

The other two experiments further restrict the conditions of the language bias specification. Here, only descriptions with variables constrained by concepts of granularity from the third level on ($minG = 3$) are considered. When the m.g.d.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

option is selected, the procedure for taxonomy building returns the following nine concepts:

C-1111 $\in \mathscr{F}_1^1$
q(A) ← A:MiddleEastCountry
{ARM, BRN, IR, IRQ, IL, JOR, KWT, RL, OM, Q, SA, SYR, TR, UAE, YE}

C-9333 $\in \mathscr{F}_3^3$
q(A) ← speaks(A,B) & A:MiddleEastCountry, B:ArabicLanguage
{IR, SA, YE}

C-2333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,'Druze') & A:MiddleEastCountry
{IL, SYR}

C-3333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:JewishReligion
{IR, IL, SYR}

C-4333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:ChristianReligion
{ARM, IR, IRQ, IL, JOR, RL, SYR}

C-5333 $\in \mathscr{F}_3^3$
q(A) ← believes(A,B) & A:MiddleEastCountry, B:MuslimReligion
{BRN, IR, IRQ, IL, JOR, KWT, RL, OM, Q, SA, SYR, TR, UAE}

C-33356 $\in \mathscr{F}_5^3$
q(A) ← speaks(A,B), believes(A,C) &
A:MiddleEastCountry, B:ArabicLanguage, C:MuslimReligion
{IR, SA}

C-14356 $\in \mathscr{F}_5^3$
q(A) ← believes(A,B), believes(A,C) &
A:MiddleEastCountry, B:ChristianReligion, C:MuslimReligion
{IR, IRQ, IL, JOR, RL, SYR}

C-5356 $\in \mathscr{F}_5^3$
q(A) ← believes(A,B), believes(A,C) &
A:MiddleEastCountry, B:MuslimReligion, C:MuslimReligion
{BRN, IR, SYR}

organized in a DAG $\mathscr{G}''_{\mathtt{CIA}}$ (see Figure 5), which partially reproduces $\mathscr{G}'_{\mathtt{CIA}}$. Note that the stricter conditions set in the language bias cause three concepts occurring in $\mathscr{G}'_{\mathtt{CIA}}$ not to appear in $\mathscr{G}''_{\mathtt{CIA}}$: the scarsely significant C-5233 and C-6256, and the quite interesting C-3233. Therefore, the language bias can prune the space of clusters. Note that the other concepts of $\mathscr{G}'_{\mathtt{CIA}}$ emerged at $l = 2$ do remain in $\mathscr{G}''_{\mathtt{CIA}}$ as clusters, but with a different characterization: C-9333 and C-33356 instead of C-2233 and C-8256, respectively.

When the m.s.d. condition is chosen (see Figure 6), the intensions for the concepts C-2333 and C-3333 change analogously to $\mathscr{G}'_{\mathtt{CIA}}$. Note that both $\mathscr{G}''_{\mathtt{CIA}}$ and $\mathscr{G}'''_{\mathtt{CIA}}$ are hierarchical taxonomies. It can be empirically observed that the possibility of producing a hierarchy increases as the conditions of the language bias become stricter.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

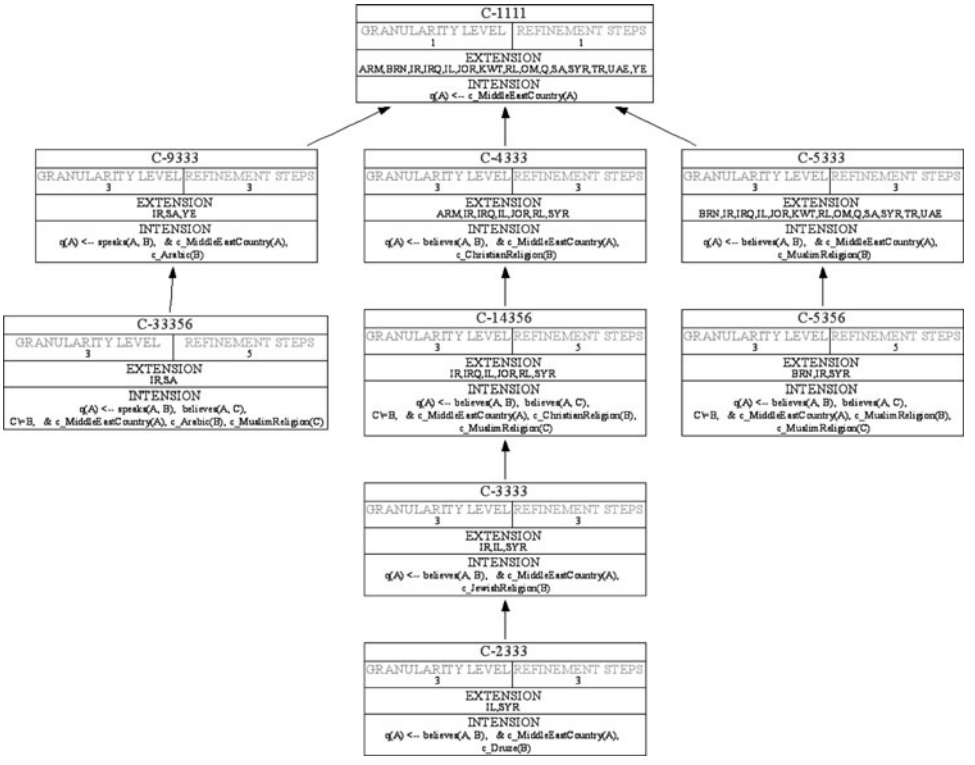


Fig. 5. Taxonomy $\mathscr{G}''_{\texttt{CIA}}$ obtained with the m.g.d. criterion for $minG = 3$.

5 Conclusions

Building rules on top of ontologies for the Semantic Web is a task that can be automated by applying Machine Learning algorithms to data expressed with hybrid formalisms combining DLs and Horn clauses. Learning in DL-based hybrid languages has very recently attracted attention in the ILP community. In (Rouveirol and Ventos 2000) the chosen language is CARIN-$\mathscr{ALN}$, therefore, example coverage and subsumption between two hypotheses are based on the existential entailment algorithm of CARIN (Levy and Rousset 1998). Following (Rouveirol and Ventos 2000), Kietz studies the learnability of CARIN-$\mathscr{ALN}$, thus providing a preprocessing method which enables ILP systems to learn CARIN-$\mathscr{ALN}$ rules (Kietz 2003). Closely related to DL-based hybrid systems are the proposals arising from the study of many-sorted logics, where a first-order language is combined with a sort language which can be regarded as an elementary DL (Frisch 1991). In this respect, the study of a sorted downward refinement (Frisch 1999) can be also considered a contribution to learning in hybrid languages. In this paper, we have proposed a general framework for learning in $\mathscr{AL}$-log. We would like to emphasize that the *DL-safeness* and the *decidability* of $\mathscr{AL}$-log are two desirable properties which are particularly appreciated both in ILP and in the Semantic Web application area.

As an instantiation of the framework, we have considered the case of characteristic induction from interpretations, more precisely, the task of Frequent Pattern

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

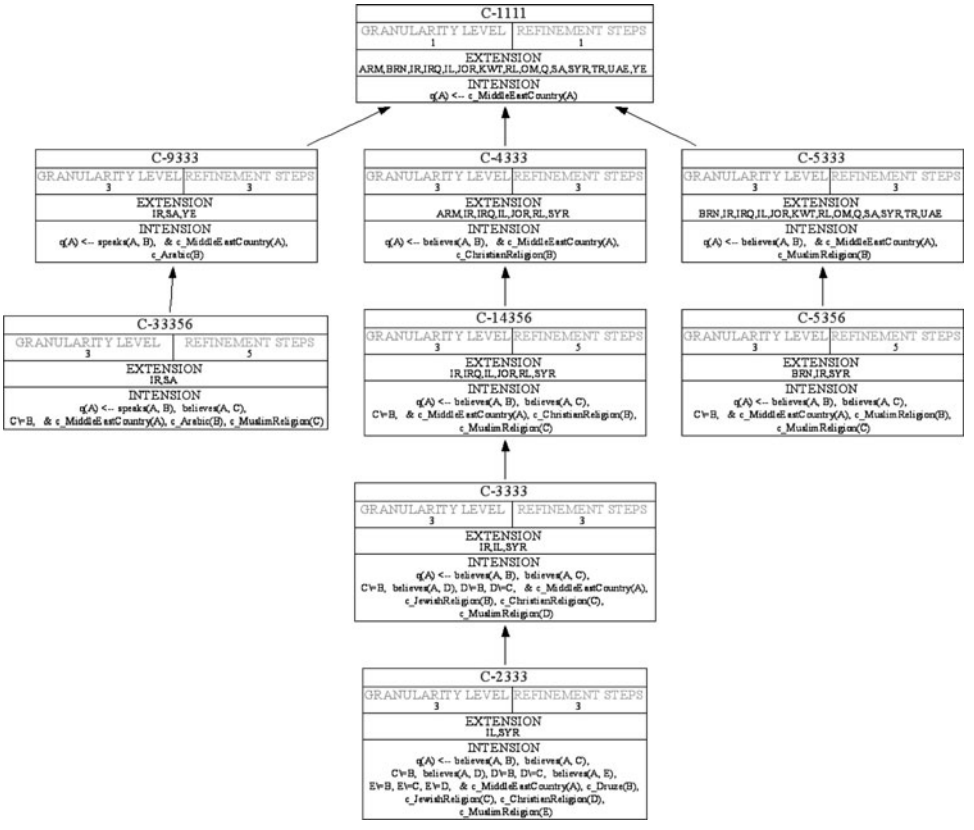


Fig. 6. Taxonomy $\mathscr{G}'''_{\mathtt{CIA}}$ obtained with the m.s.d. criterion for $minG = 3$.

Discovery, and an application to Ontology Refinement. The specific problem at hand takes an ontology as input and returns subconcepts of one of the concepts in the ontology. A distinguishing feature of our setting for this problem is that the intensions of these subconcepts are in the form of rules that are automatically built by discovering strong associations between concepts in the input ontology. The idea of resorting to Frequent Pattern Discovery in Ontology Learning has been already investigated in (Maedche and Staab 2000). Yet, there are several differences between Maedche and Staab (2000) and the present work. Maedche and Staab (2000) is conceived for Ontology Extraction instead of Ontology Refinement, uses generalized association patterns (bottom-up search), instead of multi-level association patterns (top-down search), adopts propositional logic instead of FOL. Within the same application area, Maedche and Zacharias (2002) propose a distance-based method for clustering in RDF which is not conceptual. Also, the relation between Frequent Pattern Discovery and Concept Formation as such has never been investigated. Rather, our pattern-based approach to clustering is inspired by (Xiong *et al.* 2005). Some contact points can be also found with Zimmermann and Raedt (2004) that defines the problem of cluster-grouping and a solution to it that integrates Subgroup Discovery, Correlated Pattern Mining and Conceptual Clustering. Note that neither

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Xiong *et al.* (2005) nor Zimmermann and Raedt (2004) deal with (fragments of) FOL. Conversely, Stumme (2004) combines the notions of frequent DATALOG query and iceberg concept lattices to upgrade Formal Concept Analysis (a well established and widely used approach for Conceptual Clustering) to FOL. Generally speaking, very few works on Conceptual Clustering and Concept Formation in FOL can be found in the literature. They vary as for the approaches (distance-based, probabilistic, etc.) and/or the representations (description logics, conceptual graphs, E/R models, etc.) adopted. The closest work to ours is Vrain's proposal (Vrain 1996) of a top-down incremental, but distance-based method for Conceptual Clustering in a mixed object-logical representation.

For the future, we plan to extensively evaluate this approach on significantly big and expressive ontologies. Without doubt, there is a lack of evaluation standards in Ontology Learning. Comparative work in this field would help an ontology engineer to choose the appropriate method. One step in this direction is the framework presented in (Bisson *et al.* 2000), but it is conceived for Ontology Extraction. The evaluation of our approach can follow the criteria outlined in (Dellschaft and Staab 2006) or criteria from the ML tradition like measuring the cluster validity (Halkidi *et al.* 2001), or the category utility (Fisher 1987). Anyway, due to the peculiarities of our approach, the evaluation itself requires a preliminary work from the methodological point of view. Regardless of performance, each approach has its own benefits. Our approach has the advantages of dealing with expressive ontologies and being conceptual. One such approach, and in particular, its ability of forming concepts with an intensional description in the form of rule, can support many of the use cases defined by the *W3C Rule Interchange Format Working Group*. Another direction of future work can be the extension of the present work toward hybrid formalisms, (e.g., Motik *et al.* 2004), that are more expressive than $\mathcal{AL}$-log and more inspiring for prototipical SWRL reasoners. Also, we would like to investigate other instantiations of the framework, e.g., the ones in the case of discriminant induction to learn predictive rules.

References

BAADER, F., CALVANESE, D., MCGUINNESS, D., NARDI, D., AND PATEL-SCHNEIDER, P., Eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press, Cambridge.

BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. 2001. The Semantic Web. *Scientific American May*.

BISSON, G., NEDELLEC, C., AND CAÑAMERO, D. 2000. Designing clustering methods for ontology building—the Mo'K workbench. In *ECAI Workshop on Ontology Learning*, S. Staab, A. Maedche, C. Nedellec, and P. Wiemer-Hastings, Eds. CEUR Workshop Proceedings, vol. 31. CEUR-WS.org.

BLOCKEEL, H., DE RAEDT, L., JACOBS, N., AND DEMOEN, B. 1999. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery 3*, 59–93.

BORGIDA, A. 1996. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence 82,* 1–2, 353–367.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Buntine, W. 1988. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence 36,* 2, 149–176.

Ceri, S., Gottlob, G., and Tanca, L. 1990. *Logic Programming and Databases*. Springer, New York.

De Raedt, L. 1997. Logical settings for concept-learning. *Artificial Intelligence 95,* 1, 187–201.

De Raedt, L. and Dehaspe, L. 1997. Clausal discovery. *Machine Learning 26,* 2–3, 99–146.

De Raedt, L. and Džeroski, S. 1994. First order jk-clausal theories are PAC-learnable. *Artificial Intelligence 70*, 375–392.

Dellschaft, K. and Staab, S. 2006. On how to perform a gold standard based evaluation of ontology learning. In *The Semantic Web*, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, Eds. Lecture Notes in Computer Science, vol. 4273. Springer, Berlin/Heidelberg, 228–241.

Donini, F., Lenzerini, M., Nardi, D., and Schaerf, A. 1998. $\mathcal{AL}$-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems 10,* 3, 227–252.

Fisher, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2,* 2, 139–172.

Flach, P. and Lavrač, N. 2002. Learning in clausal logic: A perspective on inductive logic programming. In *Computational Logic: Logic Programming and Beyond*, A. Kakas and F. Sadri, Eds. Lecture Notes in Computer Science, vol. 2407. Springer, Berlin/Heidelberg, 437–471.

Frazier, M. and Page, C. 1993. Learnability in inductive logic programming. In *Proceedings of the 10st National Conference on Artificial Intelligence*. The AAAI Press/The MIT Press, 93–98.

Frisch, A. 1991. The substitutional framework for sorted deduction: Fundamental results on hybrid reasoning. *Artificial Intelligence 49*, 161–198.

Frisch, A. 1999. Sorted downward refinement: Building background knowledge into a refinement operator for inductive logic programming. In *Inductive Logic Programming*, S. Džeroski and P. Flach, Eds. Lecture Notes in Artificial Intelligence, vol. 1634. Springer, Berlin/Heidelberg, 104–115.

Frisch, A. and Cohn, A. 1991. Thoughts and afterthoughts on the 1988 workshop on principles of hybrid reasoning. *AI Magazine 11,* 5, 84–87.

Gennari, J., Langley, P., and Fisher, D. 1989. Models of incremental concept formation. *Artificial Intelligence 40,* 1–3, 11–61.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. 2004. *Ontological Engineering*. Springer, London.

Gruber, T. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition 5*, 199–220.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. 2001. On clustering validation techniques. *Journal of Intelligent Information Systems 17,* 2–3, 107–145.

Han, J. and Fu, Y. 1999. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering 11,* 5, 798–804.

Hartigan, J. 2001. Statistical clustering. In *International Encyclopedia of the Social and Behavioral Sciences*, N. Smelser and P. Baltes, Eds. Oxford Press, Oxford, 15014–15019.

Horrocks, I., Patel-Schneider, P., and van Harmelen, F. 2003. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics 1,* 1, 7–26.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Horrocks, I., Sattler, U., and Tobies, S. 2000. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL 8,* 3, 239–263.

Kietz, J. 2003. Learnability of description logic programs. In *Inductive Logic Programming*, S. Matwin and C. Sammut, Eds. Lecture Notes in Artificial Intelligence, vol. 2583. Springer, Berlin/Heidelberg, 117–132.

Langley, P. 1987. Machine learning and concept formation. *Machine Learning 2,* 2, 99–102.

Levy, A. and Rousset, M.-C. 1998. Combining Horn rules and description logics in CARIN. *Artificial Intelligence 104*, 165–209.

Lisi, F. 2006. Practice of inductive reasoning on the semantic web: A system for semantic web mining. In *Principles and Practice of Semantic Web Reasoning*, J. Alferes, J. Bailey, W. May and U. Schwertel, Eds. Lecture Notes in Computer Science, vol. 4187. Springer, Berlin/Heidelberg, 242–256.

Lisi, F. and Esposito, F. 2004. Efficient evaluation of candidate hypotheses in $\mathscr{AL}$-log. In *Inductive Logic Programming*, R. Camacho, R. King, and A. Srinivasan, Eds. Lecture Notes in Artificial Intelligence, vol. 3194. Springer, Berlin, 216–233.

Lisi, F. and Esposito, F. 2006. Two orthogonal biases for choosing the intensions of emerging concepts in ontology refinement. In *ECAI 2006. Proceedings of the 17th European Conference on Artificial Intelligence*, G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, Eds. IOS Press, Amsterdam, 765–766.

Lisi, F. and Esposito, F. 2007. On the missing link between frequent pattern discovery and concept formation. In *Inductive Logic Programming*, S. Muggleton, R. Otero, and A. Tamaddoni-Nezhad, Eds. Lecture Notes in Artificial Intelligence, vol. 4455. Springer, Berlin/Heidelberg, 305–319.

Lisi, F. and Malerba, D. 2003a. Bridging the gap between horn clausal logic and description logics in inductive learning. In *AI*IA 2003: Advances in Artificial Intelligence*, A. Cappelli and F. Turini, Eds. Lecture Notes in Artificial Intelligence, vol. 2829. Springer, Berlin/Heidelberg, 49–60.

Lisi, F. and Malerba, D. 2003b. Ideal refinement of descriptions in $\mathscr{AL}$-log. In *Inductive Logic Programming*, T. Horvath and A. Yamamoto, Eds. Lecture Notes in Artificial Intelligence, vol. 2835. Springer, Berlin/Heidelberg, 215–232.

Lisi, F. and Malerba, D. 2004. Inducing multi-level association rules from multiple relations. *Machine Learning 55*, 175–210.

Maedche, A. and Staab, S. 2000. Discovering conceptual relations from text. In *Proceedings of the 14th European Conference on Artificial Intelligence*, W. Horn, Ed. IOS Press, Amsterdam, 321–325.

Maedche, A. and Staab, S. 2004. Ontology learning. In *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Springer, Berlin.

Maedche, A. and Zacharias, V. 2002. Clustering ontology-based metadata in the semantic web. In *Principles of Data Mining and Knowledge Discovery*, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Lecture Notes in Computer Science, vol. 2431. Springer, Berlin/Heidelberg, 348–360.

Mannila, H. and Toivonen, H. 1997. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery 1,* 3, 241–258.

Michalski, R. and Stepp, R. 1983. Learning from observation: Conceptual clustering. In *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonell, and T. Mitchell, Eds. Vol. I. Morgan Kaufmann, San Mateo, CA.

Mitchell, T. 1982. Generalization as search. *Artificial Intelligence 18*, 203–226.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Motik, B., Sattler, U., and Studer, R. 2004. Query answering for OWL-DL with rules. In *The Semantic Web*, S. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. Lecture Notes in Computer Science, vol. 3298. Springer, Berlin/Heidelberg, 549–563.

Nédellec, C., Rouveirol, C., Adé, H., Bergadano, F., and Tausend, B. 1996. Declarative bias in ILP. In *Advances in Inductive Logic Programming*, L. D. Raedt, Ed. IOS Press, Amsterdam, 82–103.

Nienhuys-Cheng, S. and de Wolf, R. 1997. *Foundations of Inductive Logic Programming*. Lecture Notes in Artificial Intelligence, vol. 1228. Springer, New York.

Reiter, R. 1980. Equality and domain closure in first order databases. *Journal of ACM 27*, 235–249.

Rosati, R. 2005. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics 3,* 1, 61–73.

Rouveirol, C. and Ventos, V. 2000. Towards Learning in CARIN-$\mathcal{ALN}$. In *Inductive Logic Programming*, J. Cussens and A. Frisch, Eds. Lecture Notes in Artificial Intelligence, vol. 1866. Springer, Berlin/Heidelberg, 191–208.

Schmidt-Schauss, M. and Smolka, G. 1991. Attributive concept descriptions with complements. *Artificial Intelligence 48,* 1, 1–26.

Semeraro, G., Esposito, F., Malerba, D., Fanizzi, N., and Ferilli, S. 1998. A logic framework for the incremental inductive synthesis of Datalog theories. In *Proceedings of 7th International Workshop on Logic Program Synthesis and Transformation*, N. Fuchs, Ed. Lecture Notes in Computer Science, vol. 1463. Springer, Berlin, 300–321.

Staab, S. and Studer, R., Eds. 2004. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, Berlin.

Stumme, G. 2004. Iceberg query lattices for Datalog. In *Conceptual Structures at Work*, K. Wolff, H. Pfeiffer, and H. Delugach, Eds. Lecture Notes in Artificial Intelligence, vol. 3127. Springer, Berlin/Heidelberg, 109–125.

Utgoff, P. and Mitchell, T. 1982. Acquisition of appropriate bias for inductive concept learning. In *Proceedings of the 2nd National Conference on Artificial Intelligence*. Morgan Kaufmann, Los Altos, CA, 414–418.

Vrain, C. 1996. Hierarchical conceptual clustering in a first order representation. In *Foundations of Intelligent Systems*, Z. Ras and M. Michalewicz, Eds. Lecture Notes in Computer Science, vol. 1079. Springer, Berlin, 643–652.

Xiong, H., Steinbach, M., Ruslim, A., and Kumar, V. 2005. Characterizing pattern-based clustering. Technical Report TR 05-015, Dept. of Computer Science and Engineering, University of Minnesota, Minneapolis, USA.

Zimmermann, A. and Raedt, L. D. 2004. Cluster-grouping: From subgroup discovery to clustering. In *Machine Learning: ECML 2004*, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Lecture Notes in Artificial Intelligence, vol. 3201. Springer, Berlin/Heidelberg, 575–577.

Appendix A The semantic markup language OWL

The Web Ontology Language (OWL) is a semantic mark-up language for publishing and sharing ontologies on the World Wide Web (Horrocks *et al*. 2003). An OWL ontology is an RDF graph, which is in turn a set of RDF triples. As with any RDF graph, an OWL ontology graph can be written in many different syntactic forms. However, the meaning of an OWL ontology is solely determined by the RDF graph.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

Table A1. *Mapping from $\mathcal{ALC}$ to OWL*

$\neg C$	`<owl:Class>`
	`  <owl:complementOf><owl:Class rdf:ID="C" /></owl:complementOf>`
	`</owl:Class>`

$C \sqcap D$	`<owl:Class>`
	`  <owl:intersectionOf rdf:parseType="Collection">`
	`  <owl:Class rdf:ID="C" /><owl:Class rdf:ID="D" />`
	`  </owl:intersectionOf>`
	`</owl:Class>`

$C \sqcup D$	`<owl:Class>`
	`  <owl:unionOf rdf:parseType="Collection">`
	`  <owl:Class rdf:ID="C" /><owl:Class rdf:ID="D" />`
	`  </owl:unionOf>`
	`</owl:Class>`

$\exists R \cdot C$	`<owl:Restriction>`
	`  <owl:onProperty rdf:resource="#R" />`
	`  <owl:someValuesFrom rdf:resource="#C" />`
	`</owl:Restriction>`

$\forall R \cdot C$	`<owl:Restriction>`
	`  <owl:onProperty rdf:resource="#R" />`
	`  <owl:allValuesFrom rdf:resource="#C" />`
	`</owl:Restriction>`

$C \equiv D$	`<owl:Class rdf:ID="C">`
	`  <owl:sameAs rdf:resource="#D" />`
	`</owl:Class>`

$C \sqsubseteq D$	`<owl:Class rdf:ID="C">`
	`  <rdfs:subClassOf rdf:resource="#D" />`
	`</owl:Class>`

$a : C$	`<C rdf:ID="a" />`
$\langle a, b \rangle : R$	`<C rdf:ID="a"><R rdf:resource="#b" />`

Thus, it is allowable to use other syntactic RDF/XML forms, as long as these result in the same underlying set of RDF triples.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. OWL Lite also has a lower formal complexity than OWL DL.
- *OWL DL* supports those users who want the maximum expressiveness while retaining computational completeness and decidability. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (e.g., while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press

with the very expressive DL $\mathcal{SHOIN}(\mathbf{D})$ (Horrocks *et al.* 2000), which thus provides a logical foundation to OWL. The mapping from $\mathcal{ALC}$ to OWL is reported in Table A1.

- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full, a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the predefined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded.

https://doi.org/10.1017/S1471068407003195 Published online by Cambridge University Press