UiO : Department of Informatics
University of Oslo

Mapping Relational Database Constraints to SHACL (Extended Version)

Ratan Bahadur Thapa and Martin Giese

Research Report Nr 503, July 25, 2022

ISBN 978-82-7368-603-9
ISSN 0806-3036

Mapping Relational Database Constraints to SHACL

Ratan Bahadur Thapa and Martin Giese

Dept. of Informatics, University of Oslo, Norway
{ratanbt|martingi}@ifi.uio.no

Abstract. Most structured data today is still stored in relational databases, which makes it important to provide a translation between relational and semantic data. A relational to RDF mapping, such as R2RML [14], provides a way to view existing relational data in the RDF data model through declarative mappings. While relational to RDF mapping translates relational instance data to RDF, it does not specify any translation of existing relational constraints such as primary and foreign key constraints. Since the introduction of R2RML, interest in RDF constraint languages has increased and SHACL [17] has been standardised. This raises the question of which SHACL constraints are guaranteed to be valid on a dataset produced by a relational to RDF mapping. For arbitrary SQL constraints and relational to RDF mappings, this is a hard problem, but we introduce a number of restrictions on the mappings that allow us to introduce a *constraint rewriting* for relational to RDF mappings that faithfully transfers SQL integrity constraints to SHACL constraints. We define and prove two fundamental properties, namely *maximal semantics preservation* and *monotonicity*.

1 Introduction

In relational database theory, one can restrict data to a set of relations that are considered to be useful to applications at hand by imposing relevant integrity constraints upon them, i.e., the semantics properties, also known as data dependencies, that the data in the database must obey. However, such integrity constraints of relational data are not explicit when mapped into RDF. A relational to RDF (R2R) mapping outputs an RDF graph that no longer contains the integrity constraints information. To overcome the problem, one can restore the semantic properties of R2R transformed data by using a semantics preserving constraint rewriting [28,7,25] that maps the integrity constraints of relational data into a well-behaved constraint formalism, which provides a closed-world description for the mapped RDF graph. The integrity constraints of the dataset that is being stored or represented in the RDF graph are a critical piece of information in practice, both to detect problems in the RDF dataset and provide data quality guarantees for RDF data exchange and interoperability.

In this paper, we study constraint rewriting for R2R mapping to make it more faithful by transforming the integrity constraints, such as primary and foreign keys, unique and not null integrity constraints as well as data types, from SQL database to RDF graph. In an attempt to transfer such integrity constraints of relational data, such as key constraints and functional dependency in *direct mapping* [2] to a larger perspective of relational constraints [1, Sect. 10] in more expressive *ontology-based mapping* [20] of relational data, into OWL DL axioms [22] as well as Epistemic DL axioms [15], the problem has recently been studied in [25,7] and [10,11,23] respectively. However, for our work, we follow the constraint rewriting technique proposed in [28] that explicitly transforms integrity constraints of SQL database into integrity constraints on the RDF graph, expressed in SHACL [17] as opposed to OWL/Epistemic DL axioms. Contrary to OWL, SHACL, the Shapes Constraint Language recommended by W3C since 2017, has a closed world semantics and uses the unique name assumption, which makes it a more suitable candidate than OWL for expressing as well as detecting the violations of integrity constraints on an RDF graph.

For arbitrary SQL constraints and relational to RDF mappings, constraint rewriting is a hard problem. For simplicity, we restrict ourselves to (a) the most common SQL constraints, namely keys, uniqueness and not null constraints, and (2) *simple* R2R mappings (Defn. 4), which are restricted in such a way that the resulting RDF is structurally close enough to the source that it remains possible to analyse the propagation of source constraints to the target. Thus, once the SHACL descriptions of the mapped RDF graph are available, they can be used to validate that the facts in the graph are compatible with the constraints of the relational source and the mapping, using the SHACL validation engine. However, R2R mappings are also known for their mapping inconsistency and redundancy anomalies [18,9], thus one-to-one semantics correspondence such as *semantics preservation* proposed in [28, Defn. 6] and [25, Defn. 12] between the relational and the mapped RDF data can not be established in general [28,25, Prop. 1]. One of the prominent reasons behind such flaws is that R2R mappings often imply SHACL constraints that satisfy the mapped RDF graph with respect to database constraints even if the key constraints are violated in the source database, which can not be easily fixed as the mappings rely on the values of database keys to produce RDF terms [28, Exam. 4 and 5]. We can thus not hope for semantic *equivalence* between the SQL and SHACL constraints. In this work we instead define a notion of *maximal semantics preservation* to express that any additional SHACL constraints are either implied by the generated ones, or not implied by the SQL constraints.

Example 1 *Consider the following database instance $\mathcal{D}$ with schemas that describes students and their enrollment in courses being offered by a university :*

create table `course` *(*`C_id` *varchar primary key,* `Title` *varchar unique);*
create table `student` *(*`S_id` *integer primary key,* `Name` *varchar,* `Code` *varchar not null foreign key references* `course(C_id)`*);*

S_id	Name	Code
011	Ida	CS40
012		CS20

FK →

C_id	Title
CS40	Logic
CS20	Database
CS50	Data Eng

In general, an R2R mapping is an assertion of the form $Q \longrightarrow \psi$ that transforms a set of tuples projected by SQL query Q, called *source query*, over a relational source $\mathcal{D}$ into a set of RDF triples defined by graph triple patterns ψ. Assume an R2R mapping M to retrieve students and their enrollment in the university's courses,

$$\text{Select } \texttt{S\_id} \text{ from } \texttt{student} \longrightarrow \langle \texttt{iri}_1(\texttt{S\_id}), \texttt{rdf:type}, \texttt{Student}\rangle.$$
$$\text{Select } \texttt{C\_id} \text{ from } \texttt{course} \longrightarrow \langle \texttt{iri}_2(\texttt{C\_id}), \texttt{rdf:type}, \texttt{Course}\rangle.$$
$$\text{Select } \texttt{S\_id, C\_id} \text{ from } \texttt{student}, \texttt{course} \longrightarrow \langle \texttt{iri}_1(\texttt{S\_id}), \texttt{enrolledFor}, \texttt{iri}_2(\texttt{C\_id})\rangle.$$
$$\text{where } \texttt{student.Code} = \texttt{course.C\_id}$$

where $\texttt{iri}_1$ and $\texttt{iri}_2$ are injective functions that construct iris for students and courses from their respective id's. The mapping M yields the following RDF graph G (on the left) from the database instance $\mathcal{D}$:

$\langle \texttt{iri}_1(011), \texttt{rdf:type}, \texttt{Student}\rangle.$
$\langle \texttt{iri}_1(012), \texttt{rdf:type}, \texttt{Student}\rangle.$
$\langle \texttt{iri}_2(\text{CS40}), \texttt{rdf:type}, \texttt{Course}\rangle.$
$\langle \texttt{iri}_2(\text{CS20}), \texttt{rdf:type}, \texttt{Course}\rangle.$
$\langle \texttt{iri}_2(\text{CS50}), \texttt{rdf:type}, \texttt{Course}\rangle.$
$\langle \texttt{iri}_1(011), \texttt{enrolledFor}, \texttt{iri}_2(\text{CS40})\rangle.$
$\langle \texttt{iri}_1(012), \texttt{enrolledFor}, \texttt{iri}_2(\text{CS20})\rangle.$

```
:Student a sh:NodeShape, rdfs:Class;
  sh:property [ sh:path :enrolledFor;
     sh:maxCount 1;  sh:minCount 1;
     sh:nodeKind sh:IRI;  sh:class :Course ].
:Course a sh:NodeShape, rdfs:Class;
  sh:property [ sh:path [sh:inversePath
     :enrolledFor];
    sh:nodeKind sh:IRI;  sh:class :Student ].
```

Next, consider a SHACL document S (on the right), which consists of node shapes :Student and :Course with implicit target class [1] that define the constraints, intuitively, all students must be enrolled for exactly one course, and all courses must be enrolled by zero or more students. Now observe that the document S not only validates the graph G but also guarantee the validation of every RDF graphs that can be generated via mappings M from any valid instance $\mathcal{D}$ of the schemas in Example 1, i.e., semantics preservation. Moreover, any further restrictions on the property paths of S, such as all courses must be enrolled by at least one students, would easily be violated, meaning that a valid database instance $\mathcal{D}$ can be found such that mapped RDF graphs would not validate the document S. Thus, we say that S is a maximally implied set of SHACL shapes for the given relational source and the mappings M. For proof details, we refer the readers to the extended version [30].

Example 1 illustrates that an assessment of R2R mapping is necessary to guarantee whether the integrity constraints of relational data are maximally propagated via mappings to the RDF. We thus take the process of R2R transformation into account and define constraint rewriting as a function from constraints in SQL database to the sets of SHACL shapes over RDF graph. We first introduce two fundamental properties of constraint rewriting, namely maximal semantics preservation and monotonicity. Finally, we show that our proposed constraint rewriting is both maximal semantics preserving and monotone, even in the most general and practical scenario where relational databases contain null values. A constraint rewriting for R2R mappings is monotonic if it assures that the result of constraint rewriting that is already computed no longer requires alteration after the addition of new mappings.

2 Preliminaries

In this section, we fix notions and notations fundamental to the definition of R2R mapping, and SHACL constraints [17].

Databases. Let Δ be a countably infinite set of constants, including the reserved symbol `null`. A *relational schema* $\mathcal{R}$ is a finite set of relation names, known as *relation schemas*.

[1] `https://www.w3.org/TR/shacl/#implicit-targetClass`.

We associate with each relation schema $R \in \mathcal{R}$ a finite, non-empty *set of named attributes*, denoted by $\mathsf{att}(R)$. An *instance* $\mathcal{D}$ of $\mathcal{R}$ assigns each relation schema $R \in \mathcal{R}$ a finite set of tuples $R^{\mathcal{D}}$, where each *tuple* $t \in R^{\mathcal{D}}$ is a function that assigns to each attribute in $\mathsf{att}(R)$ a value from domain Δ.

We write X as shorthand for a non-empty set $\{x_1, \ldots, x_n\}$ of attributes for $n \geq 1$, and $x \in X$ to say that x is one of the elements of the set. $|X| = n$ denotes the cardinality of the set. We further write $X \triangleleft R$ to denote that X is a non-empty subset of $\mathsf{att}(R)$. We write $t(x)$ to denote the restriction of a tuple $t \in R^{\mathcal{D}}$ to an attribute $x \in \mathsf{att}(R)$, which can be extended to a set $X \triangleleft R$, i.e., $t(X)$. Finally, we define a *relational database* as a pair of $\mathcal{R}$ and $\mathcal{D}$, where $\mathcal{R}$ is a relational schema and $\mathcal{D}$ is a database instance of $\mathcal{R}$. The *active domain* $\Gamma_{\mathcal{D}}$ of a database is the set of constants appearing in $\mathcal{D}$, i.e., $\Gamma_{\mathcal{D}} \subseteq \Delta \setminus \{\mathtt{null}\}$.

SQL Constraints. We consider declarations of the SQL: (a) *primary* (PK) and *foreign* (FK) keys, (b) *not null* (NN) and *unique* (UNQ) integrity, and (c) *data types*, constraints on the relational schema $\mathcal{R}$. We write Σ for the set of SQL constraints. NN, UNQ and PK constraints on a relational schema $\mathcal{R}$ are expressions of the form $\mathtt{NN}(X, R)$, $\mathtt{UNQ}(X, R)$ and $\mathtt{PK}(X, R)$, resp., for any $X \triangleleft R$ such that $R \in \mathcal{R}$. An instance $\mathcal{D}$ of $\mathcal{R}$ satisfies:

- □ $\mathtt{NN}(X, R)$ if for every $t \in R^{\mathcal{D}}$ and $x \in X$, $t(x) \neq \mathtt{null}$.
- □ $\mathtt{UNQ}(X, R)$ if for every $t, t' \in R^{\mathcal{D}}$, if $t(x) = t'(x) \neq \mathtt{null}$ for every $x \in X$ then $t = t'$.
- □ $\mathtt{PK}(X, R)$ if: (a) for every $t \in R^{\mathcal{D}}$ and $x \in X$, $t(x) \neq \mathtt{null}$, and (b) for every $t, t' \in R^{\mathcal{D}}$, if $t(X) = t'(X)$ then $t = t'$.

An FK constraint on $\mathcal{R}$ is an expression of the form $\mathtt{FK}(X, R, Y, S)$ for any $X \triangleleft R$ and $Y \triangleleft S$ with $|X| = |Y|$ and $R, S \in \mathcal{R}$. An instance $\mathcal{D}$ of $\mathcal{R}$ satisfies $\mathtt{FK}(X, R, Y, S)$ if for every $t \in R^{\mathcal{D}}$: either (a) $t(x) = \mathtt{null}$ for some $x \in X$, or (b) there exists a tuple $t' \in S^{\mathcal{D}}$ such that $t(X) = t'(Y)$. Next, to handle SQL data types, let the domain of an SQL data type ν be a subset $\Delta_\nu \subseteq \Delta$. An SQL data type declaration on $\mathcal{R}$ is an expression of the form $\mathtt{Type}(x, \nu, R)$ for every $x \in \mathsf{att}(R)$ such that $R \in \mathcal{R}$, where ν is an SQL data type. An instance $\mathcal{D}$ of $\mathcal{R}$ satisfies $\mathtt{Type}(x, \nu, R)$ for an attribute $x \in \mathsf{att}(R)$, if $t(x) \in \Delta_\nu$ for every $t \in R^{\mathcal{D}}$.

A *relational schema* $\mathcal{R}$ *with source constraints* Σ consists of the relational schema $\mathcal{R}$ and a set Σ of SQL constraints on $\mathcal{R}$, such that $\mathtt{UNQ}(Y, R) \in \Sigma$ for all $\mathtt{FK}(X, R, Y, S) \in \Sigma$, as usual in all SQL implementations. W.l.o.g., we also assume that for every $X \triangleleft R$: (a) if $\mathtt{PK}(X, R) \in \Sigma$, then $\mathtt{UNQ}(X, R) \in \Sigma$ and $\mathtt{NN}(X, R) \in \Sigma$, (b) if $\mathtt{NN}(X, R) \in \Sigma$, then $\mathtt{NN}(x, R) \in \Sigma$ for every $x \in X$ and (c) if $\mathtt{NN}(x, R) \in \Sigma$ for every $x \in X$, then $\mathtt{NN}(X, R) \in \Sigma$. Finally, given a relational schema $\mathcal{R}$ with constraints Σ, and an instance $\mathcal{D}$ of $\mathcal{R}$, we call $\mathcal{D}$ a *legal instance* of $\mathcal{R}$ with Σ, denoted by $\mathcal{D} \models \Sigma$, if $\mathcal{D}$ satisfies all constraints in Σ.

Queries. Assume relational algebra with Selection $\sigma_{\neg\mathsf{isNull}}$, Projection π, Equi Join $\bowtie_{\mathtt{equality}}$, Right Outer Join $⟖_{\mathtt{equality}}$, Left Outer Join $⟕_{\mathtt{equality}}$ and Full Outer Join $⟗_{\mathtt{equality}}$ operations as query language that corresponds to a sub-class of *basic fragment of SQL* standard. We use notation $\sigma_{\neg\mathtt{isNull}}$ for the select condition 'IS NOT NULL' over an attribute as in SQL, which can be extended to a set of attributes. Assume that $\mathcal{R}$ is a relational schema, $\mathcal{D}$ is an instance of $\mathcal{R}$ and Q is a relational algebra expression over $\mathcal{R}$. Then $\mathsf{att}(Q)$, the set of attributes of Q, is recursively defined as follows, where we write $X \triangleleft Q$ to denote that X is a non-empty subset of $\mathsf{att}(Q)$:

1. If $Q = R$ such that $R \in \mathcal{R}$, then $\mathsf{att}(Q) = \mathsf{att}(R)$.
2. If Q' is a relational algebra expression over $\mathcal{R}$, $X \lhd Q'$ and $Q = \sigma_{\neg\mathtt{isNull}(X)}(Q')$, i.e., $\sigma_{\neg\mathtt{isNull}(x_1)\wedge \ldots \wedge\neg\mathtt{isNull}(x_n)}(Q')$, then $\mathsf{att}(Q) = \mathsf{att}(Q')$.
3. If Q' is a relational algebra expression over $\mathcal{R}$, $X \lhd Q'$ and $Q = \pi_X(Q')$, then $\mathsf{att}(Q) = X$.
4. Let Q_1, Q_2 be relational algebra expressions over $\mathcal{R}$ such that $X \lhd Q_1$ and $Y \lhd Q_2$ have compatible data types. If $Q = Q_1 \, \mathtt{OP}_{X=Y} \, Q_2$ s.t. $\mathtt{OP} \in \{\bowtie, ⟕, ⟖, ⟗\}$, then $\mathsf{att}(Q) = \mathsf{att}(Q_1) \cup \mathsf{att}(Q_2)$.

The evaluation of Q over $\mathcal{D}$, a set of tuples denoted by $Q^{\mathcal{D}}$, is recursively defined as follows,

1. If $Q = R$ such that $R \in \mathcal{R}$, then $Q^{\mathcal{D}} = R^{\mathcal{D}}$.
2. If Q' is a relational algebra expression over $\mathcal{R}$, $X \lhd Q'$ and $Q = \sigma_{\neg\mathtt{isNull}(X)}(Q')$, then $Q^{\mathcal{D}} = \{t \in Q'^{\mathcal{D}} \mid t(x) \neq \mathtt{null} \text{ for every } x \in X\}$.
3. If Q' is a relational algebra expression over $\mathcal{R}$, $X \lhd Q'$ and $Q = \pi_X(Q')$ then, for every $t \in Q^{\mathcal{D}}$ there exists $t' \in Q'^{\mathcal{D}}$ such that $t(X) = t'(X)$.
4. Let Q_1, Q_2 be relational algebra expressions over $\mathcal{R}$ such that $X \lhd Q_1$ and $Y \lhd Q_2$ have compatible data types.
   a. If $Q = Q_1 \bowtie_{X=Y} Q_2$ then for every $t \in Q^{\mathcal{D}}$: (i) there exist $t_1 \in {Q_1}^{\mathcal{D}}$ and $t_2 \in {Q_2}^{\mathcal{D}}$ s.t. $t(x) = t_1(x) = t_2(y) \neq \mathtt{null}$ for every $x \in X$ and $y \in Y$, (ii) $t(u) = t_1(u)$ for every $u \in (\mathsf{att}(Q_1) \setminus \mathsf{att}(Q_2))$, and (iii) $t(v) = t_2(v)$ for every $v \in (\mathsf{att}(Q_2) \setminus \mathsf{att}(Q_1))$.
   b. If $Q = Q_1 ⟕_{X=Y} Q_2$ then for every $t \in Q^{\mathcal{D}}$: either (i) there exist $t_1 \in {Q_1}^{\mathcal{D}}$ and $t_2 \in {Q_2}^{\mathcal{D}}$ s.t. $t(x) = t_1(x) = t_2(y) \neq \mathtt{null}$ for every $x \in X$ and $y \in Y$, $t(u) = t_1(u)$ for every $u \in (\mathsf{att}(Q_1) \setminus \mathsf{att}(Q_2))$ and $t(v) = t_2(v)$ for every $v \in (\mathsf{att}(Q_2) \setminus \mathsf{att}(Q_1))$, or (ii) there exist $t_1 \in {Q_1}^{\mathcal{D}}$ s.t. $t(u) = t_1(u)$ for every $u \in (\mathsf{att}(Q_1) \setminus \mathsf{att}(Q_2))$ and $t(v) = \mathtt{null}$ for every $v \in (\mathsf{att}(Q_2) \setminus \mathsf{att}(Q_1))$.
   c. If $Q = Q_1 ⟖_{X=Y} Q_2$ then for every $t \in Q^{\mathcal{D}}$: either (i) there exist $t_1 \in {Q_1}^{\mathcal{D}}$ and $t_2 \in {Q_2}^{\mathcal{D}}$ s.t. $t(x) = t_1(x) = t_2(y) \neq \mathtt{null}$ for every $x \in X$ and $y \in Y$, $t(u) = t_1(u)$ for every $u \in (\mathsf{att}(Q_1) \setminus \mathsf{att}(Q_2))$ and $t(v) = t_2(v)$ for every $v \in (\mathsf{att}(Q_2) \setminus \mathsf{att}(Q_1))$, or (ii) there exist $t_2 \in {Q_2}^{\mathcal{D}}$ s.t. for every $t(v) = t_2(v)$ for every $v \in (\mathsf{att}(Q_2) \setminus \mathsf{att}(Q_1))$ and $t(u) = \mathtt{null}$ for every $u \in (\mathsf{att}(Q_1) \setminus \mathsf{att}(Q_2))$.
   d. If $Q = Q_1 ⟗_{X=Y} Q_2$ then $Q^{\mathcal{D}} = {Q_a}^{\mathcal{D}} \cup {Q_b}^{\mathcal{D}}$ s.t. $Q_a = Q_1 ⟕_{X=Y} Q_2$ and $Q_b = Q_1 ⟖_{X=Y} Q_2$.

Henceforth, we denote by SP the relational expression containing only select-project relational operations, and SPJ the relational expression containing select-project-(outer)join relational operations, respectively.

Definition 1 *Let Q be a relational expression over a relational schema $\mathcal{R}$. Then, we say that the Q is a* `valid query` *if and only if there exist foreign key references between every two sets of attributes participating in an equality join condition in the Q.*

RDF Graphs. Assume that $\mathcal{I}, \mathcal{B}$ and $\mathcal{L}$ are countably infinite disjoint sets of *Internationalized Resource Identifiers* (IRIs), *Blank nodes* and *Literals*, respectively. The set of RDF terms $\mathcal{T}$ is $\mathcal{I} \cup \mathcal{L} \cup \mathcal{B}$. A *well-defined RDF triple* is defined as a triple $\langle s, p, o\rangle$ where $s \in \mathcal{I} \cup \mathcal{B}$ is called the subject, $p \in \mathcal{I}$ is called the predicate and $o \in \mathcal{T}$ is called the object. An RDF graph $G \subseteq (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times \mathcal{T}$ is a finite subset of RDF triples.

Definition 2 *The set of nodes of an RDF graph G is the set of subjects and objects of triples in the graph, i.e.,* $\{s, o \mid \langle s, p, o\rangle \in G\}$.

Assume a countably infinite set $\mathcal{V}$ of variables disjoint from $\mathcal{T}$. A triple pattern is defined as a triple in $(\mathcal{I} \cup \mathcal{B} \cup \mathcal{V}) \times (\mathcal{I} \cup \mathcal{V}) \times (\mathcal{T} \cup \mathcal{V})$. A *basic graph pattern (BGP)* is a finite set of triple patterns. The schema $sch(\psi)$ of a triple pattern ψ is the *RDF property and class* predicates [19] from the ψ.

Mappings. Formally, we adopt R2R mapping [6,24] that generate RDF triples from the active domain of a database $\Gamma_{\mathcal{D}}$. Assume countably infinite and disjoint sets $\mathbb{F}$ and $\mathbb{T}$ of iri-template and typing *functions* respectively, with each function $\alpha \in \mathbb{F} \cup \mathbb{T}$ has an associated arity $n > 0$. W.l.o.g., we assume that functions $\mathbb{F} \cup \mathbb{T}$ are injective, and map only `null` to `null`.

Definition 3 *We specify R2R-mapping* $\mathcal{M}$*, from relational database-to-RDF, partitioned into three disjoint sets:* $\mathcal{M}_\mathcal{C}$*,* $\mathcal{M}_\mathcal{P}$ *and* $\mathcal{M}_\mathcal{U}$ *such that*

i. $\mathcal{M}_\mathcal{C}$ *is a set of data-to-RDF concept mappings, each one of the form*

$$Q_X \longrightarrow \langle \mathbf{f}(X), \texttt{rdf:type}, C\rangle,$$

where
a. Q_X *is a source query Q over* $\mathcal{R}$ *with* $X \triangleleft Q$*,*
b. $\mathbf{f} \in \mathbb{F}$ *and C is an RDF concept.*

ii. $\mathcal{M}_\mathcal{P}$ *is a set of data-to-RDF object property mappings, each one of the form*

$$Q_{X,Y} \longrightarrow \langle \mathbf{f}(X), P, \mathbf{f}'(Y)\rangle,$$

where
a. $Q_{X,Y}$ *is a source query Q over* $\mathcal{R}$ *with* $X, Y \triangleleft Q$*,*
b. $\mathbf{f}, \mathbf{f}' \in \mathbb{F}$ *and P is an RDF object property.*

iii. $\mathcal{M}_\mathcal{U}$ *is a set of data-to-RDF datatype property mappings, each one of the form*

$$Q_{X,Y} \longrightarrow \langle \mathbf{f}(X), U, \mathbf{t}(Y)\rangle,$$

where
a. $Q_{X,Y}$ *is a source query Q over* $\mathcal{R}$ *with* $X, Y \triangleleft Q$*,*
b. $\mathbf{f} \in \mathbb{F}$*,* $\mathbf{t} \in \mathbb{T}$ *and U is an RDF datatype property.*

Let m be a mapping $Q \longrightarrow \psi$ of a triple pattern ψ, as in Defn. 3. The source query Q is the *body*(m) of m, whereas the triple pattern ψ is the *head*(m). The schema $sch(\mathcal{M})$ of a mapping set $\mathcal{M}$ is the union of $sch(head(m))$ of each $m \in \mathcal{M}$. For any two mapping sets $\mathcal{M}$ and $\mathcal{M}'$ defined over a relational schema $\mathcal{R}$ with source constraint Σ, we write $\mathcal{M}' \subseteq \mathcal{M}$, if for every mapping definition m, if $m \in \mathcal{M}'$ then $m \in \mathcal{M}$.

Definition 4 *Let* Q_C*,* Q_P *and* Q_U *be the source queries of mappings of an RDF concept C, object property P and datatype property U, respectively. Then, we say that a mapping set* $\mathcal{M}$ *(according to Defn. 3) is a* simple mapping *if: (a)* $\mathcal{M}$ *contains exactly one mapping definition per concept C, object property P and datatype property U predicates* *in sch($\mathcal{M}$); (b) each Q_P is a valid* SPJ *query with one join operation, (c) each Q_U is an* SP *query, d) if C and C' are the concepts whose instances are subject and object of an object property P, then the Q_C and $Q_{C'}$ are either equal to Q_P or* SP *queries with a projected set of attributes whose (tuple) values are mapped to instances of C and C', and (e) if C is the concept whose instances are the subject of a datatype property U, then Q_C is either equal to Q_U or an* SP *query with a projected set of attributes whose (tuple) values are mapped into the instances of C.*

Example 2 *Consider the mapping of object property '`EnrolledFor`' in Example 1. Instances of concepts '`Student`' and '`Course`' are mapped to subject and object of the property '`EnrolledFor`', respectively. Then, according to simple mapping in Defn. 4, the source queries used in the mappings of those '`Student`' and '`Course`' concepts must be either the exact same source query used in the mapping of the property '`EnrolledFor`' or the* SP *source queries as in Example 1. Thus, a distinct simple mapping could be defined for the same purpose that maps RDF concepts '`Student`' and '`Course`' using the same* SPJ *source query Q_P,*

$$Q_P ::= \textit{Select } \texttt{S\_id, C\_id} \textit{ from } \texttt{student}, \texttt{course} \\ \textit{where } \texttt{student.Code} = \texttt{course.C\_id}$$

as used in the mapping of object property '`EnrolledFor`' as follows:

$$Q_P \longrightarrow \langle \mathtt{iri}_1(\mathtt{S\_id}), \mathtt{rdf{:}type}, \mathtt{Student}\rangle.$$
$$Q_P \longrightarrow \langle \mathtt{iri}_2(\mathtt{C\_id}), \mathtt{rdf{:}type}, \mathtt{Course}\rangle.$$
$$Q_P \longrightarrow \langle \mathtt{iri}_1(\mathtt{S\_id}), \mathtt{enrolledFor}, \mathtt{iri}_2(\mathtt{C\_id})\rangle.$$

Let $t \in Q^{\mathcal{D}}$ be a tuple of constants, and let $\mathbf{f}$(X) be a term such that $\mathbf{f} \in \mathbb{F}$ and $X \triangleleft Q$. Then, $\mathbf{f}(t(X))$ is a ground term of $\mathbf{f}$(X) obtained by substituting occurrence of every $x \in X$ with $t(x)$.

Definition 5 *Let $\mathcal{M}_C \cup \mathcal{M}_{\mathcal{P}} \cup \mathcal{M}_{\mathcal{U}}$ be an R2R mapping set $\mathcal{M}$ defined over a relational schema $\mathcal{R}$, and $\mathcal{D}$ an instance of $\mathcal{R}$. Then, we call the set of well-defined RDF triple assertions $\mathcal{M}(\mathcal{D})$, i.e.,*

$$\begin{aligned}\mathcal{M}(\mathcal{D}) = &\{\langle \mathbf{f}(t(X)), \mathtt{rdf{:}type}, C\rangle \mid \{Q \longrightarrow \langle \mathbf{f}(X), \mathtt{rdf{:}type}, C\rangle\} \in \mathcal{M}_C, X \triangleleft Q \text{ and } t \in Q^{\mathcal{D}}\} \\ &\cup \{\langle \mathbf{f}(t(X)), P, \mathbf{f}'(t(Y))\rangle \mid \{Q \longrightarrow \langle \mathbf{f}(X), P, \mathbf{f}'(Y)\rangle\} \in \mathcal{M}_{\mathcal{P}}, X, Y \triangleleft Q \text{ and } t \in Q^{\mathcal{D}}\} \\ &\cup \{\langle \mathbf{f}(t(X)), U, \mathbf{t}(t(Y))\rangle \mid \{Q \longrightarrow \langle \mathbf{f}(X), U, \mathbf{t}(Y)\rangle\} \in \mathcal{M}_{\mathcal{U}}, X, Y \triangleleft Q \text{ and } t \in Q^{\mathcal{D}}\},\end{aligned}$$

the RDF graph projected by the mapping set $\mathcal{M}$ and the instance $\mathcal{D}$.

We recall that R2R mappings in Defn. 3 generate RDF triples from the active domain of a database $\Gamma_{\mathcal{D}}$, i.e., `null` cannot appear in the output RDF triples. Therefore, in this paper, we explicitly consider that (a) mappings $\mathcal{M}$ is simple, and (b) w.l.o.g., source query Q of each mapping in $\mathcal{M}$ contains $\sigma_{\neg\mathtt{isNull}(X)}$ and $\sigma_{\neg\mathtt{isNull}(Y)}$ filters over every projected set of $X, Y \triangleleft \mathsf{att}(Q)$.

SHACL. Our formal treatment of the *core constraints* of SHACL [17] is based on the approach of Corman et al. [13]. Each SHACL constraint is a set of conditions, usually referred to as shape, defined as a triple $\langle s, \tau_s, \phi_s\rangle$ consisting of a shape IRI s, a *target* *definition* τ_s, and a *constraint definition* ϕ_s. The τ_s and ϕ_s are expressions that determine for every RDF graph G and node n of G, whether n is a target of the shape, $G \models \tau_s(n)$, respectively, whether n satisfies the constraint, $G \models \phi_s(n)$. All shapes generated by our transformation have an 'implicit target class,' which means that s is also the IRI of a class and $G \models \tau(n)$ iff n is a SHACL instance of class s.[2] For the purpose of our work, the constraint ϕ_s is an expression defined according to the following grammar:

$$\begin{aligned} \phi &::= \phi \wedge \phi \;\mid \geq_n P^{\pm}.\,\alpha \;\mid \leq_n P^{\pm}.\,\alpha \;\mid \rhd_C P^{\pm} \\ \alpha &::= \top \;\mid\; \ell \;\mid\; \neg\ell \;\mid\; C \;\mid\; \neg C \end{aligned} \tag{1}$$

where $\top$ stands for truth, ℓ is an XML schema datatype, C and P are an RDF concept and property names respectively, the superscript $\pm$ stands for a property or its inverse, $n \in \mathbb{N}$, $\neg$ for negation, $(\geq_n P^{\pm}.\alpha)$ means 'must have at least n $P^{\pm}$-successor verifying α' for any $n \in \mathbb{N}$ and $(\rhd_C P^{\pm})$ means 'all values of $P^{\pm}$-successor must be unique [3] among instances of concept C'. As syntactic sugar, we use $(=_n P^{\pm}.\alpha)$ for $(\geq_n P^{\pm}.\alpha) \wedge (\leq_n P^{\pm}.\alpha)$, $(\rhd_C P^{\pm}.\alpha)$ for $(\leq_1 P^{\pm}.\alpha) \wedge (\rhd_C P^{\pm})$ and $(\unrhd_C P^{\pm}.\alpha)$ for $(=_1 P^{\pm}.\alpha) \wedge (\rhd_C P^{\pm})$.

A SHACL document is a set of SHACL shapes. An RDF graph G validates against a shape $\langle s, \tau_s, \phi_s \rangle$ if for every nodes n of G, if $G \models \tau_s(n)$ then $G \models \phi_s(n)$. An RDF graph G validates against a SHACL document S, written $G \models S$, iff G validates against all shapes in S. The schema $sch(s)$ of a SHACL shape s is the set of RDF concept and property predicates [19] used in the target τ_s and constraint ϕ_s definition. The schema $sch(S)$ of a SHACL document S is the union of $sch(s)$ of every shape $s \in S$.

3 Constraint rewriting: Definition and Properties

Our goal is to generate a set of SHACL constraints that is as strong as possible while being guaranteed to hold for all RDF graphs resulting from valid database instances. Let $\mathcal{M}$ be a mapping set defined over a relational schema $\mathcal{R}$ with source constraints Σ.

Definition 6 *A SHACL document S is an Σ-implied set of shapes with respect to $\mathcal{M}$, written as $\Sigma \models_{\mathcal{M}} S$, if for every instance $\mathcal{D}$ of $\mathcal{R}$:*

$$\mathcal{D} \models \Sigma \rightarrow \mathcal{M}(\mathcal{D}) \models S.$$

Definition 7 *Let $\Sigma \models_{\mathcal{M}} S$. Then, we say that S is a maximally Σ-implied set of shapes with respect to $\mathcal{M}$, written as $\Sigma \models^*_{\mathcal{M}} S$, if for every $\Sigma \models_{\mathcal{M}} S'$ s.t. $sch(S') \subseteq sch(\mathcal{M})$ and every RDF graph $\mathcal{G}$:*

$$\mathcal{G} \models S \rightarrow \mathcal{G} \models S'.$$

We now formalise a constraint rewriting and some desirable properties. Let $\mathbb{S}$ be the set of all SHACL shapes and $\mathbb{Q}$ be the set of all pairs $(\mathcal{M}, \Sigma)$ such that $\mathcal{M}$ is a mapping set defined over a relational schema $\mathcal{R}$ with source constraints Σ.

Definition 8 (Constraint rewriting) *A* constraint rewriting *is a function* $\mathcal{T} : \mathbb{Q} \rightarrow \mathcal{P}(\mathbb{S})$.

[2] `https://www.w3.org/TR/shacl/#implicit-targetClass`
[3] `dash:uniqueValueForClassConstraintComponent` from `http://datashapes.org`

We next introduce central properties of a constraint rewriting $\mathcal{T}$.

Definition 9 (Semantics preservation) *A constraint rewriting $\mathcal{T}$ is* semantics preserving *if for every mapping set $\mathcal{M}$ and every source constraints Σ:*

$$\Sigma \models_{\mathcal{M}} \mathcal{T}(\mathcal{M}, \Sigma).$$

Definition 10 (Maximal semantics preservation) *A constraint rewriting $\mathcal{T}$ is* maximal semantics preserving *if for every mapping set $\mathcal{M}$ and every source constraints Σ:*

$$\Sigma \models^{*}_{\mathcal{M}} \mathcal{T}(\mathcal{M}, \Sigma).$$

Definition 11 (Monotonicity) *A constraint rewriting $\mathcal{T}$ is monotone if for any mapping sets $\mathcal{M}' \subseteq \mathcal{M}$ defined over a relational schema $\mathcal{R}$ with source constraint Σ and every RDF graph $\mathcal{G}$:*

$$\mathcal{G} \models \mathcal{T}(\mathcal{M}, \Sigma) \rightarrow \mathcal{G} \models \mathcal{T}(\mathcal{M}', \Sigma).$$

4 View Constraint: Definitions

As introduced in Sect. 2, R2R mapping relies on database views based on a source query to compute RDF terms from the database values. As a first step of our constraint transformation, we have to analyse the propagation of database constraints to these views.

Let $\mathcal{R}$ be a relational schema with source constraints Σ, and $R \in \mathcal{R}$. The constraint Σ restricted to the set of $\mathsf{att}(R)$, denoted by $\Sigma|_R$, is the set of constraints such that for every constraint $\sigma \in \Sigma$ on any $X \triangleleft R$, there is $\sigma \in \Sigma|_R$. For example, if $\mathtt{FK}(X, R, Y, S) \in \Sigma$ (resp., $\mathtt{FK}(Y, S, X, R) \in \Sigma$) on any $X \triangleleft R$, then there is $\mathtt{FK}(X, R, Y, S) \in \Sigma|_R$ (resp., $\mathtt{FK}(Y, S, X, R) \in \Sigma|_R$).

Definition 12 *Let Q be a relational expression over a relational schema $\mathcal{R}$ with source constraints Σ. Then, the set Σ propagated to the set of $\mathsf{att}(Q)$, denoted by $\Sigma|_Q$, is recursively defined as follows,*

a. If $Q = R$ such that $R \in \mathcal{R}$, then $\Sigma|_Q = \Sigma|_R$.
b. $Q = \sigma_{\neg \mathtt{isNull}(X)}(Q')$ where $X \triangleleft Q'$, then $\Sigma|_Q = \Sigma|_{Q'}$.
c. If $Q = \pi_X(Q')$ where $X \triangleleft Q'$ then $\Sigma|_Q = \{\mathtt{PK}(Y, R), \mathtt{UNQ}(Y, R), \mathtt{NN}(Y, R), \mathtt{FK}(Y, R, Z, S), \mathtt{FK}(Z, S, Y, R) \in \Sigma|_{Q'} \mid Y \subseteq X \text{ and } R, S \in \mathcal{R}\}$.
d. If $Q = Q_1 \; \mathtt{OP}_{X=Y} \; Q_2$ where $X \triangleleft Q_1$ and $Y \triangleleft Q_2$ have compatible data types, and $\mathtt{OP} \in \{\bowtie, \ltimes, \rtimes, ⟗\}$, then $\Sigma|_Q = \Sigma|_{Q_1} \cup \Sigma|_{Q_2}$.

SQL constraints are not well suited to direct translation to SHACL, so we introduce an intermediate representation similar to functional dependencies. Let R be a relation name with $X, Y \triangleleft R$. Then, we write a functional dependency as an expression of the form $\mathtt{FD}_{X \rightarrow Y}$, i.e., meaning $X \triangleleft R$ functionally determines $Y \triangleleft R$. Relational data dependencies, such as functional, multi-value and others, are originally defined on databases without `null` [3,5]. However, we need notions of data dependencies that also apply to databases with `null`, such as in [4], which we define as follows:

Definition 13 *Let Q be a source query over a relational schema $\mathcal{R}$ with source constraints Σ, $R \in \mathcal{R}$ a relation name and $\mathcal{D}$ an arbitrary instance of $\mathcal{R}$. Let V be the pair $(Q^{\mathcal{D}}, \Sigma|_Q)$ of projected view $Q^{\mathcal{D}}$ and propagated constraints $\Sigma|_Q$. Then, for any $X, Y \lessdot Q$,*

a. $V \models FP_{X \to Y}$ *if for every* $t, t' \in Q^{\mathcal{D}}$*, if* $t(X) = t'(X)$ *then* $t(Y) = t'(Y)$.
b. $V \models UF_{X \to Y}$ *if* $Q^{\mathcal{D}} \models FP_{X \to Y}$ *and* $Q^{\mathcal{D}} \models FP_{Y \to X}$.
c. $V \models FD_{X \to Y}$ *if* $Q^{\mathcal{D}} \models FP_{X \to Y}$ *and* $NN(X, R), NN(Y, R) \in \Sigma|_Q$.
d. $V \models UFD_{X \to Y}$ *if* $Q^{\mathcal{D}} \models FD_{X \to Y}$ *and* $Q^{\mathcal{D}} \models FD_{Y \to X}$.

Henceforth, we will keep the SQL notations intuitively simple in examples, i.e., we write $\mathtt{NN}(X) \in \Sigma|_{X \lessdot R}$ instead of $\mathtt{NN}(X, R) \in \Sigma|_{X \lessdot R}$ for the propagated $\mathtt{NN}(X, R) \in \Sigma$ to $\Sigma|_{X \lessdot R}$.

Example 3 *Following Example 1, assume a mapping set $\mathcal{M}$ with $\mathbf{f}_S$ and $\mathbf{f}_C$ iri-templates and a typing function $\mathbf{t}_v$* [4] *as follows:*

a. $\pi_{\mathtt{S\_id, Name}} \sigma_{\neg \mathtt{isNull(S\_id)} \wedge \neg \mathtt{isNull(Name)}}(\mathtt{student}) \longrightarrow \langle \mathbf{f}_S(\mathtt{S\_id}), \mathtt{hasName}, \mathbf{t}_v(\mathtt{Name}) \rangle$.
b. $\pi_{\mathtt{C\_id, Title}} \sigma_{\neg \mathtt{isNull(C\_id)} \wedge \neg \mathtt{isNull(Title)}}(\mathtt{course}) \longrightarrow \langle \mathbf{f}_C(\mathtt{C\_id}), \mathtt{hasTitle}, \mathbf{t}_v(\mathtt{Title}) \rangle$.

Let $Q_1 = \pi_{\mathtt{S\_id, Name}} \sigma_{\neg \mathtt{isNull(S\_id)} \wedge \neg \mathtt{isNull(Name)}}(\mathtt{student})$*, and* $V_1 = ({Q_1}^{\mathcal{D}}, \Sigma|_{Q_1})$*. Then,*

- $\mathsf{att}(Q_1) = \{\mathtt{S\_id, Name}\}$ *and* $\Sigma|_{\mathsf{att}(Q_1)} = \{PK(\mathtt{S\_id}), UNQ(\mathtt{S\_id}), NN(\mathtt{S\_id}), Type(\mathtt{S\_id}, v), Type(\mathtt{Name}, v)\}$*, i.e., from assumption in Sect. 2, if* $PK(\mathtt{S\_id})$ *then* $UNQ(\mathtt{S\_id})$ *and* $NN(\mathtt{S\_id})$.
- $V_1 \models FP_{\mathtt{S\_id} \to Name}$ *since for every* $t, t' \in {Q_1}^{\mathcal{D}}$*, if* $t(\mathtt{S\_id}) = t'(\mathtt{S\_id})$ *then* $t(\mathtt{Name}) = t'(\mathtt{Name})$.

Filter $\sigma_{\neg \mathtt{isNull(Name)}}$ *excludes tuples from* ${Q_1}^{\mathcal{D}}$ *that contains null for the* $\mathtt{Name} \in \mathsf{att}(Q_1)$.
Similarly, let $Q_2 = \pi_{\mathtt{C\_id,\ Title}} \sigma_{\neg \mathtt{isNull(C\_id)} \wedge \neg \mathtt{isNull(Title)}}(\mathtt{course})$*, and* $V_2 = ({Q_2}^{\mathcal{D}}, \Sigma|_{Q_2})$*. Then,*

- $\mathsf{att}(Q_2) = \{\mathtt{C\_id,\ Title}\}$ *and* $\Sigma|_{\mathsf{att}(Q_2)} = \{PK(\mathtt{C\_id}), UNQ(\mathtt{C\_id}), NN(\mathtt{C\_id}), Type(\mathtt{C\_id}, v), UNQ(\mathtt{Title}), Type(\mathtt{Title}, v), FK(\mathtt{Code}, \mathtt{student}, \mathtt{C\_id}, \mathtt{course})\}$
- $V_2 \models FP_{\mathtt{C\_id} \to \mathtt{Title}}$ *since for any* $t, t' \in {Q_2}^{\mathcal{D}}$*, if* $t(\mathtt{C\_id}) = t'(\mathtt{C\_id})$ *then* $t(\mathtt{Title}) = t'(\mathtt{Title})$.
- $V_2 \models FP_{\mathtt{Title} \to \mathtt{C\_id}}$ *since for any* $t, t' \in {Q_2}^{\mathcal{D}}$*, if* $t(\mathtt{Title}) = t'(\mathtt{Title})$ *then* $t(\mathtt{C\_id}) = t'(\mathtt{C\_id})$.
- $V_2 \models UF_{\mathtt{C\_id} \to \mathtt{Title}}$ *since* ${Q_2}^{\mathcal{D}} \models FP_{\mathtt{C\_id} \to \mathtt{Title}}$ *and* ${Q_2}^{\mathcal{D}} \models FP_{\mathtt{Title} \to \mathtt{C\_id}}$.

5 Source to View Constraint Implication

The next step is to determine which of the data dependencies from Defn. 13 hold for the view defined by the source queries, i.e., they are implied by the propagated SQL constraints.

Let Q be a source query over a relational schema $\mathcal{R}$ with source constraints Σ. Then, we say that Σ implies a data dependency $\sigma_{X \to Y}$ s.t. $\sigma \in \{\mathtt{UFD, FD, UFP, FP}\}$ on $X, Y \lessdot Q$, denoted by $\Sigma_Q \Vdash \sigma_{X \to Y}$, if $V \models \sigma_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, where $V = (Q^{\mathcal{D}}, \Sigma|_Q)$ is the pair of projected view $Q^{\mathcal{D}}$ and propagated constraints $\Sigma|_Q$. We now concentrate on SP source queries.

[4] $\mathbf{t}_v$ specify XML Schema datatype of RDF literal $\mathbf{t}_v(d)$ corresponding to the SQL data type v of the database constant $d \in \Delta_v$, e.g., $\mathbf{t}_v$ is an xsd:string IRI term if v is varchar SQL data type.

Lemma 1. *Let Q be a source query $\pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R)$ over a relational schema $\mathcal{R}$ with source constraints Σ, $R \in \mathcal{R}$ a relation name and $\Sigma|_Q$ the set Σ propagated to set of* $\mathsf{att}(Q)$. *Then, for any $X, Y \triangleleft Q$,*

a. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $UNQ(X,R) \in \Sigma|_Q$.
b. $\Sigma_Q \Vdash UF_{X\to Y}$ *if* $UNQ(X,R), UNQ(Y,R) \in \Sigma|_Q$.
c. $\Sigma_Q \Vdash FD_{X\to Y}$ *if* $UNQ(X,R) \in \Sigma|_Q$ *and* $NN(X,R), NN(Y,R) \in \Sigma|_Q$.
d. $\Sigma_Q \Vdash UFD_{X\to Y}$ *if* $UNQ(X,R), UNQ(Y,R) \in \Sigma|_Q$ *and* $NN(X,R), NN(Y,R) \in \Sigma|_Q$.

Corollary 1. *Let Q be a source query $\pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R)$ over a relational schema $\mathcal{R}$ with source constraints Σ, $R \in \mathcal{R}$ a relation name and $\Sigma|_Q$ the set Σ propagated to set of* $\mathsf{att}(Q)$. *Then, for any $X, Y \triangleleft Q$,*

a. $\Sigma_Q \Vdash UFD_{X\to Y} \rightarrow \Sigma_Q \Vdash FD_{X\to Y}$ *and* $\Sigma_Q \Vdash FD_{X\to Y} \rightarrow \Sigma_Q \Vdash FP_{X\to Y}$
b. $\Sigma_Q \Vdash UFD_{X\to Y} \rightarrow \Sigma_Q \Vdash UF_{X\to Y}$ *and* $\Sigma_Q \Vdash UF_{X\to Y} \rightarrow \Sigma_Q \Vdash FP_{X\to Y}$

We next concentrate on `SPJ` source queries. An `SPJ` source query Q over a relational schema $\mathcal{R}$ with source constraints Σ is a relational algebra expression of the form,

$$Q \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R_1 \,\mathsf{OP}_{U=V}\, R_2),$$

where $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $Y, V \triangleleft R_2$, $|U| = |V|$ and $\mathsf{OP} \in \{\bowtie, \rightouterjoin, \leftouterjoin, \fullouterjoin\}$. Since mapping in Defn. 3 generates RDF triples from the active domain $\Gamma_{\mathcal{D}} \subseteq \Delta \setminus \{\mathtt{null}\}$ of the database, w.l.o.g., we equivalently express the stated `SPJ` source query Q, that yields the same set of RDF triples as the original Q, as follows,

$$\pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(U)}(R_1)\,\mathsf{OP}_{U=V}\,\sigma_{\neg\mathtt{isNull}(V)\wedge\neg\mathtt{isNull}(Y)}(R_2)).$$

Note that the `SPJ` query Q is valid if and only if $\mathsf{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\mathsf{FK}(V, R_2, U, R_1) \in \Sigma|_Q$, see Defn. 1. Henceforth, we use symbol $\rightarrow^*$ to express dependency in the opposite direction of foreign key reference, i.e., we write $\mathrm{FD}_{X\rightarrow^* Y}$ to state functional dependency from $X \triangleleft Q$ to $Y \triangleleft Q$ if $\mathsf{FK}(Y, R_2, X, R_1) \in \Sigma|_Q$ or $\mathsf{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ s.t. $X, U \triangleleft R_1$ and $Y, V \triangleleft R_2$.

Lemma 2. *Let $\mathcal{R}$ be a relational schema with source constraints Σ, and let Q be an* `SPJ` *source query over $\mathcal{R}$,*

$$Q \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(Q_1 \; OP_{U=V} \; Q_2)$$

s.t. Q_1 and Q_2 are SP expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ with $X, U \triangleleft Q_1$ and $Y, V \triangleleft Q_2$ respectively, $OP \in \{\bowtie, \rightouterjoin, \leftouterjoin, \fullouterjoin\}$ and $FK(U, R_1, V, R_2) \in \Sigma|_Q$. Then, for any $X, Y \triangleleft Q$:

a. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ *s.t.* $\sigma \in \{UFD, FD, UF\}$.
b. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash UFD_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ *s.t.* $\sigma \in \{FD, UF\}$.
c. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *s.t.* $\sigma \in \{FD, UF\}$ *and* $\Sigma_{Q_2} \Vdash UFD_{V\to Y}$.
d. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash FD_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash UF_{V\to Y}$.
e. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash UF_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash FD_{V\to Y}$.
f. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash FP_{X\to U}$.

g. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash FP_{V\to Y}$ *s.t.* $\sigma \in \{UFD, FD, UF\}$.
h. $\Sigma_Q \Vdash \sigma_{Y\to^* X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ *s.t.* $\sigma \in \{UFD, UF\}$.
i. $\Sigma_Q \Vdash FP_{Y\to^* X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *s.t.* $\sigma \in \{UFD, FD, FP\}$ *and* $\Sigma_{Q_2} \Vdash UF_{Y\to V}$.
j. $\Sigma_Q \Vdash \sigma_{Y\to * X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *s.t.* $\sigma \in \{FD, UF, FP\}$ *and* $\Sigma_{Q_2} \Vdash UFD_{Y\to V}$.

On the correctness of Lemma 2, e.g., assume the case (f). Then, $\mathtt{UNQ}(V, R_2) \in \Sigma|_{Q_2}$ since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Thus, $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$ is the set of all possible constraints implication. Hence, the case (f) of Lemma 2 covers the following possible cases of constraints implication:

- $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$.

Further, by applying similar arguments and the implication rules stated in Corollary 1 to the rest of cases in Lemma 2, the correctness proof of the Lemma can be enumerated.

Example 4 *Following Example 1 and 4, assume an R2R mapping:*

$$Q \longrightarrow \langle \mathbf{f}_S(\mathtt{S\_id}), \mathtt{enrolledFor}, \mathbf{f}_C(\mathtt{C\_id})\rangle,$$

where Q is a source query $\pi_{\mathtt{S\_id}, \mathtt{C\_id}}\sigma_{\neg \mathtt{isNull}(\mathtt{S\_id}) \wedge \neg \mathtt{isNull}(\mathtt{C\_id})}(Q_1 \bowtie_{\mathtt{Code}=\mathtt{C\_id}} Q_2)$ *such that* $Q_1 = \sigma_{\neg \mathtt{isNull}(\mathtt{S\_id}) \wedge \neg \mathtt{isNull}(\mathtt{Code})}(\mathtt{student})$ *and* $Q_2 = \sigma_{\neg \mathtt{isNull}(\mathtt{C\_id})}(\mathtt{course})$. *Then,*

a. *for SP expression Q_1 :*
   - $\mathrm{att}(Q_1) = \{\mathtt{S\_id}, \mathtt{Code}\}$ *and* $\{UNQ(\mathtt{S\_id}), NN(\mathtt{S\_id}), NN(\mathtt{Code})\} \subseteq \Sigma|_{Q_1}$ *from Defn. 12.*
   - $\Sigma_{Q_1} \Vdash FD_{\mathtt{S\_id}\to\mathtt{Code}}$ *from the case (c) of Lemma 1*
b. *for SP expression Q_2 :*
   - $\mathrm{att}(Q_2) = \{\mathtt{C\_id}\}$ *and* $\{UNQ(\mathtt{C\_id}), NN(\mathtt{C\_id})\} \subseteq \Sigma|_{Q_2}$ *from Defn. 12.*
   - $\Sigma_{Q_2} \Vdash UFD_{\mathtt{C\_id}\to\mathtt{C\_id}}$ *from the case (d) of Lemma 1*
c. *finally, for SPJ expression Q:*
   - $\mathrm{att}(Q) = \{\mathtt{S\_id}, \mathtt{C\_id}\}$
   - $FK(\mathtt{Code}, \mathtt{student}, \mathtt{C\_id}, \mathtt{course}) \in \Sigma|_{Q_1} \cap \Sigma|_{Q_2}$, *i.e., Q is a valid SPJ query.*
   - $\Sigma_Q \Vdash FD_{\mathtt{S\_id}\to\mathtt{C\_id}}$ *from case (c) of Lemma 2, since*
     i. $\Sigma_{Q_1} \Vdash FD_{\mathtt{S\_id}\to\mathtt{Code}}$, *and*
     ii. $\Sigma_{Q_2} \Vdash FD_{\mathtt{C\_id}\to\mathtt{C\_id}}$ *from* $\Sigma \Vdash UFD_{\mathtt{C\_id}\to\mathtt{C\_id}} \to \Sigma \Vdash FD_{\mathtt{C\_id}\to\mathtt{C\_id}}$ *following the case (a) of Corollary 1*

6 The Constraint Rewriting

We now introduce a constraint rewriting Γ for a simple mapping $\mathcal{M}$ (Defn. 4), and prove the properties defined in Sect. 3. The constraint rewriting Γ in Defn. 15 transforms the view constraints implied by the relational source Σ (as introduced in Sect. 4 and 5) into sets of SHACL shapes. Since the semantic equivalence of generated SHACL constraints to the source constraints Σ also depends on the combination of source queries used in mappings of RDF triples, we first introduce the classification functions ι and κ to distinguish between the various cases that can occur.

Let $\mathbf{f}_C$ and $\mathbf{f}_{C'}$ be `iri` mapping templates for the respective RDF concepts C and C', and let $\mathbf{t}$ be an `iri` typing template. Let Q_C, Q_P and Q_U be the source queries of mapping Defn. 3 of an RDF concept C, object property P and datatype property U, respectively.

Definition 14 *Let $\mathcal{M}$ be a simple mapping with RDF predicates $C, C', P, U \in sch(\mathcal{M})$. Let ι and κ be classification functions that take a triple pattern of the form $\langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y)\rangle$ and $\langle \mathbf{f}_C(X), U, \mathbf{t}(Y)\rangle$ respectively, and the mapping set $\mathcal{M}$ as input, and classifies the groups of the respective source queries $(Q_C, Q_P, Q_{C'})$ and (Q_C, Q_U) as follows,*

$$\iota(\langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y)\rangle, \mathcal{M}) = \begin{cases} A & \textit{if } Q_C \neq Q_P \\ B & \textit{otherwise.} \end{cases} \quad \textit{and} \quad \kappa(\langle \mathbf{f}_C(X), U, \mathbf{t}(Y)\rangle, \mathcal{M}) = \begin{cases} A & \textit{if } Q_C \neq Q_U \\ B & \textit{otherwise.} \end{cases}$$

Let Q be a source query over a relational schema $\mathcal{R}$ with source constraint Σ. Then, we write $\Sigma_Q \Vdash \sigma_{X \twoheadrightarrow Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$ to express the dependency that is either $\Sigma_Q \Vdash \sigma_{X\to Y}$ or $\Sigma_Q \Vdash \sigma_{X \to^* Y}$ on $X, Y \triangleleft Q$.

Definition 15 (Constraint rewriting Γ) *Let $\mathcal{M}$ be a simple mapping defined over a relational schema $\mathcal{R}$ with source constraint Σ, and let ι and κ be the classification functions. Then, the constraint rewriting $\Gamma(\mathcal{M},\Sigma)$ of Σ w.r.t. $\mathcal{M}$ is a set of SHACL shapes that for each RDF concept C with mapping $Q_X \longrightarrow \langle \mathbf{f}_C(X), \mathtt{rdf{:}type}, C\rangle$, contains $\langle C, \tau_C, \phi_C\rangle$ with an implicit targetClass τ_C and conjunctive set of constraints $\phi_C = \bigwedge_{1\le i\le 3} \Phi_i$, where*

1. *for mapping m of each object property P such as $Q_{X,Y} \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y)\rangle$,*

$$\Phi_1 = \begin{cases} (\le_0\ P.\neg C') \wedge (\ge_0\ P.C') \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \lambda_1(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = A \\ (\le_0\ P.\neg C') \wedge (\ge_1\ P.C') \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \lambda_2(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = B \end{cases}$$

where

$$\lambda_1(\sigma) = \begin{cases} (\unrhd_C\ P.C') & \textit{if } \sigma = \mathtt{UFD}_{X\to Y} \\ (=_1\ P.C') & \textit{if } \sigma = \mathtt{FD}_{X\to Y} \\ (\rhd_C\ P.C') & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (\le_1\ P.C') & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases} \quad \textit{and} \quad \lambda_2(\sigma) = \begin{cases} (\unrhd_C\ P.C') & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (=_1\ P.C') & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases}$$

2. *for mapping m of each object property P such as $Q_{X,Y} \longrightarrow \langle \mathbf{f}_{C'}(X), P, \mathbf{f}_C(Y)\rangle$,*

$$\Phi_2 = \begin{cases} (\le_0\ P^-.\neg C') \wedge (\ge_0\ P^-.C') \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \delta_1(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = A \\ (\le_0\ P^-.\neg C') \wedge (\ge_1\ P^-.C') \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \delta_2(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = B \end{cases}$$

where

$$\delta_1(\sigma) = \begin{cases} (\unrhd_C\ P^-.C') & \textit{if } \sigma = \mathtt{UFD}_{X\to Y} \\ (=_1\ P^-.C') & \textit{if } \sigma = \mathtt{FD}_{X\to Y} \\ (\rhd_C\ P^-.C') & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (\le_1\ P^-.C') & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases} \quad \textit{and} \quad \delta_2(\sigma) = \begin{cases} (\unrhd_C\ P^-.C') & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (=_1\ P^-.C') & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases}$$

3. *for mapping m of each datatype property U such as $Q_{X,Y} \longrightarrow \langle \mathbf{f}_C(X), U, \mathbf{t}(Y)\rangle$,*

$$\Phi_3 = \begin{cases} (\le_0\ U.\neg \mathbf{t}) \wedge (\ge_0\ U.\mathbf{t}) \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \mu_1(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = A \\ (\le_0\ U.\neg \mathbf{t}) \wedge (\ge_1\ U.\mathbf{t}) \wedge (\bigwedge_{\Sigma_Q \Vdash \sigma} \mu_2(\sigma)) & \textit{if } \iota(head(m), \mathcal{M}) = B \end{cases}$$

where

$$\mu_1(\sigma) = \begin{cases} (\unrhd_C\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{UFD}_{X\to Y} \\ (=_1\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{FD}_{X\to Y} \\ (\rhd_C\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (\le_1\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases} \quad \textit{and} \quad \mu_2(\sigma) = \begin{cases} (\unrhd_C\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{UF}_{X\twoheadrightarrow Y} \\ (=_1\ U.\mathbf{t}) & \textit{if } \sigma = \mathtt{FP}_{X\twoheadrightarrow Y} \end{cases}$$

Observe that in Defn. 15, the first constraint components, such as $(\leq_0\ P^{\pm}.\neg C')$ and $(\leq_0\ U.\neg\mathbf{t})$ in the definitions of Φ_i, are implied by the restriction on the mapping set $\mathcal{M}$, i.e., by the fact that $\mathcal{M}$ contains exactly one mapping defining per object and datatype property predicates. The second constraint components, such as $(\geq_0\ P^{\pm}.\neg C')$ or $(\geq_1\ P^{\pm}.\neg C')$ and $(\geq_0\ U.\neg\mathbf{t})$ or $(\geq_1\ U.\neg\mathbf{t})$, in the Φ_i are implied by the combination of ι- and κ-classifications. Finally, the third constraints components $\bigwedge_{\Sigma_Q \Vdash \sigma} f(\sigma)$ s.t. $f \in \lambda_i \cup \delta_i \cup \mu_i$ for $1 \leq i \leq 2$ are implied by the source constraint Σ w.r.t. $\mathcal{M}$.

The constraint definition $\phi_C \coloneqq (\leq_0\ P^{\pm}.\neg C')$ requires all nodes n' in the graph that that are reachable from a node n s.t. $\langle n, \texttt{rdf:type}, C\rangle$ via property path $P^{\pm}$ to have a typing triple s.t. $\langle n', \texttt{rdf:type}, C'\rangle$, which is exactly what we needed for the mapped object property paths $P^{\pm}$ in the RDF graph given the restriction that set $\mathcal{M}$ contains exactly one mapping definitions per object property predicates. Thus, to extend the constraint rewriting Γ Defn. in 15 beyond the simple mapping $\mathcal{M}$, the rewriting Γ must: (i) not generate constraint components such as $(\leq_0\ P^{\pm}.\neg C')$ and $(\leq_0\ U.\neg\mathbf{t})$ when there exist more than one mapping definition per object P and datatype U properties, respectively, in the set $\mathcal{M}$, (ii) accommodate classification of all possible combinations of sources queries in the definitions of ι and κ, and (iii) revise the definitions of λ_i, δ_i and μ_i for additional consequences of Σ-implications w.r.t. the extended $\mathcal{M}$.

We now state the properties of the constraint Γ rewriting. Theorem 1 is a soundness statement that guarantees that all constraints produced by Γ will be validated by the RDF graph mapped from any valid database instance.

Theorem 1. *The constraint rewriting Γ is semantics preserving.*

Theorem 2 expresses the completeness of Γ, i.e., every SHACL constraint expressible with the schema $sch(\mathcal{M})$ of the mappings, and that is implied by Σ is implied by the generated shapes $\Gamma(\mathcal{M},\Sigma)$. This does not hold in general for SHACL constraints on predicates not in $sch(\mathcal{M})$. Finally, Theorem 3 expresses that adding mappings will never invalidate generated constraints.

Theorem 2. *The constraint rewriting Γ is maximal semantics preserving.*

Theorem 3. *The constraint rewriting Γ is monotone.*

7 Discussion

We have presented a constraint rewriting Γ for simple R2R mapping that is useful in the context of relational to RDF data transformation [14,25,21] and data integration [24,33]. Observe that simple R2R mappings can express a comprehensive catalog of useful mapping patterns studied in [8,27,26]. Simplifying simple R2R mapping further yields direct mapping [2] since that requires additional restrictions on Defn. 4; therefore, the results for our constraint rewriting for simple mappings also seamlessly holds for direct mapping [2,25,29]. In future work, we believe that it would interesting to extend our constraint rewriting Γ in two different directions: (a) for arbitrary R2R mappings, e.g., admitting the full relational algebra or arbitrary `SPJ` expressions as the source query in mapping Defn. 3, and (b) for a broader class of relational constraints such as (disjunctive) tuple and equality generating dependencies [1].

There are several approaches that map relational schemas and constraints to RDFS and OWL/Epistemic DL axioms since, with an appropriate closed world semantics, OWL can express integrity constraints. In particular, we first refer the reader to the implications of constraints in ontology-based data access platform under different names, such as protection and faithfulness in [10,11], which is equivalent to relational constraints-to-OWL, i.e., to check whether the mapped RDF of every source dataset satisfying the source constraints can be extended to a model of the mapped $\texttt{DL-Lite}_{\texttt{A}}$ axioms, and OWL-to-relational constraints, i.e., opposite of former, constraints implication in [23]. Even though these proposals for combining OWL/Epistemic DL axioms with integrity constraints have some promising results for target constraints specification in the OBDA setting, there has been no unanimity on the correct semantics.

The problem of direct mapping of source schemas and constraints into RDFS/OWL axioms has been studied in [25,7]. Sequeda et al. [25] attempted to capture the database constraints on the RDF graph resulting from direct mapping using OWL. However, the bootstrapped OWL axioms did not trigger the unsatisfiability of the directly mapped graph whenever keys are violated in the source database unless the database instance is explicitly encoded in the constraint rewriting. Further, Sequeda et al. [25, Theorem 3] established that the desirable monotonicity property of direct mapping is an obstacle to obtain a semantics preserving OWL axioms even if the database instance is explicitly encoded in the constraint rewriting. To accomplish the desired one-to-one semantics correspondence between legal relational data and RDF graph satisfying OWL axioms, Calvanese et al. [7] further extended the direct mapping of relational schemas into $\texttt{DL-Lite}_{\texttt{RDFS}}$ with disjointness – as constraints over mapped RDF graphs.

Finally, Thapa et al. [28] have studied the problem of translating database constraints into SHACL, instead of OWL/Epistemic DL, giving a direct transformation from SQL constraints to SHACL, preserving their semantics when source key constraints are satisfied [28, Theorem 2]. The present work improves on this by a) not being restricted to direct mappings, and b) lifting the requirement on satisfied key constraints.

8 Conclusion

In this paper, we study the problem of constraint rewriting for relational to RDF data transformation based on the central property of maximal semantics preservation. We translate standard SQL database constraints to shapes in the SHACL constraint language for RDF graphs. We show that our proposed rewriting Γ for the simple relational to RDF mappings satisfies the central properties of a constraint rewriting.

We believe that the propose constraint rewriting constitutes a core component of R2R mapping tools for the crucial task of constructing and maintaining a quality-assured RDF graph with SHACL constraints. The SHACL description of the generated RDF graph provides a data quality guarantee for data exchange, interoperability and query optimization. Hence, an important direction for future work will be the implementation and practical evaluation of our rewriting for relational to RDF data transformation and query optimization [32] in an ontology-based data access platform [31,33].

Acknowledgements This work is supported by the Norwegian Research Council via the SIRIUS SFI (237898). We thank Egor Kostylev for many constructive suggestions.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*, volume 8. Addison-Wesley Reading, 1995.
2. Marcelo Arenas, Alexandre Bertails, Eric Prud'hommeaux, and Juan Sequeda. A direct mapping of relational data to RDF. *W3C recommendation*, 27:1–11, 2012.
3. William Ward Armstrong. Dependency structures of data base relationships. In *IFIP congress*, volume 74, pages 580–583. Geneva, Switzerland, 1974.
4. Antonio Badia and Daniel Lemire. Functional dependencies with null markers. *The Computer Journal*, 58(5):1160–1168, 2015.
5. Catriel Beeri, Ronald Fagin, and John H Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *Proceedings of the 1977 ACM SIGMOD international conference on Management of data*, pages 47–61, 1977.
6. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases: The DL-Lite approach. In *Reasoning Web International Summer School*, pages 255–356. Springer, 2009.
7. Diego Calvanese, Wolfgang Fischl, Reinhard Pichler, Emanuel Sallinger, and Mantas Simkus. Capturing relational schemas and functional dependencies in RDFS. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.
8. Diego Calvanese, Avigdor Gal, Davide Lanti, Marco Montali, Alessandro Mosca, and Roee Shraga. Mapping patterns for virtual knowledge graphs. *arXiv preprint arXiv:2012.01917*, 2020.
9. Cristina Civili, Jose Mora, Riccardo Rosati, Marco Ruzzi, and Valerio Santarelli. Semantic analysis of R2RML mappings for ontology-based data access. In *International Conference on Web Reasoning and Rule Systems*, pages 25–38. Springer, 2016.
10. Marco Console and Maurizio Lenzerini. Data quality in ontology-based data access: The case of consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.
11. Marco Console and Maurizio Lenzerini. Epistemic integrity constraints for ontology-based data management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2790–2797, 2020.
12. Julien Corman, Juan L. Reutter, and Ognjen Savkovic. Semantics and validation of recursive SHACL [extended version], KRDB research centre technical report: KRDB18-01, 6 november 2018. *http://www.inf.unibz.it/krdb/*.
13. Julien Corman, Juan L Reutter, and Ognjen Savković. Semantics and validation of recursive SHACL. In *International Semantic Web Conference*, pages 318–336. Springer, 2018.
14. Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF mapping language. September 2012. https://www.w3.org/TR/2012/REC-r2rml-20120927/.
15. Francesco M Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic (ToCL)*, 3(2):177–225, 2002.
16. Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. SPARQL 1.1 query language. *W3C recommendation*, 21(10):778, 2013.
17. Holger Knublauch and Dimitris Kontokostas. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017. https://www.w3.org/TR/2017/REC-shacl-20170720/.
18. Domenico Lembo, José Mora, Riccardo Rosati, Domenico Fabio Savo, and Evgenij Thorstensen. Mapping analysis in ontology-based data access: Algorithms and complexity. In *International Semantic Web Conference*, pages 217–234. Springer, 2015.

19. Frank Manola, Eric Miller, Brian McBride, et al. RDF primer. *W3C recommendation*, 10(1-107):6, 2004.
20. Giansalvatore Mecca, Guillem Rull, Donatello Santoro, and Ernest Teniente. Ontology-based mappings. *Data & Knowledge Engineering*, 98:8–29, 2015.
21. Luciano Frontino de Medeiros, Freddy Priyatna, and Oscar Corcho. MIRROR: Automatic R2RML mapping generation from relational databases. In *International Conference on Web Engineering*, pages 326–343. Springer, 2015.
22. Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between OWL and relational databases. *Journal of Web Semantics*, 7(2):74–89, 2009.
23. Charalampos Nikolaou, Bernardo Cuenca Grau, Egor V Kostylev, Mark Kaminski, and Ian Horrocks. Satisfaction and Implication of Integrity Constraints in Ontology-based Data Access. In *IJCAI*, pages 1829–1835, 2019.
24. Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. In *Journal on data semantics X*, pages 133–173. Springer, 2008.
25. Juan F. Sequeda, Marcelo Arenas, and Daniel P Miranker. On directly mapping relational databases to RDF and OWL. In *Proceedings of the 21st international conference on World Wide Web*, pages 649–658, 2012.
26. Juan F. Sequeda and Daniel P Miranker. Ultrawrap mapper: A semi-automatic relational database to RDF (RDB2RDF) mapping tool. In *International semantic web conference (posters & demos)*, 2015.
27. Juan F. Sequeda, Freddy Priyatna, and Boris Villazón-Terrazas. Relational database to RDF mapping patterns. In *WOP*, 2012.
28. Ratan Bahadur Thapa and Martin Giese. A source-to-target constraint rewriting for direct mapping. In *International Semantic Web Conference*, pages 21–38. Springer, 2021.
29. Ratan Bahadur Thapa and Martin Giese. A source-to-target constraint rewriting for direct mapping (extended version). Research Report 498, Dept. of Informatics, University of Oslo, September 2021. `http://urn.nb.no/URN:NBN:no-90764`.
30. Ratan Bahadur Thapa and Martin Giese. Mapping relational database constraints to SHACL (extended version). Research Report 503, Dept. of Informatics, University of Oslo, July 2022. `http://urn.nb.no/URN:NBN:no-35645`.
31. Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. IJCAI Organization, 2018.
32. Guohui Xiao, Roman Kontchakov, Benjamin Cogrel, Diego Calvanese, and Elena Botoeva. Efficient handling of SPARQL optional for OBDA. In *International Semantic Web Conference*, pages 354–373. Springer, 2018.
33. Guohui Xiao, Davide Lanti, Roman Kontchakov, Sarah Komla-Ebri, Elem Güzel-Kalaycı, Linfang Ding, Julien Corman, Benjamin Cogrel, Diego Calvanese, and Elena Botoeva. The virtual knowledge graph system Ontop. In *International Semantic Web Conference*, pages 259–277. Springer, 2020.

A Preliminaries

A.1 SHACL constraints.

We briefly recall the *core constraints* [17] of SHACL introduced in [13]. Each SHACL constraint in a set of conditions, usually referred to as shape, defined as a triple $\langle s, \tau_s, \phi_s \rangle$ consisting of a shape IRI s, a *target definition* τ_s, and a *constraint definition* ϕ_s. The target definition τ_s is a SPARQL [16] query with one distinguished variable whose purpose is to retrieve target nodes of node shape s from the data graph, i.e., entities occurring in the RDF graph for which the constraints ϕ_s of the node shape s_c should be validated. For the purpose of our work, the constraint ϕ_s is an expression defined according to the following grammar:

$$\phi ::= \phi \wedge \phi \mid \geq_n P^{\pm}.\,\alpha \mid \leq_n P^{\pm}.\,\alpha \mid \rhd_C P^{\pm}$$
$$\alpha ::= \top \mid \ell \mid \neg\ell \mid C \mid \neg C$$

where $\top$ stands for the Boolean truth values, ℓ is an XML schema datatype, C and P are an RDF concept and property names respectively, the superscript $\pm$ stands for a property or its inverse, $n \in \mathbb{N}$, $\neg$ for negation, $(\geq_n P^{\pm}.\alpha)$ means 'must have at least n $P^{\pm}$-successor verifying α' for any $n \in \mathbb{N}$ and $(\rhd_C P^{\pm}.\alpha)$ means 'all values of $P^{\pm}$-successor must be unique [5] among instances of concept C'. As syntactic sugar, we use $(=_n P^{\pm}.\alpha)$ for $(\geq_n P^{\pm}.\alpha) \wedge (\leq_n P^{\pm}.\alpha)$, $(\rhd_C P^{\pm}.\alpha)$ for $(\leq_1 P^{\pm}.\alpha) \wedge (\rhd_C P^{\pm})$ and $(\unrhd_C P^{\pm}.\alpha)$ for $(=_1 P^{\pm}.\alpha) \wedge (\rhd_C P^{\pm})$.

A SHACL document is a set of shapes, also referred to as *shape graph*, and an RDF graph that is supposed to be validated against a shape graph is called *data graph*. *Validation of a data graph G against a shape graph S* is a two-step process: (1) retrieve the target nodes of every shape in S from G, known as *shape assignment*, and (2) check whether or not target nodes of every shape in S satisfy their constraint definition, known as *constraint validation*. A *shape assignment* β for G and S is a total function mapping nodes in G to subsets of $(S \cup \{\neg s \mid s \in S\})$ such that s and $\neg s$ can not be both in $\beta(n)$. The assignment is called total if either $s \in \beta(n)$ or $\neg s \in \beta(n)$ for each node n in G and shape $\langle s, \tau_s, \phi_s \rangle \in S$. The *semantics of constraint ϕ_s validation* is given in terms of a function $[\phi_s]^{G,n}$, i.e., an evaluation $[\phi_s]^{G,n}$ of constraint ϕ_s at node n in graph G. Given the *total shape assignment* β for G and S, and ϕ_s a constraint formula, for each nodes n in G, the evaluation $[\phi_s]^{G,n}$, i.e., either $[\phi_s]^{G,n} = 1$ (true) or $[\phi_s]^{G,n} = 0$ (false), is inductively defined as follows:

$$\begin{aligned}
[\top]^{G,n} &= 1 \\
[\ell]^{G,n} &= 1 \text{ iff } n \text{ has datatype } \ell \\
[C]^{G,n} &= 1 \text{ iff } n \text{ is the instance of SHACL class } C \\
[\neg C]^{G,n} &= 1 - [C]^{G,n} \\
[\phi_1 \wedge \phi_2]^{G,n} &= \min\{[\phi_1]^{G,n}, [\phi_2]^{G,n}\} \\
[\geq_n P^{\pm}.\alpha]^{G,n} &= 1 \text{ iff } |\{n' \mid (n,n') \in [\![P^{\pm}]\!]^G \text{ and } [\alpha]^{G,n'} = 1\}| \geq n
\end{aligned}$$

[5] `dash:uniqueValueForClassConstraintComponent` from `http://datashapes.org`

$$[\leq_n P^{\pm}.\alpha]^{G,n} = 1 \text{ iff } |\{n' \mid \text{if}(n,n') \in [\![P^{\pm}]\!]^G \text{ and } [\alpha]^{G,n'} = 1\}| \leq n$$
$$[=_n P^{\pm}.\alpha]^{G,n} = 1 \text{ iff } [\geq_n P^{\pm}.\alpha]^{G,n} = 1 \text{ and } [\leq_n P^{\pm}.\alpha]^{G,n} = 1$$
$$[\rhd_C P^{\pm}]^{G,n} = 1 \text{ iff for every node } n' \text{ s.t. } n' \neq n \text{ and } [C]^{G,n'} = 1,$$
$$\text{if } (n,n_1) \in [\![P^{\pm}]\!]^G \text{ and } (n',n_2) \in [\![P^{\pm}]\!]^G \text{ then } n_1 \neq n_2.$$
$$[\rhd_C P^{\pm}.\alpha]^{G,n} = 1 \text{ iff } [\leq_1 P^{\pm}.\alpha]^{G,n} = 1 \text{ and } [\rhd_C P^{\pm}]^{G,n} = 1$$
$$[\unrhd_C P^{\pm}.\alpha]^{G,n} = 1 \text{ iff } [=_1 P^{\pm}.\alpha]^{G,n} = 1 \text{ and } [\rhd_C P^{\pm}]^{G,n} = 1$$

where $(n,n') \in [\![P^{\pm}]\!]^G$ represents that the nodes n and n' in $\mathcal{G}$ are connected via a SHACL property path $P^{\pm}$.

We are now ready to introduce graph validation. Intuitively, a data graph G is valid against a shape graph S if one can find an assignment β for G and S complying with targets and constraints, known as *faithful assignment*. Formally, an assignment β for G and S is *faithful* if and only if $[\tau_s]^G \subseteq \beta(n)$ for each shape $\langle s, \tau_s, \phi_s \rangle \in S$, and for each node n in G:

- if $s \in \beta(n)$, then $[\phi_s]^{G,n} = 1$
- if $\neg s \in \beta(n)$, then $[\phi_s]^{G,n} = 0$

where τ_s is a SPARQL query by definition, and $[\tau_s]^G$ is the evaluation of τ_s over G. *A data graph G is valid against a shape graph S* if there is a *faithful assignment* β for G and S.

A.2 Example of Abstract syntax-to-SHACL Syntax Translation

For the complete translation of abstract syntax to SHACL syntax, we refer to the source [12, appx A.3]. For the purpose of brief examples, consider following @prefixes:

```
@prefix dash: <http :// datashapes .org/dash#> .
@prefix rdf: <http :// www.w3.org/1999/02/22−rdf−syntax−ns#> .
@prefix rdfs: <http :// www.w3.org/2000/01/rdf−schema#> .
@prefix schema: <http :// schema.org/> .
@prefix sh: <http :// www.w3.org/ns/shacl#> .
@prefix xsd: <http :// www.w3.org/2001/XMLSchema#> .
```

Then,

- $\langle \texttt{Student}, \tau_{\texttt{Student}}, \phi_C \coloneqq (=_1 \texttt{EnrolledFor}.\texttt{Course})\rangle$ can be translated into:

```
schema:Student a sh:NodeShape, rdfs:Class ;
 sh:property [
     sh:path schema:EnrolledFor ;
     sh:qualifiedValueShape [ sh:class schema:Course ] ;
     sh:qualifiedMinCount 1 ;
     sh:qualifiedMaxCount 1 ;
  ] .
```

- $\langle \texttt{Student}, \tau_{\texttt{Student}}, \phi_C \coloneqq (\leq_0 \texttt{EnrolledFor}.\neg\texttt{Course})\rangle$ can be translated into:

```
schema:Student a sh:NodeShape, rdfs:Class ;
 sh:property [
     sh:path schema:EnrolledFor ;
     sh:nodeKind sh:IRI;
     sh:class schema:Course
 ] .
```

– $\langle \texttt{Student}, \tau_{\texttt{Student}}, \phi_C \coloneqq (=_1 \texttt{EnrolledFor.Course}) \land (\leq_0 \texttt{EnrolledFor.}\neg\texttt{Course})\rangle$ can be translated into:

```
schema:Student a sh:NodeShape, rdfs:Class ;
 sh:property [
     sh:path schema:EnrolledFor ;
     sh:minCount 1;
     sh:maxCount 1;
     sh:nodeKind sh:IRI ;
     sh:class schema:Course;
 ] .
```

For the translation of abstract syntax $\rhd_C$ to dash:uniqueValueForClassConstraintComponent syntax definition, we refer to the page `http://datashapes.org`. For the purpose of brief examples,

– $\langle \texttt{Student}, \tau_{\texttt{Student}}, \phi_C \coloneqq (\rhd_C \texttt{HasName.xsd:string})\rangle$ can be translated into:

```
schema:Student a sh:NodeShape, rdfs:Class ;
sh:property [
     sh:path schema:HasName ;
     sh:datatype xsd:string ;
     sh:maxCount 1 ;
     dash:uniqueValueForClass schema:Student ;
 ] .
```

B Proof of Section 5

Lemma 1. *Let Q be a source query $\pi_{X,Y}\sigma_{\neg isNull(X)\land\neg isNull(Y)}(R)$ over a relational schema $\mathcal{R}$ with source constraints Σ, $R \in \mathcal{R}$ a relation name and $\Sigma|_Q$ the set Σ propagated to set of* att(Q)*. Then, for any $X, Y \lhd Q$,*

a. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $UNQ(X,R) \in \Sigma|_Q$.
b. $\Sigma_Q \Vdash UF_{X\to Y}$ *if* $UNQ(X,R), UNQ(Y,R) \in \Sigma|_Q$.
c. $\Sigma_Q \Vdash FD_{X\to Y}$ *if* $UNQ(X,R) \in \Sigma|_Q$ *and* $NN(X,R), NN(Y,R) \in \Sigma|_Q$.
d. $\Sigma_Q \Vdash UFD_{X\to Y}$ *if* $UNQ(X,R), UNQ(Y,R) \in \Sigma|_Q$ *and* $NN(X,R), NN(Y,R) \in \Sigma|_Q$.

Proof. Let $\mathcal{R}$ be a relational schema with source constraints Σ and $R \in \mathcal{R}$ a relation name. Let $Q = \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\land\neg\texttt{isNull}(Y)}(R)$. Then, att$(Q)$ and $\Sigma|_Q$ are recursively defined by following query declaration in Sect. 2 and Defn. 12, respectively.

a. Let $\mathcal{D}$ be an arbitrary instance of $\mathcal{R}$. Since $\neg\texttt{isNull}(X)$ and $\neg\texttt{isNull}(Y)$ filter-out the tuples $t \in R^{\mathcal{D}}$ that contain $\texttt{null}$ for any $x \in X$ and $y \in Y$ attribute from the answer set $Q^{\mathcal{D}}$ of query Q, i.e.,

$$Q^{\mathcal{D}} = \{t \in R^{\mathcal{D}} \mid t(x) \neq \texttt{null} \text{ for each } x \in X \text{ and } t(y) \neq \texttt{null} \text{ for each } y \in Y\},$$

there must be $t(X) \neq \texttt{null}$ and $t(Y) \neq \texttt{null}$ for every $t \in Q^{\mathcal{D}}$. Next, assume $\texttt{UNQ}(X, R) \in \Sigma|_Q$. Then, $\texttt{UNQ}(X, R) \in \Sigma|_Q$ must have sprung from the $\texttt{UNQ}(X, R) \in \Sigma|_R$, i.e., for every tuples $t, t' \in R^{\mathcal{D}}$ of every legal instance $\mathcal{D}$ of $\mathcal{R}$, if $t(x) = t'(x) \neq \texttt{null}$ for every $x \in X$ then $t = t'$.

Hence, the filters $\neg\texttt{isNull}(X)$ and $\neg\texttt{isNull}(Y)$ on $X, Y \lessdot Q$ and the uniqueness constraint $\texttt{UNQ}(X, R) \in \Sigma|_Q$ guarantees that if $t(X) = t'(X)$ then $t(Y) = t'(Y)$ for every tuples $t, t' \in Q^{\mathcal{D}}$ of every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., $Q^{\mathcal{D}} \models \texttt{FP}_{X\to Y}$ for each legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (a) of Defn. 13. Therefore, $\Sigma_Q \Vdash \texttt{FP}_{X\to Y}$.

b. Since $\neg\texttt{isNull}(X)$ and $\neg\texttt{isNull}(Y)$ on $X, Y \lessdot Q$, we have $t(X) \neq \texttt{null}$ and $t(Y) \neq \texttt{null}$ for every $t \in Q^{\mathcal{D}}$ of every instance $\mathcal{D}$ of $\mathcal{R}$.

Next, assume $\texttt{UNQ}(X, R), \texttt{UNQ}(X, R) \in \Sigma|_Q$. Then, $\texttt{UNQ}(X, R), \texttt{UNQ}(Y, R) \in \Sigma|_R$. Hence, $Q^{\mathcal{D}} \models \texttt{FP}_{X\to Y}$ and $Q^{\mathcal{D}} \models \texttt{FP}_{Y\to X}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, following the case (a) above. That is, $Q^{\mathcal{D}} \models \texttt{UF}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (c) of Defn. 13. Therefore, $\Sigma_Q \Vdash \texttt{UF}_{X\to Y}$.

c. Assume $\texttt{UNQ}(X, R) \in \Sigma|_Q$. Then, we have $Q^{\mathcal{D}} \models \texttt{FP}_{X\to Y}$ on $X, Y \lessdot Q$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (a).

Next, assume $\texttt{NN}(X, R), \texttt{NN}(Y, R) \in \Sigma|_Q$. Then, we have $Q^{\mathcal{D}} \models \texttt{FD}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (d) of Defn. 13. Hence, $\Sigma_Q \models \texttt{FD}_{X\to Y}$.

d. Assume $\texttt{UNQ}(X, R), \texttt{UNQ}(Y, R) \in \Sigma|_Q$. Then, we have $Q^{\mathcal{D}} \models \texttt{UF}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (b).

Assume $\texttt{NN}(X, R), \texttt{NN}(Y, R) \in \Sigma|_Q$. Then, we have $Q^{\mathcal{D}} \models \texttt{FD}_{X\to Y}$ and $Q^{\mathcal{D}} \models \texttt{FD}_{Y\to X}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (c) of Defn. 13. That is, $Q^{\mathcal{D}} \models \texttt{UFD}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, from the case (d) of Defn. 13. Hence $\Sigma_Q \Vdash \texttt{UFD}_{X\to Y}$.

Corollary 1. *Let Q be a source query $\pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(R)$ over a relational schema $\mathcal{R}$ with source constraints Σ, $R \in \mathcal{R}$ a relation name and $\Sigma|_Q$ the set Σ propagated to set of* $\mathsf{att}(Q)$*. Then, for any $X, Y \lessdot Q$,*

a. $\Sigma_Q \Vdash \mathit{UFD}_{X\to Y} \to \Sigma_Q \Vdash \mathit{FD}_{X\to Y}$ *and* $\Sigma_Q \Vdash \mathit{FD}_{X\to Y} \to \Sigma_Q \Vdash \mathit{FP}_{X\to Y}$

b. $\Sigma_Q \Vdash \mathit{UFD}_{X\to Y} \to \Sigma_Q \Vdash \mathit{UF}_{X\to Y}$ *and* $\Sigma_Q \Vdash \mathit{UF}_{X\to Y} \to \Sigma_Q \Vdash \mathit{FP}_{X\to Y}$

Proof. The proof is implicit by the following inspections.

a. From case (d) of lemma 1, we have $\Sigma_Q \Vdash \texttt{UFD}_{X\to Y}$ if $\texttt{UNQ}(X, R), \texttt{UNQ}(Y, R) \in \Sigma|_Q$ and $\texttt{NN}(X, R), \texttt{NN}(Y, R) \in \Sigma|_Q$. Then, from the case (c) of lemma 1, these conditions are sufficient for the $\Sigma_Q \Vdash \texttt{FD}_{X\to Y}$. Hence, if $\Sigma_Q \Vdash \texttt{UFD}_{X\to Y}$ then $\Sigma_Q \Vdash \texttt{FD}_{X\to Y}$, i.e., $\Sigma_Q \Vdash \texttt{UFD}_{X\to Y} \to \Sigma_Q \Vdash \texttt{FD}_{X\to Y}$.

From case (c) of lemma 1, $\Sigma_Q \Vdash \texttt{FD}_{X\to Y}$ if $\texttt{UNQ}(X, R) \in \Sigma|_Q$ and $\texttt{NN}(X, R), \texttt{NN}(Y, R) \in \Sigma|_Q$. Then, from the case (a) of lemma 1, the condition $\texttt{UNQ}(X, R) \in \Sigma|_Q$ is sufficient for the $\Sigma_Q \Vdash \texttt{FP}_{X\to Y}$. Hence, if $\Sigma_Q \Vdash \texttt{FD}_{X\to Y}$ then $\Sigma_Q \Vdash \texttt{FP}_{X\to Y}$, i.e., $\Sigma_Q \Vdash \texttt{FD}_{X\to Y} \to \Sigma_Q \Vdash \texttt{FP}_{X\to Y}$.

b. From case (d) of lemma 1, we have $\Sigma_Q \Vdash \mathtt{UFD}_{X\to Y}$ if $\mathtt{UNQ}(X,R), \mathtt{UNQ}(Y,R) \in \Sigma|_Q$ and $\mathtt{NN}(X,R), \mathtt{NN}(Y,R) \in \Sigma|_Q$. Then, from the case (b) of lemma 1, the conditions $\mathtt{UNQ}(X,R), \mathtt{UNQ}(Y,R) \in \Sigma|_Q$ is sufficient for the $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$. Hence, if $\Sigma_Q \Vdash \mathtt{UFD}_{X\to Y}$ then $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$, i.e., $\Sigma_Q \Vdash \mathtt{UFD}_{X\to Y} \to \Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$.

From case (b) of lemma 1, $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$ if $\mathtt{UNQ}(X,R), \mathtt{UNQ}(Y,R) \in \Sigma|_Q$. Then, from the case (a) of lemma 1, the condition $\mathtt{UNQ}(X,R) \in \Sigma|_Q$ is sufficient for the $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$. Hence, if $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$ then $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$, i.e., $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y} \to \Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$.

Note 1. Consider an R2R mapping over a relational schema $\mathcal{R}$ with source constraints Σ,

$$\pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R_1 \ \mathtt{OP}_{U=V}\ R_2) \longrightarrow \langle \mathbf{f}(X), \mathrm{MarriedTo}, \mathbf{f}'(Y)\rangle,$$

where $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $Y, V \triangleleft R_2$, $\mathtt{OP} \in \{\bowtie, ⟕, ⟖, ⟗\}$ and $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma$. Next, consider the source query $\pi_{X,Y}(R_1 \ \mathtt{OP}_{U=V}\ R_2)$ with/without $\sigma_{\neg\mathsf{isNull}}$ filters as follows,

- $Q_1 := \pi_{X,Y}(R_1 \ \mathtt{OP}_{U=V}\ R_2)$ and $Q_2 := \pi_{X,Y}(\sigma_{\neg\mathtt{isNull}(U)}(R_1)\ \mathtt{OP}_{U=V}\ \sigma_{\neg\mathtt{isNull}(V)}(R_2))$
- $Q_3 := \pi_{X,Y}(\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(U)}(R_1)\ \mathtt{OP}_{U=V}\ \sigma_{\neg\mathtt{isNull}(V)\wedge\neg\mathtt{isNull}(Y)}(R_2))$
- $Q_4 := \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(\sigma_{\neg\mathtt{isNull}(U)}(R_1)\ \mathtt{OP}_{U=V}\ \sigma_{\neg\mathtt{isNull}(V)}(R_2))$
- $Q_5 := \pi_{X,Y}(\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(U)}(R_1)\ \mathtt{OP}_{U=V}\ \sigma_{\neg\mathtt{isNull}(V)\wedge\neg\mathtt{isNull}(Y)}(R_2))$
- $Q_6 := \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(U)}(R_1)\ \mathtt{OP}_{U=V}\ \sigma_{\neg\mathtt{isNull}(V)\wedge\neg\mathtt{isNull}(Y)}(R_2))$

Since R2R mapping in Defn. 3 generates RDF triples from the active domain of database $\Gamma_{\mathcal{D}}$, i.e., $\Gamma_{\mathcal{D}} \subseteq \Delta\backslash\{\mathtt{null}\}$, the stated mapping with source queries from Q_1-to-Q_6 for each $\mathtt{OP} \in \{\bowtie, ⟕, ⟖, ⟗\}$ equijoin operation yields the same set of RDF triples as with the original query $\pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R_1 \ \mathtt{OP}_{U=V}\ R_2)$. Thus, henceforth, w.l.o.g., [6] we assume that every $\mathtt{SPJ}$ source query is a relational expression of the form Q_6.

Theorem 4. *Lemma 2 covers the complete space of constraints $\Sigma_Q \Vdash \sigma_{X\to Y}$ and $\Sigma_Q \Vdash \sigma_{Y\to *X}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UFP}, \mathtt{FP}\}$ implications for a valid $\mathtt{SPJ}$ source query Q.*

Proof. The proof involves a syntactic enumeration of the complete space of constraint implications for the $\mathtt{SPJ}$ source query. We show that Lemma 2 exhaustively consider the all possible case combinations for constraints $\Sigma_Q \Vdash \sigma_{X\to Y}$ and $\Sigma_Q \Vdash \sigma_{Y\to *X}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UFP}, \mathtt{FP}\}$ implications on any $X, Y \triangleleft Q$.

Let Q be an $\mathtt{SPJ}$ source query over a relational schema $\mathcal{R}$ with source constraint Σ as in Lemma 2, i.e.,

$$Q := \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(Q_1 \ \mathtt{OP}_{U=V}\ Q_2)$$

s.t. Q_1 and Q_2 are $\mathtt{SP}$ expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ [7] with $X, U \triangleleft Q_1$ and $Y, V \triangleleft Q_2$ respectively, $\mathtt{OP} \in \{\bowtie, ⟖, ⟕, ⟗\}$ and $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Then, following our assumption for the $\mathtt{SP}$-expression, i.e., $\mathtt{SP}$ source-query definition, let

$$Q_1 := \pi_{X,U}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(U)}(R_1)$$

[6] from our assumption for the (simple) mappings, see paragraph below Defn. 5, i.e. only selections considered are those that filter out non-null values.

[7] i.e., not necessarily $R_1 \neq R_2$

and

$$Q_2 \coloneqq \pi_{V,Y}\sigma_{\neg\mathtt{isNull}(V)\wedge\neg\mathtt{isNull}(Y)}(R_2).$$

For any $\Sigma_Q \Vdash \sigma_{X\to Y}$ or $\Sigma_Q \Vdash \sigma_{Y\to *X}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UP},\mathtt{FP}\}$ on $X, Y \lhd Q$, we need to consider following cases.

A. For $\Sigma_Q \Vdash \sigma_{X\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UP},\mathtt{FP}\}$, we exhaustively consider all possible $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ case combinations. Note that $\mathtt{UNQ}(V, R_2) \in \Sigma|_{Q_2}$ since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Thus, $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UF},\mathtt{FP}\}$ represents the complete space of constraint implications from $V \lhd Q_2$ to $Y \lhd Q_2$. Hence, the possible case combinations are:
   a. $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UF},\mathtt{FP}\}$,
      i. $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$
      ii. $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$
      iii. $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$
      iv. $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$
   b. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UF},\mathtt{FP}\}$,
      i. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$
      ii. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$
      iii. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$
      iv. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$
   c. $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UF},\mathtt{FP}\}$,
      i. $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$
      ii. $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$
      iii. $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$
      iv. $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$
   d. $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UF},\mathtt{FP}\}$,
      i. $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$
      ii. $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$
      iii. $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$
      iv. $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$

   The cases a(i.), b(ii.) and c(iii.) are covered by the case(a) of Lemma 2. The cases a(ii.) and a(iii.) are covered by the case(b) of Lemma 2, e.g., consider the case a(iii.). We have $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U} \to \Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ from the case (b) of Corollary 1, and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$. Then, the case (a) of Lemma 2 s.t. $\sigma = \text{UF}$ implies $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$, which exactly correspond to the result,

   ► $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$.

   Further, the cases b(i.) and c(i.) are covered by the case (c) of Lemma 2. The cases b(iii.) and c(ii.) are covered by the case (d) and (e) of Lemma 2, respectively. The cases a(iv.), b(iv.) and c(iv.) are covered by the case (g) of Lemma 2. Finally, the remaining cases d(i.)-to-d(iv.) are covered by the case (f) of Lemma 2.

B. For $\Sigma_Q \Vdash \sigma_{Y\to *X}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{FD},\mathtt{UP},\mathtt{FP}\}$, we exhaustively consider all possible $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ case combinations. Again, we have $\mathtt{UNQ}(V, R_2) \in \Sigma|_{Q_2}$ since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Hence, $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ s.t. $\sigma \in \{\mathtt{UFD},\mathtt{UF}\}$ represents the complete space of constraint implications from $Y \lhd Q_2$ to $V \lhd Q_2$. Thus, the possible case combinations are:

a. $\Sigma_{Q_1} \Vdash \texttt{UFD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ s.t. $\sigma \in \{\texttt{UFD},\texttt{UF}\}$,
   i. $\Sigma_{Q_1} \Vdash \texttt{UFD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{Y\to V}$
   ii. $\Sigma_{Q_1} \Vdash \texttt{UFD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UF}_{Y\to V}$
b. $\Sigma_{Q_1} \Vdash \texttt{FD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ s.t. $\sigma \in \{\texttt{UFD},\texttt{UF}\}$,
   i. $\Sigma_{Q_1} \Vdash \texttt{FD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{Y\to V}$
   ii. $\Sigma_{Q_1} \Vdash \texttt{FD}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UF}_{Y\to V}$
c. $\Sigma_{Q_1} \Vdash \texttt{UF}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ s.t. $\sigma \in \{\texttt{UFD},\texttt{UF}\}$,
   i. $\Sigma_{Q_1} \Vdash \texttt{UF}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{Y\to V}$
   ii. $\Sigma_{Q_1} \Vdash \texttt{UF}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UF}_{Y\to V}$
d. $\Sigma_{Q_1} \Vdash \texttt{FP}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ s.t. $\sigma \in \{\texttt{UFD},\texttt{UF}\}$,
   i. $\Sigma_{Q_1} \Vdash \texttt{FP}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{Y\to V}$
   ii. $\Sigma_{Q_1} \Vdash \texttt{FP}_{U\to X}$ and $\Sigma_{Q_2} \Vdash \texttt{UF}_{Y\to V}$

The cases a(i.) and c(ii.) are covered by the case (h) of Lemma 2. The cases a(ii.), b(ii.) and d(ii.) are covered by the case (i) of Lemma 2. Finally, the cases b(i), c(i) and d(i) are covered by the case (j) of Lemma 2.

Hence, the complete space of constraint implications for the valid SPJ query Q is covered by the Lemma 2, which concludes the correctness of the lemma.

Remark 1. Lemma 2 can be extended for an SPJ source query Q containing (outer)equijoins between more than two SP expressions, i.e.,

$$Q \coloneqq \pi_A \sigma_{\neg\texttt{isNull}(A)}(Q_1 \, \texttt{OP}_{U_1=V_2} \, Q_2 \, \texttt{OP}_{U_2=V_3} \, Q_3 \ldots \texttt{OP}_{U_{n-1}=V_n} \, Q_n)$$

such that $Q_1, Q_2, \ldots$ and Q_n are SP queries, $\texttt{OP} \in \{\bowtie, ⟗, \ltimes, \rtimes\}$ and there exist foreign key references in an order from $U_1 \triangleleft Q_1$ to $V_1 \triangleleft Q_2$, $U_2 \triangleleft Q_2$ to $V_3 \triangleleft Q_3$, etc. Note that relaxing clause (b) in Defn. 4 of simple mapping, i.e., allowing (outer)equijoins between more than two tables or SP expressions in SPJ source queries of mapping, requires such extension of lemma 2 for the (maximal) semantics-preserving rewriting in Defn. 15. We left this as future work.

Lemma 2. *Let $\mathcal{R}$ be a relational schema with source constraints Σ, and let Q be an SPJ source query over $\mathcal{R}$,*

$$Q \coloneqq \pi_{X,Y} \sigma_{\neg\texttt{isNull}(X) \wedge \neg\texttt{isNull}(Y)}(Q_1 \, OP_{U=V} \, Q_2)$$

s.t. Q_1 and Q_2 are SP expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ with $X, U \triangleleft Q_1$ and $Y, V \triangleleft Q_2$ respectively, $OP \in \{\bowtie, ⟗, \ltimes, \rtimes\}$ and $FK(U, R_1, V, R_2) \in \Sigma|_Q$. Then, for any $X, Y \triangleleft Q$:

a. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ *s.t.* $\sigma \in \{UFD, FD, UF\}$.
b. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash UFD_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ *s.t.* $\sigma \in \{FD, UF\}$.
c. $\Sigma_Q \Vdash \sigma_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *s.t.* $\sigma \in \{FD, UF\}$ *and* $\Sigma_{Q_2} \Vdash UFD_{V\to Y}$.
d. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash FD_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash UF_{V\to Y}$.
e. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash UF_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash FD_{V\to Y}$.
f. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash FP_{X\to U}$.
g. $\Sigma_Q \Vdash FP_{X\to Y}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ *and* $\Sigma_{Q_2} \Vdash FP_{V\to Y}$ *s.t.* $\sigma \in \{UFD, FD, UF\}$.
h. $\Sigma_Q \Vdash \sigma_{Y\to^* X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *and* $\Sigma_{Q_2} \Vdash \sigma_{Y\to V}$ *s.t.* $\sigma \in \{UFD, UF\}$.

i. $\Sigma_Q \Vdash FP_{Y\to^* X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *s.t.* $\sigma \in \{UFD, FD, FP\}$ *and* $\Sigma_{Q_2} \Vdash UF_{Y\to V}$.
j. $\Sigma_Q \Vdash \sigma_{Y\to * X}$ *if* $\Sigma_{Q_1} \Vdash \sigma_{U\to X}$ *s.t.* $\sigma \in \{FD, UF, FP\}$ *and* $\Sigma_{Q_2} \Vdash UFD_{Y\to V}$.

Proof. Let $\mathcal{R}$ be a relational schema with source constraints Σ, and let Q be an SPJ source query over $\mathcal{R}$ as in Lemma 2, i.e.,

$$Q := \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_1 \; \texttt{OP}_{U=V} \; Q_2).$$

Then, following our assumption for the SP relational expressions, i.e., SP source query, in the mapping definition 3, let

$$Q_1 := \pi_{X,U}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(U)}(R_1)$$

and

$$Q_2 := \pi_{V,Y}\sigma_{\neg\texttt{isNull}(V)\wedge\neg\texttt{isNull}(Y)}(R_2),$$

where $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $Y, V \triangleleft R_2$. Next observe that $\texttt{UNQ}(U, R_2) \in \Sigma|_{Q_2}$ since $\texttt{FK}(V, R_1, U, R_2) \in \Sigma|_Q$. Thus, $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\texttt{UFD}, \texttt{FD}, \texttt{UF}, \texttt{FP}\}$ and $\Sigma_{Q_2} \Vdash \sigma_{Y\to^* V}$ s.t. $\sigma \in \{\texttt{UFD}, \texttt{UF}\}$, respectively, represent the complete space of constraint implications on $Y, V \triangleleft R_2$.

Let $\mathcal{D}$ be an arbitrary instance of $\mathcal{R}$. Then, observe that the filters $\neg\texttt{isNull}(X)$ and $\neg\texttt{isNull}(Y)$ on the $X, Y \triangleleft Q$ filter-out the tuples $t \in Q^{\mathcal{D}}$ that contain null for any $x \in X$ and $y \in Y$ attribute from the answer set $Q^{\mathcal{D}}$, i.e.,

$$Q^{\mathcal{D}} = \{t \in Q'^{\mathcal{D}} \mid \; t(x) \neq \texttt{null} \text{ for each } x \in X \text{ and } t(y) \neq \texttt{null} \text{ for each } y \in Y\},$$

where $Q' = (Q_1 \texttt{OP}_{U=V} Q_2)$. In addition, $X \subseteq \mathsf{att}(Q) \cap \mathsf{att}(Q_1)$ and $Y \subseteq \mathsf{att}(Q) \cap \mathsf{att}(Q_2)$, therefore, the answer set $Q^{\mathcal{D}}$ of the SPJ source query Q with any $\texttt{OP} \in \{⟕, ⟖, ⟗\}$ operations contain same set of tuples as of the query Q with equi-join $\bowtie$ operation [8], i.e., indeed, w.l.o.g., $Q = \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_1 \;\bowtie_{U=V}\; Q_2)$.

The proof of each case (a)-to (k) of Lemma 2 is as follows:

a. For the case σ = UFD, the claim $\Sigma_Q \Vdash \texttt{UFD}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \texttt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{V\to Y}$ holds as follows.

   Let $\Sigma_{Q_1} \Vdash \texttt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \texttt{UFD}_{V\to Y}$. Then, referential integrity constraint $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ from $U \triangleleft Q_1$ to $V \triangleleft Q_2$ guarantees that for each tuple $t_1 \in {Q_1}^{\mathcal{D}}$ of every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists a unique tuple $t_2 \in {Q_2}^{\mathcal{D}}$ such that

[8] i.e., observe that $\neg\texttt{isNull}(X)$ and $\neg\texttt{isNull}(Y)$ filter-out all dangling tuples of $Q^{\mathcal{D}}$ if exist, i.e., every tuple in ${Q_1}^{\mathcal{D}}$ that has no matching tuple in ${Q_2}^{\mathcal{D}}$ and vice-versa. In addition, note that the Q is a valid SPJ source query, i.e., $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Hence, the Q with $⟕_{U=V}$ left-outer join operation either does not contain the dangling tuples or filter-out by the filter $\neg\texttt{isNull}(Y)$ over the $Y \subseteq \mathsf{att}(Q) \cap \mathsf{att}(Q_2)$, thus, equivalent to the Q with join $\bowtie_{U=V}$. Similarly, the dangling tuples of Q with right-outer join $⟖_{U=V}$ operation, if exist,i.e., emerging from the Q_2, filter-out by the filter $\neg\texttt{isNull}(X)$ over the $X \subseteq \mathsf{att}(Q) \cap \mathsf{att}(Q_1)$, thus, equivalent to the Q with $\bowtie_{U=V}$. Finally, the equivalence of the query Q with full-outer join $⟗_{U=V}$ operation to the Q with equi-join $\bowtie_{U=V}$ follows from the previous arguments.

$t_1(U) = t_2(V)$. This is a semantic restriction imposed on the evaluation of every extension of R_1 and R_2 over every legal instance $\mathcal{D}$ of $\mathcal{R}$.

W.l.o.g., let $Q = \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}Q'$ such that $Q' = (Q_1 \bowtie_{U=V} Q_2)$. Then, for each tuple t in the answer set $Q'^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exist $t_1 \in Q_1{}^{\mathcal{D}}$ and $t_2 \in Q_2{}^{\mathcal{D}}$ with $t(U) = t_1(U) = t_2(V)$, i.e.,

$$
\begin{aligned}
Q'^{\mathcal{D}} = \{t \mid\ & t_1 \in Q_1{}^{\mathcal{D}} \text{ and } t_2 \in Q_2{}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \text{ for every } u \in U \text{ and } v \in V, \\
& t(x) = t_1(x) \text{ for every } x \in \mathsf{att}(Q_1) \text{ and} \\
& t(y) = t_2(y) \text{ for every } y \in \mathsf{att}(Q_2)\}.
\end{aligned}
$$

Next, $\mathsf{att}(Q') = \mathsf{att}(Q_1) \cup \mathsf{att}(Q_2)$ since $Q' = (Q_1 \bowtie_{U=V} Q_2)$. Then,

i. $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to U}$ since $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$.
ii. $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to V}$ since $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to U}$, and for each $t \in Q'^{\mathcal{D}}$, there exists a unique tuple $t_2 \in Q_2{}^{\mathcal{D}}$ such that $t(U) = t_2(V)$.
iii. $\Sigma_{Q'} \Vdash \mathtt{UFD}_{V\to Y}$ since $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$.
iv. $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to Y}$ since $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to V}$ and $\Sigma_{Q'} \Vdash \mathtt{UFD}_{V\to Y}$.

Thus, $\Sigma_Q \Vdash \mathtt{UFD}_{X\to Y}$ since $Q = \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}Q'$ and $\Sigma_{Q'} \Vdash \mathtt{UFD}_{X\to Y}$.

For the cases $\sigma = \{\mathtt{FD},\mathtt{UF}\}$, i.e., the claims $\Sigma_Q \Vdash \sigma_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$, hold as in the case σ = UFD, by following similar arguments.

b. For the case σ = FD, the claim $\Sigma_Q \Vdash \mathtt{FD}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$ holds as follows.

Note that $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U} \to \Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ from the case (a) of Corollary 1. Then, $\Sigma_Q \Vdash \mathtt{FD}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$, holds by following similar argument as in the previous case (a).

For the case σ = UF, the claim $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$ holds as follows. We have $\Sigma_{Q_1} \Vdash \mathtt{UFD}_{X\to U} \to \Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ from the case (b) of Corollary 1. Then, $\Sigma_Q \Vdash \mathtt{UF}_{X\to Y}$ whenever $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$ holds by following similar arguments as in the previous case.

c. For the cases $\sigma \in \{\mathtt{FD},\mathtt{UF}\}$, the claims $\Sigma_Q \Vdash \sigma_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{V\to Y}$ hold by following similar arguments as in the case (b).

d. The claim, $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{UF}_{V\to Y}$ holds as follows.

Note that $\Sigma_{Q_1} \Vdash \mathtt{FD}_{X\to U} \to \Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ and $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U} \to \Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ from the case (a) and (b) of Corollary 1, respectively. Next, as in the case (a) above, w.l.o.g., let $Q = \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}Q'$ s.t. $Q' = (Q_1 \bowtie_{U=V} Q_2)$. We have $\mathsf{att}(Q') = \mathsf{att}(Q_1) \cup \mathsf{att}(Q_2)$ since $Q' = (Q_1 \bowtie_{U=V} Q_2)$. Then,

i. $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to U}$ since $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$.
ii. $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to V}$ since $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to U}$, and for each $t \in Q'^{\mathcal{D}}$, there exists a unique tuple $t_2 \in Q_2{}^{\mathcal{D}}$ such that $t(U) = t_2(V)$.

iii. $\Sigma_{Q'} \Vdash \mathtt{FP}_{V\to Y}$ since $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$.
iv. $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to Y}$ since $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to V}$ and $\Sigma_{Q'} \Vdash \mathtt{FP}_{V\to Y}$.

Thus, $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ since $Q = \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}Q'$ and $\Sigma_{Q'} \Vdash \mathtt{FP}_{X\to Y}$.

e. The claim $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{UF}_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FD}_{V\to Y}$ holds by following similar arguments as in the previous case (d).

f. The claims $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ hold as follows.

Recall that $\mathtt{UNQ}(U, R_2) \in \Sigma|_{Q_2}$ since $\mathtt{FK}(V, R_1, U, R_2) \in \Sigma|_Q$. Hence $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$ represents the complete space of constraint implications from $V \triangleleft Q_2$ to $Y \triangleleft Q_2$. Then, $\Sigma_Q \Vdash \mathtt{FP}_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \mathtt{FP}_{X\to U}$ (i.e., and $\Sigma_{Q_2} \Vdash \sigma_{V\to Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$) hold by following similar arguments as in the case (b)-to-(e) above.

g. For the case $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}\}$, the claims $\Sigma_Q \Vdash \sigma_{X\to Y}$ if $\Sigma_{Q_1} \Vdash \sigma_{X\to U}$ and $\Sigma_{Q_2} \Vdash \mathtt{FP}_{V\to Y}$ hold by following similar observations as in the case (f).

h. The claims hold by follows similar arguments as in the case (a) when $\sigma = \{\mathtt{UFD}, \mathtt{UF}\}$.
i. The claims hold by following similar observations as in the case (d).
j. The claims holds by following similar observations as in the case (c) and (e).

C Proof of Section 6

Theorem 1. *The constraint rewriting Γ is semantics preserving.*

Proof. The proof requires that constraint rewriting Γ in Defn. 15 satisfies the condition stated in Definition 9, i.e., given a mapping set $\mathcal{M}$ defined over a relational schema $\mathcal{R}$ with source constraint Σ and an arbitrary instance $\mathcal{D}$ of $\mathcal{R}$:

$$\mathcal{D} \models \Sigma \longrightarrow \mathcal{M}(\mathcal{D}) \models \mathcal{S},$$

where $\mathcal{M}(\mathcal{D})$ from Definition 5, $\mathcal{S} = \Gamma(\mathcal{M}, \Sigma)$ from Definition 8 and Γ is constraint rewriting in Defn. 15.

Let $\mathcal{D}$ be an instance of relational schema $\mathcal{R}$ with $\mathcal{D} \models \Sigma$. Then, there must be $\mathcal{M}(\mathcal{D}) \models \mathcal{S}$. In order to prove $\mathcal{M}(\mathcal{D}) \models \mathcal{S}$, we next show that there exists an *assignment* β for $\mathcal{G}$ and $\mathcal{S}$ complying with both *targets* and *constraints*, known as *faithful assignment*.

- For each shape s in the SHACL document $\mathcal{S}$ s.t. $\langle C, \tau_C, \phi_C \rangle$ [9], there exists a mapping of an RDF concept C s.t.,

$$Q_C \longrightarrow \langle \mathbf{f}_C(X), \mathtt{rdf{:}type}, C \rangle,$$

[9] Note that τ_C is implicit targetClass declaration, i.e., C a sh:NodeShape, rdfs:Class .

where Q_C is an SP/SPJ source query Q over $\mathcal{R}$ with $X \triangleleft Q$, see the rewriting Γ in Defn. 15.

Next, given the $\mathcal{M}(\mathcal{D})$ graph and shape assignment β, if there exists a mapping of a node n in graph $\mathcal{M}(\mathcal{D})$ s.t. $\langle n, \texttt{rdf:type}, C\rangle$, then there exists shape assignment $s \in \beta(n)$ for the node n and shape s by following the target definition τ_C. An inspection of $\mathcal{M}(\mathcal{D})$, i.e.,

$$\{\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C\rangle \in \mathcal{G} \mid \{Q \longrightarrow \langle \mathbf{f}_C(X), \texttt{rdf:type}, C\rangle\} \in \mathcal{M}_C, X \triangleleft Q \;\&\; t \in Q^{\mathcal{D}}\}$$

further reveals that the node n is generated by the injective mapping template $\mathbf{f}_C$ from $t(X)$, i.e., a tuple $t \in Q^{\mathcal{D}}$ restricted to $X \triangleleft Q$. Then, from the *semantics of shape validation*, every node n in $\mathcal{M}(\mathcal{D})$ s.t. $s \in \beta(n)$ validates against the shape s if and only if $[\phi_C]^{\mathcal{G},n}$=1. Hence, for every node n in $\mathcal{M}(\mathcal{D})$ s.t. $s \in \beta(n)$, we next show that $[\phi_C]^{\mathcal{G},n}$=1 for all possible definition of the constraint ϕ_C.

Starting by the base cases:

1. Let ϕ_C be a property shape definition for an object property P. Then, from the rewriting Γ in Defn. 15, there exist mapping of an object property $P \in sch(\mathcal{M})$ s.t.,

$$Q_P \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y)\rangle,$$

   where Q_P is an SPJ source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$, and mapping of an RDF concept C' s.t.,

$$Q_{C'} \longrightarrow \langle \mathbf{f}_{C'}(Y), \texttt{rdf:type}, C'\rangle,$$

   where $Q_{C'}$ is an SP/SPJ source query Q over $\mathcal{R}$ with $Y \triangleleft Q$. Further, from the Defn. of SPJ source query in Lemma 2, there must be

$$Q_P \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_C \;\texttt{OP}_{U=V}\; Q_{C'})$$

   s.t. Q_C and $Q_{C'}$ are SP relational expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ with $X, U \triangleleft Q_C$ and $Y, V \triangleleft Q_{C'}$ respectively, and $\texttt{OP} \in \{\bowtie, ⟗, ⟕, ⟖\}$. In addition, there must be either $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ since every Q_P is a valid SPJ query.

   a. For object property path P:
      i. Let $\phi_C \coloneqq (\trianglerighteq_C\; P.\, C')$. Then, from the definition of classification ι function, there are two mapping cases to consider:

         A. Let $Q_C \neq Q_P$, i.e., either $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$. For the mapping case $Q_C \neq Q_P \neq Q_{C'}$, let

$$Q_C \coloneqq \pi_X\sigma_{\neg\texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} \coloneqq \pi_Y\sigma_{\neg\texttt{isNull}(Y)}(R_2),$$

         such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $V, Y \triangleleft R_2$. Then, from the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X\rightarrow Y}$. Further from the case(a) of Lemma 2, there must be

$\Sigma_{Q_C} \Vdash \texttt{UFD}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{UFD}_{V \to Y}$, and $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$.

Since $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X \to Y}$, ${Q_P}^{\mathcal{D}} \models \texttt{UFD}_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_P}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$,
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$, and
- $\texttt{NN}(X, R_1), \texttt{NN}(Y, R_2) \in \Sigma|_{Q_P}$.

Then, w.l.o.g., $Q_P \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X) \wedge \neg\texttt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'})$ following arguments from the proof of Lemma 2. Thus, for every legal instance $\mathcal{D}$ of $\mathcal{R}$,

$$\begin{aligned} {Q_P}^{\mathcal{D}} = \{t \mid t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \texttt{null} \\ \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \texttt{null} \text{ and} \\ t(y) = t_2(y) \neq \texttt{null} \text{ for every } x \in X \text{ and } y \in Y\}. \end{aligned}$$

Next, since $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\texttt{NN}(X, R_1), \texttt{NN}(U, R_1) \in \Sigma|_{Q_C}$, for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists a tuple $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$ and vice-versa. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is an injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of exactly one unique node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(X)), \texttt{rdf:type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is an injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ to $Y \triangleleft Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Therefore, every node n in $\mathcal{M}(\mathcal{D})$ s.t. $s \in \beta(n)$ satisfies both the $\trianglerighteq_C$ cardinality and the range typing C' requirement for the property path P, i.e., $[(\trianglerighteq_C P.C')]^{\mathcal{G},n}$=1.

Now, for the mapping case $Q_C \neq Q_P = Q_{C'}$, let

$$Q_C \coloneqq \pi_X\sigma_{\neg\texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same } \texttt{SPJ}\text{-query},$$

where $R_1 \in \mathcal{R}$ is relation name with $X, U \triangleleft R_1$. As in the mapping case above, from the rewriting function λ_1 in Defn. 15, we have $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X \to Y}$, i.e., for every tuples $t, t' \in {Q_P}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$,

- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$, and
- $\mathtt{NN}(X, R_1), \mathtt{NN}(Y, R_2) \in \Sigma|_{Q_P}$.

In addition [10],

$$Q_P{}^{\mathcal{D}} = \{t \mid t_1 \in Q_C{}^{\mathcal{D}} \text{ and } t_2 \in Q_{C'}{}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \mathtt{null} \\ \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \mathtt{null} \text{ and} \\ t(y) = t_2(y) \neq \mathtt{null} \text{ for every } x \in X \text{ and } y \in Y\}$$

and for every tuple $t_1 \in Q_C{}^{\mathcal{D}}$ there exists a tuple $t \in Q_P{}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$ and vice-versa. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \mathtt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is an injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of exactly one unique node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(X)), \mathtt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, $[(\trianglerighteq_C P.C')]^{\mathcal{G},n}$=1.

B. Let $Q_C = Q_P$, i.e., either $Q_C = Q_P \neq Q_{C'}$ or $Q_C = Q_P = Q_{C'}$. Assume the mapping case $Q_C = Q_P \neq Q_{C'}$ such that,

$$Q_C = Q_P, \text{ i.e., same } \mathtt{SPJ} \text{ query, and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg \mathtt{isNull}(Y)}(R_2),$$

where $R_2 \in \mathcal{R}$ is relation name with $V, Y \triangleleft R_2$. Then, from the rewriting function λ_2 in Defn. 15, we have $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \twoheadrightarrow Y}$. First, assume $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$. Then, from the case(a) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \mathtt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{UF}_{V \to Y}$, and $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Note that, since $\Sigma_{Q_C} \Vdash \mathtt{UFD}_{X \to U} \to \Sigma_{Q_C} \Vdash \mathtt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{UFD}_{V \to Y} \to \Sigma_{Q_{C'}} \Vdash \mathtt{UF}_{V \to Y}$, the case(a) also covers the relevant cases (b) and (c) of Lemma 2.

Since $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$, i.e., for every tuples $t, t' \in Q_P{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:
- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$ and
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$,

and

$$Q_P{}^{\mathcal{D}} = \{t \mid t_1 \in Q_C{}^{\mathcal{D}} \text{ and } t_2 \in Q_{C'}{}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \mathtt{null} \\ \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \mathtt{null} \text{ and} \\ t(y) = t_2(y) \neq \mathtt{null} \text{ for every } x \in X \text{ and } y \in Y\}$$

[10] w.l.o.g., $Q_P \coloneqq \pi_{X,Y} \sigma_{\neg \mathtt{isNull}(X) \wedge \neg \mathtt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'})$.

for every $t \in Q_P{}^{\mathcal{D}}$ there exist a tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ s.t. $t(Y) = t_2(Y)$. Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of exactly one unique node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ to $Y \lessdot Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lessdot Q_P$. Hence, $[(\trianglerighteq_C P. C')]^{\mathcal{G},n}$=1.

Next, assume $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to^* Y}$. Then, following the case (h) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \texttt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{UF}_{V \to Y}$, and $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$. Note that the case (h) also covers relevant case (i) of Lemma 2 since $\Sigma_{Q_{C'}} \Vdash \texttt{UFD}_{V \to Y} \to \Sigma_{Q_{C'}} \Vdash \texttt{UF}_{V \to Y}$.

As in the previous case, since $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to Y}$ i.e., for every tuples $t, t' \in Q_P{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:
- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$ and
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$

and

$$\begin{aligned} Q_P{}^{\mathcal{D}} = \{t \mid\ & t_1 \in Q_C{}^{\mathcal{D}} \text{ and } t_2 \in Q_{C'}{}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \texttt{null} \\ & \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \texttt{null} \text{ and} \\ & t(y) = t_2(y) \neq \texttt{null} \text{ for every } x \in X \text{ and } y \in Y\} \end{aligned}$$

for every $t \in Q_P{}^{\mathcal{D}}$ there exist a tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ s.t. $t(Y) = t_2(Y)$. Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of exactly one unique node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ to $Y \lessdot Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lessdot Q_P$. Hence, $[(\trianglerighteq_C P. C')]^{\mathcal{G},n}$=1.

Similarly, assume $Q_C = Q_P = Q_{C'}$. Then, by following the same reasoning arguments as in the case above, we get $[(\trianglerighteq_C\ P.C')]^{\mathcal{G},n}$=1.

ii. Let $\phi_C \coloneqq (=_1\ P.C')$. Then, from the function ι definition, there are two mapping cases to consider:

A. Let $Q_C \neq Q_P$, i.e., $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$. For the former case, assume $Q_C \neq Q_P \neq Q_{C'}$, s.t.,

$$Q_C \coloneqq \pi_X \sigma_{\neg \mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg \mathtt{isNull}(Y)}(R_2),$$

where $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $V, Y \triangleleft R_2$. Similarly, for the latter case, assume $Q_C \neq Q_P = Q_{C'}$ s.t.,

$$Q_C \coloneqq \pi_X \sigma_{\neg \mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same } \mathtt{SPJ}\text{-query}.$$

Then, in both cases, from the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathtt{FD}_{X \to Y}$. Further from the case (a) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \mathtt{FD}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{FD}_{V \to Y}$, and $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Note that, since $\Sigma_{Q_C} \Vdash \mathtt{UFD}_{X \to U} \to \Sigma_{Q_C} \Vdash \mathtt{FD}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{UFD}_{V \to Y} \to \Sigma_{Q_{C'}} \Vdash \mathtt{FD}_{V \to Y}$, the case (a) also covers relevant cases (b) and (c) of Lemma 2.

Since $\Sigma_{Q_P} \Vdash \mathtt{FD}_{X \to Y}$, ${Q_P}^{\mathcal{D}} \models \mathtt{FD}_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_P}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$, and
- $\mathtt{NN}(X, R_1), \mathtt{NN}(Y, R_2) \in \Sigma|_{Q_P}$.

Then, w.l.o.g., $Q_P \coloneqq \pi_{X,Y} \sigma_{\neg \mathtt{isNull}(X) \wedge \neg \mathtt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'})$ following arguments from the proof of Lemma 2. Thus, for every legal instance $\mathcal{D}$ of $\mathcal{R}$,

$$\begin{aligned} {Q_P}^{\mathcal{D}} = \{t \mid t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \mathtt{null} \\ \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \mathtt{null} \text{ and} \\ t(y) = t_2(y) \neq \mathtt{null} \text{ for every } x \in X \text{ and } y \in Y\}. \end{aligned}$$

In addition, since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\mathtt{NN}(X, R_1), \mathtt{NN}(U, R_1) \in \Sigma|_{Q_C}$, for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists a tuple $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$ and vice-versa. Thus, in the former case, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \mathtt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of exactly one node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(X)), \mathtt{rdf:type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ to $Y \triangleleft Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Likewise, in the latter case, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of exactly one node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(X)), \texttt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Therefore, in the both case, $[(=_1\ P.\, C')]^{\mathcal{G},n}$=1.

B. Let $Q_C = Q_P$, i.e., $Q_C = Q_P \neq Q_{C'}$ or $Q_C = Q_P = Q_{C'}$. Assume $Q_C = Q_P \neq Q_{C'}$ such that,

$$Q_C \text{ and } Q_P \text{ are same } \texttt{SPJ}\text{-query, and } Q_{C'} := \pi_Y \sigma_{\neg\texttt{isNull}(Y)}(R_2),$$

where $R_2 \in \mathcal{R}$ is relation name with $V, Y \triangleleft R_2$. Then, from the rewriting function λ_2 in Defn. 15, we have $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \twoheadrightarrow Y}$.

First, assume $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to Y}$. Then, from the cases (d), (e) and (f) of Lemma 2, there could be any of the following case with $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$:
- $\Sigma_{Q_C} \Vdash \texttt{FD}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{UF}_{V \to Y}$,
- $\Sigma_{Q_C} \Vdash \texttt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{FD}_{V \to Y}$.
- $\Sigma_{Q_C} \Vdash \texttt{FD}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \sigma_{V \to Y}$ s.t. $\sigma \in \{\texttt{UFD}, \texttt{FD}, \texttt{UF}, \texttt{FP}\}$,

For the simplicity, w.l.o.g., we concentrate on the case
- $\Sigma_{Q_C} \Vdash \texttt{FP}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{FP}_{V \to Y}$,

and exclude the rest of cases since the same arguments apply on those cases as well.

Since $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to Y}$, i.e., for every tuples $t, t' \in Q_P{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:
- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$,

and

$$
\begin{aligned}
{Q_P}^{\mathcal{D}} = \{t \mid\ & t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \texttt{null} \\
& \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \texttt{null} \text{ and} \\
& t(y) = t_2(y) \neq \texttt{null} \text{ for every } x \in X \text{ and } y \in Y\}
\end{aligned}
$$

with $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\texttt{UNQ}(V, R_2) \in \Sigma|_{Q_C}$ [11], for every $t \in {Q_P}^{\mathcal{D}}$ there exist exactly one tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ s.t. $t(Y) = t_2(Y)$. Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of exactly one node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ to $Y \lessdot Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \lessdot Q_P$.

Next, assume $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to^* Y}$. Then, following the case (i) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \texttt{FP}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{FP}_{V \to Y}$, and $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$. Note that, since $\Sigma_{Q_{C'}} \Vdash \texttt{UFD}_{V \to Y} \to \Sigma_{Q_{C'}} \Vdash \texttt{UF}_{V \to Y} \to \Sigma_{Q_{C'}} \Vdash \texttt{FP}_{V \to Y}$, the case (i) covers rest and the case (j) of Lemma 2.

Since $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to^* Y}$ i.e., for every tuples $t, t' \in {Q_P}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$,

and

$$
\begin{aligned}
{Q_P}^{\mathcal{D}} = \{t \mid\ & t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \texttt{null} \\
& \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \texttt{null} \text{ and} \\
& t(y) = t_2(y) \neq \texttt{null} \text{ for every } x \in X \text{ and } y \in Y\},
\end{aligned}
$$

for every $t \in {Q_P}^{\mathcal{D}}$ there exist a tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ s.t. $t(Y) = t_2(Y)$. Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of exactly one node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle$$

[11] arguments hold for both cases: (a) $\texttt{NN}(U, R_1) \in \Sigma|_{Q_C}$, i.e., in case of $\Sigma_{Q_C} \Vdash \texttt{FD}_{X \to U}$, and (b) $\texttt{NN}(U, R_1) \notin \Sigma|_{Q_C}$, i.e., in case of $\Sigma_{Q_C} \Vdash \texttt{UF}_{X \to U}$ and $\Sigma_{Q_C} \Vdash \texttt{FP}_{X \to U}$.

where $\mathbf{f}_{C'}$ is an injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ to $Y \triangleleft Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_C$ is an injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Hence, in the case $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \twoheadrightarrow Y}$, there is $[(=_1 P. C')]^{\mathcal{G},n}$=1.

Similarly, for the latter mapping case $Q_C = Q_P = Q_{C'}$, we obtain $[(=_1 P. C')]^{\mathcal{G},n}$=1 by following the same arguments as in the previous mapping case.

iii. Let $\phi_C \coloneqq (\rhd_C P. C')$. Then, from the constraint rewriting Γ in Defn. 15, there is only two mapping cases to consider $Q_C \neq Q_P$, i.e., either $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$.

For the former case $Q_C \neq Q_P \neq Q_{C'}$, assume

$$Q_C \coloneqq \pi_X \sigma_{\neg\mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg\mathtt{isNull}(Y)}(R_2),$$

such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $V, Y \triangleleft R_2$. Similarly, for the latter case $Q_C \neq Q_P = Q_{C'}$, assume

$$Q_C \coloneqq \pi_X \sigma_{\neg\mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same SPJ-query.}$$

Then, in both cases, from the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \twoheadrightarrow Y}$.

First, assume the case $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$. Further from the case (a) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \mathtt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{UF}_{V \to Y}$, and $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$. Note that the case (a) also covers relevant cases (b) and (c) of Lemma 2, i.e., case $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$.

Since $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$, there is ${Q_P}^{\mathcal{D}} \models \mathtt{UF}_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_P}^{\mathcal{D}}$:
- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$, and
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$.

Note that, w.l.o.g. $Q_P \coloneqq \pi_{X,Y} \sigma_{\neg\mathtt{isNull}(X) \wedge \neg\mathtt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'})$. Then, for every legal instance $\mathcal{D}$ of $\mathcal{R}$,

$$\begin{aligned} {Q_P}^{\mathcal{D}} = \{t \mid\, & t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \mathtt{null} \\ & \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \mathtt{null} \text{ and} \\ & t(y) = t_2(y) \neq \mathtt{null} \text{ for every } x \in X \text{ and } y \in Y\}. \end{aligned}$$

Further, since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\Sigma_C \Vdash \mathtt{UF}_{X \to U}$ (i.e., $\mathtt{NN}(U, R_1) \notin \Sigma|_{Q_C}$), for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, in the former case, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is an injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of at most one unique node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(X)), \texttt{rdf:type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is an injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ to $Y \triangleleft Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Similarly, in the latter mapping case $Q_C \neq Q_P = Q_{C'}$, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of at most one unique node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(X)), \texttt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Hence, in the both case, $[(\triangleright_C\ P.\, C')]^{\mathcal{G},n}$=1.

Second, assume the case $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to^* Y}$. Further from the case (h) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \texttt{UF}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{UF}_{V \to Y}$, and $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$. Note that the case (h) also covers the relevant case (j) of Lemma 2.

Then, following the same procedure as in the previous constraint case, given $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ and $\Sigma_C \Vdash \texttt{UF}_{X \to U}$ (i.e., $\texttt{UNQ}(U, R_1) \in \Sigma|_{Q_C}$), for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$.

Hence, with the same arguments as in the first constraint case in the both former and latter mapping cases, we arrive at $[(\triangleright_C\ P.\, C')]^{\mathcal{G},n}$=1.

iv. Let $\phi_C \coloneqq (\leq_1 P.C')$. Then, there is only one mapping case $Q_C \neq Q_P$ to consider based on the rewriting λ_1 rules, i.e., either $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$. For the former case $Q_C \neq Q_P \neq Q_{C'}$, let

$$Q_C \coloneqq \pi_X \sigma_{\neg \mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg \mathtt{isNull}(Y)}(R_2),$$

such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \lhd R_1$ and $V, Y \lhd R_2$. Similarly, for the latter case $Q_C \neq Q_P = Q_{C'}$, let

$$Q_C \coloneqq \pi_X \sigma_{\neg \mathtt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same } \mathtt{SPJ}\text{-query}.$$

Then, in both cases, based on the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \twoheadrightarrow Y}$.

First, assume $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \rightarrow Y}$. Then, from the cases (d), (e) and (f) of Lemma 2, there could be any of the following case with $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$:
- $\Sigma_{Q_C} \Vdash \mathtt{FD}_{X \rightarrow U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{UF}_{V \rightarrow Y}$,
- $\Sigma_{Q_C} \Vdash \mathtt{UF}_{X \rightarrow U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{FD}_{V \rightarrow Y}$.
- $\Sigma_{Q_C} \Vdash \mathtt{FD}_{X \rightarrow U}$ and $\Sigma_{Q_{C'}} \Vdash \sigma_{V \rightarrow Y}$ s.t. $\sigma \in \{\mathtt{UFD}, \mathtt{FD}, \mathtt{UF}, \mathtt{FP}\}$,

For the simplicity, w.l.o.g., we concentrate on the case
- $\Sigma_{Q_C} \Vdash \mathtt{FP}_{X \rightarrow U}$ and $\Sigma_{Q_{C'}} \Vdash \mathtt{FP}_{V \rightarrow Y}$,

and exclude the rest of cases since the same arguments apply on those cases as well.

Since $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \rightarrow Y}$ there is ${Q_P}^{\mathcal{D}} \models \mathtt{UF}_{X \rightarrow Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_P}^{\mathcal{D}}$:
- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$.

Next, w.l.o.g. $Q_P \coloneqq \pi_{X,Y} \sigma_{\neg \mathtt{isNull}(X) \wedge \neg \mathtt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'})$. Thus, for every legal instance $\mathcal{D}$ of $\mathcal{R}$,

$$\begin{aligned} {Q_P}^{\mathcal{D}} = \{t \mid\ & t_1 \in {Q_C}^{\mathcal{D}} \text{ and } t_2 \in {Q_{C'}}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \mathtt{null} \\ & \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \mathtt{null} \text{ and} \\ & t(y) = t_2(y) \neq \mathtt{null} \text{ for every } x \in X \text{ and } y \in Y\}. \end{aligned}$$

In addition, since $\mathtt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\mathtt{UNQ}(V, R_2) \in \Sigma|_{Q_C}$, for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, in the former case $Q_C \neq Q_P \neq Q_{C'}$, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \mathtt{rdf{:}type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \lhd Q_C$, there exist mapping of at most one node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(X)), \mathtt{rdf{:}type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ to $Y \lhd Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lhd Q_P$.

Similarly, in the latter case $Q_C \neq Q_P = Q_{C'}$, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \lhd Q_C$, there exist mapping of at most one unique node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(X)), \texttt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lhd Q_P$.

Hence, in the both case, $[(\leq_1 P.C')]^{\mathcal{G},n}{=}1$.

Second, assume the case $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to^* Y}$. Further from the case (i) of Lemma 2, there must be $\Sigma_{Q_C} \Vdash \texttt{FP}_{X \to U}$ and $\Sigma_{Q_{C'}} \Vdash \texttt{FP}_{V \to Y}$, and $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$. Note that the case (i) also covers the relevant case (j) of Lemma 2.

As in the previous case, given $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ and $\Sigma_C \Vdash \texttt{UF}_{X \to U}$ (i.e., $\texttt{UNQ}(U, R_1) \in \Sigma|_{Q_C}$), for every tuple $t_1 \in Q_C{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in Q_P{}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$.

Hence, by following same arguments as in the previous case in the both mapping setting, we arrive at $[(\leq_1 P.C')]^{\mathcal{G},n}{=}1$.

v. Let $\phi_C \coloneqq (\geq_1 P.C')$. Then, from the definition of classification function ι associated with the constraint rewriting Γ in Defn. 15, there is mapping case $Q_C = Q_P$ to consider, i.e., either $Q_C = Q_P \neq Q_{C'}$ or $Q_C = Q_P = Q_{C'}$. For the first case $Q_C = Q_P \neq Q_{C'}$, let

$$Q_C = Q_P, \text{ i.e., same } \texttt{SPJ}\text{-query, and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg\texttt{isNull}(Y)}(R_2),$$

such that $R_2 \in \mathcal{R}$ is relation name with $V, Y \lhd R_2$.

In both mapping cases, w.l.o.g., we have

$$Q_P \coloneqq \pi_{X,Y} \sigma_{\neg\texttt{isNull}(X) \wedge \neg\texttt{isNull}(Y)}(Q_C \bowtie_{U=V} Q_{C'}).$$

Then, for every legal instance $\mathcal{D}$ of $\mathcal{R}$,

$$\begin{aligned} Q_P{}^{\mathcal{D}} = \{t \mid t_1 \in Q_C{}^{\mathcal{D}} \text{ and } t_2 \in Q_{C'}{}^{\mathcal{D}} \text{ s.t. } t(u) = t_1(u) = t_2(v) \neq \texttt{null} \\ \text{for every } u \in U \text{ and } v \in V, t(x) = t_1(x) \neq \texttt{null} \text{ and} \\ t(y) = t_2(y) \neq \texttt{null} \text{ for every } x \in X \text{ and } y \in Y\}. \end{aligned}$$

In addition, since $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ and filter $\neg\texttt{isNull}(X)$ on $X \lessdot Q_{C'}$, for every tuple $t \in Q_P{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exist one or more tuples $t_2 \in Q_{C'}{}^{\mathcal{D}}$ s.t. $t(Y) = t_2(Y)$. Thus, in the former mapping case, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of zero or more node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in Q_{C'}{}^{\mathcal{D}}$ to $Y \lessdot Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lessdot Q_P$.

Similarly, in the latter mapping case $Q_C = Q_P = Q_{C'}$, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of one or more node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(Y)), \texttt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \lessdot Q_P$.

Hence, in the both mapping cases, $[(\geq_1 P.C')]^{\mathcal{G},n}$=1.

vi. Let $\phi_C \coloneqq (\geq_0 P.C')$. Then, from the constraint rewriting Γ Defn. 3, there is the mapping case $Q_C \neq Q_P$ to consider, i.e., either $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$. For the former case $Q_C \neq Q_P \neq Q_{C'}$, let

$$Q_C \coloneqq \pi_X \sigma_{\neg\texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} \coloneqq \pi_Y \sigma_{\neg\texttt{isNull}(Y)}(R_2),$$

such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $V, Y \triangleleft R_2$. Similarly, for the latter case $Q_C \neq Q_P = Q_{C'}$, assume

$$Q_C \coloneqq \pi_X \sigma_{\neg\texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same } \texttt{SPJ}\text{-query.}$$

In both cases, since $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ and filter $\neg\texttt{isNull}(X)$ on $X \triangleleft Q_C$, for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exist zero or more tuples $t \in {Q_P}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, in the former case, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of zero or more node $\mathbf{f}_{C'}(t_2(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t_2(Y)), \texttt{rdf:type}, C' \rangle,$$

where $\mathbf{f}_{C'}$ is injective template and $t_2(Y)$ is the restriction of tuple $t_2 \in {Q_{C'}}^{\mathcal{D}}$ to $Y \triangleleft Q_{C'}$, connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Similarly, in the latter case $Q_C \neq Q_P = Q_{C'}$, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of zero or more node $\mathbf{f}_{C'}(t(Y))$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_{C'}(t(Y)), \texttt{rdf:type}, C' \rangle$$

connected via the property path P s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{f}_{C'}(t(Y)) \rangle,$$

where $\mathbf{f}_{C'}$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$.

Hence, in the both mapping cases, $[(\geq_0 P. C')]^{\mathcal{G},n}$=1.

vii. Let $\phi_C \coloneqq (\leq_0 P. \neg C')$. Then, following the constraint rewriting Γ in Defn. 15, there exists mapping of an object property P s.t.,

$$Q_P \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y) \rangle,$$

where Q_P is an SPJ-source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$, and mapping of an RDF concept C' s.t.,

$$Q_{C'} \longrightarrow \langle \mathbf{f}_{C'}(Y), \texttt{rdf:type}, C' \rangle,$$

where $Q_{C'}$ is an SP/SPJ source query Q over $\mathcal{R}$ with $Y \triangleleft Q$.

Further, from the definition of simple mapping set $\mathcal{M}$, there exists only one mapping definition per object property P. Hence, if there exists any node in the graph $\mathcal{M}(\mathcal{D})$ that is reachable from a node n s.t. $\langle n, \texttt{rdf:type}, C \rangle$ via property path $P^{\pm}$ then that must be no other than the node n' s.t. $\langle n', \texttt{rdf:type}, C \rangle$, i.e., $[(\leq_0 P. \neg C')]^{\mathcal{G},n}$=1.

b. For inverse object property path P^-, from the constraint rewriting Γ in Defn. 15, observe that there exist mapping of an object property P s.t.,

$$Q_P \longrightarrow \langle \mathbf{f}_{C'}(X), P, \mathbf{f}_C(Y) \rangle,$$

where Q_P is an SPJ-source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$, and mapping of an RDF concept C' s.t.,

$$Q_{C'} \longrightarrow \langle \mathbf{f}_{C'}(Y), \texttt{rdf:type}, C' \rangle,$$

where $Q_{C'}$ is an SP/SPJ source query Q over $\mathcal{R}$ with $Y \triangleleft Q$.

Then, like in the mappings of different property shapes ϕ_C definition with path P in the previous case (a), there exist analogous arguments based on the rewriting functions δ_1 and δ_2 in Defn. 15 for the following inverse property path P^- constraints ϕ_C definitions:

i. $\phi_C := (\trianglerighteq_C P^-. C')$.
ii. $\phi_C := (\triangleright_C P^-. C')$
iii. $\phi_C := (=_1 P^-. C')$
iv. $\phi_C := (\leq_1 P^-. C')$
v. $\phi_C := (\geq_1 P^-. C')$
vi. $\phi_C := (\geq_0 P^-. C')$
vii. $\phi_C := (\leq_0 P^-. \neg C')$

2. Let ϕ_C be a property shape definition for a datatype property U. Then, from the constraint rewriting Γ in Defn. 15, there exist mapping of a datatype property U s.t.,

$$Q_U \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{t}(Y)(Y) \rangle,$$

where Q_U is an SP source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$.

i. Let $\phi_C := (\trianglerighteq_C U. \mathbf{t})$. Then, from the definition of classification κ function, there are two mapping cases to consider, i.e., $Q_C \neq Q_U$ and $Q_C = Q_U$.

First, let $Q_C \neq Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U := \pi_{X,Y} \sigma_{\neg \texttt{isNull}(X) \wedge \neg \texttt{isNull}(Y)} R \text{ and } Q_C := \pi_X \sigma_{\neg \texttt{isNull}(X)} R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \lhd R$. Next, from the rewriting function μ_1 in Defn. 15, there must be $\Sigma_{Q_U} \Vdash \mathtt{UFD}_{X \to Y}$. Thus, ${Q_U}^{\mathcal{D}} \models \mathtt{UFD}_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_U}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$,
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$, and
- $\mathtt{NN}(X, R_1), \mathtt{NN}(Y, R_2) \in \Sigma|_{Q_U}$.

Then, for every tuple $t_1 \in {Q_C}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists a tuple $t \in {Q_U}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$ and vice-versa. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \mathtt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in {Q_C}^{\mathcal{D}}$ to $X \lhd Q_C$, there exist mapping of exactly one unique literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \lhd Q_P$.

Therefore, i.e., $[(\trianglerighteq_C U.\mathbf{t})]^{\mathcal{G},n}$=1.

Second, let $Q_C = Q_U$, i.e., there is relation name $R \in \mathcal{R}$ s.t.,

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X) \wedge \neg\mathtt{isNull}(Y)}(R) \text{ and } Q_C = Q_U, \text{i.e., are same SP-query.}$$

As in the previous case, from the rewriting function μ_2 in Defn. 15, we have $\Sigma_{Q_U} \Vdash \mathtt{UF}_{X \to Y}$, i.e., for every tuples $t, t' \in {Q_U}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$, and
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$.

Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \mathtt{rdf:type}, C \rangle,$$

there exist mapping of exactly one unique literal value $t(Y)$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle,$$

where $\mathbf{f}_C$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \lhd Q_P$. Therefore, $[(\trianglerighteq_C U.\mathbf{t})]^{\mathcal{G},n}$=1.

ii. Let $\phi_C \coloneqq (=_1 U.\mathbf{t})$. Then, from the function κ definition, there are two mapping cases to consider, i.e., $Q_C \neq Q_U$ and $Q_C = Q_U$.

First, let $Q_C \neq Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X) \wedge \neg\mathtt{isNull}(Y)} R \text{ and } Q_C \coloneqq \pi_X \sigma_{\neg\mathtt{isNull}(X)} R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \lhd R$. Next, from the rewriting function μ_1 in Defn. 15, there must be $\Sigma_{Q_U} \Vdash \mathtt{FD}_{X \to Y}$. Thus, ${Q_U}^{\mathcal{D}} \models \mathtt{FD}_{X \to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in {Q_U}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$ and
- $\mathtt{NN}(X, R_1), \mathtt{NN}(Y, R_2) \in \Sigma|_{Q_U}$.

Then, for every tuple $t_1 \in Q_C{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists a tuple $t \in Q_U{}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \mathtt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of exactly one literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, i.e., $[(=_1\ U.\mathbf{t})]^{\mathcal{G},n}$=1.

Second, let $Q_C = Q_U$, i.e., there is relation name $R \in \mathcal{R}$ s.t.,

$Q_U := \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}(R)$ and $Q_C = Q_U$, i.e., are same SP-query.

As in the previous case, from the rewriting function μ_2 in Defn. 15, we have $\Sigma_{Q_U} \Vdash \mathtt{FP}_{X\to Y}$, i.e., for every tuples $t, t' \in Q_U{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$.

Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \mathtt{rdf:type}, C \rangle,$$

there exist mapping of exactly one literal value $t(Y)$ in $\mathcal{M}(\mathcal{D})$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle,$$

where $\mathbf{f}_C$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, $[(=_1\ U.\mathbf{t})]^{\mathcal{G},n}$=1.

iii. Let $\phi_C := (\rhd_C\ U.\mathbf{t})$. Then, based on the rewriting μ_1 rules, there is only one mapping cases to consider, i.e., $Q_C \neq Q_U$.

Let $Q_C \neq Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U := \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}R \text{ and } Q_C := \pi_X\sigma_{\neg\mathtt{isNull}(X)}R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \triangleleft R$. From the rewriting function μ_1 in Defn. 15, there must be $\Sigma_{Q_U} \Vdash \mathtt{UF}_{X\to Y}$. Thus, $Q_U{}^{\mathcal{D}} \models \mathtt{UF}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in Q_U{}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$ and
- if $t(Y) = t'(Y)$ then $t(X) = t'(X)$.

Then, for every tuple $t_1 \in Q_C{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in Q_U{}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of at most one unique literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, i.e., $[(\triangleright_C\ U.\,\mathbf{t})]^{\mathcal{G},n}$=1.

iv. Let $\phi_C \coloneqq (\leq_1\ U.\,\mathbf{t})$. Then, similar to the case (iii) above, there is only one mapping cases to consider based on the rewriting μ_1 rules, i.e., $Q_C \neq Q_U$.

Let $Q_C \neq Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}R \text{ and } Q_C \coloneqq \pi_X\sigma_{\neg\texttt{isNull}(X)}R,$$

where $R \in \mathcal{R}$ is a relation name with $X, Y \triangleleft R$. From the rewriting function μ_1 in Defn. 15, there must be $\Sigma_{Q_U} \Vdash \texttt{FP}_{X\to Y}$. Thus, $Q_U{}^{\mathcal{D}} \models \texttt{FP}_{X\to Y}$ for every legal instance $\mathcal{D}$ of $\mathcal{R}$, i.e., for every $t, t' \in Q_U{}^{\mathcal{D}}$:

- if $t(X) = t'(X)$ then $t(Y) = t'(Y)$.

Then, for every tuple $t_1 \in Q_C{}^{\mathcal{D}}$ over every legal instance $\mathcal{D}$ of $\mathcal{R}$, there exists at most one tuple $t \in Q_U{}^{\mathcal{D}}$ s.t. $t_1(X) = t(X)$. Thus, for mapping of each node $\mathbf{f}_C(t_1(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t_1(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t_1(X)$ is the restriction of tuple $t_1 \in Q_C{}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of at most one literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in Q_P{}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, i.e., $[(\leq_1\ U.\,\mathbf{t})]^{\mathcal{G},n}$=1.

v. Let $\phi_C \coloneqq (\geq_1\ U.\,\mathbf{t})$. Then, from the constraint rewriting function Γ, there is mapping case $Q_C = Q_U$ to consider.

Let $Q_C = Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}R \text{ and } Q_C = Q_P,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \triangleleft R$. Thus, for mapping of each node $\mathbf{f}_C(t(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t(X)), \texttt{rdf:type}, C \rangle,$$

there exist mapping of at least one literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $\mathbf{f}_C$ is injective template, and $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, i.e., $[(\geq_1 U. \mathbf{t})]^{\mathcal{G},n}=1$.

vi. Let $\phi_C \coloneqq (\geq_0 U. \mathbf{t})$. Then, from the constraint rewriting function Γ , there is mapping case $Q_C \neq Q_U$ to consider.

Let $Q_C \neq Q_U$. Then, based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y} \sigma_{\neg \texttt{isNull}(X) \land \neg \texttt{isNull}(Y)} R \text{ and } Q_C \coloneqq \pi_X \sigma_{\neg \texttt{isNull}(X)} R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \triangleleft R$. Thus, for mapping of each node $\mathbf{f}_C(t'(X))$ in $\mathcal{M}(\mathcal{D})$ s.t.

$$\langle \mathbf{f}_C(t'(X)), \texttt{rdf:type}, C \rangle,$$

where $\mathbf{f}_C$ is injective template and $t'(X)$ is the restriction of tuple $t' \in {Q_C}^{\mathcal{D}}$ to $X \triangleleft Q_C$, there exist mapping of zero or more literal value $t(Y)$ s.t.,

$$\langle \mathbf{f}_C(t(X)), P, \mathbf{t}(t(Y)) \rangle \in \mathcal{M}(\mathcal{D}),$$

where $t(X)$ and $t(Y)$ are the restriction of tuple $t \in {Q_P}^{\mathcal{D}}$ to $X, Y \triangleleft Q_P$. Therefore, i.e., $[(\geq_0 U. \mathbf{t})]^{\mathcal{G},n}=1$.

vii. Let $\phi_C \coloneqq (\leq_0 U. \neg \mathbf{t})$. Then, following the constraint rewriting Γ in Defn. 15, there exists mapping of a datatype property U s.t.,

$$Q_U \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{t}(Y) \rangle,$$

where Q_U is an SP-source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$, and mapping of an RDF concept C s.t.,

$$Q_C \longrightarrow \langle \mathbf{f}_C(X), \texttt{rdf:type}, C \rangle,$$

where Q_C is also an SP source query Q over $\mathcal{R}$ with $X \triangleleft Q$.

Further, from the definition of simple mapping set $\mathcal{M}$, there exists only one mapping definition per datatype property $Usch(\mathcal{M})$. Hence, if there exists a node n in the graph $\mathcal{M}(\mathcal{D})$ s.t. $\langle n, \texttt{rdf:type}, C \rangle$ and $\langle n, U, v \rangle$, then the RDF literal v has no other XML datatype than the $\mathbf{t}$, i.e., $[(\leq_0 U. \neg t)]^{\mathcal{G},n}=1$.

This concludes the proof of the theorem.

Theorem 2. *The constraint rewriting Γ is maximal semantics preserving.*

Proof. The proof involves showing that the constraint rewriting Γ introduced in Definition 15 satisfies the condition stated in Definition 10, i.e., for every mapping set $\mathcal{M}$ and every source constraint Σ,

$$\Sigma \models^{*}_{\mathcal{M}} \Gamma(\mathcal{M}, \Sigma).$$

Let $\mathcal{M}$ be a mapping set defined over a relational schema $\mathcal{R}$ with source constraint Σ. Then, as per the Definition 7, $S = \Gamma(\mathcal{M}, \Sigma)$ is maximally Σ-implied set of SHACL shapes with respect to $\mathcal{M}$, $\Sigma \models^{*}_{\mathcal{M}} S$, that is:

I. First, $\Sigma \models_{\mathcal{M}} S$, means S is Σ-implied with respect to $\mathcal{M}$ as per the Definition 6. More precisely, for every instance $\mathcal{D}$ of $\mathcal{R}$,

$$\mathcal{D} \models \Sigma \longrightarrow \mathcal{M}(\mathcal{D}) \models S.$$

II. Then, for every $\Sigma \models_{\mathcal{M}} S'$ s.t. $sch(S') \subseteq sch(\mathcal{M})$ and every RDF graph $\mathcal{G}$,

$$\mathcal{G} \models S \longrightarrow \mathcal{G} \models S'.$$

Theorem 1 establish that the constraint rewriting Γ is semantic preserving, i.e., we have

$$\Sigma \models_{\mathcal{M}} S, \quad \text{where } S = \Gamma(\mathcal{M}, \Sigma) \text{ from Definition 8.}$$

In order to prove the next statement (II), we first identify every Σ-implied sets S' of SHACL shapes w.r.t. $\mathcal{M}$, i.e., $\forall S'.\Sigma \models_{\mathcal{M}} S'$ s.t. $sch(S') \subseteq sch(\mathcal{M})$, and subsequently, establish that $\mathcal{G} \models S \rightarrow \mathcal{G} \models S'$ for every RDF graph $\mathcal{G}$.

– Let $s \in \mathcal{S}$ be a shape s.t. $\langle C, \tau_C, \phi_C \rangle$ [12]. Then, there exists a mapping of an RDF concept $C \in sch(\mathcal{M})$ s.t.,

$$Q_C \longrightarrow \langle \mathbf{f}_C(X), \texttt{rdf:type}, C \rangle,$$

where Q_C is an SP/SPJ source query Q over $\mathcal{R}$ with $X \triangleleft Q$, see the rewriting Γ in Defn. 15.

Starting by the base cases:

i. We first consider the object (SHACL) property path P. Let $P \in sch(\langle C, \tau_C, \phi_C \rangle)$ be an object property path. Then, from the rewriting Γ in Defn. 15, there exists mapping of an object property $P \in sch(\mathcal{M})$ s.t.,

$$Q_P \longrightarrow \langle \mathbf{f}_C(X), P, \mathbf{f}_{C'}(Y) \rangle,$$

where Q_P is an SPJ source query Q over $\mathcal{R}$ with $X, Y \triangleleft Q$, and mapping of an RDF concept $C' \in sch(\mathcal{M})$ s.t.,

$$Q_{C'} \longrightarrow \langle \mathbf{f}_{C'}(Y), \texttt{rdf:type}, C' \rangle,$$

[12] i.e., τ_C is implicit targetClass declaration, i.e., C a sh:NodeShape, rdfs:Class .

where $Q_{C'}$ is an SP/SPJ source query Q over $\mathcal{R}$ with $Y \triangleleft Q$. Further, from the Defn. of SPJ source query in Lemma 2, the query Q_P must be

$$Q_P := \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_C \ \texttt{OP}_{U=V}\ Q_{C'})$$

such that Q_C and $Q_{C'}$ are SP expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ with $X, U \triangleleft Q_C$ and $Y, V \triangleleft Q_{C'}$ respectively, and $\texttt{OP} \in \{\bowtie, ⟖, ⟕, ⟗\}$. In addition, there must be either $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ since every Q_P is valid SPJ query. From the proof arguments of Lemma 2, w.l.o.g.,

$$Q_P := \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_C \ \bowtie_{U=V}\ Q_{C'}).$$

a. Let $\phi_C := (\leq_0 P.\neg C')$. Note that the rewriting Γ in Defn. 15 generates constraint $(\leq_0 P.\neg C') \in \phi_C$ since there exists only one mapping definition per property path predicate P in $sch(\mathcal{M})$, see the proof case (1).(a).(vii) of Theorem 1 for details. Following, the Q_P above, and the assumption for SP source query, let

$$Q_C := \pi_X\sigma_{\neg\texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} := \pi_Y\sigma_{\neg\texttt{isNull}(Y)}(R_2),$$

such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \triangleleft R_1$ and $V, Y \triangleleft R_2$.

Next, for the purpose of constructing $s' := \langle C, \tau_C, \phi_C\rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept [13]. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:

(1). $\phi_C := (\geq_0 P^{\pm}.B)$: since the minimum zero cardinality is always satisfied.
(2). $\phi_C := (\geq_0 P^{\pm}.\neg B)$: by following similar argument as in the case (1).
(3). $\phi_C := (\geq_0 P^{\pm}.\top)$: it follows from the previous cases (1) and (2).
(4). $\phi_C := (\leq_0 P.\neg C')$: it follows from arguments of the case (1).(a).(vii) of Theorem 1.
(5). $\phi_C := (\leq_n P.D)$ s.t. $D \in \mathbb{C} \setminus C'$ and $n \geq 0$: since there will not an instance of D that is reachable from any instance of C via property path P, which is enough to imply the constraint.
(6). $\phi_C := (=_0 P^{\pm}.D)$ s.t. $D \in \mathbb{C} \setminus C'$: it follows from the previous case (5).
(7). $\phi_C := (\leq_n P^{-}.B)$ for $n \geq 0$: since there will not an instance of B that is reachable from any instance of C via property path P^{-}, which is enough to imply the constraint.

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:

(1). $\phi_C := (\geq_n P^{\pm}.B)$ for $n \geq 1$:
Let $\mathcal{D} = \{R_1(a, \texttt{null}), R_2(c, d)\}$ [14]. Then $Q_C{}^{\mathcal{D}} = \{(a)\}$ and $Q_P{}^{\mathcal{D}} = \{\emptyset\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C :=\geq_n P^{\pm}.B)\rangle$ for $n \geq 1$.

[13] i.e., $B \in \mathbb{C}$ s.t. $B \neq C'$ or $B = C'$.

[14] we ignore Σ in the counter-example since we are reasoning for the constraints implied by the restriction on the mapping $\mathcal{M}$ (i.e., not by the source constraint Σ).

(2). $\phi_C := (\geq_n P^{\pm}. \neg B)$ for $n \geq 1$: similar to the previous case.
(3). $\phi_C := (\geq_n P^{\pm}. \top)$ for $n \geq 1$: it follows from the previous cases (1) and (2).
(4). $\phi_C := (\leq_n P. C)$ for $n \geq 0$: it follow from the argument as in the previous case(1), there exist counter-examples for each constraint when n is fixed.
(5). $\phi_C := (=_n P^{\pm}. B)$ for $n \geq 0$: it follows from the previous cases.
(6). $\phi_C := (=_n P^{\pm}. \neg B)$ for $n \geq 0$: it follows from the previous cases.
(7). $\phi_C := (=_n P^{\pm}. \top)$ for $n \geq 0$: it follows from the previous cases.
(8). $\phi_C := (\rhd_C P^{\pm}. B)$:
Let $\mathcal{D} = \{R_1(a, c), R_1(b, c), R_2(c, d)\}$. Then $Q_C{}^{\mathcal{D}} = \{(a), (b)\}$ and $Q_P{}^{\mathcal{D}} = \{(a, d), (b, d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C\rangle, \langle b, \texttt{rdf:type}, C\rangle, \langle a, P, d\rangle, \langle b, P, d\rangle, \langle d, \texttt{rdf:type}, C'\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C := (\rhd_C P^{\pm}. B)\rangle$ (even if $B = C'$).
(9). $\phi_C := (\unrhd_C P^{\pm}. B)$: it follows from the case (8).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C := (\geq_0 P. C')\rangle$. Then, there must be $\mathcal{G} \models s' := \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s'$ and $sch(\phi_C) \subseteq sch(\mathcal{M})$, as follows:

- Base cases:
  (1) $\phi_C := (\geq_0 P^{\pm}. B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C := (\geq_0 P^{\pm}. \neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C := (\geq_0 P^{\pm}. \top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C := (\leq_0 P. \neg C')$: trivial.
  (5) $\phi_C := (\leq_n P^{\pm}. D)$ s.t. $D \in \mathbb{C} \setminus C'$ and $n \geq 0$: $\mathcal{G} \models s'$ since the case (4).
  (6) $\phi_C := (=_0 P^{\pm}. D)$ s.t. $D \in \mathbb{C} \setminus C'$: $\mathcal{G} \models s'$ since the case (5).
  (7) $\phi_C := (\leq_n P^{-}. B)$ for $n \geq 0$: $\mathcal{G} \models s'$ since there will not be any instance of C with property path P^{-}.
- Inductive cases:

  Let $\phi_C{}^1$ and $\phi_C{}^2$ be two arbitrary base constraints (enumerated above) s.t. $\Sigma \models_{\mathcal{M}} s' := \langle C, \tau_C, \phi_C{}^1\rangle$ and $\Sigma \models_{\mathcal{M}} s' := \langle C, \tau_C, \phi_C{}^2\rangle$ with $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$ and $i = 1, 2$. Then, for any graph $\mathcal{G}$, we have

  $$\text{if } \mathcal{G} \models \langle C, \tau_C, \phi_C{}^1\rangle \text{ and } \mathcal{G} \models \langle C, \tau_C, \phi_C{}^2\rangle \text{ then } \mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2\rangle.$$

  Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n\rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i\rangle$ and $i = 1, \ldots, n$.

b. Let $\phi_C := (\geq_0 P. C')$. Then, from the constraint rewriting Γ Defn. 15, there is the mapping case $Q_C \neq Q_P$ to consider, i.e., $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$. See the proof case (1).(a).(vi) of Theorem 1 for details. Assume $Q_C \neq Q_P \neq Q_{C'}$. Then,

$$Q_C := \pi_X \sigma_{\neg \texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} := \pi_Y \sigma_{\neg \texttt{isNull}(Y)}(R_2),$$

such that $R_1, R_2 \in \mathcal{R}$ are relation names with $X, U \lhd R_1$ and $V, Y \lhd R_2$. Similarly, assume $Q_C \neq Q_P = Q_{C'}$. Then,

$$Q_C := \pi_X \sigma_{\neg \texttt{isNull}(X)}(R_1) \text{ and } Q_{C'} = Q_P, \text{i.e., same } \texttt{SPJ}\text{-query}.$$

Next, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' := \langle C, \tau_C, \phi_C \rangle$, let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' := \langle C, \tau_C, \phi_C \rangle$ [15] with $sch(\phi_C) \subseteq sch(\mathcal{M})$:

(1). $\phi_C := (\geq_0 P^{\pm}.B)$: since the minimum zero cardinality is always satisfied.
(2). $\phi_C := (\geq_0 P^{\pm}.\neg B)$: by following similar argument as in the case (1).
(3). $\phi_C := (\geq_0 P^{\pm}.\top)$: it follows from the previous cases (1) and (2).

Contrarily, following are the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$: [16]

(1). $\phi_C := (\geq_n P.C')$ s.t. $n \geq 1$:
let $\mathcal{D} = \{R_1(a, \texttt{null}), R_2(c, d)\}$. Then, $Q_C{}^{\mathcal{D}} = \{(a)\}$ and $Q_P{}^{\mathcal{D}} = \{\emptyset\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C ::= (\geq_n P.C') \rangle$ for $n \geq 1$.
(2). $\phi_C := (\leq_n P.C')$ s.t. $n \geq 0$: similar to the case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C := (=_n P.C')$ s.t. $n \geq 0$: it follows from the argument of case (1) and (2).
(4). $\phi_C := (\triangleright_C P^{\pm}.C')$: similar counter example as in the case (a) above exists.
(5). $\phi_C := (\trianglerighteq_C P^{\pm}.C')$: it follows from the case (3) and (4).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s := \langle C, \tau_C, \phi_C := (\geq_0 P.C') \rangle$. Then, there must be $\mathcal{G} \models s' := \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C := (\geq_0 P^{\pm}.B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C := (\geq_0 P^{\pm}.\neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C := (\geq_0 P^{\pm}.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
- Inductive case: Since, for every graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n \rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i \rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

c. Let $\phi_C := (\geq_1 P.C')$. Then, from the constraint rewriting Γ Defn. 15, there is the mapping case $Q_C = Q_P$ to consider, i.e., either $Q_C = Q_P \neq Q_{C'}$ or $Q_C = Q_P = Q_{C'}$. See the proof case (1).(a).(v) of Theorem 1 for details.

[15] We only have to cover the constraint ϕ_C definition involving the property path predicate P for this particular case. The constraint ϕ_C definition involving different property path predicates will eventually be covered when we reason for the rest of the (SHACL) property path predicates $\mathbb{P}$ in the $sch(\mathcal{M})$.

[16] Note that we can't claim the rest of the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$ based on the available facts, i.e., $\mathcal{M}$ and Σ. However, those unlisted constraints will eventually be enumerated when we reason for the maximality of rest of the Defn. ϕ_C s.t. $\Sigma \models s := \langle C, \tau_C, \phi_C \rangle$.

Next, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$:
(1). $\phi_C \coloneqq (\geq_0 P^{\pm}. B)$: since the minimum zero cardinality is always satisfied.
(2). $\phi_C \coloneqq (\geq_0 P^{\pm}. \neg B)$: similar to the case (1).
(3). $\phi_C \coloneqq (\geq_0 P^{\pm}. \top)$: it follows from the arguments of cases (1) and (2).
(4). $\phi_C \coloneqq (\geq_1 P. C')$: it follows from arguments of case (1).(a).(v) of Theorem 1.

Contrarily, following are the rest of the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_n P. C')$ s.t. $n \geq 2$:
let $\mathcal{D} = \{R_1(a, \texttt{null}), R_1(a, c), R_2(c, d)\}$ s.t. $\mathsf{att}(R_1) = \{X, U\}$ and $\mathsf{att}(R_2) = \{V, Y\}$. Then, $Q_P{}^{\mathcal{D}} = \{(a, d)\}$ and $Q_{C'}{}^{\mathcal{D}} = \{(d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, P, d\rangle, \langle a, \texttt{rdf:type}, C\rangle, \langle d, \texttt{rdf:type}, C'\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models (\geq_n P. C')$ for $n \geq 2$.
(2). $\phi_C \coloneqq (\leq_n P. C')$ s.t. $n \geq 0$: similar to the case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C \coloneqq (=_n P. C')$ s.t. $n \geq 0$: it follows from the argument of case (1) and (2).
(4). $\phi_C \coloneqq (\rhd_C P^{\pm}. C')$: similar counter example as in the case (a) above exists.
(5). $\phi_C \coloneqq (\unrhd_C P^{\pm}. C')$: it follows from the case (3) and (4).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0 P. C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0 P^{\pm}. B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 P^{\pm}. \neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0 P^{\pm}. \top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\geq_1 P. C')$: $\mathcal{G} \models s'$ is trivial.
- Inductive case:
  Since, for every graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n \rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i \rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

d. Let $\phi_C \coloneqq (\leq_1 P. C')$. Then, from the constraint rewriting Γ in Defn. 15 and the classification function ι in Defn. 14, there is mapping case $Q_C \neq Q_P$, i.e., $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$, to consider. See the proof case (1).(a).(iv) of Theorem 1 for details. Further, from the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathrm{FP}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \rightarrow Y}$ or $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \rightarrow^* Y}$ depending on the foreign key reference between join-attributes.

Next, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$:
(1). $\phi_C \coloneqq (\geq_0 P^{\pm}. B)$: trivial.
(2). $\phi_C \coloneqq (\geq_0 P^{\pm}. \neg B)$: trivial.

(3). $\phi_C \coloneqq (\geq_0 P^{\pm}.\top)$: trivial.
(4). $\phi_C \coloneqq (\leq_1 P.C')$: it follows from arguments of case (1).(a).(iv) of Theorem 1.
(5). $\phi_C \coloneqq (\leq_n P.C')$ s.t. $n \geq 2$: it follows from the same argument as in the case (4).

Contrarily, following are the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:
(1). $\phi_C \coloneqq (\geq_n P.C')$ s.t. $n \geq 1$:
let $\mathcal{D} = \{R_1(a, \texttt{null}), R_2(c, d)\}$ s.t. $\mathsf{att}(R_1) = \{X, U\}$, $\mathsf{att}(R_2) = \{V, Y\}$ and $\Sigma_{Q_P} \Vdash \texttt{FP}_{X \to Y}$, and $Q_C \neq Q_P \neq Q_{C'}$. Then, $Q_C{}^{\mathcal{D}} = \{(a)\}$, $Q_P{}^{\mathcal{D}} = \{\emptyset\}$ and $Q_{C'}{}^{\mathcal{D}} = \{(d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C\rangle, \langle d, \texttt{rdf:type}, C'\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models (\geq_n P.C')$ for $n \geq 1$.
(2). $\phi_C \coloneqq (=_n P.C')$ s.t. $n \geq 0$:
Similar to the previous case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C \coloneqq (\rhd_C P^{\pm}.C')$: similar counter example as in the case (a) above exists.
(4). $\phi_C \coloneqq (\unrhd_C P^{\pm}.C')$: it follows from the case (2) and (3).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0 P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0 P^{\pm}.B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 P^{\pm}.\neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0 P^{\pm}.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\leq_1 P.C')$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_n P.C')$ s.t. $n \geq 2$: since $\mathcal{G} \models s'$ in the previous case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n\rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i\rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

e. Let $\phi_C \coloneqq (\rhd_C P.C')$. Then, from the constraint rewriting Γ in Defn. 15 and the classification function ι in Defn. 14, there is mapping case $Q_C \neq Q_P$, i.e., $Q_C \neq Q_P \neq Q_{C'}$ or $Q_C \neq Q_P = Q_{C'}$, to consider. See the proof case (1).(a).(iii) of Theorem 1 for details. Further, from the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to Y}$ or $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to^* Y}$ depending on the foreign key reference between attributes participating in the join condition.

Now, let s' be a shape $\langle C, \tau_C, \phi_C\rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$:
(1). $\phi_C \coloneqq (\geq_0 P^{\pm}.B)$: trivial.
(2). $\phi_C \coloneqq (\geq_0 P^{\pm}.\neg B)$: trivial.
(3). $\phi_C \coloneqq (\geq_0 P^{\pm}.\top)$: trivial.

(4). $\phi_C \coloneqq (\triangleright_C\ P.C')$: it follows from arguments of case (1).(a).(iii) of Theorem 1.
(5). $\phi_C \coloneqq (\leq_1\ P.C')$: it follows from the same argument as in the case (4).

Contrarily, following are the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_n\ P.C')$ s.t. $n \geq 1$:
let $\mathcal{D} = \{R_1(a, \texttt{null}), R_2(c, d)\}$ s.t. $\mathsf{att}(R_1) = \{X, U\}$, $\mathsf{att}(R_2) = \{V, Y\}$ and $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \to Y}$, and $Q_C \neq Q_P \neq Q_{C'}$. Then, $Q_C{}^{\mathcal{D}} = \{(a)\}$, $Q_P{}^{\mathcal{D}} = \{\emptyset\}$ and $Q_{C'}{}^{\mathcal{D}} = \{(d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C\rangle, \langle d, \texttt{rdf:type}, C'\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models (\geq_n\ P.C')$ for $n \geq 1$.

(2). $\phi_C \coloneqq (=_n\ P.C')$ s.t. $n \geq 0$:
Similar to the previous case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively. For example, let $\mathcal{D} = \{R_1(a, \texttt{null}), R_1(b, c), R_2(c, d)\}$ s.t. $\Sigma_{Q_P} \Vdash \mathtt{UF}_{X \to Y}$, and $Q_C \neq Q_P = Q_{C'}$. Then, $Q_C{}^{\mathcal{D}} = \{(a), (b)\}$ and $Q_P{}^{\mathcal{D}} = \{(b, d)\} = Q_{C'}{}^{\mathcal{D}}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C\rangle, \langle d, \texttt{rdf:type}, C'\rangle, \langle b, P, d\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (=_1\ P.C')\rangle$.

(3). $\phi_C \coloneqq (\trianglerighteq_C\ P.C')$: it follows from the case (2).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0\ P^{\pm}.B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0\ P^{\pm}.\neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0\ P^{\pm}.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\triangleright_C\ P.C')$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_1\ P.C')$: since $\mathcal{G} \models s'$ in the previous case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n\rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i\rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

f. Let $\phi_C \coloneqq (=_1\ P.C')$. Then, from the constraint rewriting Γ in Defn. 15 and the classification function ι in Defn. 14, there is two mapping cases $Q_C \neq Q_P$ and $Q_C = Q_P$ to consider. See the proof case (1).(a).(ii) of Theorem 1 for details. Consider the case $Q_C \neq Q_P$. Then, based on the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathtt{FD}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \mathtt{FD}_{X \to Y}$ or $\Sigma_{Q_P} \Vdash \mathtt{FD}_{X \to^* Y}$ depending on the foreign key reference between attributes participating in the join condition. Similarly, assume the case $Q_C = Q_P$. Then, based on the rewriting function λ_2 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \to Y}$ or $\Sigma_{Q_P} \Vdash \mathtt{FP}_{X \to^* Y}$. For details, we refer to the case (a).(ii).(A) and (a).(ii).(B) of Theorem 1.

Next, let s' be a shape $\langle C, \tau_C, \phi_C\rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, in either case of mappings, i.e., $Q_C \neq Q_P$

or $Q_C = Q_P$, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$:

(1). $\phi_C \coloneqq (\geq_0 P^{\pm}. B)$: trivial.

(2). $\phi_C \coloneqq (\geq_0 P^{\pm}. \neg B)$: trivial.

(3). $\phi_C \coloneqq (\geq_0 P^{\pm}. \top)$: trivial.

(4). $\phi_C \coloneqq (=_1 P. C')$: it follows from arguments of case (1).(a).(ii) of Theorem 1.

(5). $\phi_C \coloneqq (\leq_1 P. C')$: it follows from the same argument as in the case (4).

(6). $\phi_C \coloneqq (\geq_1 P. C')$: it follows from the same argument as in the case (4).

Contrarily, following are the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_n P. C')$ s.t. $n \geq 2$:
Let $Q_C \neq Q_P = Q_{C'}$, and let $\mathcal{D} = \{R_1(a, c), R_2(c, d)\}$ s.t. $\Sigma_{Q_P} \Vdash \texttt{FD}_{X \to Y}$. Then, $Q_C{}^{\mathcal{D}} = \{(a)\}$ and $Q_P{}^{\mathcal{D}} = \{(a, d)\} = Q_{C'}{}^{\mathcal{D}}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, P, d \rangle, \langle a, \texttt{rdf:type}, C \rangle, \langle d, \texttt{rdf:type}, C' \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\geq_n P. C') \rangle$ for $n \geq 2$.

(2). $\phi_C \coloneqq (=_n P. C')$ s.t. $n \geq 2$ or $n = 0$: it follows from the previous case (1).

(3). $\phi_C \coloneqq (\triangleright_C P. C')$:
Let $Q_C \neq Q_P = Q_{C'}$, and let $\mathcal{D} = \{R_1(a, c), R_1(b, c), R_2(c, d)\}$ s.t. $\mathsf{att}(R_1) = \{X, U\}$, $\mathsf{att}(R_2) = \{V, Y\}$ and $\Sigma_{Q_P} \Vdash \texttt{FD}_{X \to Y}$. Then, $Q_C{}^{\mathcal{D}} = \{(a), (b)\}$ and $Q_P{}^{\mathcal{D}} = \{(a, d), (b, d)\} = Q_{C'}{}^{\mathcal{D}}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C \rangle, \langle b, \texttt{rdf:type}, C \rangle, \langle d, \texttt{rdf:type}, C' \rangle, \langle a, P, d \rangle, \langle b, P, d \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\triangleright_C P. C') \rangle$.

(4). $\phi_C \coloneqq (\trianglerighteq_C P. C')$: it follows from the case (2) and (3).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0 P. C') \rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0 P^{\pm}. B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 P^{\pm}. \neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0 P^{\pm}. \top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (=_1 P. C')$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_1 P. C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (6) $\phi_C \coloneqq (\leq_1 P. C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n \rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i \rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

g. Let $\phi_C \coloneqq (\trianglerighteq_C P. C')$. Then, from the constraint rewriting Γ in Defn. 15 and the classification function ι in Defn. 14, there is two mapping cases $Q_C \neq Q_P$ and $Q_C = Q_P$ to consider. See the proof case (1).(a).(i) of Theorem 1 for details. Consider the case $Q_C \neq Q_P$. Then, based on the rewriting function λ_1 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X \to Y}$ or $\Sigma_{Q_P} \Vdash \texttt{UFD}_{X \to^* Y}$ depending on the foreign key reference between attributes participating in the join condition. Similarly, assume the case $Q_C = Q_P$. Then, based on the rewriting function λ_2 in Defn. 15, there must be $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \twoheadrightarrow Y}$, i.e., $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to Y}$ or $\Sigma_{Q_P} \Vdash \texttt{UF}_{X \to^* Y}$. For details, we refer to the case (a).(i).(A) and (a).(i).(B) of

Theorem 1.

Next, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Let $\mathbb{C}$ and $\mathbb{P}$ be the set of all RDF concepts and object property predicates in $sch(\mathcal{M})$ respectively, and let $B \in \mathbb{C}$ be an RDF concept. Then, in either case of mappings, i.e., $Q_C \neq Q_P$ or $Q_C = Q_P$, following are the constraint Defn. ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s' := \langle C, \tau_C, \phi_C \rangle$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$:

(1). $\phi_C := (\geq_0 P^{\pm}.B)$: trivial.
(2). $\phi_C := (\geq_0 P^{\pm}.\neg B)$: trivial.
(3). $\phi_C := (\geq_0 P^{\pm}.\top)$: trivial.
(4). $\phi_C := (\trianglerighteq_C P.C')$: it follows from arguments of case (1).(a).(i) of Theorem 1.
(5). $\phi_C := (\triangleright_C P.C')$: it follows from the same argument as in the case (4).
(6). $\phi_C := (=_1 P.C')$: it follows from the same argument as in the case (4).
(7). $\phi_C := (\leq_1 P.C')$: it follows from the same argument as in the case (4).
(8). $\phi_C := (\geq_1 P.C')$: it follows from the same argument as in the case (4).

Contrarily, following are the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C := (\geq_n P.C')$ s.t. $n \geq 2$:
Let $Q_C \neq Q_P = Q_{C'}$, and let $\mathcal{D} = \{R_1(a, c), R_2(c, d)\}$ s.t. $\Sigma_{Q_P} \Vdash \mathtt{UFD}_{X \to Y}$. Then, $Q_C{}^{\mathcal{D}} = \{(a)\}$ and $Q_P{}^{\mathcal{D}} = \{(a, d)\} = Q_{C'}{}^{\mathcal{D}}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, P, d \rangle, \langle a, \mathtt{rdf{:}type}, C \rangle, \langle d, \mathtt{rdf{:}type}, C' \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C := (\geq_n P.C') \rangle$ for $n \geq 2$.
(2). $\phi_C := (=_n P.C')$ s.t. $n \geq 2$ or $n = 0$: it follows from the previous case (1).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C := (\geq_0 P.C') \rangle$. Then, there must be $\mathcal{G} \models s' := \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C := (\geq_0 P^{\pm}.B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C := (\geq_0 P^{\pm}.\neg B)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C := (\geq_0 P^{\pm}.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C := (\trianglerighteq_C P.C')$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C := (\triangleright_C P.C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (6) $\phi_C := (=_1 P.C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (7) $\phi_C := (\leq_1 P.C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (8) $\phi_C := (\leq_1 P.C')$: $\mathcal{G} \models s'$ is trivial from the case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n \rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i \rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

ii. We now consider the inverse object (SHACL) property path P^-. Let $P^- \in sch(\langle C, \tau_C, \phi_C \rangle)$ be an object property path. Then, from the rewriting Γ in Defn. 15, there exists mapping of an object property $P \in sch(\mathcal{M})$ s.t.,

$$Q_P \longrightarrow \langle \mathbf{f}_{C'}(Y), P, \mathbf{f}_C(X) \rangle,$$

where Q_P is an SPJ source query Q over $\mathcal{R}$ with $X, Y \lhd Q$, and mapping of an RDF concept $C' \in sch(\mathcal{M})$ s.t.,

$$Q_{C'} \longrightarrow \langle \mathbf{f}_{C'}(Y), \texttt{rdf:type}, C' \rangle,$$

where $Q_{C'}$ is an SP/SPJ source query Q over $\mathcal{R}$ with $Y \lhd Q$. Further, from the Defn. of SPJ source query in Lemma 2, the query Q_P must be

$$Q_P \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_C \; \texttt{OP}_{U=V} \; Q_{C'})$$

such that Q_C and $Q_{C'}$ are SP expressions over $R_1 \in \mathcal{R}$ and $R_2 \in \mathcal{R}$ with $X, U \lhd Q_C$ and $Y, V \lhd Q_{C'}$ respectively, and $\texttt{OP} \in \{\bowtie, ⟗, ⟕, ⟖\}$. In addition, there must be either $\texttt{FK}(U, R_1, V, R_2) \in \Sigma|_Q$ or $\texttt{FK}(V, R_2, U, R_1) \in \Sigma|_Q$ since every Q_P is valid SPJ query. From the proof arguments of Lemma 2, w.l.o.g.,

$$Q_P \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}(Q_C \; \bowtie_{U=V} \; Q_{C'}).$$

Similar to the mapping of property path $P \in sch(\mathcal{M})$ in the previous cases (i), there exist similar proof-arguments for the following $\Sigma \models^*_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$ s.t. $P^- \in sch(\phi_C)$ based on the rewriting rules δ_1 and δ_2 in Defn. 15.

a. $\phi_C \coloneqq (\leq_0 P^- . \neg C')$
b. $\phi_C \coloneqq (\geq_0 P^- . C')$
c. $\phi_C \coloneqq (\leq_1 P^- . C')$
d. $\phi_C \coloneqq (\geq_1 P^- . C')$
e. $\phi_C \coloneqq (=_1 P^- . C')$
f. $\phi_C \coloneqq (\rhd_C P^- . C')$
g. $\phi_C \coloneqq (\unrhd_C P^- . C')$

iii. We next consider the datatype (SHACL) property. Let $U \in sch(\langle C, \tau_C, \phi_C \rangle)$ be a datatype property path. Then, from the rewriting Γ in Defn. 15, there exists mapping of a datatype property $U \in sch(\mathcal{M})$ s.t.,

$$Q_U \longrightarrow \langle \mathbf{f}_C(X), U, \mathbf{t}(Y) \rangle,$$

where Q_P is an SP source query Q over $\mathcal{R}$ with $X, Y \lhd Q$, see proof case (2) of Theorem 1 for details. Hence, there must be either $Q_C = Q_U$ or $Q_C \neq Q_U$, i.e.,

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}R \text{ and } Q_C = Q_U,$$

or

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\texttt{isNull}(X)\wedge\neg\texttt{isNull}(Y)}R \text{ and } Q_C \coloneqq \pi_X\sigma_{\neg\texttt{isNull}(X)}R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \lhd R$.

Next, similarly to the maximality proof arguments for the mapping of object property path $P^{\pm} \in sch(\mathcal{M})$ in the previous case (1), there exist similar proof-arguments for the datatype property U, i.e., $\Sigma \models^*_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$ s.t. $U \in sch(\phi_C)$, based on the rewriting rules μ_1 and μ_2 in Defn. 15.

a. Let $\phi_C \coloneqq (\leq_0 U.\neg\mathbf{t})$. Recall that the constraints rewriting Γ in Defn. 15 generates constraint $(\leq_0 U.\neg\mathbf{t}) \in \phi_C$ since there exists only one mapping definition per datatype property $U \in sch(\mathcal{M})$, see the proof case (2).(vii) of Theorem 1 for details.

For the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:

(1). $\phi_C \coloneqq (\geq_0 U.\mathbf{t}')$: since the minimum zero cardinality is always satisfied.

(2). $\phi_C \coloneqq (\geq_0 U.\neg\mathbf{t}')$: by following similar argument as in the case (1).

(3). $\phi_C \coloneqq (\geq_0 U.\top)$: it follows from the previous cases (1) and (2).

(4). $\phi_C \coloneqq (\leq_0 U.\neg\mathbf{t})$: it follows from arguments of the case (2).(vii) of Theorem 1.

(5). $\phi_C \coloneqq (\leq_n U.\mathbf{X})$ s.t. $\mathbf{X} \in \mathbb{T}\setminus\mathbf{t}$ and $n \geq 0$: since the value of datatype property path U will not be a literal of datatype $\mathbf{X}$, which is enough to imply the constraint.

(6). $\phi_C \coloneqq (=_0 U.\mathbf{X})$ s.t. $X \in \mathbb{T} \setminus \mathbf{t}$: it follows from the previous case (5).

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:

(1). $\phi_C \coloneqq (\geq_n U.\mathbf{t}')$ for $n \geq 1$:
Let $\mathcal{D} = \{R(a, \mathtt{null}), R(c, d)\}$ [17], where $\mathsf{att}(R) = \{X, Y\}$. Then ${Q_C}^{\mathcal{D}} = \{(a), (c)\}$ and ${Q_U}^{\mathcal{D}} = \{(c)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf\!:\!type}, C\rangle, \langle c, \mathtt{rdf\!:\!type}, C\rangle, \langle c, U, d\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq \geq_n U.\mathbf{t}'\rangle\rangle$ for $n \geq 1$.

(2). $\phi_C \coloneqq (\geq_n U.\neg\mathbf{t}')$ for $n \geq 1$: similar to the previous case.

(3). $\phi_C \coloneqq (\geq_n U.\top)$ for $n \geq 1$: it follows from the previous cases (1) and (2).

(4). $\phi_C \coloneqq (\leq_n U.\mathbf{t})$ for $n \geq 0$: it follow from the argument as in the previous case(1), there exist counter-examples for each constraint when n is fixed.

(5). $\phi_C \coloneqq (=_n U.\mathbf{t}')$ for $n \geq 0$: it follows from the previous cases.

(6). $\phi_C \coloneqq (=_n U.\neg\mathbf{t}')$ for $n \geq 0$: it follows from the previous cases.

(7). $\phi_C \coloneqq (=_n U.\top)$ for $n \geq 0$: it follows from the previous cases.

(8). $\phi_C \coloneqq (\rhd_C U.\mathbf{t}')$:
Let $\mathcal{D} = \{R(a, c), R(b, c)\}$. Then ${Q_C}^{\mathcal{D}} = \{(a), (b)\}$ and ${Q_U}^{\mathcal{D}} = \{(a, c), (b, c)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf\!:\!type}, C\rangle, \langle b, \mathtt{rdf\!:\!type}, C\rangle, \langle a, U, c\rangle, \langle b, U, c\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\rhd_C U.\mathbf{t}')\rangle$ (as if $\mathbf{t}' = \mathbf{t}$).

(9). $\phi_C \coloneqq (\unrhd_C U.\mathbf{t}')$: it follows from the case (8).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0 P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s'$ and $sch(\phi_C) \subseteq sch(\mathcal{M})$, as follows:

- Base cases:
  (1) $\phi_C \coloneqq (\geq_0 U.\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 U.\neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.

[17] we ignore source constraints in the counter-example since we are reasoning for the constraints implied by the restriction on the mapping $\mathcal{M}$.

(3) $\phi_C \coloneqq (\geq_0\ U.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
(4) $\phi_C \coloneqq (\leq_0\ U.\neg\mathbf{t})$: trivial.
(5) $\phi_C \coloneqq (\leq_n\ U.\mathbf{X})$ s.t. $\mathbf{X} \in \mathbb{T} \setminus \mathbf{t}$ and $n \geq 0$: $\mathcal{G} \models s'$ since the case (4).
(6) $\phi_C \coloneqq (=_0\ U.\mathbf{X})$ s.t. $\mathbf{X} \in \mathbb{T} \setminus \mathbf{t}$: $\mathcal{G} \models s'$ since the case (5).

– Inductive cases: Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n\rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, {\phi_C}^i\rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

b. Let $\phi_C \coloneqq (\geq_0\ U.\mathbf{t})$. Then, from the constraint rewriting function Γ , there is mapping case $Q_C \neq Q_U$ to consider, see proof case (2).(vi) of Theorem 1. Based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}R \text{ and } Q_C \coloneqq \pi_X\sigma_{\neg\mathtt{isNull}(X)}R,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \triangleleft R$.

Next, let s' be a shape $\langle C, \tau_C, \phi_C\rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:
(1). $\phi_C \coloneqq (\geq_0\ U.\mathbf{t}')$: since the minimum zero cardinality is always satisfied.
(2). $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t}')$: by following similar argument as in the case (1).
(3). $\phi_C \coloneqq (\geq_0\ U.\top)$: it follows from the previous cases (1) and (2).

Contrarily, following are the Definitions ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$: [18]

(1). $\phi_C \coloneqq (\geq_n\ U.\mathbf{t})$ s.t. $n \geq 1$:
let $\mathcal{D} = \{R(a, \mathtt{null}), R(c, d)\}$, where $\mathsf{att}(R) = \{X, Y\}$. Then, ${Q_C}^{\mathcal{D}} = \{(a), (c)\}$ and ${Q_U}^{\mathcal{D}} = \{(c, d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C\rangle, \langle c, \mathtt{rdf{:}type}, C\rangle, \langle c, U, d\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C ::= (\geq_n\ U.\mathbf{t})\rangle$ for $n \geq 1$.
(2). $\phi_C \coloneqq (\leq_n\ U.\mathbf{t})$ s.t. $n \geq 0$: similar to the case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C \coloneqq (=_n\ U.\mathbf{t})$ s.t. $n \geq 0$: it follows from the argument of case (1) and (2).
(4). $\phi_C \coloneqq (\triangleright_C\ U.\mathbf{t})$: similar counter example as in the case (a) above exists.
(5). $\phi_C \coloneqq (\trianglerighteq_C\ U.\mathbf{t})$: it follows from the case (3) and (4).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s \coloneqq \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(\phi_C) \subseteq sch(\mathcal{M})$.

– Base case:
(1) $\phi_C \coloneqq (\geq_0\ U.\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
(2) $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.

[18] Note that we can't claim the rest of the Defn. ϕ_C s.t. $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$ based on the available facts, i.e., $\mathcal{M}$ and Σ. However, those unlisted constraints will eventually be enumerated when we reason for the maximality of rest of the Defn. ϕ_C s.t. $\Sigma \models s \coloneqq \langle C, \tau_C, \phi_C\rangle$.

(3) $\phi_C \coloneqq (\geq_0\ U.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
– Inductive case: Since, for every graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n\rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, {\phi_C}^i\rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

c. Let $\phi_C \coloneqq (\geq_1\ U.\mathbf{t})$. Then, from the constraint rewriting function Γ, there is mapping case $Q_C = Q_U$ to consider, see proof case (2).(v) of Theorem 1. Based on the assumption for SP source query in the mapping Defn. 3, there must be

$$Q_U \coloneqq \pi_{X,Y}\sigma_{\neg\mathtt{isNull}(X)\wedge\neg\mathtt{isNull}(Y)}R \text{ and } Q_C = Q_P,$$

such that $R \in \mathcal{R}$ is a relation name with $X, Y \lhd R$.

Let s' be a shape $\langle C, \tau_C, \phi_C\rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:
(1). $\phi_C \coloneqq (\geq_0\ U.\mathbf{t}')$: since the minimum zero cardinality is always satisfied.
(2). $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t}')$: by following similar argument as in the case (1).
(3). $\phi_C \coloneqq (\geq_0\ U.\top)$: it follows from the previous cases (1) and (2).
(4). $\phi_C \coloneqq (\geq_1\ U.\mathbf{t})$: it follows from arguments of case (2).(v) of Theorem 1.

Contrarily, following are the rest of the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:

(1). $\phi_C \coloneqq (\geq_n\ U.\mathbf{t})$ s.t. $n \geq 2$:
let $\mathcal{D} = \{R(a, \mathtt{null}), R(a, c)\}$, where $\mathsf{att}(R) = \{X, Y\}$. Then, ${Q_U}^{\mathcal{D}} = \{(a, c)\}$ and ${Q_C}^{\mathcal{D}} = \{(a)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C\rangle, \langle a, P, c\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C ::= (\geq_n\ U.\mathbf{t})$ for $n \geq 2$.
(2). $\phi_C \coloneqq (\leq_n\ U.\mathbf{t})$ s.t. $n \geq 0$: similar to the case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C \coloneqq (=_n\ U.\mathbf{t})$ s.t. $n \geq 0$: it follows from the argument of case (1) and (2).
(4). $\phi_C \coloneqq (\rhd_C\ U.\mathbf{t})$: similar counter example as in the case (a) above exists.
(5). $\phi_C \coloneqq (\unrhd_C\ U.\mathbf{t})$: it follows from the case (3) and (4).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.
– Base case:
(1) $\phi_C \coloneqq (\geq_0\ U.\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
(2) $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
(3) $\phi_C \coloneqq (\geq_0\ U.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
(4) $\phi_C \coloneqq (\geq_1\ U.\mathbf{t})$: $\mathcal{G} \models s'$ is trivial.
– Inductive case:
Since, for every graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2\rangle$ then

$\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n \rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, {\phi_C}^i \rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

d. Let $\phi_C \coloneqq (\leq_1\ U.\mathbf{t})$. Then, similar to the case (c) above, there is only one mapping cases to consider based on the rewriting μ_1 rules in Defn. 15, that is, $Q_C \neq Q_U$. See the proof case (2).(iv) of Theorem 1 for details.

Let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_0 U.\mathbf{t}')$: Trivial.
(2). $\phi_C \coloneqq (\geq_0 U.\neg\mathbf{t}')$: Trivial.
(3). $\phi_C \coloneqq (\geq_0 U.\top)$: Trivial following the cases (1) and (2).
(4). $\phi_C \coloneqq (\leq_1 U.\mathbf{t})$: it follows from arguments of case (2).(iv) of Theorem 1.
(5). $\phi_C \coloneqq (\leq_n U.\mathbf{t})$ s.t. $n \geq 2$: it follows from the same argument as in the case (4).

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_n U.\mathbf{t})$ s.t. $n \geq 1$:
let $\mathcal{D} = \{R(a, \mathtt{null}), R(c, d)\}$ s.t. $\mathsf{att}(R) = \{X, Y\}$ and $\Sigma_{Q_U} \Vdash \mathtt{FP}_{X \to Y}$. Let $Q_C \neq Q_U$. Then, ${Q_C}^{\mathcal{D}} = \{(a), (c)\}$ and ${Q_U}^{\mathcal{D}} = \{(c, d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C \rangle, \langle c, \mathtt{rdf{:}type}, C \rangle, \langle c, U, d \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \Coloneqq (\geq_n U.\mathbf{t}') \rangle$ for $n \geq 1$.
(2). $\phi_C \coloneqq (=_n U.\mathbf{t})$ s.t. $n \geq 0$:
Similar to the previous case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively.
(3). $\phi_C \coloneqq (\rhd_C U.\mathbf{t})$: similar counter example as in the case (a) above exists.
(4). $\phi_C \coloneqq (\unrhd_C U.\mathbf{t})$: it follows from the case (2) and (3).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.C') \rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0 U.\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 U.\neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0 U.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\leq_1 U.\mathbf{t})$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_n U.\mathbf{t})$ s.t. $n \geq 2$: since $\mathcal{G} \models s'$ in the previous case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n \rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, {\phi_C}^i \rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

e. Let $\phi_C \coloneqq (\rhd_C\ U.\mathbf{t})$. Then, similar to the case (c) above, there is only one mapping cases to consider based on the rewriting μ_1 rules in Defn. 15, that is,

$Q_C \neq Q_U$. See the proof case (2).(iii) of Theorem 1 for details.

Now, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ s.t. $\Sigma \models_\mathcal{M} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_\mathcal{M} \langle C, \tau_C, \phi_C \rangle$:
(1). $\phi_C \coloneqq (\geq_0 U. \mathbf{t}')$: Trivial.
(2). $\phi_C \coloneqq (\geq_0 U. \neg\mathbf{t}')$: Trivial.
(3). $\phi_C \coloneqq (\geq_0 U. \top)$: Trivial.
(4). $\phi_C \coloneqq (\triangleright_C U. \mathbf{t})$: it follows from arguments of case (2).(iii) of Theorem 1.
(5). $\phi_C \coloneqq (\leq_1 U. \mathbf{t})$: it follows from the same argument as in the case (4).

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_\mathcal{M} \langle C, \tau_C, \phi_C \rangle$:
(1). $\phi_C \coloneqq (\geq_n U. \mathbf{t})$ s.t. $n \geq 1$: as for the case (f) above.

(2). $\phi_C \coloneqq (=_n U. \mathbf{t})$ s.t. $n \geq 0$:
Similar to the previous case (1), there exist counter-examples for each constraint when $n > 0$ or $n = 0$, respectively. For example, let $\mathcal{D} = \{R(a, \texttt{null}), R(b, c)\}$ s.t. $\mathsf{att}(R) = \{X, Y\}$ and $\Sigma_{Q_U} \Vdash \texttt{UF}_{X \to Y}$. Let $Q_C \neq Q_U$. Then, ${Q_C}^\mathcal{D} = \{(a), (b)\}$ and ${Q_U}^\mathcal{D} = \{(b, c)\} = {Q_{C'}}^\mathcal{D}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \texttt{rdf:type}, C \rangle, \langle b, \texttt{rdf:type}, C \rangle, \langle b, U, c \rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (=_1 U. \mathbf{t}) \rangle$.
(3). $\phi_C \coloneqq (\trianglerighteq_C U. \mathbf{t})$: it follows from the case (2).
Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0 P. C') \rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0 U. \mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0 U. \neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0 U. \top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\triangleright_C U. \mathbf{t})$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_1 U. \mathbf{t})$: since $\mathcal{G} \models s'$ in the previous case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2 \rangle$ then $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2 \rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n \rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_\mathcal{M} \langle C, \tau_C, {\phi_C}^i \rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

f. Let $\phi_C \coloneqq (=_1 U. \mathbf{t})$. Then, from the μ_1 and μ_2 rules in Defn. 15, there are two mapping cases to consider, that is, $Q_C \neq Q_U$ and $Q_C = Q_U$. See the proof case (2).(ii) of Theorem 1 for details.

Next, let s' be a shape $\langle C, \tau_C, \phi_C \rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ s.t. $\Sigma \models_\mathcal{M} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t}' \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_\mathcal{M} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_0\ U.\mathbf{t'})$: Trivial.
(2). $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t'})$: Trivial.
(3). $\phi_C \coloneqq (\geq_0\ U.\top)$: Trivial.
(4). $\phi_C \coloneqq (=_1\ U.\mathbf{t})$: it follows from arguments of case (2).(ii) of Theorem 1.
(5). $\phi_C \coloneqq (\leq_1\ U.\mathbf{t})$: it follows from the same argument as in the case (4).
(6). $\phi_C \coloneqq (\geq_1\ U.\mathbf{t})$: it follows from the same argument as in the case (4).

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:
(1). $\phi_C \coloneqq (\geq_n\ U.\mathbf{t})$ s.t. $n \geq 2$:
Let $Q_C \neq Q_U$, and let $\mathcal{D} = \{R(a,c), R(b,d)\}$ s.t. $\mathsf{att}(R) = \{X, Y\}$ and $\Sigma_{Q_U} \Vdash \mathtt{FD}_{X\rightarrow Y}$. Then, ${Q_C}^{\mathcal{D}} = \{(a),(b)\}$ and ${Q_U}^{\mathcal{D}} = \{(a,c),(b,d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C\rangle, \langle b, \mathtt{rdf{:}type}, C\rangle, \langle a, U, c\rangle, \langle b, U, d\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\geq_n\ U.\mathbf{t})\rangle$ for $n \geq 2$.
(2). $\phi_C \coloneqq (=_n\ U.\mathbf{t})$ s.t. $n \geq 2$ or $n = 0$: it follows from the previous case (1).
(3). $\phi_C \coloneqq (\rhd_C\ P.C')$:
Let $Q_C \neq Q_U$, and let $\mathcal{D} = \{R(a,c), R(b,c)\}$ s.t. $\mathsf{att}(R) = \{X, Y\}$ and $\Sigma_{Q_U} \Vdash \mathtt{FD}_{X\rightarrow Y}$. Then, ${Q_C}^{\mathcal{D}} = \{(a),(b)\}$ and ${Q_U}^{\mathcal{D}} = \{(a,c),(b,c)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C\rangle, \langle b, \mathtt{rdf{:}type}, C\rangle, \langle a, U, c\rangle, \langle b, U, c\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\rhd_C\ U.\mathbf{t})\rangle$.
(4). $\phi_C \coloneqq (\unrhd_C\ U.\mathbf{t})$: it follows from the case (2) and (3).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0\ U.\mathbf{t'})$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t'})$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0\ U.\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (=_1\ U.\mathbf{t})$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\leq_1\ U.\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (6) $\phi_C \coloneqq (\leq_1\ U.\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge {\phi_C}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, {\phi_C}^1 \wedge \cdots \wedge {\phi_C}^n\rangle$ for all ${\phi_C}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, {\phi_C}^i\rangle$ and $sch({\phi_C}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

g. Let $\phi_C \coloneqq (\unrhd_C\ U.\mathbf{t})$. Then, from the μ_1 and μ_2 rules in Defn. 15, there are two mapping cases to consider, that is, $Q_C \neq Q_U$ and $Q_C = Q_U$. See the proof case (2).(i) of Theorem 1 for details.

Let s' be a shape $\langle C, \tau_C, \phi_C\rangle$ s.t. $sch(s') \subseteq sch(\mathcal{M})$. Then, for the purpose of constructing $s' \coloneqq \langle C, \tau_C, \phi_C\rangle$ s.t. $\Sigma \models_{\mathcal{M}} s'$ with $sch(s') \subseteq sch(\mathcal{M})$, let $\mathbb{U}$ be the set of all RDF datatype predicates in $sch(\mathcal{M})$, $\mathbb{T}$ be set of all XML Schema datatypes, and let $\mathbf{t'} \in \mathbb{T}$ be an XML schema datatype. Then, following are the Definitions ϕ_C such that $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C\rangle$:
(1). $\phi_C \coloneqq (\geq_0\ U.\mathbf{t'})$: Trivial.
(2). $\phi_C \coloneqq (\geq_0\ U.\neg\mathbf{t'})$: Trivial.
(3). $\phi_C \coloneqq (\geq_0\ U.\top)$: Trivial.

(4). $\phi_C \coloneqq (\trianglerighteq_C\ U.\,\mathbf{t})$: it follows from arguments of case (2).(i) of Theorem 1.
(5). $\phi_C \coloneqq (\triangleright_C\ U.\,\mathbf{t})$: it follows from the same argument as in the case (4).
(6). $\phi_C \coloneqq (=_1\ U.\,\mathbf{t})$: it follows from the same argument as in the case (4).
(7). $\phi_C \coloneqq (\leq_1\ U.\,\mathbf{t})$: it follows from the same argument as in the case (4).
(8). $\phi_C \coloneqq (\geq_1\ U.\,\mathbf{t})$: it follows from the same argument as in the case (4).

Contrarily, following are the Definitions ϕ_C such that $\Sigma \not\models_{\mathcal{M}} \langle C, \tau_C, \phi_C \rangle$:

(1). $\phi_C \coloneqq (\geq_n\ U.\,\mathbf{t})$ s.t. $n \geq 2$:
Let $Q_C \neq Q_U$. Let $\mathcal{D} = \{R(a,c), R(b,d)\}$ s.t. $\mathsf{att}(R) = \{X, Y\}$ and $\Sigma_{Q_U} \Vdash \mathtt{UFD}_{X \to Y}$. Then, $Q_C{}^{\mathcal{D}} = \{(a), (b)\}$ and $Q_U{}^{\mathcal{D}} = \{(a,c), (b,d)\}$. Thus, $\mathcal{M}(\mathcal{D}) = \{\langle a, \mathtt{rdf{:}type}, C\rangle, \langle b, \mathtt{rdf{:}type}, C\rangle, \langle a, U, c\rangle, \langle b, U, d\rangle\}$ and $\mathcal{M}(\mathcal{D}) \not\models \langle C, \tau_C, \phi_C \coloneqq (\geq_n\ U.\,\mathbf{t})\rangle$ for $n \geq 2$.

(2). $\phi_C \coloneqq (=_n\ P.\,C')$ s.t. $n \geq 2$ or $n = 0$: it follows from the previous case (1).

Let $\mathcal{G}$ be an arbitrary graph with $\mathcal{G} \models s$, where $s = \langle C, \tau_C, \phi_C \coloneqq (\geq_0\ P.\,C')\rangle$. Then, there must be $\mathcal{G} \models s' \coloneqq \langle C, \tau_C, \phi_C \rangle$ for all the ϕ_C s.t. $sch(\phi_C) \subseteq sch(\mathcal{M})$.

- Base case:
  (1) $\phi_C \coloneqq (\geq_0\ U.\,\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (2) $\phi_C \coloneqq (\geq_0\ U.\,\neg\mathbf{t}')$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (3) $\phi_C \coloneqq (\geq_0\ U.\,\top)$: $\mathcal{G} \models s'$ since zero cardinality is always satisfied.
  (4) $\phi_C \coloneqq (\trianglerighteq_C\ U.\,\mathbf{t})$: $\mathcal{G} \models s'$ is trivial.
  (5) $\phi_C \coloneqq (\triangleright_C\ U.\,\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (6) $\phi_C \coloneqq (=_1\ U.\,\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (7) $\phi_C \coloneqq (\leq_1\ U.\,\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
  (8) $\phi_C \coloneqq (\leq_1\ U.\,\mathbf{t})$: $\mathcal{G} \models s'$ is trivial from the case (4).
- Inductive case:
  Since, for any graph $\mathcal{G}$, if $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1\rangle$ and $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^2\rangle$ then $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \phi_C{}^2\rangle$. Thus, $\mathcal{G} \models \langle C, \tau_C, \phi_C{}^1 \wedge \cdots \wedge \phi_C{}^n\rangle$ for all $\phi_C{}^i$ s.t. $\Sigma \models_{\mathcal{M}} \langle C, \tau_C, \phi_C{}^i\rangle$ and $sch(\phi_C{}^i) \subseteq sch(\mathcal{M})$, where $i = 1, \ldots, n$.

This concludes the proof of the theorem.

Finally, we claim that $\mathcal{M}'$ extending a given simple mapping $\mathcal{M}$ does not invalidate already translated $\Gamma(\mathcal{M}, \Sigma)$ constraints, i.e., $\Gamma(\mathcal{M}, \Sigma)$ is contained in $\Gamma(\mathcal{M}', \Sigma)$ for every simple mappings $\mathcal{M} \subseteq \mathcal{M}'$. However, interest in having a monotone constraint rewriting for arbitrary R2R mappings lies purely in the formulation of the 'rewriting rules' that generate disjunctive constraints when clause (a) in Defn. 4 of simple mapping is relaxed.

Theorem 3. *The constraint rewriting Γ is monotone.*

Proof. It is straightforward to see that Γ is monotone, because all the rewriting steps, i.e., including rules $\lambda_i \cup \delta_i \cup \mu_i$ for $1 \leq i \leq 2$ in the constraints $\Phi_{1 \leq i \leq 3}$ definitions as well as the classifications functions ι and κ, defining Γ in Defn. 15 refer to the source constraints Σ and the mappings $\mathcal{M} \subseteq \mathcal{M}'$, and these elements are kept fixed when checking monotonicity.

The proof involves showing that the constraint rewriting Γ satisfies the condition stated in Defn. 11, i.e., for every mapping sets $\mathcal{M} \subseteq \mathcal{M}'$ defined over a relational schema $\mathcal{R}$ with source constraint Σ and every RDF graph $\mathcal{G}$,

$$\mathcal{G} \models \Gamma(\mathcal{M}', \Sigma) \rightarrow \mathcal{G} \models \Gamma(\mathcal{M}, \Sigma),$$

where Γ is constraint rewriting from Definition 15, and $\Gamma(\mathcal{M}', \Sigma)$ and $\Gamma(\mathcal{M}, \Sigma)$ from Definition 8.

Let $\mathcal{M} \subseteq \mathcal{M}'$ be simple mappings defined over a relational schema $\mathcal{R}$ with source constraints Σ, i.e., for every mapping m , if $m \in \mathcal{M}$ then $m \in \mathcal{M}'$. Note that database schema $\mathcal{R}$ with source constraints Σ is fixed for the both mappings $\mathcal{M}'$ and $\mathcal{M}$ and all the rewriting steps defining Γ in Defn. 15 are negation-free. Hence, by structural induction on Γ, for every SHACL shape $\langle s, \tau_s, \phi_s \rangle$ on $sch(M)$, if $\langle s, \tau_s, \phi_s \rangle \in \Gamma(\mathcal{M}, \Sigma)$ then $\langle s', \tau_{s'}, \phi_{s'} \rangle \in \Gamma(\mathcal{M}', \Sigma)$ s.t. $\tau_s = \tau_{s'}$ and, for each shape constraint ϕ, if $\phi \in \phi_s$ then $\phi \in \phi_{s'}$. Thus, for every graph $\mathcal{G}$, if $\mathcal{G} \models \Gamma(\mathcal{M}', \Sigma)$ then $\mathcal{G} \models \Gamma(\mathcal{M}, \Sigma)$.

D Proof of Sect. 1

Lemma 3. *Let M be a mapping set defined over the relational schema $\mathcal{R}$ with source constraints Σ and S a set of SHACL shapes, known as SHACL document, from Example 1. Then, the document S is maximally Σ-implied set of shapes with respect to M.*

Proof. To ease the presentation, consider the relational schema $\mathcal{R}$ with source constraints Σ from Example 1 as follows,

cu {C_id varchar PK, Title varchar UNQ},
st {S_id integer PK, Name varchar, Code
varchar NN and FK to cu(C_id)}.

Henceforth, we use atom P(a, b) for the RDF triple $\langle a, P, b \rangle$ and $C(a)$ for the class membership triple $\langle a, \texttt{rdf:type}, C \rangle$. Then, the mapping set M from Example 1:

Select S_id from st $\longrightarrow$ St(S_id).
Select C_id from cu $\longrightarrow$ Cu(C_id).
Select S_id, C_id from st, cu $\longrightarrow$ enrolledFor(S_id, C_id).
where st.Code = cu.C_id

The SHACL document S from Example 1 in abstract syntax [13] :

$$S = \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle, \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle\}, \text{ with } ^{19}$$

$$\phi_{\texttt{St}} \coloneqq (=_1 \ \texttt{enrolledFor} \ .\texttt{Cu}) \wedge (\leq_0 \ \texttt{enrolledFor} \ .\neg\texttt{Cu})$$

$$\text{and } \phi_{\texttt{Cu}} \coloneqq (\leq_0 \ \texttt{enrolledFor}^{-}.\neg\texttt{St}).$$

To prove that S is maximally Σ-implied set of SHACL shapes with respect to M, we need to establish:

[19] Note that node shapes are defined with implicit targetClass

- □ $\Sigma \models_M S$ as per required by the Defn. 6, and
- □ $\Sigma \models^*_M S$ as per required by the Defn. 7.

First, to prove $\Sigma \models_M S$, we show:

$$\mathcal{D} \models \Sigma \rightarrow M(\mathcal{D}) \models S$$

for every instance $\mathcal{D}$ of $\mathcal{R}$, where $M(D)$ is graph from Definition 5.

Let $\mathcal{D}$ be an instance of $\mathcal{R}$ that is legal for the source constraints Σ, i.e., $\mathcal{D} \models \Sigma$. Then, there must be,

$$\bigwedge_{\langle s,\tau_s,\phi_s\rangle \in S} \; \bigwedge_{\text{node } n \text{ in } M(\mathcal{D})} . \, (M(\mathcal{D}) \models \tau_c(n) \rightarrow M(\mathcal{D}) \models \phi_C(n)),$$

where $S = \{\langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}} \coloneqq (=_1 \; \mathtt{enrolledFor} \; .\mathtt{Cu}) \wedge (\leq_0 \; \mathtt{enrolledFor} \; .\neg\mathtt{Cu})\rangle, \langle \mathtt{Cu}, \tau_{\mathtt{C}}, \phi_{\mathtt{Cu}} \coloneqq (\leq_0 \; \text{enrolledFor}^-.\neg\mathtt{St})\rangle\}$.

To show that $M(\mathcal{D})$ satisfies the shape constraint $\langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}} \coloneqq (=_1 \; \text{enrolledFor.Cu}) \wedge (\leq_0 \; \mathtt{enrolledFor} \; .\neg\mathtt{Cu})\rangle \in S$. For the component $(=_1$ enrolledFor.Cu$) \in \phi_{\mathtt{St}}$, let n be an arbitrary node with $\mathtt{St}(n) \in M(\mathcal{D})$. A $\mathtt{St}(n)$ triple must come from a tuple $t_1 \in \mathtt{st}^{\mathcal{D}}$ with $t_1(\mathtt{S\_id}) \neq \mathtt{null}$. The NN constraint on Code ensures that $t_1(\mathtt{Code}) \neq \mathtt{null}$, and the FK constraint in turn guarantees that there is a tuple $t_2 \in \mathtt{cu}^{\mathcal{D}}$ with $t_1(\mathtt{Code}) = t_2(\mathtt{C\_id})$. There is therefore a triple enrolledFor$(n, t_2(\mathtt{C_id})) \in M(\mathcal{D})$ with $\mathtt{Cu}(t_2(\mathtt{C\_id})) \in M(\mathcal{D})$. Thus, there cannot be more than one triple enrolledFor$(n, t_2(\mathtt{C_id})) \in M(\mathcal{D})$ for each $\mathtt{St}(n) \in M(\mathcal{D})$, since that would either require several tuples $t_1 \in \mathtt{st}^{\mathcal{D}}$ with the same $t_1(\mathtt{S\_ID})$ but different $t_1(\mathtt{Code})$, contradicting the primary key constraint $\mathtt{PK}(\mathtt{S\_id}, \text{st}) \in \Sigma$. Similarly, for the component $(\leq_0$ enrolledFor.$\neg$Cu$) \in \phi_{\mathtt{St}}$, observe that there exists only one mapping definition per property path enrolledFor in $sch(M)$. Thus, M assures that if there exist any node in $M(\mathcal{D})$ that is reachable from a node n s.t. $\mathtt{St}(n) \in M(\mathcal{D})$ via property path enrolledFor then that must no other than the node n' with $\mathtt{Cu}(n') \in M(\mathcal{D})$.

To show that $M(\mathcal{D})$ satisfies the shape constraint $\langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, \phi_{\mathtt{Cu}} \coloneqq (\leq_0 \; \text{enrolledFor}^-.\neg\mathtt{St})\rangle \in S$. Let n be an arbitrary node with $\mathtt{Cu}(n) \in M(\mathcal{D})$. A $\mathtt{Cu}(n)$ triple must come from a tuple $t_2 \in \mathtt{cu}^{\mathcal{D}}$ with $t_2(\mathtt{C\_id}) \neq \mathtt{null}$. Since there is no UNQ constraint on the $\mathtt{Code} \in \mathsf{att}(\mathtt{cu})$ and $\mathtt{FK}(\mathtt{Code}, \mathtt{st}, \mathtt{C\_id}, \mathtt{Cu})$ constraint, for each $t_2 \in \mathtt{cu}^{\mathcal{D}}$ with $t_2(\mathtt{C\_id}) \neq \mathtt{null}$ there exist zero or more tuples $t_1 \in \mathtt{st}^{\mathcal{D}}$ with $t_1(\mathtt{Code}) = t_2(\mathtt{C\_id})$. Therefore, there is zero or more triples enrolledFor$(t_1(\mathtt{S_id}), n) \in M(\mathcal{D})$ with $\mathtt{St}(t(\mathtt{S\_id})) \in M(\mathcal{D})$ for each $\mathtt{Cu}(n) \in M(\mathcal{D})$. In addition, there cannot be a triple enrolledFor$(t_1(\mathtt{S_id}), n) \in M(\mathcal{D})$ without $\mathtt{St}(t(\mathtt{S\_id})) \in M(\mathcal{D})$, since that contradict the mapping M and the $\mathtt{PK}(\mathtt{S\_id}, \text{st}) \in \Sigma$ constraint.

Thus, for each legal instance $\mathcal{D}$ of $\mathcal{R}$, we have $M(\mathcal{D}) \models S$, i.e.,

- □ $(M(\mathcal{D}) \models \tau_{\mathtt{St}}(n) \rightarrow M(\mathcal{D}) \models \phi_{\mathtt{St}}(n))$ for every St(n) $\in M(\mathcal{D})$ and
- □ $(M(\mathcal{D}) \models \tau_{\mathtt{Cu}}(n') \rightarrow M(\mathcal{D}) \models \phi_{\mathtt{Cu}}(n'))$ for every Cu(n') $\in M(\mathcal{D})$,

where $S = \{\langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle, \langle \mathtt{Cu}, \tau_{\mathtt{C}}, \phi_{\mathtt{Cu}}\rangle\}$. Hence, $\Sigma \models_M S$.

Next, we proceed to prove $\Sigma \models^*_M S$, i.e., according to Defn. 7, for every SHACL document $\Sigma \models_M S'$ such that $sch(S') \subseteq sch(M)$ and every RDF graph G:

$$G \models S \rightarrow G \models S'.$$

For the purpose, we first identify all the SHACL documents $\Sigma \models_M S'$ such that $sch(S') \subseteq sch(M)$, i.e., according to the Defn. 6, all the SHACL documents S' such that $sch(S') \subseteq sch(M)$ and for every database instance $\mathcal{D}$:

$$\mathcal{D} \models \Sigma \rightarrow M(\mathcal{D}) \models S'.$$

Let S' be a SHACL document such that $sch(S') \subseteq sch(M)$, and let s be an arbitrary shape in S'. Then, s must be defined on $sch(M)$, i.e., $sch(s) \subseteq sch(M) = \{\texttt{St}, \texttt{Cu}, \texttt{enrolledFor}^{\pm}\}$. Therefore,[20]

$$s := \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle \mid \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle,$$

whose constraints ϕ definition is given by the following grammar:[21]

$$\begin{aligned} \phi &::= \ \geq_n P^{\pm}.\alpha \ \mid \leq_n P^{\pm}.\alpha \ \mid \rhd_C P^{\pm} \ \mid \ \phi \wedge \phi \\ \alpha &::= \top \ \mid \ C \ \mid \ \neg C \end{aligned}$$

where P stands for the property path '$\texttt{enrolledFor}$', the superscript $\pm$ refers to the path '$\texttt{enrolledFor}$' or its inverse '$\texttt{enrolledFor}^{-}$', $C := \texttt{St} \mid \texttt{Cu}$, $n \in \mathbb{N}$ and rest of the notations are as introduced in Sect 2.

Thus,

A. For $\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle \in S'$ – with implicit targetClass $\tau_{\texttt{St}}$ that declares the instances of RDF concept St as target nodes, there exist following $\phi_{\texttt{St}}$ definitions on RDF vocabularies $\{\texttt{St}, \texttt{Cu}, \texttt{enrolledFor}^{\pm}\}$:
   i. Defn. $\phi_{\texttt{St}}$ on $\{\texttt{St}, \texttt{Cu}, \texttt{enrolledFor}\}$:
      a. Defn. $\phi_{\texttt{St}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}\}$ typing [22]:
         1. Defn. $\phi_{\texttt{St}} := (\leq_0 \texttt{enrolledFor}.\top)$:
            For the constraint, we have $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_0 \texttt{enrolledFor}.\top) \rangle$.
            Assume $\mathcal{D}$ = {st(001,_, CS40), cu(CS40, Logic)} such that $\mathcal{D} \models \Sigma$. Then,

            $$\mathcal{M}(\mathcal{D}) = \{\texttt{St}(001), \texttt{Cu}(\texttt{CS40}), \texttt{enrolledFor}(001, \texttt{CS40})\}.$$

            For the $\phi_{\texttt{St}} := (\leq_0 \texttt{enrolledFor}.\top)$ definition, there is $M(\mathcal{D}) \models \tau_{\texttt{St}}(001)$, but $M(\mathcal{D}) \not\models \phi_{\texttt{St}}(001)$ since $\texttt{St}(001) \in M(\mathcal{D})$ violates the at most zero cardinality.
         2. Defn. $\phi_{\texttt{St}} := (\leq_0 \texttt{enrolledFor}.\texttt{St})$:
            There is $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_0 \texttt{enrolledFor}.\texttt{St}) \rangle$ since M ensures that there will not be an instance $\texttt{St}(n)$ in $M(\mathcal{D})$ for any $\mathcal{D}$ that is reachable from the target $\tau_{\texttt{St}}$ node with the property path '$\texttt{enrolledFor}$'.
         3. Defn. $\phi_{\texttt{St}} := (\leq_0 \texttt{enrolledFor}.\texttt{Cu})$:
            For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_0 \texttt{enrolledFor}.\texttt{Cu}) \rangle$ follows from the counter-example of case (1).
         4. Defn. $\phi_{\texttt{St}} := (\geq_0 \texttt{enrolledFor}.\top)$:
            For the constraint, we have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_0 \texttt{enrolledFor}.\top) \rangle$ since the minimum zero cardinality is always satisfied.

[20] For simplicity, w.l.o.g. we ignore these cases $s :=$ $\langle \exists\,\texttt{enrolledFor}, \tau_{\exists\texttt{enrolledFor}}, \phi_{\exists\texttt{enrolledFor}} \rangle \mid \langle \exists\,\texttt{enrolledFor}^{-}, \tau_{\exists\texttt{enrolledFor}^{-}}, \phi_{\exists\texttt{enrolledFor}^{-}} \rangle$. for now, i.e., constraints on the domain and range of the property path '$\texttt{enrolledFor}$' and its inverse '$\texttt{enrolledFor}^{-}$'.

[21] ℓ is removed from sub-grammar α because we do not have datatype predicates in $sch(M)$, and $\bot$ (false, i.e. constraints that will always be violated) is shorthand for $\neg\top$.

[22] i.e. 'typing' for the value node in the data graph $M(\mathcal{D})$ that can be reached from the target $\tau_{\texttt{St}}$ node with the property path '$\texttt{enrolledFor}$'.

5. Defn. $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.St})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\geq_0\ \mathtt{enrolledFor.St})\rangle$ follows from the same reason as in the previous case (4).
6. Defn. $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.Cu})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\geq_0\ \mathtt{enrolledFor.Cu})\rangle$ follows from the same reason as in the previous case (4).
7. Defn $\phi_{\mathtt{St}} := (=_0\ \mathtt{enrolledFor.}\top)$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_0\ \mathtt{enrolledFor.}\top)\rangle$ follows from the previous cases (1) and (4).
8. Defn. $\phi_{\mathtt{St}} := (=_0\ \mathtt{enrolledFor.St})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_0\ \mathtt{enrolledFor.St})\rangle$ follows from the previous cases (2) and (5).
9. Defn. $\phi_{\mathtt{St}} := (=_0\ \mathtt{enrolledFor.Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_0\ \mathtt{enrolledFor.Cu})\rangle$ follows from the previous cases (3) and (6).

b. Defn. $\phi_{\mathtt{St}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
1. Defn. $\phi_{\mathtt{St}} := (\leq_0\ \mathtt{enrolledFor.}\neg\mathtt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0\ \mathtt{enrolledFor.}\neg\mathtt{St})\rangle$ follows from the previous case (a).(3).
2. Defn. $\phi_{\mathtt{St}} := (\leq_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})$:
   The $\phi_{\mathtt{St}} := (\leq_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})$ declares that if there exists any node in the graph that is reachable from a node n s.t. $\langle n, \mathtt{rdf{:}type}, \mathtt{St}\rangle$ via property path enrolledFor then that must be no other than the node n' s.t. $\langle n', \mathtt{rdf{:}type}, \mathtt{Cu}\rangle$. The M ensures that for every $\mathtt{St}(n) \in M(\mathcal{D})$ s.t. $\mathtt{enrolledFor}(n, n') \in M(\mathcal{D})$ with $\mathtt{Cu}(n') \in M(\mathcal{D})$ following as reason as in the previous case (a).(2). Thus, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})\rangle$.
3. Defn. $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.}\neg\mathtt{St})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\geq_0\ \mathtt{enrolledFor.}\neg\mathtt{St})\rangle$. Note that the constraints is equivalent to previous $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.Cu})$ in the case (a).(6).
4. Defn. $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\geq_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})\rangle$. Note that the constraints is equivalent to previous $\phi_{\mathtt{St}} := (\geq_0\ \mathtt{enrolledFor.St})$ in the case (a).(5).
5. Defn. $\phi_{\mathtt{St}} := (=_0\ \mathtt{enrolledFor.}\neg\mathtt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_0\ \mathtt{enrolledFor.}\neg\mathtt{St})\rangle$ follows from the previous cases (1) and (3).
6. Defn. $\phi_{\mathtt{St}} := (=_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_0\ \mathtt{enrolledFor.}\neg\mathtt{Cu})\rangle$ follows from the previous cases (2) and (4).

c. Defn. $\phi_{\mathtt{St}}$ with $\{\leq_1, \geq_1, =_1\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}\}$ typing:
1. Defn. $\phi_{\mathtt{St}} := (\leq_1\ \mathtt{enrolledFor.}\top)$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_1\ \mathtt{enrolledFor.}\top)\rangle$ follows from the arguments for the case $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_1\ \mathtt{enrolledFor.Cu})\rangle$, see details in the proof section $\Sigma \models_M S''$ above.
2. Defn. $\phi_{\mathtt{St}} := (\leq_1\ \mathtt{enrolledFor.St})$:
   We have $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_1\ \mathtt{enrolledFor.St})\rangle$ following same arguments as for the $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0\ \mathtt{enrolledFor.St})\rangle$ in the case (a).(2).
3. Defn. $\phi_{\mathtt{St}} := (\leq_1\ \mathtt{enrolledFor.Cu})$:

We have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_1 \texttt{enrolledFor}.\texttt{Cu})\rangle$ following arguments from the previous case (1).

4. Defn. $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\top)$:
   There is $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{enrolledFor}.\top)\rangle$. It also follows from the arguments in the previous case (1).
5. Defn. $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\texttt{St})$:
   There is $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{enrolledFor}.\texttt{St})\rangle$ following arguments from the case (a).(2).
6. Defn. $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\texttt{Cu})$:
   The $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{enrolledFor}.\texttt{Cu})\rangle$ follows from the arguments of the previous case (1).
7. Defn. $\phi_{\texttt{St}} := (=_1 \texttt{enrolledFor}.\top)$:
   For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{enrolledFor}.\top)\rangle$ follows from the previous cases (1) and (4).
8. Defn. $\phi_{\texttt{St}} := (=_1 \texttt{enrolledFor}.\texttt{St})$:
   The $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{enrolledFor}.\texttt{St})\rangle$ follows from the previous cases (2) and (5).
9. Defn. $\phi_{\texttt{St}} := (=_1 \texttt{enrolledFor}.\texttt{Cu})$:
   The $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{enrolledFor}.\texttt{Cu})\rangle$ follows from the previous cases (3) and (6).

d. Defn. $\phi_{\texttt{St}}$ with $\{\leq_1, \geq_1, =_1\}$ cardinality and $\{\neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
   1. Defn. $\phi_{\texttt{St}} := (\leq_1 \texttt{enrolledFor}.\neg\texttt{St})$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_1 \texttt{enrolledFor}.\neg\texttt{St})\rangle$. Note that the constraint is equivalent to $\phi_{\texttt{St}} := (\leq_1 \texttt{enrolledFor}.\texttt{Cu})$ in the case (c.3).
   2. Defn. $\phi_{\texttt{St}} := (\leq_1 \texttt{enrolledFor}.\neg\texttt{Cu})$:
      We have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_1 \texttt{enrolledFor}.\neg\texttt{Cu})\rangle$. Note that the constraint is equivalent to $\phi_{\texttt{St}} := (\leq_1 \texttt{enrolledFor}.\texttt{St})$, see the case (c.2).
   3. Defn. $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\neg\texttt{St})$:
      We have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{enrolledFor}.\neg\texttt{St})\rangle$. Note that the constraint is equivalent to $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\texttt{Cu})$, see the previous case (c.6).
   4. Defn. $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\neg\texttt{Cu})$:
      We have $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{enrolledFor}.\neg\texttt{Cu})\rangle$. Note that the constraint is equivalent to $\phi_{\texttt{St}} := (\geq_1 \texttt{enrolledFor}.\texttt{St})$, see the case (c.5).
   5. Defn. $\phi_{\texttt{St}} := (=_1 \texttt{enrolledFor}.\neg\texttt{St})$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{enrolledFor}.\neg\texttt{St})\rangle$ follows from the previous cases (1) and (3).
   6. Defn. $\phi_{\texttt{St}} := (=_1 \texttt{enrolledFor}.\neg\texttt{Cu})$:
      For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{enrolledFor}.\neg\texttt{Cu})\rangle$ follows from the previous cases (2) and (4).

e. Defn. $\phi_{\texttt{St}}$ with $\{\leq_n, \geq_n, =_n\}$ such that $n \geq 2$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}\}$ typing:
   1. Defn. $\phi_{\texttt{St}} := (\leq_n \texttt{enrolledFor}.\top)$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_n \texttt{enrolledFor}.\top)\rangle$ follows from the previous case (c.1).
   2. Defn. $\phi_{\texttt{St}} := (\leq_n \texttt{enrolledFor}.\texttt{St})$:
      We have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_n \texttt{enrolledFor}.\texttt{St})\rangle$ following the same arguments as for the case (c.2).
   3. Defn. $\phi_{\texttt{St}} := (\leq_n \texttt{enrolledFor}.\texttt{Cu})$:
      We have $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_n \texttt{enrolledFor}.\texttt{Cu})\rangle$ following the same arguments as for the case (c.3).
   4. Defn. $\phi_{\texttt{St}} := (\geq_n \texttt{enrolledFor}.\top)$:

We have $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_n \texttt{ enrolledFor}. \top) \rangle$ since $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{ enrolledFor}. \top) \rangle$ from the case (c.7).

5. Defn. $\phi_{\texttt{St}} := (\geq_n \texttt{ enrolledFor}. \texttt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_n \texttt{ enrolledFor}. \texttt{St}) \rangle$ since $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_1 \texttt{ enrolledFor}. \texttt{St}) \rangle$ from the case (c.5).
6. Defn. $\phi_{\texttt{St}} := (\geq_n \texttt{ enrolledFor}. \texttt{Cu})$:
   We have $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_n \texttt{ enrolledFor}. \texttt{Cu}) \rangle$ since $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_1 \texttt{ enrolledFor}. \top) \rangle$ from the case (c.7).
7. Defn. $\phi_{\texttt{St}} := (=_n \texttt{ enrolledFor}. \top)$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_n \texttt{ enrolledFor}. \top) \rangle$ following the previous case (4).
8. Defn. $\phi_{\texttt{St}} := (=_n \texttt{ enrolledFor}. \texttt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_n \texttt{ enrolledFor}. \texttt{St}) \rangle$ following the previous case (5).
9. Defn. $\phi_{\texttt{St}} := (=_n \texttt{ enrolledFor}. \texttt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_n \texttt{ enrolledFor}. \texttt{Cu}) \rangle$ following the previous case (6).

f. Defn. $\phi_{\texttt{St}}$ with $\{\leq_n, \geq_n, =_n, \}$ such that $n \geq 2$ cardinality and $\{\neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
   1. Defn. $\phi_{\texttt{St}} := (\leq_n \texttt{ enrolledFor}. \neg\texttt{St})$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_n \texttt{ enrolledFor}. \neg\texttt{St}) \rangle$ follows from the previous case (e.3).
   2. Defn. $\phi_{\texttt{St}} := (\leq_n \texttt{ enrolledFor}. \neg\texttt{Cu})$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\leq_n \texttt{ enrolledFor}. \neg\texttt{Cu}) \rangle$ follows from the previous case (e.2).
   3. Defn. $\phi_{\texttt{St}} := (\geq_n \texttt{ enrolledFor}. \neg\texttt{St})$:
      For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_n \texttt{ enrolledFor}. \neg\texttt{St}) \rangle$ follows from the previous case (e.6).
   4. Defn. $\phi_{\texttt{St}} := (\geq_n \texttt{ enrolledFor}. \neg\texttt{Cu})$:
      For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\geq_n \texttt{ enrolledFor}. \neg\texttt{Cu}) \rangle$ follows from the previous case (e.5).
   5. Defn. $\phi_{\texttt{St}} := (=_n \texttt{ enrolledFor}. \neg\texttt{St})$:
      For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_n \texttt{ enrolledFor}. \neg\texttt{St}) \rangle$ follows from the previous case (e.9).
   6. Defn. $\phi_{\texttt{St}} := (=_n \texttt{ enrolledFor}. \neg\texttt{Cu})$:
      For the constraint, $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (=_n \texttt{ enrolledFor}. \neg\texttt{Cu}) \rangle$ follows from the previous case (e.8).

g. Defn. $\phi_{\texttt{St}}$ with $\{\rhd_{\texttt{St}}, \unrhd_{\texttt{St}}\}$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}\}$ typing:
   1. Defn. $\phi_{\texttt{St}} := (\rhd_{\texttt{St}} \texttt{ enrolledFor}. \top)$:
      For the constraint, we have $\Sigma \not\models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\rhd_{\texttt{St}} \texttt{ enrolledFor}. \top) \rangle$.
      Assume $\mathcal{D} = \{\text{st}(001,\_, \text{CS40}), \text{st}(002,\_, \text{CS40}), \text{cu}(\text{CS40}, \text{Logic})\}$ such that $\mathcal{D} \models \Sigma$. Then,

      $$\mathcal{M}(\mathcal{D}) = \{\texttt{St}(001), \texttt{St}(002), \texttt{Cu}(\text{CS40}), \texttt{enrolledFor}(001, \text{CS40}), \\ \texttt{enrolledFor}(002, \text{CS40})\}.$$

      From the semantics of $\rhd_{\texttt{Cu}}$ cardinality, we have $M(\mathcal{D}) \not\models \phi_{\texttt{St}}$ since there are two '$\texttt{enrolledFor}^-$'- successor for the $\texttt{Cu}(\text{CS40}) \in M(\mathcal{D})$.
   2. Defn. $\phi_{\texttt{St}} := (\rhd_{\texttt{St}} \texttt{ enrolledFor}. \texttt{St})$:
      For the constraint, $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, (\rhd_{St} \texttt{ enrolledFor}. \texttt{St}) \rangle$ follows from the mapping M definition.

3. Defn. $\phi_{\mathtt{St}} := (\rhd_{\mathtt{St}}\ \mathtt{enrolledFor.Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\rhd_{St}\ \mathtt{enrolledFor.Cu})\rangle$ follows from the case (1).
4. Defn. $\phi_{\mathtt{St}} := (\unrhd_{\mathtt{St}}\ \mathtt{enrolledFor}.\top)$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\unrhd_{St}\ \mathtt{enrolledFor}.\top)\rangle$ follows from the case (1).
5. Defn. $\phi_{\mathtt{St}} := (\unrhd_{\mathtt{St}}\ \mathtt{enrolledFor.St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\unrhd_{St}\ \mathtt{enrolledFor.St})\rangle$ follows from the semantics $\rhd_{\mathtt{St}}$ and the $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_1\ \mathtt{enrolledFor.St})\rangle$ in the case c.8.
6. Defn. $\phi_{\mathtt{St}} := (\unrhd_{\mathtt{St}}\ \mathtt{enrolledFor.Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\unrhd_{St}\ \mathtt{enrolledFor.Cu})\rangle$ follows from the case (1).

h. Defn. $\phi_{\mathtt{St}}$ with $\{\rhd_{\mathtt{St}}, \unrhd_{\mathtt{St}}\}$ cardinality and $\{\neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
1. Defn. $\phi_{\mathtt{St}} := (\rhd_{\mathtt{St}}\ \mathtt{enrolledFor}.\neg\mathtt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\rhd_{St}\ \mathtt{enrolledFor}.\neg\mathtt{St})\rangle$ follows from the case (g.3).
2. Defn. $\phi_{\mathtt{St}} := (\rhd_{\mathtt{St}}\ \mathtt{enrolledFor}.\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\rhd_{St}\ \mathtt{enrolledFor}.\neg\mathtt{Cu})\rangle$ follows from the case (g.2).
3. Defn. $\phi_{\mathtt{St}} := (\unrhd_{\mathtt{St}}\ \mathtt{enrolledFor}.\neg\mathtt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\unrhd_{St}\ \mathtt{enrolledFor}.\neg\mathtt{St})\rangle$ follows from the case (g.6).
4. Defn. $\phi_{\mathtt{St}} := (\unrhd_{\mathtt{St}}\ \mathtt{enrolledFor}.\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, (\unrhd_{St}\ \mathtt{enrolledFor}.\neg\mathtt{Cu})\rangle$ follows from the case (g.5).

ii. Defn. $\phi_{\mathtt{St}}$ on $\{\mathtt{St}, \mathtt{Cu}, \mathtt{enrolledFor}^-\}$

a. Defn. $\phi_{\mathtt{St}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   For the constraints, we have $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle$ since M ensures that there will not be an instance in $M(\mathcal{D})$ for any $\mathcal{D}$ that can be reached from the $\tau_{\mathtt{St}}$ (i.e., any $\mathtt{St}(n) \in M(\mathcal{D})$) node with the property path '$\mathtt{enrolledFor}^-$'.
b. Defn. $\phi_{\mathtt{St}}$ with $\{\leq_n\}$ such that $n \geq 1$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   For the constraints, we have $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle$ following the same arguments from previous case (a).
c. Defn. $\phi_{\mathtt{St}}$ with $\{\geq_n, =_n\}$ such that $n \geq 1$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   For the constraints, we have $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle$.
   Let $\phi_{\mathtt{St}} := (\geq_1\ \mathtt{enrolledFor}^-.\top)$. Assume $\mathcal{D} = \{\mathrm{st}(001,\_,\mathrm{CS40}), \mathrm{cu}(\mathrm{CS40}, \mathrm{Logic})\}$ such that $\mathcal{D} \models \Sigma$. Then,

$$\mathcal{M}(\mathcal{D}) = \{\mathtt{St}(001), \mathtt{Cu}(\mathrm{CS40}), \mathtt{enrolledFor}(001, \mathrm{CS40})\}.$$

   Then $M(\mathcal{D}) \not\models \phi_{\mathtt{St}}(001)$ since $\mathtt{St}(001) \in M(\mathcal{D})$ violates the at least one cardinality requirement for the property path '$\mathtt{enrolledFor}^-$'. Similar arguments exist for the rest of the constraints $\phi_{\mathtt{St}}$ definitions as well.
d. Defn. $\phi_{\mathtt{St}}$ with $\{\rhd_{\mathtt{St}}\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   For the constraints, we have $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle$ following the same arguments as in the first case (a).
e. Defn. $\phi_{\mathtt{St}}$ with $\{\unrhd_{\mathtt{St}}\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   For the constraints, we have $\Sigma \not\models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}}\rangle$ following the previous case (c).

B. For $\langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, \phi_{\mathtt{Cu}}\rangle \in S'$ – with implicit targetClass $\tau_{\mathtt{Cu}}$ that declares the instances of RDF concept *Cu* as target nodes, there exist following $\phi_{\mathtt{Cu}}$ definitions on RDF vocabularies $\{\mathtt{St}, \mathtt{Cu}, \mathtt{enrolledFor}^{\pm}\}$:

i. Defn. $\phi_{\mathtt{Cu}}$ on $\{\mathtt{St}, \mathtt{Cu}, \mathtt{enrolledFor}^{-}\}$:

a. Defn. $\phi_{\mathtt{Cu}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}\}$ typing [23]:

1. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\top)$:
We have $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\leq_0 \mathtt{enrolledFor}^{-}.\top)\rangle$.
Assume $\mathcal{D} = \{\text{st}(001, \_, \text{CS40}), \text{cu}(\text{CS40}, \text{Logic})\}$ such that $\mathcal{D} \models \Sigma$. Then,

$$\mathcal{M}(\mathcal{D}) = \{\mathtt{St}(001), \mathtt{Cu}(\text{CS40}), \mathtt{enrolledFor}^{-}(001, \text{CS40})\}.$$

There is $M(\mathcal{D}) \models \tau_{\mathtt{Cu}}(\text{CS40})$, but $M(\mathcal{D}) \not\models \phi_{\mathtt{Cu}}(\text{CS40})$ since $\mathtt{Cu}(\text{CS40})$ in $M(\mathcal{D})$ violates the at most zero cardinality.

2. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\mathtt{St})$:
The $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\leq_0 \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ follows from previous case (1).

3. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\leq_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})\rangle$ since M ensures that there will not be an instance $\mathtt{Cu}(n)$ in $M(\mathcal{D})$ for any $\mathcal{D}$ that is reachable from the target $\tau_{\mathtt{Cu}}$ node with the property path '$\mathtt{enrolledFor}^{-}$'.

4. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\top)$:
For the constraint, we have $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_0 \mathtt{enrolledFor}^{-}.\top)\rangle$ since the minimum zero cardinality is always satisfied.

5. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{St})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ follows from the same reason as in the previous case (4).

6. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})\rangle$ follows from the same reason as in the previous case (4).

7. Defn $\phi_{\mathtt{Cu}} \coloneqq (=_0 \mathtt{enrolledFor}^{-}.\top)$:
For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_0 \mathtt{enrolledFor}^{-}.\top)\rangle$ follows from the previous cases (1) and (4).

8. Defn. $\phi_{\mathtt{Cu}} \coloneqq (=_0 \mathtt{enrolledFor}^{-}.\mathtt{St})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_0 \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ follows from the previous cases (2) and (5).

9. Defn. $\phi_{\mathtt{Cu}} \coloneqq (=_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})$:
For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})\rangle$ follows from the previous cases (3) and (6).

b. Defn. $\phi_{\mathtt{Cu}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:

1. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{St})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\leq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{St})\rangle$ follows from the previous case (a).(3).

2. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
The $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\leq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ follows from the previous case (a).(2).

3. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{St})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{St})\rangle$ since the minimum zero cardinality is always satisfied. Note that the constraints is equivalent to $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{Cu})$.

4. Defn. $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ since the minimum zero cardinality is always satisfied. Note that the constraints is equivalent to $\phi_{\mathtt{Cu}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{St})$.

[23] i.e. 'typing' for the value node in the data graph $M(\mathcal{D})$ that can be reached from the target node with the property path '$\mathtt{enrolledFor}^{-}$'.

5. Defn. $\phi_{\texttt{Cu}} := (=_0 \texttt{ enrolledFor}^-.\neg\texttt{St})$:
   For the constraint, $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (=_0 \texttt{ enrolledFor}^-.\neg\texttt{St})\rangle$ follows from the previous cases (1) and (3).
6. Defn. $\phi_{\texttt{Cu}} := (=_0 \texttt{ enrolledFor}^-.\neg\texttt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (=_0 \texttt{ enrolledFor}^-.\neg\texttt{Cu})\rangle$ follows from the previous cases (2) and (4).

c. Defn. $\phi_{\texttt{Cu}}$ with $\{\leq_n, \geq_n, =_n\}$ such that $n \geq 1$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}\}$ typing:
1. Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{ enrolledFor}^-.\top)$:
   For the constraints $\phi_{\texttt{Cu}}$ when n is any fixed natural number, we have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_n \texttt{ enrolledFor}^-.\top)\rangle$. There always exists a counter-example for any fixed at most n cardinality restriction since $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{ enrolledFor}^-.\neg\texttt{St})\rangle$ in the previous case (b.1).
2. Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{ enrolledFor}^-.\texttt{St})$:
   We have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_n \texttt{ enrolledFor}^-.\texttt{St})\rangle$ following the previous case (1).
3. Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{ enrolledFor}^-.\texttt{Cu})$:
   We have $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_n \texttt{ enrolledFor}^-.\texttt{Cu})\rangle$ following the previous case (a.3).
4. Defn. $\phi_{\texttt{Cu}} := (\geq_n \texttt{ enrolledFor}^-.\top)$:
   For any fixed n, we have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\geq_n \texttt{ enrolledFor}^-.\top)\rangle$ since $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{ enrolledFor}^-.\neg\texttt{St})\rangle$ in the previous case (b.1).
5. Defn. $\phi_{\texttt{Cu}} := (\geq_n \texttt{ enrolledFor}^-.\texttt{St})$:
   For the constraints, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\geq_n \texttt{ enrolledFor}^-.\texttt{St})\rangle$ following the same arguments as in the previous case (1).
6. Defn. $\phi_{\texttt{Cu}} := (\geq_n \texttt{ enrolledFor}^-.\texttt{Cu})$:
   For the constraints, we have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\geq_n \texttt{ enrolledFor}^-.\texttt{Cu})\rangle$.
   Let $\phi_{\texttt{Cu}} := (\geq_1 \texttt{ enrolledFor}^-.\texttt{Cu})$. Assume $\mathcal{D} = \{\text{cu(CS40, Logic)}\}$ such that $\mathcal{D} \models \Sigma$. Then,
   $$\mathcal{M}(\mathcal{D}) = \{\texttt{Cu}(\text{CS40})\}.$$
   There is $M(\mathcal{D}) \models \tau_{\texttt{Cu}}(\text{CS40})$, but $M(\mathcal{D}) \not\models \phi_{\texttt{Cu}}(\text{CS40})$ since $\texttt{Cu}(\text{CS40}) \in M(\mathcal{D})$ violates the at least one cardinality.
7. Defn. $\phi_{\texttt{Cu}} := (=_n \texttt{ enrolledFor}^-.\top)$:
   For the constraints, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (=_n \texttt{ enrolledFor}^-.\top)\rangle$ following the previous case (1).
8. Defn. $\phi_{\texttt{Cu}} := (=_n \texttt{ enrolledFor}^-.\texttt{St})$:
   For the constraints, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (=_n \texttt{ enrolledFor}^-.\texttt{St})\rangle$ following the previous cases (2) and (5).
9. Defn. $\phi_{\texttt{Cu}} := (=_n \texttt{ enrolledFor}^-.\texttt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (=_n \texttt{ enrolledFor}^-.\texttt{Cu})\rangle$ following the previous case (6).

d. Defn. $\phi_{\texttt{Cu}}$ with $\{\leq_n, \geq_n, =_n, \}$ such that $n \geq 1$ cardinality and $\{\neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
1. Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{ enrolledFor}^-.\neg\texttt{St})$:
   For the constraint, $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_n \texttt{ enrolledFor}^-.\neg\texttt{St})\rangle$ follows from the previous case (c.3).
2. Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{ enrolledFor}^-.\neg\texttt{Cu})$:
   For the constraint, $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_n \texttt{ enrolledFor}^-.\neg\texttt{Cu})\rangle$ follows from the previous case (c.2).
3. Defn. $\phi_{\texttt{Cu}} := (\geq_n \texttt{ enrolledFor}^-.\neg\texttt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\geq_n \texttt{ enrolledFor}^-.\neg\texttt{St})\rangle$ follows from the previous case (c.6).

4. Defn. $\phi_{\mathtt{Cu}} := (\geq_n \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\geq_n \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ follows from the previous case (c.5).
5. Defn. $\phi_{\mathtt{Cu}} := (=_n \mathtt{enrolledFor}^{-}.\neg\mathtt{St})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_n \mathtt{enrolledFor}^{-}.\neg\mathtt{St})\rangle$ follows from the previous case (c.9).
6. Defn. $\phi_{\mathtt{Cu}} := (=_n \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
   For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_n \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ follows from the previous case (c.8).

e. Defn. $\phi_{\mathtt{Cu}}$ with $\{\rhd_{\mathtt{Cu}}, \unrhd_{\mathtt{Cu}}\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}\}$ typing:
   1. Defn. $\phi_{\mathtt{Cu}} := (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\top)$:
      For the constraint, we have $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\top)\rangle$. Assume $\mathcal{D}$ = {st(001,_, CS40), st(002,_, CS40), cu(CS40, Logic)} such that $\mathcal{D} \models \Sigma$. Then,

$$\mathcal{M}(\mathcal{D}) = \{\mathtt{St}(001), \mathtt{St}(002), \mathtt{Cu}(\mathrm{CS40}), \mathtt{enrolledFor}^{-}(001, \mathrm{CS40}), \mathtt{enrolledFor}^{-}(002, \mathrm{CS40})\}.$$

      From the semantics of $\rhd_{\mathtt{Cu}}$ cardinality, we have $M(\mathcal{D}) \not\models \phi_{\mathtt{Cu}}$, i.e., there are two '$\mathtt{enrolledFor}^{-}$'-successor for the $\mathtt{Cu}(\mathrm{CS40}) \in M(\mathcal{D})$.
   2. Defn. $\phi_{\mathtt{Cu}} := (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\mathtt{St})$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\rhd_{Cu} \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ follows the previous case (1).
   3. Defn. $\phi_{\mathtt{Cu}} := (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\mathtt{Cu})$:
      For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\rhd_{Cu} \mathtt{enrolledFor}^{-}.\mathtt{Cu})\rangle$ follows the case (a.3).
   4. Defn. $\phi_{\mathtt{Cu}} := (\unrhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\top)$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\unrhd_{Cu} \mathtt{enrolledFor}^{-}.\top)\rangle$ follows the case (1).
   5. Defn. $\phi_{\mathtt{Cu}} := (\unrhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\mathtt{St})$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\unrhd_{Cu} \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ follows from the semantics $\rhd_{\mathtt{Cu}}$ and the $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (=_1 \mathtt{enrolledFor}^{-}.\mathtt{St})\rangle$ in the case c.8.
   6. Defn. $\phi_{\mathtt{Cu}} := (\unrhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\mathtt{Cu})$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\unrhd_{Cu} \mathtt{enrolledFor}^{-}.\mathtt{Cu})\rangle$ follows from the case (c.9).

f. Defn. $\phi_{\mathtt{Cu}}$ with $\{\rhd_{\mathtt{Cu}}, \unrhd_{\mathtt{Cu}}\}$ cardinality and $\{\neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:
   1. Defn. $\phi_{\mathtt{Cu}} := (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\neg\mathtt{St})$:
      For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\rhd_{St} \mathtt{enrolledFor}^{-}.\neg\mathtt{St})\rangle$ follows from the case (e.3).
   2. Defn. $\phi_{\mathtt{Cu}} := (\rhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
      For the constraint, $\Sigma \models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\rhd_{Cu} \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ follows from the case (e.2).
   3. Defn. $\phi_{\mathtt{Cu}} := (\unrhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\neg\mathtt{St})$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\unrhd_{Cu} \mathtt{enrolledFor}^{-}.\neg\mathtt{St})\rangle$ follows from the previous case (1).
   4. Defn. $\phi_{\mathtt{Cu}} := (\unrhd_{\mathtt{Cu}} \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})$:
      For the constraint, $\Sigma \not\models_M \langle \mathtt{Cu}, \tau_{\mathtt{Cu}}, (\unrhd_{Cu} \mathtt{enrolledFor}^{-}.\neg\mathtt{Cu})\rangle$ follows from the previous case (2).

ii. Defn. $\phi_{\mathtt{Cu}}$ on $\{\mathtt{St}, \mathtt{Cu}, \mathtt{enrolledFor}\}$
   a. Defn. $\phi_{\mathtt{Cu}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:

For the constraints $\phi_{\texttt{Cu}}$, we have $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$ since M ensures that there will not be an instance in $M(\mathcal{D})$ for any $\mathcal{D}$ that can be reached from $\tau_{\texttt{Cu}}$ (i.e., any $\texttt{Cu}(n) \in M(\mathcal{D})$) node with the property path '`enrolledFor`'.

b. Defn. $\phi_{\texttt{Cu}}$ with $\{\leq_n\}$ such that $n \geq 1$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For the constraints, we have $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$ following the same arguments from previous case (a).

c. Defn. $\phi_{\texttt{Cu}}$ with $\{\geq_n, =_n\}$ such that $n \geq 1$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For the constraints, we have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$.
Let $\phi_{\texttt{Cu}} := (\geq_1 \texttt{enrolledFor}.\top)$. Assume $\mathcal{D} = \{\text{cu(CS40, Logic)}\}$ such that $\mathcal{D} \models \Sigma$. Then,

$$\mathcal{M}(\mathcal{D}) = \{\texttt{Cu}(\text{CS40})\}.$$

Then $M(\mathcal{D}) \not\models \phi_{\texttt{Cu}}(\text{CS40})$ since $\texttt{Cu}(\text{CS40}) \in M(\mathcal{D})$ violates the at least one cardinality requirement for the property path '`enrolledFor`'. Similar arguments exist for the rest of the constraints $\phi_{\texttt{Cu}}$ as well.

d. Defn. $\phi_{\texttt{Cu}}$ with $\{\rhd_{\texttt{Cu}}\}$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For the constraints, we have $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$ following the same arguments as in the previous case (a).

e. Defn. $\phi_{\texttt{Cu}}$ with $\{\unrhd_{\texttt{Cu}}\}$ cardinality and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For the constraints, we have $\Sigma \not\models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$ following the previous case (c).

Next, observe that all the SHACL document S' such that $\Sigma \models_M S'$ are :

$$S' \subseteq \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle, \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle\}$$

where $\phi_{\texttt{St}}$ and $\phi_{\texttt{Cu}}$ are the sets of constraints (i.e., including their possible conjunctions) such that $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle$ and $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$, see the listed $\phi_{\texttt{St}}$ and $\phi_{\texttt{Cu}}$ constraints above.

Finally, we now establish $\Sigma \models_M^* S$, i.e., for every SHACL document $\Sigma \models_M S'$ and every RDF graph G : $G \models S \rightarrow G \models S'$. Recall that $S = \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} := (=_1 \ \texttt{enrolledFor} \ .\text{Cu}) \wedge (\leq_0 \ \texttt{enrolledFor} \ .\neg\text{Cu})\rangle, \langle \texttt{Cu}, \tau_{\texttt{C}}, \phi_{\texttt{Cu}} := (\leq_0 \ \text{enrolledFor}^-.\neg\texttt{St})\rangle\}$. Then, there are the following three cases to consider for the $\Sigma \models_M S'$:

1. $\Sigma \models_M S'$ such that $S' = \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle\}$:
Observe that, for any RDF graph G, whenever there is

$$\bigwedge_{\text{node } n \text{ in } \in G} .(G \models \tau_{\texttt{St}}(n) \rightarrow G \models \phi_{\texttt{St}}(n))$$

for the $\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} := (=_1 \ . \ \texttt{enrolledFor}.\ \text{Cu}) \wedge (\leq_0 \ . \ \texttt{enrolledFor}.\ \neg\text{Cu})\rangle \in S$, the sentence

$$\bigwedge_{\text{node } n \text{ in } \in G} .(G \models \tau_{\texttt{St}}(n) \rightarrow G \models \phi_{\texttt{St}}(n))$$

must hold for the $S' = \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle\}$.

Let G be an arbitrary graph with $G \models S$, i.e.,

$$G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} := (=_1 \ \ \texttt{enrolledFor} \ .\text{Cu}) \wedge (\leq_0 \ \ \texttt{enrolledFor} \ .\neg\text{Cu})\rangle$$

and

$$G \models \langle \texttt{Cu}, \tau_{\texttt{C}}, \phi_{\texttt{Cu}} := (\leq_0 \ \text{enrolledFor}^-.\neg\texttt{St})\rangle.$$

Then, there must be $G \models S'$.

- Base case: In this part of the proof, we need to consider the following Defn. $\phi_{\mathtt{St}}$ such that $\Sigma \models_M \langle \mathtt{St}, \tau_{\mathtt{St}}, \phi_{\mathtt{St}} \rangle$.

  (A).(i).(a).(2) Defn. $\phi_{\mathtt{St}} := (\leq_0 \mathtt{enrolledFor}.\mathtt{St})$:
  For the $\phi_{\mathtt{St}}$, $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0 \mathtt{enrolledFor}.\mathtt{St}) \rangle$ is trivial since $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0 \mathtt{enrolledFor}.\neg\mathtt{Cu}) \rangle$, i.e., there will not be any node n s.t. $\mathtt{St}(n) \in G$ reachable from $\tau_{\mathtt{St}}$ node with the property path '`enrolledFor`' other than the node n' s.t. $\langle n', \mathtt{rdf:type}, \mathtt{Cu} \rangle$.

  (A).(i).(a).(4) Defn. $\phi_{\mathtt{St}} := (\geq_0 \mathtt{enrolledFor}.\top)$:
  For the $\phi_{\mathtt{St}}$, $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\geq_0 \mathtt{enrolledFor}.\top) \rangle$ since the minimum zero cardinality is always satisfied.

  (A).(i).(a).(5) Defn. $\phi_{\mathtt{St}} := (\geq_0 \mathtt{enrolledFor}.\mathtt{St})$: same as for the case (A).(i).(a).(4)
  (A).(i).(a).(6) Defn. $\phi_{\mathtt{St}} := (\geq_0 \mathtt{enrolledFor}.\mathtt{Cu})$: same as for the case (A).(i).(a).(4)
  (A).(i).(a).(8) Defn. $\phi_{\mathtt{St}} := (=_0 \mathtt{enrolledFor}.\mathtt{St})$: same as for the case (A).(i).(a).(2)
  (A).(i).(b).(2) Defn. $\phi_{\mathtt{St}} := (\leq_0 \mathtt{enrolledFor}.\neg\mathtt{Cu})$: same as for the case (A).(i).(a).(2)
  (A).(i).(b).(3) Defn. $\phi_{\mathtt{St}} := (\geq_0 \mathtt{enrolledFor}.\neg\mathtt{St})$:
  For $\mathtt{St}(n) \in G$ of any G, if $G \models \phi_{\mathtt{St}}(n)$ for the $\phi_{\mathtt{St}} := (=_1 \mathtt{enrolledFor}.\mathtt{Cu})$ then $G \models \phi_{\mathtt{St}}(n)$ for the $(\leq_0 \mathtt{enrolledFor}.\neg\mathtt{St})$ is trivial. Thus, $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_0 \mathtt{enrolledFor}.\neg\mathtt{St}) \rangle$.

  (A).(i).(b).(4) Defn. $\phi_{\mathtt{St}} := (\geq_0 \mathtt{enrolledFor}.\neg\mathtt{Cu})$: trivial from the case (A).(i).(a).(4)
  (A).(i).(b).(6) Defn. $\phi_{\mathtt{St}} := (=_0 \mathtt{enrolledFor}.\neg\mathtt{Cu})$: same as for the case (A).(i).(a).(2)
  (A).(i).(c).(1) Defn. $\phi_{\mathtt{St}} := (\leq_1 \mathtt{enrolledFor}.\top)$: same as for the case (A).(i).(b).(3)
  (A).(i).(c).(2) Defn. $\phi_{\mathtt{St}} := (\leq_1 \mathtt{enrolledFor}.\mathtt{St})$: same as for the case (A).(i).(a).(2)
  (A).(i).(c).(3) Defn. $\phi_{\mathtt{St}} := (\leq_1 \mathtt{enrolledFor}.\mathtt{Cu})$: same as for the case (A).(i).(b).(3)
  (A).(i).(c).(4) Defn. $\phi_{\mathtt{St}} := (\geq_1 \mathtt{enrolledFor}.\top)$: same as for the case (A).(i).(b).(3)
  (A).(i).(c).(6) Defn. $\phi_{\mathtt{St}} := (\geq_1 \mathtt{enrolledFor}.\mathtt{Cu})$: same as for the case (A).(i).(b).(3)
  (A).(i).(c).(7) Defn. $\phi_{\mathtt{St}} := (=_1 \mathtt{enrolledFor}.\top)$: same as for the case (A).(i).(b).(3)
  (A).(i).(c).(9) Defn. $\phi_{\mathtt{St}} := (=_1 \mathtt{enrolledFor}.\mathtt{Cu})$: same as for the case (A).(i).(b).(3)
  (A).(i).(d).(1) Defn. $\phi_{\mathtt{St}} := (\leq_1 \mathtt{enrolledFor}.\neg\mathtt{St})$: same as for the case (A).(i).(b).(3)
  (A).(i).(d).(2) Defn. $\phi_{\mathtt{St}} := (\leq_1 \mathtt{enrolledFor}.\neg\mathtt{Cu})$: same as for the case (A).(i).(a).(3)
  (A).(i).(d).(3) Defn. $\phi_{\mathtt{St}} := (\geq_1 \mathtt{enrolledFor}.\neg\mathtt{St})$: same as for the case (A).(i).(b).(3)
  (A).(i).(d).(5) Defn. $\phi_{\mathtt{St}} := (=_1 \mathtt{enrolledFor}.\neg\mathtt{St})$: same as for the case (A).(i).(b).(3)
  (A).(i).(e).(1) Defn. $\phi_{\mathtt{St}} := (\leq_n \mathtt{enrolledFor}.\top)$ for $n \geq 2$:
  same as for the case (A).(i).(b).(2). Since $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_1 \mathtt{enrolledFor}.\mathtt{Cu}) \rangle$, the $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_n \mathtt{enrolledFor}.\top)$ is trivial. Note that, for any graph G,

  $$G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (=_1 \mathtt{enrolledFor}.\top) \rangle \rightarrow G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_1 \mathtt{enrolledFor}.\top) \rangle,$$

  and

  $$G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_1 \mathtt{enrolledFor}.\top) \rangle \rightarrow G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_n \mathtt{enrolledFor}.\top) \rangle.$$

  (A).(i).(e).(2) Defn. $\phi_{\mathtt{St}} := (\leq_n \mathtt{enrolledFor}.\mathtt{St})$ for $n \geq 2$: same as for the case (A).(i).(a).(2)
  (A).(i).(e).(3) Defn. $\phi_{\mathtt{St}} := (\leq_n \mathtt{enrolledFor}.\mathtt{Cu})$ for $n \geq 2$: same as for the case (A).(i).(b).(3)
  (A).(i).(f).(1) Defn. $\phi_{\mathtt{St}} := (\leq_n \mathtt{enrolledFor}.\neg\mathtt{St})$ for $n \geq 2$: same as for the case (A).(i).(b).(3)
  (A).(i).(f).(2) Defn. $\phi_{\mathtt{St}} := (\leq_n \mathtt{enrolledFor}.\neg\mathtt{Cu})$ for $n \geq 2$: same as for the case (A).(i).(a).(2)
  (A).(i).(g).(2) Defn. $\phi_{\mathtt{St}} := (\rhd_{\mathtt{St}} \mathtt{enrolledFor}.\mathtt{St})$:
  Since $G \models \langle \mathtt{St}, \tau_{\mathtt{St}}, (\leq_n \mathtt{enrolledFor}.\neg\mathtt{Cu}) \rangle$, there will not be any instance $\mathtt{St}(n) \in G$ that is reachable from $\tau_{\mathtt{St}}$ node with the property path '`enrolledFor`'.
  (A).(i).(h).(2) Defn. $\phi_{\mathtt{St}} := (\rhd_{\mathtt{St}} \mathtt{enrolledFor}.\neg\mathtt{Cu})$: same as for the case (A).(i).(g).(2)
  (A).(ii).(a) Defn. $\phi_{\mathtt{St}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality on path $\mathtt{enrolledFor}^-$ and $\{\top, \mathtt{St}, \mathtt{Cu}, \neg\mathtt{St}, \neg\mathtt{Cu}\}$ typing:

For all $\phi_{\texttt{St}}$, same as for the case (A).(i).(a).(2), i.e., there will not be any $\texttt{St}(n) \in G$ or $\texttt{Cu}(n) \in G$ in any G s.t. $G \models S'$ reachable from $\tau_{\texttt{St}}$ node with the property path '$\texttt{enrolledFor}^{-}$'.

(A).(ii).(b) Defn. $\phi_{\texttt{St}}$ with $\{\leq_n\}$ such that $n \geq 1$ cardinality on path $\texttt{enrolledFor}^{-}$ and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For all $\phi_{\texttt{St}}$, same as for the case (A).(ii).(a).

(A).(ii).(d) Defn. $\phi_{\texttt{St}}$ with $\{\triangleright_{\texttt{St}}\}$ cardinality on path $\texttt{enrolledFor}^{-}$ and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For all $\phi_{\texttt{St}}$, same as for the case (A).(ii).(a).

- Inductive Case: In this part of the proof, we need to consider conjunction of the Defn. $\phi_{\texttt{St}}$ such that $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \coloneqq \phi^1_{\texttt{St}} \wedge \phi^2_{\texttt{St}} \rangle$, i.e., $S' = \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \coloneqq \phi^1_{\texttt{St}} \wedge \phi^2_{\texttt{St}} \rangle$.

  Let $\phi^1_{\texttt{St}}$ and $\phi^2_{\texttt{St}}$ be two arbitrary base constraints (i.e. from the base cases listed above) such that $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, \phi^1_{\texttt{St}} \rangle$ and $\Sigma \models_M \langle \texttt{St}, \tau_{\texttt{St}}, \phi^2_{\texttt{St}} \rangle$. Then, for any graph G, we have

  $$G \models S \rightarrow G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi^1_{\texttt{St}} \rangle$$

  and

  $$G \models S \rightarrow G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi^2_{\texttt{St}} \rangle.$$

  Thus, $G \models S \rightarrow G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi^1_{\texttt{St}} \wedge \phi^2_{\texttt{St}} \rangle$.

2. $\Sigma \models_M S'$ such that $S' = \{\langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle\}$:
Let $\mathcal{D}$ be an arbitrary instance of $\mathcal{R}$ such that $G \models S$. That is,

$$G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \coloneqq (=_1 \texttt{ enrolledFor }.\texttt{Cu}) \wedge (\leq_0 \texttt{ enrolledFor }.\neg\texttt{Cu}) \rangle$$

and

$$G \models \langle \texttt{Cu}, \tau_{\texttt{C}}, \phi_{\texttt{Cu}} \coloneqq (\leq_0 \text{ enrolledFor}^{-}.\neg\texttt{St}) \rangle.$$

Then, there must be $G \models S'$.

- Base Case: In this part of the proof, we need to consider the following Defn. $\phi_{\texttt{Cu}}$ such that $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$.

  (B).(i).(a).(3) Defn. $\phi_{\texttt{Cu}} \coloneqq (\leq_0 \texttt{enrolledFor}^{-}.\texttt{Cu})$:
  The $G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{enrolledFor}^{-}.\texttt{Cu}) \rangle$ follows from the case $G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{enrolledFor}^{-}.\neg\texttt{St}) \rangle$, i.e.,

  $$G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{enrolledFor}^{-}.\neg\texttt{St}) \rangle \rightarrow G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\leq_0 \texttt{enrolledFor}^{-}.\texttt{Cu}) \rangle.$$

  (B).(i).(a).(4) Defn. $\phi_{\texttt{Cu}} \coloneqq (\geq_0 \texttt{enrolledFor}^{-}.\top)$:
  There is $G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, (\geq_0 \texttt{enrolledFor}^{-}.\top) \rangle$ since the minimum zero cardinality is always satisfied.

  (B).(i).(a).(5) Defn. $\phi_{\texttt{Cu}} \coloneqq (\geq_0 \texttt{enrolledFor}^{-}.\texttt{St})$: same as for the case (B).(i).(a).(4)
  (B).(i).(a).(6) Defn. $\phi_{\texttt{Cu}} \coloneqq (\geq_0 \texttt{enrolledFor}^{-}.\texttt{Cu})$: same as for the case (B).(i).(a).(4)
  (B).(i).(a).(9) Defn. $\phi_{\texttt{Cu}} \coloneqq (=_0 \texttt{enrolledFor}^{-}.\texttt{Cu})$: same as for the case (B).(i).(a).(3)
  (B).(i).(b).(1) Defn. $\phi_{\texttt{Cu}} \coloneqq (\leq_0 \texttt{enrolledFor}^{-}.\neg\texttt{St})$: same as for the case (B).(i).(a).(3)
  (B).(i).(b).(3) Defn. $\phi_{\texttt{Cu}} \coloneqq (\geq_0 \texttt{enrolledFor}^{-}.\neg\texttt{St})$: same as for the case (B).(i).(a).(4)
  (B).(i).(b).(4) Defn. $\phi_{\texttt{Cu}} \coloneqq (\geq_0 \texttt{enrolledFor}^{-}.\neg\texttt{Cu})$: same as for the case (B).(i).(a).(4)
  (B).(i).(b).(5) Defn. $\phi_{\texttt{Cu}} \coloneqq (=_0 \texttt{enrolledFor}^{-}.\neg\texttt{St})$: same as for the case (B).(i).(a).(3)
  (B).(i).(c).(3) Defn. $\phi_{\texttt{Cu}} \coloneqq (\leq_n \texttt{enrolledFor}^{-}.\texttt{Cu})$ for $n \geq 1$:
  same as for the case (B).(i).(a).(3)

(B).(i).(d).(1) Defn. $\phi_{\texttt{Cu}} := (\leq_n \texttt{enrolledFor}^-.\neg\texttt{St})$ for $n \geq 1$:
same as for the case (B).(i).(a).(8)

(B).(i).(e).(3) Defn. $\phi_{\texttt{Cu}} := (\triangleright_{\texttt{Cu}} \texttt{enrolledFor}^-.\texttt{Cu})$: same as for the case (B).(i).(a).(3)

(B).(i).(f).(1) Defn. $\phi_{\texttt{Cu}} := (\triangleright_{\texttt{Cu}} \texttt{enrolledFor}^-.\neg\texttt{St})$: same as for the case (B).(i).(a).(3)

(B).(ii).(a) Defn. $\phi_{\texttt{Cu}}$ with $\{\leq_0, \geq_0, =_0\}$ cardinality on path $\texttt{enrolledFor}$ and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For all $\phi_{\texttt{Cu}}$, same as for the case (B).(i).(a).(3), i.e., there will not be any $\texttt{Cu}(n) \in G$ or $\texttt{St}(n) \in G$ in any G reachable from $\tau_{\texttt{Cu}}$ node with the property path '$\texttt{enrolledFor}$'.

(B).(ii).(b) Defn. $\phi_{\texttt{Cu}}$ with $\{\leq_n\}$ such that $n \geq 1$ cardinality on path $\texttt{enrolledFor}$ and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For all $\phi_{\texttt{Cu}}$, same as for the case (B).(ii).(a)

(B).(ii).(d) Defn. $\phi_{\texttt{Cu}}$ with $\{\triangleright_{\texttt{Cu}}\}$ cardinality on path $\texttt{enrolledFor}$ and $\{\top, \texttt{St}, \texttt{Cu}, \neg\texttt{St}, \neg\texttt{Cu}\}$ typing:
For all $\phi_{\texttt{Cu}}$, same as for the case (B).(ii).(a)

- Inductive Case: In this part of the proof, we consider conjunction of the Defn. $\phi_{\texttt{Cu}}$ such that $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} := \phi^1_{\texttt{Cu}} \wedge \phi^2_{\texttt{Cu}} \rangle$, i.e., $S' = \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} := \phi^1_{\texttt{Cu}} \wedge \phi^2_{\texttt{Cu}} \rangle$.

  Let $\phi^1_{\texttt{Cu}}$ and $\phi^2_{\texttt{Cu}}$ be two arbitrary base constraints (i.e. from the base cases listed above) such that $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi^1_{\texttt{Cu}} \rangle$ and $\Sigma \models_M \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi^2_{\texttt{Cu}} \rangle$. Then, for any graph G, we have

  $$G \models S' \rightarrow G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi^1_{\texttt{Cu}} \rangle$$

  and

  $$G \models S' \rightarrow G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi^2_{\texttt{Cu}} \rangle.$$

  Thus, $G \models S \rightarrow G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi^1_{\texttt{Cu}} \wedge \phi^2_{\texttt{Cu}} \rangle$.

3. $\Sigma \models_M S'$ such that $S' = \{\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle, \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle\}$:
From the case (1) and (2) above, observe that whenever we have $G \models S$ for any G:

$$G \models \langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle \text{ and } G \models \langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle$$

for any $\langle \texttt{St}, \tau_{\texttt{St}}, \phi_{\texttt{St}} \rangle \in S'$ and $\langle \texttt{Cu}, \tau_{\texttt{Cu}}, \phi_{\texttt{Cu}} \rangle \in S'$. Thus, for every graph G and every $\Sigma \models_M S'$,

$$G \models S \rightarrow G \models S'.$$

Therefore, $\Sigma \models^*_M S$. This concludes the proof of the Lemma.

E A Complete Example of Constraint Rewriting Γ

E.1 Relational Database and Mappings $\mathcal{M}$ from Example 1

create table `course` (`C_id` varchar primary key, `Title` varchar unique);
create table `student` (`S_id` integer primary key, `Name` varchar, `Code` varchar not null foreign key references `course(C_id)`);

S_id	Name	Code
011	Ida	CS40
012		CS20

FK

C_id	Title
CS40	Logic
CS20	Database
CS50	Data Eng

$$\text{Select } \texttt{S\_id} \text{ from } \texttt{student} \longrightarrow \langle \texttt{iri}_{\texttt{Student}}(\texttt{S\_id}), \texttt{rdf:type}, \texttt{Student} \rangle.$$

$$\text{Select } \texttt{C\_id} \text{ from } \texttt{course} \longrightarrow \langle \texttt{iri}_{\texttt{Course}}(\texttt{C\_id}), \texttt{rdf:type}, \texttt{Course} \rangle.$$

$$\text{Select } \texttt{S\_id, C\_id} \text{ from } \texttt{student}, \texttt{course} \longrightarrow \langle \texttt{iri}_{\texttt{Student}}(\texttt{S\_id}), \texttt{enrolledFor}, \texttt{iri}_{\texttt{Course}}(\texttt{C\_id}) \rangle.$$
$$\text{where } \texttt{student.Code} = \texttt{course.C\_id}$$

$\mathtt{iri}_{\mathtt{Student}}$ and $\mathtt{iri}_{\mathtt{Course}}$ are injective functions that construct iris for students and courses from their respective id's.

E.2 Translation of SQL-to-Relational Algebra

Following our assumption for the source query, SQL query

Select `S_id, C_id` from `student`, `course` where `student.Code` = `course.C_id`

can be translated into:

$$\pi_{\mathtt{S\_id},\mathtt{C\_id}}\sigma_{\neg\mathtt{isNull(S\_id)}\wedge\neg\mathtt{isNull(C\_id)}}(Q_1 \bowtie_{\mathtt{Code=C\_id}} Q_2)$$

s.t. $Q_1 = \sigma_{\neg\mathtt{isNull(S\_id)}\wedge\neg\mathtt{isNull(Code)}}(\mathtt{student})$ and $Q_2 = \sigma_{\neg\mathtt{isNull(C\_id)}}(\mathtt{course})$

E.3 SQL-to-View Constraint Implication

Following Example 1 and 4, we have:

$$Q \longrightarrow \langle \mathtt{iri}_{\mathtt{Student}}(\mathtt{S\_id}), \mathtt{enrolledFor}, \mathtt{iri}_{\mathtt{Course}}(\mathtt{C\_id})\rangle,$$

where Q is a source query $\pi_{\mathtt{S\_id},\mathtt{C\_id}}\sigma_{\neg\mathtt{isNull(S\_id)}\wedge\neg\mathtt{isNull(C\_id)}}(Q_1 \bowtie_{\mathtt{Code=C\_id}} Q_2)$ such that $Q_1 = \sigma_{\neg\mathtt{isNull(S\_id)}\wedge\neg\mathtt{isNull(Code)}}(\mathtt{student})$ and $Q_2 = \sigma_{\neg\mathtt{isNull(C\_id)}}(\mathtt{course})$. Then,

a. For SP expression Q_1 :
   - $\mathsf{att}(Q_1) = \{\mathtt{S\_id},\mathtt{Code}\}$ and $\{\mathtt{UNQ(S\_id)}, \mathtt{NN(S\_id)}, \mathtt{NN(Code)}\} \subseteq \Sigma|_{Q_1}$, from Defn. 12.
   - $\Sigma_{Q_1} \Vdash \mathtt{FD}_{\mathtt{S\_id}\to\mathtt{Code}}$ from the case (c) of Lemma 1
b. For SP expression Q_2 :
   - $\mathsf{att}(Q_2) = \{\mathtt{C\_id}\}$ and $\{\mathtt{UNQ(C\_id)}, \mathtt{NN(C\_id)}\} \subseteq \Sigma|_{Q_2}$ from Defn. 12.
   - $\Sigma_{Q_2} \Vdash \mathtt{UFD}_{\mathtt{C\_id}\to\mathtt{C\_id}}$ from the case (d) of Lemma 1
c. Finally, for SPJ expression Q:
   - $\mathsf{att}(Q) = \{\mathtt{S\_id}, \mathtt{C\_id}\}$
   - Q is a valid SPJ expression since $\mathtt{FK(Code, student, C\_id, course)} \in \Sigma|_{Q_1} \cap \Sigma|_{Q_2}$.
   - $\Sigma_Q \Vdash \mathtt{FD}_{\mathtt{S\_id}\to\mathtt{C\_id}}$ from case (c) of Lemma 2, since
     i. $\Sigma_{Q_1} \Vdash \mathtt{FD}_{\mathtt{S\_id}\to\mathtt{Code}}$, and
     ii. $\Sigma_{Q_2} \Vdash \mathtt{FD}_{\mathtt{C\_id}\to\mathtt{C\_id}}$ from $\Sigma \Vdash \mathtt{UFD}_{\mathtt{C\_id}\to\mathtt{C\_id}} \to \Sigma \Vdash \mathtt{FD}_{\mathtt{C\_id}\to\mathtt{C\_id}}$ following the case (a) of Corollary 1

E.4 Result of Constraint Rewriting Γ

$\langle \mathtt{Student}, \tau_{\mathtt{Student}}, \phi_{\mathtt{Student}}\rangle$ s.t.

$\phi_{\mathtt{Student}} \coloneqq (\leq_0 \mathtt{enrolledFor}.\neg\mathtt{Course})$ `by rule 1 of` Γ `since` $\mathcal{M}$ `is simple`

$\phi_{\mathtt{Student}} \coloneqq (\geq_0 \mathtt{enrolledFor}.\mathtt{Course})$ `by rule 1 of` Γ `since` $\iota(\mathit{head}(m), \mathcal{M}) = A$

$\phi_{\mathtt{Student}} \coloneqq (=_1 \mathtt{enrolledFor}.\mathtt{Course})$ `by rule 1 of` Γ `since` $\delta_1(\mathtt{FD}_{\mathtt{S\_id}\to\mathtt{C\_id}})$

$\langle \mathtt{Course}, \tau_{\mathtt{Course}}, \phi_{\mathtt{Course}}\rangle$ s.t.

$\phi_{\mathtt{Course}} \coloneqq (\leq_0 \mathtt{enrolledFor}^{-}.\neg\mathtt{Student})$ `by rule 2 of` Γ `since` $\mathcal{M}$ `is simple`

$\phi_{\mathtt{Course}} \coloneqq (\geq_0 \mathtt{enrolledFor}^{-}.\mathtt{Student})$ `by rule 2 of` Γ `since` $\iota(\mathit{head}(m), \mathcal{M}) = A$

Translation to SHACL Syntax

```
:Student a sh:NodeShape, rdfs:Class;
  sh:property [ sh:path :enrolledFor;
      sh:maxCount 1;  sh:minCount 1;
      sh:nodeKind sh:IRI;  sh:class :Course ].

:Course a sh:NodeShape, rdfs:Class;
  sh:property [ sh:path [sh:inversePath
      :enrolledFor];
     sh:nodeKind sh:IRI;  sh:class :Student ].
```