Artificial Intelligence 344 (2025) 104345

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Effective and fast module extraction for nonempty ABoxes

Piero Andrea Bonatti [a,*], Francesco Magliocca [c], Iliana Mineva Petrova [b], Luigi Sauro [a,*]

[a] Dept. of Electrical Eng. and Information Technologies, Università di Napoli Federico II, Italy
[b] INRIA, Sophia Antipolis, France
[c] Dept. of Mathematics and Applications "Renato Caccioppoli", Università di Napoli Federico II, Italy

ARTICLE INFO

Keywords:
Description logics
Module extraction

ABSTRACT

A deductive module of a knowledge base $\mathcal{KB}$ is a subset of $\mathcal{KB}$ that preserves a specified class of consequences. Module extraction is applied in ontology design, debugging, and reasoning. The locality-based module extractors of the OWL API are less effective when the knowledge base contains facts such as ABox assertions. The competing module extractor PrisM computes smaller modules at the cost of higher computation time. In this paper, we introduce and study a novel module extraction technique, called *conditional module extraction*, that can be applied to satisfiable $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases. Experimental analysis shows that conditional module extraction constitutes an appealing alternative to PrisM and to the locality-based extractors of the OWL API, when the ABox is nonempty.

1. Introduction

Given an ontology $\mathcal{KB}$ formulated with some description logic (DL), and a signature Σ, a Σ-*module* is a subset of $\mathcal{KB}$ that preserves a specified class of consequences of $\mathcal{KB}$, expressed with the symbols in Σ. Several applications in ontology design [1,2], debugging [3,4] and reasoning [5–7] fostered the development of practical techniques to extract a Σ-module from a knowledge base.

Depending on the application at hand, different notions of modules can be adopted by varying the class of preserved consequences (e.g. facts, first/second-order consequences, etc.) [8,9]. Here, we focus on modules that preserve subsumption and instance checking – that is, the consequences we intend to preserve are concept and role inclusions and assertions – as well as what we call *anchored* conjunctive queries.

Ideally, modules should be as small as possible. However, computing $\subseteq$-minimal modules is often computationally unattainable in concrete applications. For example, deciding whether $\mathcal{M}$ is a minimal Σ-module of an ontology $\mathcal{KB}$ formulated with the low-complexity DLs $\mathcal{EL}$ and *DL-lite*$_{horn}$ is EXPTIME-complete and coNP-complete, respectively [10,11]; for $\mathcal{ALC}$, the problem is complete for 2EXPTIME [12], while for $\mathcal{ALCQIO}$ it is undecidable [13].

Several practical techniques have been proposed to extract modules from a knowledge base. Due to the above complexity results, such techniques do not guarantee to find minimal modules, as they have to balance effectiveness (i.e. how small the returned modules are) and performance (the execution time of the module extractors). In particular, the module extractors based on the notion of *locality* proved to be fast and effective enough to be included in the standard OWLAPI [14].

* Corresponding authors.
E-mail address: pab@unina.it (P.A. Bonatti).

https://doi.org/10.1016/j.artint.2025.104345
Received 7 January 2023; Received in revised form 14 April 2025; Accepted 18 April 2025

Available online 24 April 2025

It has been observed that nonempty ABoxes may negatively affect the size of the output modules, turning out to be an Achilles' heel for these techniques (cf. Sec. 3.3.5 and 3.5 of [15]). Despite this, to the best of our knowledge, no module extraction methods optimized for $\mathcal{KB}$ with nonempty ABoxes have been introduced so far. We address this gap by introducing a module extraction algorithm for $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases, which, on average, discards significantly more axioms in the presence of nonempty ABoxes.

Our method almost completely decouples the extraction of ABox and TBox modules. In a first phase, it computes a Σ-module $\mathcal{A}'$ of the ABox by collecting the assertions syntactically related to Σ. In a second phase, it calls a module extractor mx that computes a Σ'-module $\mathcal{H}'$ of the TBox, where Σ' extends Σ with the symbols that occur in $\mathcal{A}'$. The module of $\mathcal{KB}$ is simply $\mathcal{A}' \cup \mathcal{H}'$. Our method is parametric in mx, which can be any extractor whose output is model inseparable from $\mathcal{KB}$. The two-phase extraction method is called *conditional* because it is correct under the assumption that the knowledge base is consistent. The consistency hypothesis, in practice, is compatible with the main intended uses of module extraction, such as importing selected parts of an already validated ontology, or speeding up the computation of its logical consequences.

Conditional module extraction will be assessed experimentally by comparing it with the module extractors of the OWLAPI, and their competitor PrisM [9], a state of the art module extractor that produces particularly small modules. We will pay equal attention to the size of modules and extraction time, because we are particularly interested in the applications that require repeated, possibly on-the-fly module extractions, as it happens – for instance – in Chainsaw's query-by-query optimizations [6], in optimized justification finding [3,4], in incremental re-classification after ontology updates [7], and in the iterative computations of secure knowledge base views [16] and nonmonotonic closures [17]. In these frameworks, if a moderate reduction of module size had to be payed for with a substantial increase in extraction time, then the advantages of reasoning with a smaller module would be overridden by the cost of module extraction.

The next section recalls the basics of description logics and module extraction needed in this paper. In Section 3, we present conditional modules in a non-constructive way, and establish their key theoretical properties, including self-containment, depletingness, and conditional completeness with respect to subsumption, instance checking, and anchored conjunctive queries (inseparability properties). In Sec. 3.3 we provide a constructive approach to the computation of conditional modules. Section 4 is devoted to the experimental assessment of the extractors, and Section 5 concludes the paper with some final remarks.

2. Preliminaries

2.1. Description logics

Description logics (DLs) are a family of formal languages that offer a variegated set of logical operators that balance expressiveness and computational complexity according to the application needs. In this work, we focus on the DL $\mathcal{SRIQ}(\mathcal{D})$. An alphabet consists of: (i) a set $\mathsf{N_C}$ of *concept names*; (ii) a set $\mathsf{N_D}$ of *datatypes* (sometimes called concrete domains); (iii) a set $\mathsf{N_R}$ of *role names*; (iv) a set $\mathsf{N_F}$ of *feature names*; (v) a set $\mathsf{N_I}$ of *individual names* (all countably infinite and mutually disjoint). We use the term *predicate* to refer to elements of $\mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_F}$. Hereafter, we use Σ to denote a *signature*, that is, any subset of $\mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_F} \cup \mathsf{N_I}$.

For the sake of simplicity, we adhere to the original formalization presented in [18], which employs a single concrete domain $\Delta^{\mathcal{D}}$. However, our methodology can be extended in a straightforward manner to accommodate multiple concrete domains (for example, the OWL2 standard inherits from XML Schema datatypes such as numbers, strings, booleans, and binary values, plus infinitely many subtypes, e.g. "integers greater than 17").[1]

The semantics of DLs is defined in terms of *interpretations*. First, each datatype $\mathtt{d} \in \mathsf{N_D}$ is associated with its interpretation $\mathtt{d}^{\mathcal{D}} \subseteq \Delta^{\mathcal{D}}$. As in [18], we assume that for each datatype $\mathtt{d}$ there exists another one, denoted with $\neg\mathtt{d}$, such that $\neg\mathtt{d}^{\mathcal{D}} = \Delta^{\mathcal{D}} \setminus \mathtt{d}^{\mathcal{D}}$. Moreover, checking the emptiness of expressions $\mathtt{d}_1^{\mathcal{D}} \cap \ldots \cap \mathtt{d}_n^{\mathcal{D}}$ is decidable. Then, an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty *domain* of individuals $\Delta^{\mathcal{I}}$, disjoint from $\Delta^{\mathcal{D}}$, and an *interpretation function* $\cdot^{\mathcal{I}}$ mapping: (a) each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$; (b) each role name $R \in \mathsf{N_R}$ to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$; (d) each feature name $F \in \mathsf{N_F}$ to a binary relation $F^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{D}}$; (e) each individual name $a \in \mathsf{N_I}$ to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Finally, datatypes always maintain the same meaning, that is, $\mathtt{d}^{\mathcal{I}} = \mathtt{d}^{\mathcal{D}}$, for all interpretations $\mathcal{I}$.

In $\mathcal{SRIQ}(\mathcal{D})$ compound concepts and roles are inductively defined with the logical operators described in Table 1. We will use metavariables A, B for concept names, C, D for (possibly compound) concepts, R, S for (possibly inverse) roles and a, b for individual names. The third column of Table 1 shows how to extend the valuation $\cdot^{\mathcal{I}}$ of an interpretation $\mathcal{I}$ to compound concept expressions. Table 1 also shows the terminological and assertional axioms of $\mathcal{SRIQ}(\mathcal{D})$. The acronym GCI stands for *general concept inclusion* and RIA stands for *role inclusion axiom*. An interpretation $\mathcal{I}$ *satisfies* an axiom α (in symbols, $\mathcal{I} \models \alpha$) if it satisfies the corresponding semantic condition in Table 1. As usual, we use the standard abbreviations $\bot$ for $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, $\forall R.C$ for $\neg\exists R.\neg C$, and $(\leq n\, R.C)$ for $\neg(\geq n+1\, R.C)$.

A finite set of terminological axioms $\mathcal{T}$ is called a TBox[2]; similarly, a finite set of assertion axioms $\mathcal{A}$ is called an ABox. Then, a knowledge base (or ontology) $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$ is the union of a TBox and an ABox. An interpretation $\mathcal{I}$ is a *model* of a knowledge base $\mathcal{KB}$ (in symbols, $\mathcal{I} \models \mathcal{KB}$) if $\mathcal{I}$ satisfies all the axioms in $\mathcal{KB}$. Moreover, $\mathcal{KB}$ *entails* an axiom α (in symbols, $\mathcal{KB} \models \alpha$) if all the models of $\mathcal{KB}$ satisfy α. Note that a $\mathcal{SRIQ}(\mathcal{D})$ knowledge base must satisfy some restrictions in order to guarantee the decidability of axiom entailment. Since such restrictions play no role in this paper, we refer the reader to [19] for further details.

1 OWL2 datatypes are defined in https://www.w3.org/TR/owl2-syntax/#Datatype_Maps.

2 Several works distinguish between TBoxes, that consists of GCIs only, and RBoxes that contain role axioms. Here, this distinction is immaterial, therefore we fuse them together to simplify notation.

Table 1
Syntax and semantics of $\mathcal{SRIQ}(\mathcal{D})$.

Name	Syntax	Semantics
Class and role expressions		
inverse role	R^-	$\{(y,x) \mid (x,y) \in R^{\mathcal{I}}\}$ $(R \in \mathsf{N_R})$
top	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
intersection	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
complement	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists (d,e) \in R^{\mathcal{I}} : e \in C^{\mathcal{I}}\}$
number restrictions	$(\geq n\ S.C)$	$\left\{d \in \Delta^{\mathcal{I}} \mid \#\{e \mid (d,e) \in S^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \geq n\right\}$
self	$\exists S.\mathsf{Self}$	$\{d \in \Delta^{\mathcal{I}} \mid (d,d) \in S^{\mathcal{I}}\}$
concrete restriction	$\exists F.\mathsf{d}$	$\{d \in \Delta^{\mathcal{I}} \mid \exists (d,v) \in F^{\mathcal{I}} : v \in \mathsf{d}^{\mathcal{I}}\}$
Terminological axioms		$\mathcal{I}$ satisfies the axiom if:
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role disjointness	$\mathsf{disj}(R_1, R_2)$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$
RIA	$R_1 \circ \ldots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
reflexivity	$\mathsf{ref}(R)$	$R^{\mathcal{I}}$ is reflexive
irreflexivity	$\mathsf{irr}(R)$	$R^{\mathcal{I}}$ is irreflexive
symmetry	$\mathsf{sym}(R)$	$R^{\mathcal{I}}$ is symmetric
transitivity	$\mathsf{tra}(S)$	$R^{\mathcal{I}}$ is transitive
Concept and role assertion axioms		
conc. assrt.	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assrt.	$R(a,b)$	$(a,b)^{\mathcal{I}} \in R^{\mathcal{I}}$
neg. role assrt.	$\neg R(a,b)$	$(a,b)^{\mathcal{I}} \notin R^{\mathcal{I}}$

Each $\mathcal{SRIQ}(\mathcal{D})$ TBox $\mathcal{T}$ enjoys the *disjoint model union property*,[3] that is, if $\mathcal{I}$ and $\mathcal{J}$ are two disjoint models of $\mathcal{T}$ (i.e. $\Delta^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = \emptyset$) then also the union $\mathcal{K}$ of $\mathcal{I}$ and $\mathcal{J}$ defined below is a model of $\mathcal{T}$:

$$\Delta^{\mathcal{K}} = \Delta^{\mathcal{I}} \cup \Delta^{\mathcal{J}},$$
$$X^{\mathcal{K}} = X^{\mathcal{I}} \cup X^{\mathcal{J}} \quad \text{for all } X \in \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_F}.$$

2.2. Conjunctive queries

A Conjunctive Query (CQ) is an existentially quantified FOL formula $q(\vec{x}) = \exists \vec{y}(conj(\vec{x}, \vec{y}))$, where $conj(\vec{x}, \vec{y}))$ is a conjunction $\alpha_1 \wedge \ldots \wedge \alpha_n$ of atoms where only the variables in $\vec{x}$ and $\vec{y}$ may occur. Given a tuple of individual constants $\vec{t}$, by $q(\vec{t})$ we mean the closed CQ which is obtained by replacing each free variable x_i in $\vec{x}$ with the corresponding t_i in $\vec{t}$. Then, $\vec{t}$ is an answer of $q(\vec{x})$ w.r.t. a knowledge base $\mathcal{KB}$ if $\mathcal{KB} \models q(\vec{t})$ according to the standard FOL semantics. Henceforth, the query's free variables will be sometimes omitted when obvious or irrelevant.

2.3. Module extraction

In short, a module of a knowledge base $\mathcal{KB}$ with respect to a signature Σ is a subset $\mathcal{M}$ of $\mathcal{KB}$ that yields the same results as $\mathcal{KB}$ in every reasoning task where only the symbols in Σ are relevant. The first reasoning tasks we take into account are subsumption and instance checking; consequently, in the following, by *query* we mean any axiom q that is a GCI, a RIA or an assertion, unless stated otherwise (later on, in Sec. 3.2, we will define and study anchored conjunctive queries). Formally, deductive Σ-modules are defined as shown below; hereafter $\mathrm{sig}(X)$ denotes the signature containing the symbols that occur in X, where X can be an expression, an axiom, or a knowledge base:

Definition 1. Let Σ be a signature. A subset $\mathcal{M}$ of $\mathcal{KB}$ is a *(deductive) Σ-module* of $\mathcal{KB}$ iff for all queries q with $\mathrm{sig}(q) \subseteq \Sigma$, it holds that $\mathcal{M} \models q$ iff $\mathcal{KB} \models q$.

[3] Unfortunately, the disjoint model union property plays an important role in our results, so we cannot generalize them to $\mathcal{SROIQ}(\mathcal{D})$ TBoxes, which do not enjoy this property due to nominals and the universal role.

Another indistinguishability criterion we will make use of is model inseparability [20,8,9]. Specifically, two knowledge bases $\mathcal{KB}_1$ and $\mathcal{KB}_2$ are *model inseparable* w.r.t. a signature Σ iff it is possible to turn a model of $\mathcal{KB}_1$ into a model of $\mathcal{KB}_2$ and viceversa by re-interpreting only symbols outside Σ. Formally, $\mathcal{KB}_1$ and $\mathcal{KB}_2$ are Σ-model inseparable iff:

$$\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{KB}_1\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{KB}_2\},$$

where $\mathcal{I}|_\Sigma$ means the restriction of $\mathcal{I}$ to the signature Σ.

If $\mathcal{KB}$ and some of its subsets $\mathcal{M}$ are Σ-model inseparable, then $\mathcal{M}$ preserves all first-order and second-order consequences of $\mathcal{KB}$ expressible with Σ [21], consequently $\mathcal{M}$ is also a deductive Σ-module.

A common technique to extract a Σ-module $\mathcal{M}$ from $\mathcal{KB}$ is based on locality [8,22,1]. The underlying idea is the following: a $\perp$-*module* of $\mathcal{KB}$ with respect to a given signature of interest Σ is a set $\mathcal{M} \subseteq \mathcal{KB}$ such that all axioms $\alpha \in \mathcal{KB} \setminus \mathcal{M}$ are $\perp$-*local* with respect to $\Sigma \cup \text{sig}(\mathcal{M})$. This means that if all concept names and role names that do not belong to $\Sigma \cup \text{sig}(\mathcal{M})$ are replaced with $\perp$ and the empty role, respectively, then α becomes a tautology. A knowledge base $\mathcal{KB}$ and its $\perp$-modules are model inseparable w.r.t. $\Sigma \cup \text{sig}(\mathcal{M})$. The definition of $\top$-modules and $\top$-locality are analogous, the only difference is that the concept and role names that are not in $\Sigma \cup \text{sig}(\mathcal{M})$ are replaced with $\top$ and the universal role, respectively.

In order to make locality checking tractable, syntactic approximations of locality have been introduced [1,9,8]. This approach is implemented in the OWLAPI[4] that supports three module extractors. Two of them are denoted by x-Mod(Σ, $\mathcal{KB}$), where $x \in \{\top, \perp\}$; x indicates which notion of locality is used. The third is denoted by $\top\perp^*$-Mod(Σ, $\mathcal{KB}$), and consists in an iterative alternation of the previous two, that is, $\top\perp^*$-Mod(Σ, $\mathcal{KB}$) is the limit of the sequence:

$$\mathcal{M}_0 = \mathcal{KB},$$

$$\mathcal{M}_{i+1} = \top\text{-Mod}(\Sigma, \perp\text{-Mod}(\Sigma, \mathcal{M}_i)).$$

Clearly $\top\perp^*$-Mod(Σ, $\mathcal{KB}$) in general returns the smallest module among the three extractors.

Although these techniques do not necessarily extract a $\subseteq$-minimal module, they turn out to be effective enough to be used in many concrete cases. Nevertheless, the module extractor of the OWLAPI based on $\perp$-locality (i.e. $\perp$-Mod) may be significantly less effective when applied to knowledge bases with nonempty ABoxes. The reason is that replacing any predicate in a negation-free assertion with $\perp$ (e.g. replacing $A(c)$ with $\perp(c)$) always produces an inconsistent (therefore nonlocal) axiom. Since positive assertions are very common, the module's signature tends to contain all the predicate symbols occurring in the ABox. In turn, this weakness of $\perp$-Mod reduces the effectiveness of the overall module extractor $\top\perp^*$-Mod.

An interesting competitor of the OWLAPI's module extractors is PrisM [9], which operates by reducing module extraction to Datalog reasoning. DL ontologies shall be converted into sets of rules before applying PrisM; essentially, this requires DL ontologies to be converted to a normal form specified in [9, Def. 2]. In general, PrisM can compute different kinds of modules, each of which preserves a different class of logical consequences, namely: all first and second order formulae, positive existential queries, ground atoms, and atomic implications (of the form $A(x) \to B(x)$). The extractor of modules of the first kind (which are Σ-model inseparable from $\mathcal{KB}$) will be denoted by χ_{m}, as in [9]. The experimental comparison with the OWLAPI's $\top\perp^*$-Mod reported in [9] shows that PrisM can extract significantly smaller modules at the price of remarkably higher computation time. This is due to the cost of hyperresolution. All of PrisM's extractors operate by constructing abstract versions of the hyperresolution proofs of suitable Σ-related statements implied by $\mathcal{KB}$. Modules are constructed by collecting the rules that contribute to the abstract proofs. The formal definitions are quite long and irrelevant to our work, so the interested reader is referred to [9] for the technical details.

There are other approaches to the computation of model inseparable modules, besides the OWLAPI's extractors and PrisM, for example SLME (https://robot.obolibrary.org/extract.html), the Locality Module extractor based on [1] (https://www.cs.ox.ac.uk/isg/tools/ModuleExtractor/), CEX and MEX, based on [23] (https://www.csc.liv.ac.uk/~konev/software/).

Various techniques have been proposed for computing minimal deductive modules of TBoxes expressed in specific DL fragments. In [24], minimal deductive modules of $\mathcal{ALCH}$ TBoxes are computed using annotated interpolants. The experimental results demonstrate that, on average, the resulting modules are significantly smaller compared to the OWLAPI's $\top\perp^*$-Mod. However, the execution time is about one order of magnitude higher. In [25], the authors concentrate on $\mathcal{ELH}$ terminologies and present a methodology that relies on the system CEX to calculate the logical difference of two terminologies with respect to a signature. When applied to large-scale biomedical ontologies like Snomed CT, where signatures consist of 75 concept names and all role names, minimal modules are found to be up to 10 times smaller than $\top\perp^*$-Mod modules. An alternative approach based on subsumer/subsumee justifications is explored in [26]. The experimental evaluation on Snomed-CT reveals a 10-fold speed-up compared to the methodology proposed in [25].

3. Module extractors for ABoxes

We introduce a module extraction technique specifically tailored to nonempty ABoxes. We call such a technique *conditional* because it returns Σ-modules under the mildly restrictive assumption that $\mathcal{KB}$ is a consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge base. We start with a general, non-constructive definition – which captures the properties needed for proving the correctness of the method – and some of its theoretical properties.

[4] See https://github.com/owlcs/owlapi/wiki for further details.

Definition 2. A *conditional* Σ*-module* (c-Σ-module for short) of a $\mathcal{SRIQ}(\mathcal{D})$ knowledge base $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$ (where $\mathcal{T}$ and $\mathcal{A}$ are the TBox and the ABox of $\mathcal{KB}$, respectively) is a subset $\mathcal{M} \subseteq \mathcal{KB}$ such that:

1. there is no individual name $a \in \mathsf{N_I}$ such that $a \in \mathrm{sig}(\mathcal{A} \setminus \mathcal{M}) \cap (\Sigma \cup \mathrm{sig}(\mathcal{M}))$;
2. $\mathcal{M} \cap \mathcal{T}$ and $\mathcal{T}$ are model inseparable w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$.

Example 1. Consider the knowledge base $\mathcal{KB}$

$$\begin{array}{llll} A \sqsubseteq B & B' \sqsubseteq C & E \sqsubseteq F & \\ A(c) & R(c,d) & B'(d) & E(a)\,, \end{array}$$

and the signature $\Sigma = \{A, c\}$. Then, let $\mathcal{M} \subseteq \mathcal{KB}$ be the union of:

$$\mathcal{A}' = \{A(c),\ R(c,d), B'(d)\} \quad \mathcal{T}' = \{A \sqsubseteq B, B' \sqsubseteq C\}\,.$$

Since $E(a)$ is the only assertion in $\mathcal{A} \setminus \mathcal{M}$ and $a \notin \Sigma \cup \mathrm{sig}(\mathcal{M})$, $\mathcal{M}$ satisfies the first condition in Definition 2. Moreover, $E \sqsubseteq F$ is $\bot$-local w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$, therefore $\mathcal{T}'$ and $\mathcal{T}$ are model inseparable w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$ and, consequently, $\mathcal{M}$ is a c-Σ-module of $\mathcal{KB}$. Note that $\bot$-Mod$(\Sigma, \mathcal{KB})$ equals $\mathcal{KB}$. $\square$

If $\mathcal{KB} = \mathcal{A} \cup \mathcal{T}$ is consistent, then its c-Σ-modules are Σ-modules of $\mathcal{KB}$, as stated in the next theorem.

In summary, the theorem is proved by demonstrating that any model of $\mathcal{M}$ can be turned into a model of $\mathcal{KB}$ while mantaining the interpretation of every symbol in Σ unchanged. In essence, condition 1 in Definition 2 guarantees that $\mathcal{M}$ includes all the individual constants in $\mathcal{A}$ that are directly or indirectly connected to some individual constant in Σ. Since $\mathcal{T}$ does not contain nominals, this implies that each model $\mathcal{J}$ of $\mathcal{M}$ interprets all the individual constants that are relevant to deciding the entailment of any query q such that $\mathrm{sig}(q) \subseteq \Sigma$. Then, by model inseparability (condition 2 of Definition 2) $\mathcal{J}$ can be transformed into a model of $\mathcal{T}$ without altering the interpretation of the symbols in Σ. Finally, since $\mathcal{KB}$ is consistent, the constants that do not occur in $\mathcal{M}$ can be interpreted by a model $\mathcal{I}$ of $\mathcal{KB}$ disjoint from $\mathcal{J}$. The disjoint model union property ensures that a suitable composition of $\mathcal{I}$ and $\mathcal{J}$ is a model of $\mathcal{KB}$.

Theorem 1. *Let $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$ be a consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge base and let $\mathcal{M}$ be a c-Σ-module of $\mathcal{KB}$. Then $\mathcal{M}$ is a Σ-module of $\mathcal{KB}$.*

Proof. Let q be a query (i.e. a GCI, a RIA or an assertion) with $\mathrm{sig}(q) \subseteq \Sigma$. Since $\mathcal{M} \subseteq \mathcal{KB}$, it immediately follows by monotonicity that $\mathcal{M} \models q$ implies $\mathcal{KB} \models q$.

For the other direction, assume per absurdum that $\mathcal{KB} \models q$ and $\mathcal{M} \not\models q$. Then, for some interpretation $\mathcal{J}$ we have that

$$\mathcal{J} \models \mathcal{M} \text{ and } \mathcal{J} \not\models q\,. \tag{1}$$

Moreover, since by hypothesis $\mathcal{KB}$ is consistent, there exists a model $\mathcal{I}$ of $\mathcal{KB}$. We can assume w.l.o.g. that these two interpretations are disjoint, i.e. $\Delta^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = \emptyset$.

By definition of c-Σ-module, $\mathcal{M} \cap \mathcal{T}$ and $\mathcal{T}$ are model-inseparable w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$, therefore there exists a model $\mathcal{J}'$ of $(\mathcal{A} \cap \mathcal{M}) \cup \mathcal{T}$ which agrees with $\mathcal{J}$ on the predicate symbols in Σ (such as those occurring in q), consequently:

$$\mathcal{J}' \models \mathcal{A} \cap \mathcal{M},\ \mathcal{J}' \models \mathcal{T} \text{ and } \mathcal{J}' \not\models q\,. \tag{2}$$

We are now ready to define a model $\mathcal{K}$ of $\mathcal{KB}$ that does not satisfy q (which contradicts the assumption $\mathcal{KB} \models q$ and completes the proof). Let $\mathcal{K}$ be the composition of $\mathcal{I}$ and $\mathcal{J}'$ defined as follows:

$$\Delta^{\mathcal{K}} = \Delta^{\mathcal{I}} \cup \Delta^{\mathcal{J}'} \tag{3}$$

$$X^{\mathcal{K}} = X^{\mathcal{I}} \cup X^{\mathcal{J}'} \quad \text{for all } X \in \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_F} \tag{4}$$

$$a^{\mathcal{K}} = \begin{cases} a^{\mathcal{J}'} & \text{if } a \in \mathrm{sig}(\mathcal{M}) \cup \Sigma \\ a^{\mathcal{I}} & \text{otherwise.} \end{cases} \tag{5}$$

Note that the restriction of $\mathcal{K}$ to predicate names (i.e. where individual names are ignored) is the disjoint union of the restriction of $\mathcal{I}$ and $\mathcal{J}'$ to predicate names, and note that no role connects the two domains $\Delta^{\mathcal{I}}$ and $\Delta^{\mathcal{J}'}$.

First, we show that $\mathcal{K}$ is a model of $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$, by proving that $\mathcal{K}$ satisfies $\mathcal{T}$, $\mathcal{A} \cap \mathcal{M}$, and $\mathcal{A} \setminus \mathcal{M}$. Let us start by considering the TBox $\mathcal{T}$. Since $\mathcal{T}$ is a $\mathcal{SRIQ}(\mathcal{D})$ TBox, (i) $\mathcal{T}$ enjoys the disjoint model union property and (ii) the interpretation of $\mathcal{T}$'s axioms does not depend on the interpretation of individual names. Then, since both $\mathcal{I}$ and $\mathcal{J}'$ are models of $\mathcal{T}$, and $\mathcal{K}$ interprets predicate names like the disjoint union of the restriction of $\mathcal{I}$ and $\mathcal{J}'$ to predicates, it follows that $\mathcal{K}$ is a model of $\mathcal{T}$ by the disjoint model union property.

Next, we prove that $\mathcal{K}$ is a model of $\mathcal{A} \cap \mathcal{M}$. Let α be any assertion axiom in $\mathcal{M}$. By construction, for all individual names a occurring in $\mathrm{sig}(\alpha) \subseteq \mathrm{sig}(\mathcal{M})$, $a^{\mathcal{K}} = a^{\mathcal{J}'} \in \Delta^{\mathcal{J}'}$, and $R^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = R^{\mathcal{J}'}$. So, if α is a role assertion $R(a,b)$ or $\neg R(a,b)$, then by definition of $\mathcal{K}$,

$$\mathcal{K} \models \alpha \text{ iff } \mathcal{J}' \models \alpha . \tag{6}$$

If α is a concept assertion $C(a)$, instead, then the proof of (6) is only slightly more complex. First, by structural induction, we prove that for all $\mathcal{SRIQ}(\mathcal{D})$ concepts D,

$$D^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = D^{\mathcal{J}'} . \tag{7}$$

The base cases $D = A$ ($A \in \mathsf{N_C}$) and $D = \top$ hold by construction, whereas the case $D = \exists F.\mathtt{d}$ is a direct consequence of the facts that the projection of $F^{\mathcal{K}}$ on $\Delta^{\mathcal{J}'}$ equals $F^{\mathcal{J}'}$ and $\mathtt{d}^{\mathcal{K}} = \mathtt{d}^{\mathcal{J}'} = \mathtt{d}^{\mathcal{D}}$.

- $D = D_1 \sqcap D_2$. Then, we have that
$$D^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = (D_1^{\mathcal{K}} \cap D_2^{\mathcal{K}} \cap \Delta^{\mathcal{J}'}) = (D_1^{\mathcal{K}} \cap \Delta^{\mathcal{J}'}) \cap (D_2^{\mathcal{K}} \cap \Delta^{\mathcal{J}'}),$$
then by IH it immediately follows that $D^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = D^{\mathcal{J}'}$.
- $D = \neg C$. Since $\Delta^{\mathcal{K}}$ is by construction the union of the disjoint domains $\Delta^{\mathcal{I}}$ and $\Delta^{\mathcal{J}'}$, it follows that $(\neg C)^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = \Delta^{\mathcal{J}'} \setminus C^{\mathcal{K}}$ which is equal to $\Delta^{\mathcal{J}'} \setminus (C^{\mathcal{K}} \cap \Delta^{\mathcal{J}'})$. Then, by IH $C^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = C^{\mathcal{J}'}$ and hence $(\neg C)^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = (\neg C)^{\mathcal{J}'}$.
- $D = \exists R.C$. The membership of an individual $d \in \Delta^{\mathcal{J}'}$ to the extension of a $\mathcal{SRIQ}(\mathcal{D})$ concept $(\exists R.C)^{\mathcal{K}}$ means that there exists an individual d' connected to d by $R^{\mathcal{K}}$ which is in $C^{\mathcal{K}}$. By construction of $\mathcal{K}$, the projection of $R^{\mathcal{K}}$ on $\Delta^{\mathcal{J}'}$ equals $R^{\mathcal{J}'}$, which in particular means that $d' \in \Delta^{\mathcal{J}'}$. Then, since by IH $d' \in C^{\mathcal{K}}$ iff $d' \in C^{\mathcal{J}'}$, we have that $(\exists R.C)^{\mathcal{K}} \cap \Delta^{\mathcal{J}'} = (\exists R.C)^{\mathcal{J}'}$.
- $D = (\geq n\ S.C)$ and $D = \exists S.\mathsf{Self}$. The proof is analogous to the previous case.

Then, by (7), it follows immediately that $a^{\mathcal{K}} \in C^{\mathcal{K}}$ iff $a^{\mathcal{J}'} \in C^{\mathcal{K}}$ iff $a^{\mathcal{J}'} \in C^{\mathcal{K}} \cap \Delta^{\mathcal{J}'}$ iff $a^{\mathcal{J}'} \in C^{\mathcal{J}'}$, that is, (6) holds also when $\alpha = C(a)$. Now, since $\mathcal{J}'$ is a model of $\mathcal{A} \cap \mathcal{M}$ and α is an arbitrary assertion in $\mathcal{M}$, it follows by (6) that $\mathcal{K}$ is a model of $\mathcal{A} \cap \mathcal{M}$.

Let us now prove that $\mathcal{K}$ is a model of $\mathcal{A} \setminus \mathcal{M}$. By condition 1 in Definition 2, the individual names a occurring in an arbitrary assertion $\alpha \in \mathcal{A} \setminus \mathcal{M}$ are not in $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$. Thus, $a^{\mathcal{K}} = a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, by definition of $\mathcal{K}$. Similarly to the previous case, we have that for all role expressions R and all $\mathcal{SRIQ}(\mathcal{D})$ concepts D, $R^{\mathcal{K}} \cap \Delta^{\mathcal{I}} = R^{\mathcal{I}}$ and $D^{\mathcal{K}} \cap \Delta^{\mathcal{I}} = D^{\mathcal{I}}$ (by definition of $\mathcal{K}$). These two equalities imply that $\mathcal{K} \models \alpha$ iff $\mathcal{I} \models \alpha$. Since $\mathcal{I}$ satisfies α (because, by hypothesis, $\mathcal{I}$ is a model of $\mathcal{KB}$), $\mathcal{K}$ must satisfy α, too. This holds for any $\alpha \in \mathcal{A} \setminus \mathcal{M}$, thus we conclude that $\mathcal{K}$ is a model of $\mathcal{A} \setminus \mathcal{M}$ (which completes also the proof that $\mathcal{K}$ is a model of $\mathcal{KB}$).

It remains to show that $\mathcal{K}$ does not entail q, where by hypothesis $\mathrm{sig}(q) \subseteq \Sigma$. In case q is a GCI or a RIA, $\mathcal{K} \not\models q$ directly derives from $\mathcal{J}' \not\models q$ and the disjoint model property. Then, let q be an assertion. Since the individual constants in Σ are treated like those in $\mathrm{sig}(\mathcal{M})$ in the construction of $\mathcal{K}$, then with a proof analogous to (6), it follows that $\mathcal{K} \models q$ iff $\mathcal{J}' \models q$. Thus, by (2), we conclude that $\mathcal{K}$ does not entail q. This proves that $\mathcal{KB} \not\models q$ (a contradiction). □

Remark 1. If Σ contains no individual names, then every model inseparable Σ-module $\mathcal{M}$ of the TBox $\mathcal{T}$ alone is a c-Σ-module of the entire knowledge base $\mathcal{KB}$. In fact, model inseparability implies point 2 of Definition 2 (note that $\mathrm{sig}(\mathcal{A} \cap \mathcal{M}) = \mathrm{sig}(\emptyset) = \emptyset$), while point 1 is satisfied because $\Sigma \cup \mathrm{sig}(\mathcal{M}) = \emptyset$. Consequently, pure terminological reasoning (i.e. computing concept and role subsumptions) can be carried out in consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases by using only a model inseparable module of the TBox (disregarding the ABox completely).

3.1. Properties of conditional modules

We have proved that conditional modules preserve the answers to CGI, RIA, and assertion queries. Now, we will examine the additional properties that conditional modules satisfy and investigate whether they preserve other types of consequences.

We will begin by examining the inseparability notions described in [9], which include model, fact, implication, and query inseparability.

Model inseparability has already been introduced in Section 2.3. The other notions essentially result in variants of Definition 1, where queries consist of facts (i.e., positive atomic assertions), atomic implications (corresponding to inclusions of the form $A \sqsubseteq B$ or $R \sqsubseteq S$), and Boolean positive existential queries. A Boolean positive existential query is a function-free first-order sentence constructed using only the operators $\exists$, $\wedge$, and $\vee$.

Based on Theorem 1, it is clear that Conditional Σ-modules exhibit both fact and implication inseparability. Conversely, the following example illustrates that conditional Σ-modules do not satisfy model or query inseparability.

Example 2. Let $\mathcal{KB} = \{A(a), A(b), B(a), \neg B(b)\}$ and $\Sigma = \{A, a\}$. It is straightforward to see that the set $\mathcal{M} = \{A(a), B(a)\}$ is a c-Σ-module of $\mathcal{KB}$. Predicate A contains at least two instances in every model of $\mathcal{KB}$ (a and b must be different in order to satisfy $B(a)$ and $\neg B(b)$), while in the least Herbrand model of $\mathcal{M}$, A contains only a (so $\mathcal{M}$ and $\mathcal{KB}$ are not Σ-model inseparable). Note that model-inseparability fails precisely because $A(b)$ and $\neg B(b)$ are not included in $\mathcal{M}$ (i.e. model inseparability may conflict with the computation of minimal deductive modules).

For what concerns query inseparability, let $\mathcal{KB}$ be the knowledge base that consists of a single assertion $A(a)$, let q be the query $\exists x A(x)$, and $\Sigma = \mathrm{sig}(q) = \{A\}$. According to Definition 2, the empty set is a c-Σ-module of the $\mathcal{KB}$; it cannot derive q, while $\mathcal{KB} \models q$. Note that q is a boolean conjunctive query, which means that even this restricted class of queries is not preserved by conditional modules. Later on, however, we will show that a broad subclass of conjunctive queries is preserved. □

The literature on ontology modules has identified other two interesting properties described below: a Σ-module $\mathcal{M}$ of a knowledge base $\mathcal{KB}$ is

1. *self-contained* if $\mathcal{M}$ is also a $(\Sigma \cup \mathrm{sig}(\mathcal{M}))$-module of $\mathcal{KB}$;
2. *depleting* if the empty set is a Σ-module of $\mathcal{KB} \setminus \mathcal{M}$.

The former property guarantees that $\mathcal{M}$ can be used in place of $\mathcal{KB}$ to reason about all the symbols occurring in $\mathcal{M}$ (not only those in Σ). The latter says that $\mathcal{M}$ collects all the axioms that entail nontrivial consequences formulated with $\Sigma \cup \mathrm{sig}(\mathcal{M})$. Although the two properties seem to be related to each other, a depleting deductive module is not necessarily self-contained and vice versa [27, Examples 65 and 66].

In general, c-Σ-modules are not self-contained.[5]

Example 3. Let $\mathcal{KB} = \mathcal{T} = \{A \sqsubseteq B,\ B \sqsubseteq A\}$ and $\Sigma = \{E\}$. The module $\mathcal{M} = \{A \sqsubseteq B\}$ is model inseparable from $\mathcal{T}$ w.r.t. Σ, and contains no individuals; consequently $\mathcal{M}$ is a c-Σ-module of $\mathcal{KB}$. However, it is not self-contained, because it does not entail the subsumption $B \sqsubseteq A$ whose signature is contained in $\mathrm{sig}(\mathcal{M})$. □

However, a natural subclass of c-Σ-modules enjoys self-containment.

Definition 3. A *self-contained conditional* Σ*-module* (sc-Σ-module, for short) of $\mathcal{KB}$ is a c-Σ-module of $\mathcal{KB}$ such that

$$\mathcal{M} \cap \mathcal{T} \text{ and } \mathcal{T} \text{ are model-inseparable w.r.t. } \Sigma \cup \mathrm{sig}(\mathcal{M}).$$

Note that the above condition entails (therefore replaces) point 2 of Definition 2.

Theorem 2. *sc-Σ-modules are self-contained.*

Proof. Let $\mathcal{M}$ be any sc-Σ-module of a given $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$. By Definition 3,

1. there is no individual name $a \in \mathsf{N_I}$ such that $a \in \mathrm{sig}(\mathcal{A} \setminus \mathcal{M}) \cap (\Sigma \cup \mathrm{sig}(\mathcal{M}))$;
2. $\mathcal{M} \cap \mathcal{T}$ and $\mathcal{T}$ are model inseparable with respect to $\Sigma \cup \mathrm{sig}(\mathcal{M})$.

The above two properties clearly hold also if Σ is replaced by $\Sigma' = \Sigma \cup \mathrm{sig}(\mathcal{M})$, because $\Sigma' \cup \mathrm{sig}(\mathcal{M}) = \Sigma \cup \mathrm{sig}(\mathcal{M})$. Thus $\mathcal{M}$ is a Σ'-module of $\mathcal{KB}$, which proves that $\mathcal{M}$ is self-contained. □

In general, sc-Σ-modules are not depleting.

Example 4. Let $\mathcal{KB} = \mathcal{T} = \{A \sqsubseteq B,\ A \sqsubseteq B \sqcap B\}$ and $\Sigma = \{A, B\}$. The module $\mathcal{M} = \{A \sqsubseteq B\}$ is a (self-contained) sc-Σ-module of $\mathcal{KB}$. However, it is not depleting ($\mathcal{KB} \setminus \mathcal{M}$ is equivalent to $\mathcal{M}$). □

In our experiments, though, we will use the following class of c-Σ-modules based on $\bot$-locality, which are both (self-contained) sc-Σ-modules and depleting.

Definition 4. A *conditional, locality based* Σ*-module* (clb-Σ-module for short) of a $\mathcal{SRIQ}(\mathcal{D})$ knowledge base $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$ (where $\mathcal{T}$ and $\mathcal{A}$ are the TBox and the ABox of $\mathcal{KB}$, respectively) is a subset $\mathcal{M} \subseteq \mathcal{KB}$ such that:

1. there is no individual name $a \in \mathsf{N_I}$ such that $a \in \mathrm{sig}(\mathcal{A} \setminus \mathcal{M}) \cap (\Sigma \cup \mathrm{sig}(\mathcal{M}))$;
2. $\mathcal{T} \setminus \mathcal{M}$ is $\bot$-local w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{M})$.

The $\bot$-locality of $\mathcal{T} \setminus \mathcal{M}$ implies the model inseparability of $\mathcal{M} \cap \mathcal{T}$ and $\mathcal{T}$ w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{M})$, therefore clb-$\Sigma$-modules are sc-$\Sigma$-modules by definition. Moreover:

Theorem 3. *clb-Σ-modules are depleting.*

Proof. Let $\mathcal{M}$ be any clb-Σ-module of a given $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$.

The ABox of $\mathcal{KB} \setminus \mathcal{M}$ is $\mathcal{A} \setminus \mathcal{M}$ and its TBox is $\mathcal{T} \setminus \mathcal{M}$. Conditions 1 and 2 in Definition 4 clearly hold also if $\mathrm{sig}(\mathcal{A} \setminus \mathcal{M})$, $\mathcal{T} \setminus \mathcal{M}$, and $\Sigma \cup \mathrm{sig}(\mathcal{M})$ are replaced with the equivalent expressions $\mathrm{sig}((\mathcal{A} \setminus \mathcal{M}) \setminus \emptyset)$, $(\mathcal{T} \setminus \mathcal{M}) \setminus \emptyset$, and $(\Sigma \cup \mathrm{sig}(\mathcal{M})) \cup \mathrm{sig}(\emptyset)$, respectively. Therefore, $\emptyset$ is a clb-Σ-module of $\mathcal{KB} \setminus \mathcal{M}$ (i.e. $\mathcal{M}$ is depleting). □

[5] The following counterexamples to self-containment and depletingness are adapted from [27, Example 65].

3.2. Using conditional modules to answer conjunctive queries

So far we focused on subsumption and instance checking; it is a natural question whether conditional modules can also deal with conjunctive queries of some sort. In Example 2 we have already seen that, in general, conditional modules cannot be applied to conjunctive query answering. For this reason, we consider a restricted class of conjunctive queries that can benefit from conditional modules.

We say that two terms (variables or individual constants) e_1 and e_2 occurring in a CQ $q(\vec{x}) = \exists\vec{y}.conj(\vec{x},\vec{y})$ are *directly connected* if, for some role R, either $R(e_1, e_2)$ or $R(e_2, e_1)$ occur in $conj(\vec{x},\vec{y})$. Then, by *connected* we denote the reflexive and transitive closure of being directly connected.

Definition 5. A conjunctive query q is *anchored* if each of its variables is connected to some individual constant occurring in q.

Many interesting conjunctive queries are anchored.

Example 5. Consider an ontology where the name and age of the employees that work for department x can be retrieved with the following query:

$$q(x, y, z) = \exists v.department(v, x) \wedge name(v, y) \wedge age(v, z)\,.$$

For all individuals dep_0, the instance $q(dep_0, y, z)$ of the above query is an anchored query that returns employee data for dep_0. Similarly, given

$$q'(x, y) = \exists v.age(v, x) \wedge diagnosis(v, y)\,,$$

that retrieves all the pairs (age, diagnosis), $q'(27, y)$ is an anchored query that returns all diagnoses of patients who are 27. □

The following theorem shows that conditional modules preserve the answers of anchored conjunctive queries. Essentially, the individuals needed to answer an anchored conjunctive query q are syntactically connected to the constants occurring in q, and this forces those individuals to be included in any conditional $\mathrm{sig}(q)$-module, to satisfy condition 1 of Definition 2.

Theorem 4. *Let $\mathcal{KB} = \mathcal{T} \cup \mathcal{A}$ be a consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge base, let $\mathcal{M}$ be a c-Σ-module of $\mathcal{KB}$, and let $q(\vec{x})$ be an anchored query such that $\mathrm{sig}(q) \subseteq \Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$. Then, $\mathcal{KB} \models q(\vec{t})$ iff $\mathcal{M} \models q(\vec{t})$, for all tuples of constants $\vec{t}$ such that $\mathrm{sig}(\vec{t}) \subseteq \mathrm{sig}(\mathcal{KB})$.*

Proof. The "if" part is straightforward because $\mathcal{M} \subseteq \mathcal{KB}$. To prove the converse, assume that $\mathcal{KB} \models q(\vec{t})$ and $\mathcal{M} \not\models q(\vec{t})$ (we shall derive a contradiction).

By the second assumption, $\mathcal{M} \cup \{\neg q(\vec{t})\}$ has a model $\mathcal{J}_0$; let $\mathcal{J}$ be the restriction of $\mathcal{J}_0$ to $\Sigma \cup \mathrm{sig}(\mathcal{A} \cap \mathcal{M})$.[6] With the construction used in the proof of Theorem 1, $\mathcal{J}$ can be extended to a model $\mathcal{J}'$ of $(\mathcal{A} \cap \mathcal{M}) \cup \mathcal{T}$, that can be composed with any model $\mathcal{I}$ of $\mathcal{KB}$, disjoint from $\mathcal{J}'$, to obtain a model $\mathcal{K}$ of $\mathcal{KB}$.

Since $\mathcal{K} \models \mathcal{KB}$, the first assumption implies

$$\mathcal{K} \models q(\vec{t})\,, \tag{8}$$

so there must be a mapping σ from q's variables to $\Delta^{\mathcal{K}}$, such that $\sigma(\vec{x}) = \vec{t}$ and $\mathcal{K}, \sigma \models q$. Hereafter, for all terms e occurring in q, let

$$e^{\mathcal{K}}_{\sigma} = \begin{cases} \sigma(e) & \text{if } e \text{ is a variable} \\ e^{\mathcal{K}} & \text{if } e \text{ is a constant.} \end{cases}$$

Now there are two possibilities: either (i) $\mathrm{sig}(\vec{t}) \subseteq \Sigma \cup \mathrm{sig}(\mathcal{M})$, or (ii) some constant t_i in $\vec{t}$ is in the complement of $\Sigma \cup \mathrm{sig}(\mathcal{M})$. In the first case, by construction, $\mathcal{K}$ interprets $q(\vec{t})$ like $\mathcal{J}_0$, therefore $\mathcal{K} \not\models q(\vec{t})$, which contradicts (8). Then case (ii) must hold. Since q is anchored, the variable x_i bound to t_i (where $t_i \notin \Sigma \cup \mathrm{sig}(\mathcal{M})$) is connected to some constant $a \in \mathrm{sig}(q) \subseteq \Sigma \cup \mathrm{sig}(\mathcal{M})$, therefore:

1. $t_i^{\mathcal{K}} \in \Delta^{\mathcal{I}}$ and $a^{\mathcal{K}} \in \Delta^{\mathcal{J}'}$ (by construction of $\mathcal{K}$);
2. so q must contain an atom $r(u, w)$ such that one of $u^{\mathcal{K}}_{\sigma}$ and $w^{\mathcal{K}}_{\sigma}$ belongs to $\Delta^{\mathcal{I}}$, while the other belongs to $\Delta^{\mathcal{J}'}$, that is, either $(u^{\mathcal{K}}_{\sigma}, w^{\mathcal{K}}_{\sigma}) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}'}$ or $(u^{\mathcal{K}}_{\sigma}, w^{\mathcal{K}}_{\sigma}) \in \Delta^{\mathcal{J}'} \times \Delta^{\mathcal{I}}$.

By (8), it should be $(u^{\mathcal{K}}_{\sigma}, w^{\mathcal{K}}_{\sigma}) \in r^{\mathcal{K}}$, however, by construction of $\mathcal{K}$, $r^{\mathcal{K}}$ contains no pairs in $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}'}$ or $\Delta^{\mathcal{J}'} \times \Delta^{\mathcal{I}}$ (a contradiction). □

[6] Note that the constants in $\vec{t}$ are not necessarily contained in $\Sigma \cup \mathrm{sig}(\mathcal{M})$.

3.3. Concrete module extractors

We now provide a constructive definition for a class of conditional module extractors. Since various algorithms for extracting model inseparable modules (as required by condition 2 in Definition 2) have been developed,[7] here we propose a parametric approach.

Let mx be any module extractor for TBoxes which *enjoys model inseparability*, that is, for all signatures Σ and all TBoxes $\mathcal{T}$, $mx(\Sigma, \mathcal{T})$ and $\mathcal{T}$ are Σ-model inseparable. Then, mx can be used to design a conditional module extractor mx-cMod as follows.

Definition 6. For all $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases $\mathcal{KB}$ with TBox $\mathcal{T}$ and ABox $\mathcal{A}$, and all signatures Σ, let:

$$\mathcal{A}' = \bigcup_{i>0} \mathcal{A}_i, \text{ where}$$
$$\mathcal{A}_0 = \emptyset$$
$$\mathcal{A}_{i+1} = \{\alpha \in \mathcal{A} \mid \exists a \in \mathsf{N_I} : a \in \mathrm{sig}(\alpha) \cap (\mathrm{sig}(\mathcal{A}_i) \cup \Sigma)\}\,;$$
$$\mathcal{H}' = mx(\mathrm{sig}(\mathcal{A}') \cup \Sigma, \mathcal{T})\,.$$

Finally, let mx-cMod$(\Sigma, \mathcal{KB}) = \mathcal{H}' \cup \mathcal{A}'$. □

In the above construction, the non-decreasing sequence $\langle \mathcal{A}_i \rangle_i$ eventually reaches a fixpoint $\mathcal{A}'$ within at most $|\mathcal{A}|$ steps. $\mathcal{A}'$ encompasses the assertions from $\mathcal{KB}$ that are syntactically linked to the individual names in Σ through a chain of assertions. Thus, by construction, it satisfies condition 1 of Definition 2.

The intensional part of the c-Σ-module is constructed by mx; its model inseparability property entails condition 2 of Definition 2.

The following theorem formalizes the above considerations:

Theorem 5. *If $\mathcal{KB}$ is a $\mathcal{SRIQ}(\mathcal{D})$ knowledge base and mx is a module extractor which enjoys model inseparability, then mx-cMod(Σ, $\mathcal{KB}$) is a c-Σ-module of $\mathcal{KB}$. Moreover, if $mx(\Sigma, \mathcal{KB})$ is self-contained, then mx-cMod(Σ, $\mathcal{KB}$) is a sc-Σ-module of $\mathcal{KB}$.*

Proof. Let $\mathcal{T}$, $\mathcal{A}$, $\mathcal{A}'$, $\mathcal{A}_i$, and $\mathcal{H}'$ be defined as in Definition 6 ($i \geq 0$). First, we show by induction on i that for all $i \geq 0$, $\mathcal{A}_i \subseteq \mathcal{A}_{i+1}$. The base case $i = 0$ is trivial. For the induction step, assume that $i > 0$. By definition we have that

$$\mathcal{A}_i = \{\alpha \in \mathcal{A} \mid a \in \mathrm{sig}(\alpha) \cap (\mathrm{sig}(\mathcal{A}_{i-1}) \cup \Sigma), \text{ for some } a \in \mathsf{N_I}\}$$
$$\mathcal{A}_{i+1} = \{\alpha \in \mathcal{A} \mid a \in \mathrm{sig}(\alpha) \cap (\mathrm{sig}(\mathcal{A}_i) \cup \Sigma), \text{ for some } a \in \mathsf{N_I}\}\,.$$

The induction hypothesis entails that $\mathrm{sig}(\mathcal{A}_{i-1}) \subseteq \mathrm{sig}(\mathcal{A}_i)$, from which it immediately follows that $\mathcal{A}_i \subseteq \mathcal{A}_{i+1}$. Since $\mathcal{A}$ is finite, the sequence $\{\mathcal{A}_i\}_{i \geq 0}$ eventually reaches a fixpoint $\mathcal{A}_m$, for some $m \geq 0$. Then, we have $\mathcal{A}' = \mathcal{A}_m$ and $\mathcal{A}_m = \mathcal{A}_{m+1}$.

Now assume per absurdum that mx-cMod$(\Sigma, \mathcal{KB})$ does not satisfy condition 1 of Definition 2. Since $\mathcal{SRIQ}(\mathcal{D})$ TBoxes are nominal free, this means that there exists an individual name a in $\mathrm{sig}(\mathcal{A}') \cup \Sigma$ that occurs also in some axiom α in $\mathcal{A} \setminus \mathcal{A}'$. However, since $\mathcal{A}' = \mathcal{A}_m$, this means by construction that α should be added to $\mathcal{A}_{m+1}$, which contradicts $\mathcal{A}_m = \mathcal{A}_{m+1}$. This proves that condition 1 is satisfied.

In order to prove condition 2 of Definition 2, note that mx-cMod$(\Sigma, \mathcal{KB}) \cap \mathcal{T} = \mathcal{H}' = mx(\Sigma \cup \mathrm{sig}(\mathcal{A}'), \mathcal{T})$, which – by hypothesis – is model inseparable from $\mathcal{T}$ w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A}') = \Sigma \cup \mathrm{sig}(\mathcal{A} \cap mx\text{-cMod}(\Sigma, \mathcal{KB}))$. Then condition 2 is satisfied, too (which proves that mx-cMod$(\Sigma, \mathcal{KB})$ is a c-Σ-module of $\mathcal{KB}$).

Finally, if $mx(\Sigma \cup \mathrm{sig}(\mathcal{A}), \mathcal{T})$ is self-contained, then mx-cMod$(\Sigma \cup \mathrm{sig}(\mathcal{A}')) \cap \mathcal{T}$ (which equals $\mathcal{H}'$) is model inseparable from $\mathcal{T}$ w.r.t. $\Sigma \cup \mathrm{sig}(\mathcal{A}') \cup \mathrm{sig}(\mathcal{H}') = \Sigma \cup \mathrm{sig}(mx\text{-cMod}(\Sigma, \mathcal{KB}))$, therefore mx-cMod$(\Sigma, \mathcal{KB})$ is a sc-Σ-module of $\mathcal{KB}$. □

From Theorems 1, 2, and 5 it follows that:

Corollary 6. *For all consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases and all extractors mx that enjoy model inseparability, mx-cMod(Σ, $\mathcal{KB}$) is a Σ-module of $\mathcal{KB}$. Moreover, if mx yields self-contained modules, then mx-cMod(Σ, $\mathcal{KB}$) is self-contained, too.*

For example, since the locality-based module extractor $\bot$-Mod enjoys model inseparability and is self-contained, it can be used as mx in our construction to obtain a sc-Σ-module extractor, denoted by $\bot$-*cMod*. By definition, $\bot$-cMod is also a clb-Σ-module extractor, therefore it constructs self-contained and depleting modules. Smaller self-contained and depleting modules can be obtained with $\top\bot^*$-*cMod*, the conditional module extractor obtained with $mx = \top\bot^*$-Mod. Finally, we denote with χ_m-cMod the conditional extractor obtained with $mx = \chi_\mathrm{m}$ (the only PrisM extractor that enjoys model inseparability, and can be used to instantiate Definition 6). The output of χ_m is neither self-contained nor depleting, in general [9, Example 62], so the output of χ_m-cMod does not enjoy these properties, either.

As a concluding remark, we note that the concept of Σ-modules (Definition 1) is inherently transitive. This implies that different module extractors can always be applied in series. Consequently, in some cases, $\top\bot^*$-Mod and $\top\bot^*$-cMod can be combined to address

[7] Cf. Sec. 2.3.

Table 2
Properties of BioPortal's ontologies.

Acronym	C	R	I	TBox	ABox
INSECTH	101	0	112	101	114
ADMIN	42	42	62	176	119
SURGICAL	50	13	62	77	122
TRANS	45	36	62	157	129
NMOSP	1483	0	157	1484	157
NEMO	1851	93	120	2692	182
MWLA	32	9	228	43	230
CCON	85	28	137	219	364
BOF	188	99	315	599	768
NMOBR	2618	2	1304	2967	1312
HAO	2360	4	2757	4659	5514
OntoAD	5899	178	2465	8390	10889
MONO	18	37	3819	71	14220
PDO_CAS	380	83	12153	869	30887
CU-VO	11	19	7321	34	70697
NCIT	118941	108	46839	167024	114399
BIOMODEL	187518	61	220948	439249	220948

knowledge bases that involve nominals. Specifically, $\top\bot^*$-Mod is applied initially, and if the resulting module belongs to $\mathcal{SRIQ}(\mathcal{D})$, then $\top\bot^*$-cMod can further reduce its size.

4. Experimental evaluation

We have implemented the conditional module extractors $\bot$-cMod, $\top\bot^*$-cMod, and χ_m-cMod in Java, leveraging the OWL API's implementation (of $\bot$-Mod and $\top\bot^*$-Mod), and PrisM, respectively. All experiments were conducted on Intel i7 6700 with 16 GB RAM DDR4, running Guix System 1.4.0.

We started the experimental evaluation on a test set acquired from a snapshot of the NCBO BioPortal ontology repository[8] curated by Matentzoglu and Parsia [28] and recently adopted in [29]. From these 438 ontologies we selected those that are consistent, contain no nominals, and have a reasonable ABox size, i.e. at least 100 ABox axioms. The resulting 17 ontologies are illustrated in Table 2. Its columns show: the ontology acronym used in the NCBO BioPortal repository, the number of classes (C), the number of roles (R), the number of individuals (I), the number of TBox axioms (TBox), and the number of ABox axioms (ABox).

We have considered two kinds of randomly generated initial signatures, according to the experimental methodology first proposed in [30]:

- *genuine signatures*, that correspond to the signature of individual axioms occurring in the ontologies, and
- *random signatures*, defined simply as random subsets of the ontology signature.

Since many genuine signatures had only one symbol, we added the signature of a second axiom, when needed, to have at least two symbols in each initial signature.

Moreover, following the approach of [9] for random signature generation, we have chosen by default a subset of 0.1% symbols (predicates or individual names) of the total signature of a given ontology and then increased it by up to two orders of magnitude in cases where the resulting signatures contained less than 15 symbols and thus were too small to provide additional information w.r.t. genuine signatures.

Table 3 presents the average size of the generated genuine and random signatures, along with the average number of individuals they contain. Genuine signatures generally consist of 2 to 3 symbols. The percentage of individuals selected is influenced by the relative size of the ABox. For instance, in an ontology like NEMO, where the ABox is relatively small compared to the TBox, the percentage of individuals is around 3%. Conversely, for ontologies such as MONO, PDO_CAS, or CU-VO, individuals constitute approximately 70% of the entire signature. Random signatures are generally one or two orders of magnitude larger than genuine signatures and exhibit greater variability, as they depend on the size of the input ontology. The percentage of individuals approximately follows the same trend as genuine signatures. Note that in MONO, PDO_CAS, and CU-VO almost the entire signature consists of individuals.

For each kind of signature and each ontology in Table 2, we run 400 different experiments. In Tables 4 and 5 we compare the standard extractors $\bot$-Mod and $\top\bot^*$-Mod of the OWL API with their corresponding conditional module extractors $\bot$-cMod and $\top\bot^*$-cMod, using genuine and random signatures, respectively. Note that BioPortal ontologies frequently use datatypes, which are not supported by PrisM. Therefore, χ_m and χ_m-cMod have been tested with a separate test suite (see below). For each ontology, the upper row represents the *mean/standard deviation/median* of the modules' size, whereas the second row shows the same statistical measures for the execution time, in milliseconds. Execution time has been measured with System.currentTimeMillis(), so times below 1 ms

[8] https://bioportal.bioontology.org/.

Table 3
Composition of the signatures used in the experiments on BioPortal ontologies.

Acronym	Genuine signatures		Random signatures	
	Signature length	Num. Individuals	Signature length	Num. Individuals
INSECTH	2.3	0.8	22	12
ADMIN	2.3	0.8	15	7
SURGICAL	2.2	1.1	13	7
TRANS	2.3	0.9	15	7
NMOSP	2.2	0.1	164	16
NEMO	2.8	0.1	206	12
MWLA	2.2	1.3	29	25
CCON	2.5	0.9	26	14
BOF	2.4	0.9	62	34
NMOBR	2.3	0.5	390	129
HAO	2.7	0.8	483	260
OntoAD	2.7	1.1	482	139
MONO	2.5	1.9	384	381
PDO_CAS	2.8	1.8	219	211
CU-VO	2.5	2.0	529	528
NCIT	2.9	0.9	166	45
BIOMODEL	2.5	0.6	410	222

Table 4
BioPortal - module size and time execution statistics for genuine signatures.

Acronym	$\bot$-Mod	$\top\bot^*$-Mod	$\bot$-cMod	$\top\bot^*$-cMod	$\mathcal{A}'$
INSECTH	212.0/0.2/212 0.7/0.8/1	27.3/46.8/3 0.6/0.7/1	6.7/2.2/6 0.4/0.8/0	1.3/0.7/1 0.5/0.8/0	0.8/0.7/1
ADMIN	88/7/86 0.76/1.02/1	82/6/80 0.83/1.08/1	9/6/8 0.80/1.08/1	7/5/5 0.73/0.89/1	1.7/1.7/2
SURGICAL	86/1/86 0.683/0.832/1	83/1/83 0.677/0.998/1	6/3/6 0.540/0.859/0	5/3/5 0.596/1.033/0	2.5/1.8/3
TRANS	103/3/102 0.77/0.81/1	94/6/91 0.90/1.07/1	9/5/8 0.76/1.12/1	8/6/6 0.75/1.07/1	2.0/1.8/2
NMOSP	542/2/542 3.06/1.69/3	18/62/2 2.71/1.40/2	35/10/32 1.95/1.16/2	1/2/1 1.91/0.90/2	0.1/0.4/0
NEMO	602/6/600 5.5/1.8/5	551/6/548 6.0/2.0/6	130/56/107 4.8/1.6/4	127/54/103 4.9/1.9/5	0.3/1.2/0
MWLA	255.1/0.4/255 0.83/0.80/1	27.6/65.0/10 0.64/0.84/1	2.8/1.1/2 0.42/0.68/0	1.4/0.5/1 0.44/0.68/0	1.3/0.6/1
CCON	450/6/447 1.16/0.90/1	438/10/434 1.32/0.94/1	65/26/77 0.97/1.00/1	52/28/64 0.91/0.99/1	3.5/3.0/4
BOF	956/3/954 2.3/1.5/2	925/5/922 2.8/1.5/2	30/51/8 1.7/1.3/1	25/51/3 1.5/1.2/1	23/50/2
NMOBR	3074/2/3073 8.54/2.92/8	1706/80/1679 9.67/2.94/9	25/18/26 4.13/1.87/4	4/9/1 4.07/1.76/4	0.5/0.6/0
HAO	7393/2/7392 17.3/3.9/16	5379/1225/4635 20.0/4.8/19	53/20/52 8.4/2.4/8	10/15/2 8.9/2.5/9	1.5/1.3/2
OntoAD	10941/34/10922 30.77/6.84/30	292/515/12 28.96/6.63/28	259/619/60 15.64/5.05/15	233/626/10 15.62/4.89/15	232/627/10
MONO	14291.0/0.0/14291 22.5/4.3/21	14265.0/0.0/14265 29.4/6.2/28	29.0/12.0/29 12.0/3.4/11	28.6/12.5/29 11.8/3.6/11	11/7/10
PDO_CAS	31284/1/31284 64.9/9.3/63	26706/8538/29394 79.1/13.4/79	408/236/318 27.7/5.7/26	258/229/164 28.0/6.0/26	126/224/26
CU-VO	70724/0.0/70724 114.42/14.66/112	20760/0.0/20760 131.72/14.95/129	74/84/48 61.64/7.22/60	74/84/48 61.57/10.42/59	48/84/22
NCIT	114580/861/114407 760/107/744	32604/47921/109 758/103/745	1993/4074/55 512/61/496	1991/4075/55 500/60/488	1812/4059/0
BIOMODEL	221693/2055/221005 1919/274/1914	27575/44087/768 1794/270/1757	746/2055/58 1301/147/1284	699/2055/2 1363/148/1349	0.6/0.7/0

Table 5
BioPortal - module size and time execution statistics for random signatures.

Acronym	⊥-Mod	⊤⊥*-Mod	⊥-cMod	⊤⊥*-cMod	$\mathcal{A}'$
INSECTH	212/0/212 0.73/0.77/1	86/74/49 0.75/0.76/1	45/7/45 0.52/0.79/0	25/13/22 0.63/0.94/1	12/3/12
ADMIN	110/13/109 0.85/1.15/1	103/12/102 0.92/0.88/1	51/15/52 0.95/1.78/1	49/14/50 1.08/2.15/1	14/5/14
SURGICAL	93/3/93 0.63/0.70/1	91/3/90 0.65/0.81/1	32/9/32 0.63/1.30/1	31/10/32 0.57/0.79/1	14/6/14
TRANS	118/8/117 0.94/1.56/1	112/11/111 0.83/0.74/1	50/12/49 0.85/1.12/1	49/12/48 0.95/1.60/1	15/6/15
NMOSP	688/14/687 3.8/2.2/3	633/97/672 4.3/2.3/4	366/21/366 2.7/1.5/2	330/59/348 2.9/1.7/3	16/4/16
NEMO	1034/37/1037 7.3/2.8/7	986/41/986 8.2/2.7/7	839/48/841 6.8/2.1/6	827/48/830 7.8/2.4/7	23/7/23
MWLA	256/2/255 1.12/1.296/1	44/65/25 0.84/1.21/1	42/6/42 0.74/1.48/1	26/6/25 0.71/0.82/1	25/4/25
CCON	485/15/484 1.6/1.8/1	482/17/483 1.8/2.2/1	159/21/159 1.3/1.7/1	156/23/158 1.4/1.7/1	43/13/43
BOF	991/11/991 2.9/1.7/3	972/12/972 3.8/2.6/3	224/59/212 2.6/1.8/2	198/59/186 2.5/1.7/2	100/54/84
NMOBR	3230/18/3231 10/3/10	2076/160/2030 12/3/11	944/82/951 6/2/6	784/77/791 7/3/7	130/15/131
HAO	7728/90/7750 19/5/18	5299/910/4995 23/4/22	1866/440/1974 13/3/13	1860/451/1972 14/4/13	520/134/550
ONTOAD	12483/1255/13456 34/5/33	1861/1767/2321 36/7/37	2772/2342/4270 20/4/20	2318/2103/3393 21/6/21	1192/1146/1598
MONO	14291/0/14291 25.8/5.6/23	14268/8/14265 31.8/4.9/31	1817/197/1833 15.9/3.7/15	1817/197/1833 16.4/3.8/16	1748/195/1764
PDO_CAS	31295/23/31284 66.7/12.1/64	8692/13621/0 60.9/19.9/50	1156/1845/335 30.8/6.5/28	1010/1854/185 31.4/7.0/29	842/1802/41
CU-VO	70724/1/70724 118.7/14.3/115	20760/1/20760 136.7/15.2/134	5998/3710/8106 70.6/8.7/71	5998/3710/8106 71.3/9.2/71	5972/3710/8080
NCIT	121092/988/121253 988/87/981	6956/5261/6864 1008/100/1001	6966/1388/7024 676/49/678	6966/1388/7024 668/50/668	273/969/137
BIOMODEL	274628/7568/275109 2169/152/2144	52748/9424/52899 2108/171/2076	53902/7569/54387 1442/130/1415	52530/7579/52965 1505/143/1468	222/15/222

are sometimes rounded to 0 by Java. The last column shows the *mean/standard deviation/median* size of the filtered ABox $\mathcal{A}'$ (cf. Definition 6).

Column $\mathcal{A}'$, in Tables 4 and 5, shows that conditional modules effectively filter out a significant portion of the knowledge base's ABox, which is reduced by one or more orders of magnitude. Accordingly, the average size of the conditional modules extracted by ⊥-cMod and ⊤⊥*-cMod is mostly one or two orders of magnitude smaller, compared to their OWLAPI counterparts ⊥-Mod and ⊤⊥*-Mod. The only cases where conditional modules are larger than their counterparts can be found in Table 5 (random signatures) for OntoAD and NCIT (the latter's conditional modules are only slightly larger than OWLAPI's). This may happen when $\mathcal{A}'$ contains too many irrelevant predicates; they are forced by Definition 6 in the initial signature fed to mx, thus the module's TBox $\mathcal{H}'$ may contain more irrelevant axioms than its non-conditional counterpart. Informally speaking, the above situation becomes more likely as the logical connections between individuals and predicates in the initial signature get looser.

A quick comparison of Tables 4 and 5 clearly shows that, using random signatures, all module extractors tend to produce larger modules. This is probably due to a combination of two factors: first, the initial signatures are larger by design; second, fully random signatures more frequently contain sets of logically unrelated symbols, which leads extractors to include in their modules numerous knowledge base components. Loosely connected signatures seem to disadvantage conditional extractors, as we have already pointed out. Nonetheless, in most cases, conditional module extractors still exhibit a clear improvement over standard ones, although the extent of this improvement appears somewhat diminished compared to the results with genuine signatures.

Regarding average execution time, the conditional extractors are in most tests faster than OWLAPI's, which implies that the time spent calculating $\mathcal{A}'$ in Definition 6 is compensated during the extraction of the TBox's module $\mathcal{H}'$. There are two exceptions: on ADMIN and TRANS, using random signatures, ⊥-cMod and ⊤⊥*-cMod are marginally slower than their counterparts (but construct considerably smaller modules). It is not surprising that conditional extractors are relatively less performant, using random signatures, because such signtures yield larger $\mathcal{A}'$, therefore the overhead of ABox filtering has greater impact on performance.

Table 6
Properties of the second test set.

ID	Name	C	R	I	TBox	ABox
00001	ACGT-v1.0	1769	263	61	5307	109
00024	DOLCE	290	317	42	1865	137
00347	LUBM-one-uni	43	29	17174	108	67464
00350	OBI	2888	82	89	10563	231
00351	AERO	296	63	25	599	33
00354	NIF-gross-anatomy	4104	65	102	6788	111
00463	Fly-anatomy-XP	8022	25	22079	20005	22079
00775	Roberts-family	101	89	405	522	1091

We designed a second test suite in order to compare PrisM-based and locality-based module extractors. Among PrisM's modules, we consider only χ_{m}, because it is the only PrisM extractor that: (i) covers all the queries preserved by conditional extractors, and (ii) enjoys model inseparability, a prerequisite of mx in Definition 6. Regarding locality-based extractors, we do not report the results for $\bot$-Mod and $\bot$-cMod here, as they consistently yield larger modules compared to their counterparts $\top\bot^*$-Mod and $\top\bot^*$-cMod, without offering a significant speed-up.

The comparison is carried out on the test ontologies used to evaluate PrisM,[9] where the ontologies with empty ABoxes are excluded from the test set because, on such ontologies, $\top\bot^*$-cMod and χ_{m}-cMod are equivalent to $\top\bot^*$-Mod and χ_{m}, respectively, which have already been extensively compared in [9]. Specifically, the ontologies used in our tests are described in Table 6. Since conditional module extractors are correct only for $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases, the few axioms involving nominals have been removed.

In order to fully understand our experiments, it should be observed that PrisM does not accept individuals in the input signature Σ. This means that PrisM cannot be told which individuals are relevant: if $P \in \Sigma$, then χ_{m}-modules preserve all atomic consequences $P(c_1, \ldots, c_n)$, for all the individuals $c_1, \ldots, c_n$ occurring in the knowledge base. To accomodate PrisM's restrictions, individuals are removed from signatures before feeding them to χ_{m}.

Note that χ_{m}-cMod forces PrisM to discard irrelevant individuals by feeding χ_{m} with the restricted ABox $\mathcal{A}'$ of Definition 6; thus, by comparing χ_{m} with χ_{m}-cMod, one can assess the benefits of ignoring irrelevant individuals in PrisM.

Tables 8 and 9 (which have the same structure as Tables 4 and 5) present the experimental outcomes obtained with the genuine and random signatures illustrated in Table 7, respectively.

The first observation is that also in this test set, the average size of the modules extracted by OWLAPI's $\top\bot^*$-Mod is almost always larger than the average size of the conditional modules constructed by $\top\bot^*$-cMod. The only exception is 00463 (Fly-anatomy-XP) using random signatures. This is a peculiar ontology:

- each individual occurs once in the ABox, in a class assertion;
- the signatures of the ABox and the TBox are disjoint, so the two parts of the ontology are logically unrelated; consequently, the non-conditional extractors discard all of the ABox when the initial signature Σ contains only TBox predicates;
- the probability of selecting an ABox predicate from $\mathrm{sig}(00463)$ is about $5 \cdot 10^{-3}$, thus most random signatures contain only TBox predicates.

As a consequence, $\top\bot^*$-Mod and χ_{m} almost always discard the entire ABox, while their conditional counterparts include the assertions involving the individuals in Σ. Accordingly, the average difference between the size of conditional modules and their non-conditional counterpart almost exactly equals the average size of $\mathcal{A}'$. This is a limit case among the scenarios where the initial signatures Σ frequently contain individuals which are logically unrelated with the predicates in Σ. The above discussion provides a deeper insight of why this situation is particularly unfavorable to conditional module extractors.

Concerning time, $\top\bot^*$-Mod is sometimes negligibly faster, and frequently slower than $\top\bot^*$-cMod.

Similar relationships hold between χ_{m} and its conditional version χ_{m}-cMod. The average size of the latter's modules is almost always smaller than the average size of the former's, with the only exception of ontology 00463 using random signatures, which we have already commented. Moreover, χ_{m} is almost always slower than χ_{m}-cMod; only in one case (ontology 00024, random signatures) χ_{m} is (negligibly) faster.

The above comparison of $\top\bot^*$-Mod and χ_{m} with their conditional versions indicates that the preliminary removal of irrelevant individuals (i.e. the computation of $\mathcal{A}'$) yields tangible benefits in terms of module size, with few exceptions. Morevoer, while $\top\bot^*$-Mod and $\top\bot^*$-cMod are comparably fast, in the case of χ_{m} and χ_{m}-cMod we also observed a more systematic improvement of computation time. As expected, the higher the complexity of the module extractor mx in Definition 6, the greater the benefits of the preliminary, fast ABox filtering in terms of time.

It is interesting to compare χ_{m} with $\top\bot^*$-cMod. The latter is almost always superior with two exceptions, both of which occur in Table 9 (random signatures): on ontology 00001, χ_{m} returns marginally smaller modules, but it is remarkably slower; the other exception is 00463, which we extensively discussed.

[9] Available at https://krr-nas.cs.ox.ac.uk/2015/jair/PrisM/testOntologies.zip in PrisM's normal form [9].

Table 7
Composition of the signatures used in the experiments on the second test set.

ID	Genuine signatures		Random signatures	
	Signature length	Num. Individuals	Signature length	Num. Individuals
00001	2.1	0.1	209	6
00024	2.3	0.2	66	4
00347	2.8	1.9	61	61
00350	2.2	0.03	304	9
00351	2.4	0.1	39	2
00354	2.3	0.03	413	10
00463	2.3	0.5	31	23
00775	3.0	1.6	60	41

Table 8
A comparison between PrisM and conditional module extractors - genuine signatures.

ID	$\top\bot^*$-Mod	$\top\bot^*$-cMod	χ_m	χ_m-cMod	$\mathcal{A}'$
00001	614/50/580 8/3/8	356/193/249 10/3/10	558/48/526 402/88/381	322/177/225 237/127/179	0.1/0.6/0
00024	738/12/735 3.0/1.4/3	398/225/532 3.5/1.6/3	719/12/717 389.8/41.6/380	676/1/676 286.3/187.0/332	0.8/2.7/0
00347	32831/27487/16634 143/30/152	124/206/48 62/8/61	30538/29094/16634 6659/6190/2382	124/206/48 91/27/87	101/203/25
00350	646/41/635 31/6/30	197/77/223 33/5/31	593/41/582 432/150/377	180/69/194 166/54/175	0.1/0.7/0
00351	100/16/93 1.0/0.9/1	17/25/5 1.1/0.9/1	92/12/88 61.8/41.6/51	14/22/5 26.2/20.7/19	0.2/0.7/0
00354	467/423/222 7/2/7	347/423/102 11/3/10	457/422/212 222/201/109	346/422/101 197/199/83	0.1/0.3/0
00463	4631/4644/4513 55/7/54	94/194/1 42/9/40	4629/4646/4513 1188/1324/720	93/190/1 111/103/66	0.5/0.5/1
00775	1496/6/1495 3/2/3	268/54/291 2/2/2	1495/6/1494 656/64/638	268/54/291 171/40/178	12/9/12

Table 9
A comparison between PrisM and conditional module extractors - random signatures.

ID	$\top\bot^*$-Mod	$\top\bot^*$-cMod	χ_m	χ_m-cMod	$\mathcal{A}'$
00001	1497/61/1500 11/59/6	1449/64/1453 16/82/9	1426/61/1431 940/157/887	1385/64/1388 917/148/872	13/5/13
00024	1095/54/1097 5/21/3	961/58/963 5/4/4	1090/54/1092 594/93/570	957/58/960 598/136/553	18/10/17
00347	1573/9628/0 164/514/82	358/1430/157 59/7/56	1539/9318/0 452/2172/102	358/1430/157 93/31/87	402/1702/112
00350	2597/193/2609 41/182/24	2297/195/2309 46/110/35	2556/194/2567 1559/184/1538	2293/196/2303 1467/220/1440	25/8/24
00351	265/39/266 1/3/1	232/41/232 2/3/1	240/38/242 183/84/157	209/40/210 152/64/134	4/3/4
00354	2356/278/2406 16/90/8	2257/282/2309 16/55/10	2336/277/2388 1244/177/1253	2247/281/2299 1213/175/1219	16/5/16
00463	799/370/750 92/380/43	821/370/775 54/131/37	792/365/743 514/216/493	814/365/769 498/197/484	23/5/23
00755	1525/20/1521 4/21/3	622/53/622 3/3/2	1523/19/1519 717/127/670	621/53/621 319/105/284	213/35/210

The above comparison indicates that the benefits of removing irrelevant individuals in $\top\bot^*$-cMod essentially override those of the finer-grained (and more expensive) axiom analysis carried out by χ_m, which makes $\top\bot^*$-cMod a very interesting alternative to χ_m.

The comparison of $\top\bot^*$-cMod and χ_m-cMod leads to similar conclusions: it is evident that, after the preliminary removal of irrelevant individuals, the size of locality-based modules is very close to that of the significantly more expensive modules extracted by χ_m; thus, when speed is crucial, $\top\bot^*$-cMod constitutes an appealing alternative to χ_m-cMod.

5. Summary and conclusions

We introduced the notion of conditional module and proved that, for all consistent $\mathcal{SRIQ}(\mathcal{D})$ knowledge bases, such modules preserve subsumption and assertion checking, as well as anchored CQ answering. Then we introduced a general method to turn any extractor mx of model-inseparable modules into a conditional module extractor, by removing some certainly irrelevant ABox assertions before calling mx.

Extensive experiments with $mx \in \{\bot\text{-Mod}, \top\bot^*\text{-Mod}, \chi_{\mathsf{m}}\}$ indicate that, when the ABox is nonempty, such ABox filtering almost always improves the performance of mx in terms of module size. When $mx = \chi_{\mathsf{m}}$, computation time is reduced, too; when mx is a locality-based module extractor, no significant overhead has been observed, and often the conditional extractor is faster. These properties make conditional modules a promising technique to improve module extraction over consistent $\mathcal{SRIQ}$ knowledge bases with nonempty ABoxes.

In particular, $\top\bot^*$-cMod – besides returning self-contained and depleting modules – constitutes an appealing tradeoff between the speed of locality-based extractors and the small size of χ_{m} modules. The output size and computation time of $\top\bot^*$-cMod lie mostly between those of $\top\bot^*$-Mod and χ_{m}; module size is frequently close to that of χ_{m}, while computation time is closer to that of $\top\bot^*$-Mod.

Conditional modules are not model inseparable from the knowledge base. While this property affects the preservation of second-order sentences, it is the key to obtaining smaller modules, as shown in Example 2.

Conditional module extractors may be easily refined to deal with peculiar ontologies such as 00463. When the signatures of ABox and TBox are disjoint, and all the predicates in the initial signature Σ are TBox predicates (two conditions that can be very quickly tested), then the ABox can be entirely dropped.

We do not currently know whether the parametric algorithmic approach of Definition 6 can be extended to extractors that do not enjoy model inseparability. This is an interesting subject for further research.

CRediT authorship contribution statement

Piero Andrea Bonatti: Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Francesco Magliocca:** Writing – review & editing, Validation, Software, Investigation, Formal analysis, Data curation. **Iliana Mineva Petrova:** Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Luigi Sauro:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is taken from public repositories. A .jar file with software and data can be requested from the corresponding authors.

References

[1] B.C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, Modular reuse of ontologies: theory and practice, J. Artif. Intell. Res. (JAIR) 31 (2008) 273–318, https://doi.org/10.1613/jair.2375.
[2] E. Jiménez-Ruiz, B.C. Grau, U. Sattler, T. Schneider, R.B. Llavori, Safe and economic re-use of ontologies: a logic-based methodology and tool support, in: S. Bechhofer, M. Hauswirth, J. Hoffmann, M. Koubarakis (Eds.), The Semantic Web: Research and Applications, 5th European Semantic Web Conference, Proceedings, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, in: Lecture Notes in Computer Science, vol. 5021, Springer, 2008, pp. 185–199, https://doi.org/10.1007/978-3-540-68234-9_16.
[3] B. Suntisrivaraporn, G. Qi, Q. Ji, P. Haase, A modularization-based approach to finding all justifications for OWL DL entailments, in: J. Domingue, C. Anutariya (Eds.), The Semantic Web, 3rd Asian Semantic Web Conference, Proceedings, ASWC 2008, Bangkok, Thailand, December 8-11, 2008, in: Lecture Notes in Computer Science, vol. 5367, Springer, 2008, pp. 1–15, https://doi.org/10.1007/978-3-540-89704-0_1.
[4] M. Ludwig, Just: a tool for computing justifications w.r.t. ELH ontologies, in: S. Bail, B. Glimm, E. Jiménez-Ruiz, N. Matentzoglu, B. Parsia, A. Steigmiller (Eds.), Informal Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation (ORE 2014) Co-Located with the Vienna Summer of Logic (VSL 2014), Vienna, Austria, July 13, 2014, in: CEUR Workshop Proceedings, vol. 1207, 2014, pp. 1–7, CEUR-WS.org, http://ceur-ws.org/Vol-1207/paper_2.pdf.
[5] A.A. Romero, B.C. Grau, I. Horrocks, More: modular combination of OWL reasoners for ontology classification, in: P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J.X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, E. Blomqvist (Eds.), The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Proceedings, Part I, Boston, MA, USA, November 11-15, 2012, in: Lecture Notes in Computer Science, vol. 7649, Springer, 2012, pp. 1–16, https://doi.org/10.1007/978-3-642-35176-1_1.
[6] D. Tsarkov, I. Palmisano, Chainsaw: a metareasoner for large ontologies, in: I. Horrocks, M. Yatskevich, E. Jiménez-Ruiz (Eds.), Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012, in: CEUR Workshop Proceedings, vol. 858, 2012, CEUR-WS.org, http://ceur-ws.org/Vol-858/ore2012_paper2.pdf.
[7] B.C. Grau, C. Halaschek-Wiener, Y. Kazakov, B. Suntisrivaraporn, Incremental classification of description logics ontologies, J. Autom. Reason. 44 (4) (2010) 337–369, https://doi.org/10.1007/s10817-009-9159-0.
[8] U. Sattler, T. Schneider, M. Zakharyaschev, Which kind of module should I extract?, in: Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009, 2009, http://ceur-ws.org/Vol-477/paper_33.pdf.
[9] A. Armas Romero, M. Kaminski, B. Cuenca Grau, I. Horrocks, Module extraction in expressive ontology languages via Datalog reasoning, J. Artif. Intell. Res. 55 (1) (2016) 499–564.

[10] C. Lutz, F. Wolter, Deciding inseparability and conservative extensions in the description logic EL, J. Symb. Comput. 45 (2) (2010) 194–228, https://doi.org/10.1016/j.jsc.2008.10.007.

[11] R. Kontchakov, F. Wolter, M. Zakharyaschev, Can you tell the difference between DL-Lite ontologies?, in: G. Brewka, J. Lang (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008, AAAI Press, 2008, pp. 285–295, http://www.aaai.org/Library/KR/2008/kr08-028.php.

[12] S. Ghilardi, C. Lutz, F. Wolter, Did I damage my ontology? A case for conservative extensions in description logics, in: P. Doherty, J. Mylopoulos, C. Welty (Eds.), Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06), AAAI Press, 2006, pp. 187–197.

[13] C. Lutz, D. Walther, F. Wolter, Conservative extensions in expressive description logics, in: M.M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, 2007, pp. 453–458, http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-071.pdf.

[14] M. Horridge, S. Bechhofer, The OWL API: a Java API for OWL ontologies, Semant. Web 2 (1) (2011) 11–21.

[15] N. Matentzoglu, Module-based classification of OWL ontologies, Ph.D. thesis, Manchester University, Manchester, UK, Aug. 2016.

[16] P.A. Bonatti, I.M. Petrova, L. Sauro, Optimized construction of secure knowledge-base views, in: D. Calvanese, B. Konev (Eds.), Proceedings of the 28th International Workshop on Description Logics, Athens, Greece, June 7–10, 2015, in: CEUR Workshop Proceedings, vol. 1350, 2015, CEUR-WS.org, http://ceur-ws.org/Vol-1350/paper-44.pdf.

[17] P.A. Bonatti, I.M. Petrova, L. Sauro, Optimizing the computation of overriding, in: The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Proceedings, Part I, Bethlehem, PA, USA, October 11–15, 2015, 2015, pp. 356–372.

[18] I. Horrocks, U. Sattler, Ontology reasoning in the SHOQ(D) description logic, in: B. Nebel (Ed.), Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4–10, 2001, Morgan Kaufmann, 2001, pp. 199–204.

[19] I. Horrocks, O. Kutz, U. Sattler, The even more irresistible SROIQ, in: P. Doherty, J. Mylopoulos, C.A. Welty (Eds.), Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2–5, 2006, AAAI Press, 2006, pp. 57–67, http://www.aaai.org/Library/KR/2006/kr06-009.php.

[20] B. Konev, C. Lutz, D. Walther, F. Wolter, Semantic modularity and module extraction in description logics, in: M. Ghallab, C.D. Spyropoulos, N. Fakotakis, N.M. Avouris (Eds.), ECAI 2008 - 18th European Conference on Artificial Intelligence, Proceedings, Patras, Greece, July 21–25, 2008, in: Frontiers in Artificial Intelligence and Applications, vol. 178, IOS Press, 2008, pp. 55–59.

[21] B. Konev, C. Lutz, D. Walther, F. Wolter, Model-theoretic inseparability and modularity of description logic ontologies, Artif. Intell. 203 (2013) 66–103.

[22] F. Martin-Recuerda, D. Walther, Axiom dependency hypergraphs for fast modularisation and atomic decomposition, in: M. Bienvenu, M. Ortiz, R. Rosati, M. Simkus (Eds.), Proceedings of the 27th International Workshop on Description Logics (DL'14), in: CEUR Workshop Proceedings, vol. 1193, 2014, pp. 299–310.

[23] B. Konev, C. Lutz, D. Walther, F. Wolter, Logical difference and module extraction with CEX and MEX, in: F. Baader, C. Lutz, B. Motik (Eds.), Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13–16, 2008, in: CEUR Workshop Proceedings, vol. 353, 2008, CEUR-WS.org, https://ceur-ws.org/Vol-353/KonevLutzWaltherWolter.pdf.

[24] P. Koopmann, J. Chen, Deductive module extraction for expressive description logics, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, 2020, pp. 1636–1643.

[25] J. Chen, M. Ludwig, D. Walther, Computing minimal subsumption modules of ontologies, in: GCAI-2018, 4th Global Conference on Artificial Intelligence, Luxembourg, September 18–21, 2018, vol. 55, 2018, pp. 41–53.

[26] J. Chen, M. Ludwig, Y. Ma, D. Walther, Zooming in on ontologies: minimal modules and best excerpts, in: The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Proceedings, Part I, Vienna, Austria, October 21–25, 2017, in: Lecture Notes in Computer Science, vol. 10587, 2017, pp. 173–189.

[27] R. Kontchakov, F. Wolter, M. Zakharyaschev, Logic-based ontology comparison and module extraction, with an application to DL-lite, Artif. Intell. 174 (15) (2010) 1093–1141, https://doi.org/10.1016/j.artint.2010.06.003.

[28] N. Matentzoglu, B. Parsia, Bioportal snapshot 30.03.2017, https://doi.org/10.5281/zenodo.439510, Mar. 2017.

[29] C.D. Vescovo, M. Horridge, B. Parsia, U. Sattler, T. Schneider, H. Zhao, Modular structures and atomic decomposition in ontologies, J. Artif. Intell. Res. 69 (2020) 963–1021, https://doi.org/10.1613/jair.1.12151.

[30] C.D. Vescovo, P. Klinov, B. Parsia, U. Sattler, T. Schneider, D. Tsarkov, Empirical study of logic-based modules: cheap is cheerful, in: H. Alani, L. Kagal, A. Fokoue, P.T. Groth, C. Biemann, J.X. Parreira, L. Aroyo, N.F. Noy, C. Welty, K. Janowicz (Eds.), The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Proceedings, Part I, Sydney, NSW, Australia, October 21–25, 2013, in: Lecture Notes in Computer Science, vol. 8218, Springer, 2013, pp. 84–100, https://doi.org/10.1007/978-3-642-41335-3_6.