Examples of transformation patterns

Pattern 1 Object Property Chain Shortcutting

Shortening a property path.
Abox structure:
- Input: $x $p $y . $y $r $z .
- Output: $x $q $z .

Features: contractive; single; monotonous
Tbox structure: (incomplete; also the possibility of local restrictions might be considered)
- $p rdfs:range $a . $r rdfs:domain $a .

Semantic links: {$p, $a, $r} → $q
Possible usage:
- reduction of number of atoms in relational learning (rules that could not have been discovered in the full representation due to search space limitations can be discovered in the compacted representation)
- more compact visualization

SPARQL TBox detection:
```
SELECT ?p ?r ?a ?b ?c
WHERE {
        ?p rdfs:domain ?b .
        ?p rdfs:range ?a .
        ?r rdfs:domain ?a .
        ?r rdfs:range ?c .
}
```
SPARQL TBox update:
```
INSERT DATA {
        <newProperty> a owl:ObjectProperty;
                rdfs:domain <b>;
                rdfs:range <c>. }
```
SPARQL ABox:
```
INSERT {?x <newProperty> ?z}
WHERE{
?x ?p ?y.
?y ?r ?z.
}
```

Examples

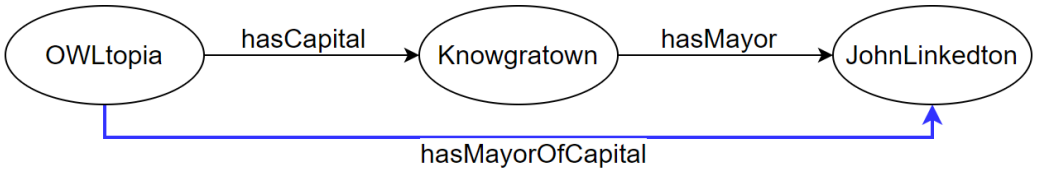
Artificial, Abox

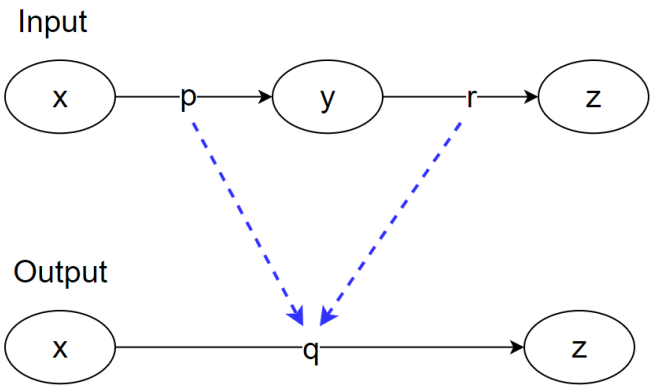
Input: :OWLtopia v:hasCapital :Knowgratown . :Knowgratown v:hasMayor :JohnLinkedton .
Output: :OWLtopia vnew:hasMayorOfCapital :JohnLinkedton .

Real

See https://people.geog.ucsb.edu/~jano/odpld.pdf

Diagrams


OWLtopia
hasCapital
Knowgratown
hasMayor
JohnLinkedton
hasMayorOfCapital


Input
x
p
y
r
z
Output
x
q
z

Pattern 2 Class by Attribute Type (CAT)

Based on "class by attribute value" (CAT1) alignment pattern in Scharffe et al. (2014)
Objects whose values for a certain property is instance of a particular class (that is just a subclass of the range of the property) become instances of a new class (subclass of their original class).

Tbox structure:

- $p rdfs:domain $a ; rdfs:range $b . $c rdfs:subClassOf $b .

Features: expansive; multiple; monotonous
Semantic links: { $p, $a, $b, $c } → $d
Possible usage:

- construction of extensive explicit taxonomy from implicit subconcepts (cf. research on focused categorization power, Svátek et al., Section 4.3 and 9.2)

Notes:

- Probably as monotonous, as the introduction of additional instantiation shouldn't usually lead to replacement of any of the previous triples
- Also consider local restrictions in the Tbox, as alternative do domain/range?
- The pattern is related to the transformation operations supported by the OReCaP tool, see Section 9.2 of the SWJ FCP paper.

SPARQL TBox detection:

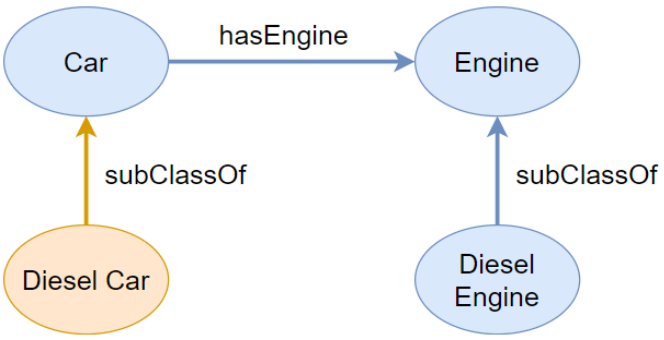
```
SELECT DISTINCT ?A ?p ?B ?C
WHERE {
  ?p rdfs:domain ?A .
  ?p rdfs:range ?B .
  ?C rdfs:subClassOf ?B
}
```

Examples

Artificial, Abox

Input: :paper123 v:hasDecision :Accept . :Accept a v:Decision .
Output: :paper123 a :AcceptedPaper .

Diagram



Pattern 3 Relationship dereification

Restoring a reified relationship as a binary property. May lead to reduction of an n-ary (n>2) relationship to a binary one.

Abox structure:

- Input: $x $p $y ; $r $z .
- Output: $y $q $z .

Features: contractive; single; monotonous

Tbox structure: (incomplete; also the possibility of local restrictions might be considered)

- $p rdfs:domain $a . $r rdfs:domain $a .

Semantic links: {$p, $a, $r} → $q

Text-based detection: the name of $a should indicate that it is a relationship (nominalized verb or gerund; other relation-indicating noun such as “...ship”, “...hood”, etc.)?#

Possible usage:

- reduction of number of atoms in relational learning (rules that could not have been discovered in the full representation due to search space limitations can be discovered in the compacted representation)
- ? more compact visualization

Inverse pattern:

- “OP reification” - probably of limited practical use for KG transformation as the intra-constructed entity is anonymous and does not link any further?

Similar patterns:

- Pattern 1 Object Property Chain Shortcutting

Notes:

- Is the TP be more relevant if the class of $y and $z is the same?
- Currently only one direction considered; an inverse one can be added as a separate transformation?
- Probably as monotonous, as the introduction of shortcut shouldn’t usually lead to replacement of the reification entity?
- We assume that the properties are other than rdf:type; that would be a different TP (Normalization?)
- Also consider local restrictions in the Tbox, as alternative do domain/range?
- Does the cardinality aspect (e.g., whether some of the properties are functional) affect the pattern relevance?

Examples

Artificial, Abox

Input: :citizenship123 v:isCitizenshipOfPerson :JohnDoe ; v:isCitizenshipOfCountry :OWLtopia .
Output: :JohnDoe vnew:citizenOf :OWLtopia .

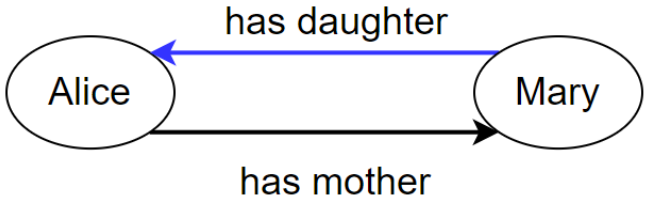
Input: :tp123 a :TradePartnership ; v:partnerInvolved :OWLtopia , RDFia .
Output: :OWLtopia vnew:hasTradePartner :RDFia .

Pattern 4 Property inversion

One property is inverse of another property.

Examples

Artificial



Pattern 5 OP to DP conversion

Abox structure:

- Input: $x $p $y . $y a $z .
- Output: $x $p1 $y1.

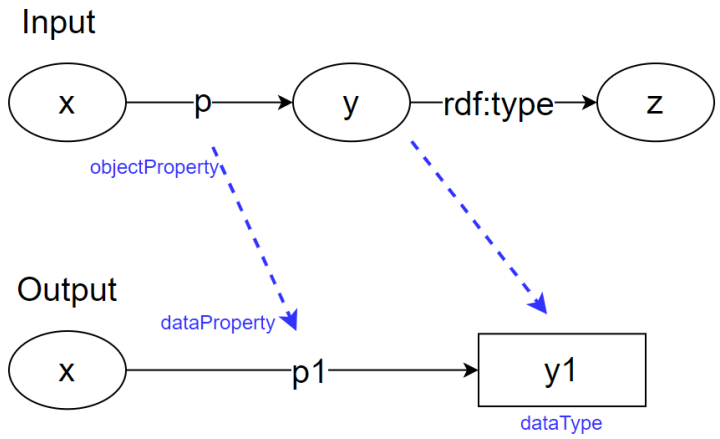
Links: $p → $p1 $y → $y1

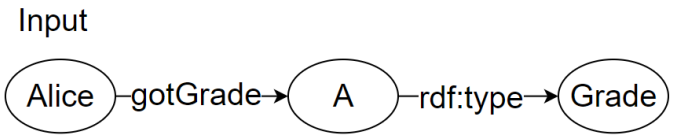
Tbox hints:

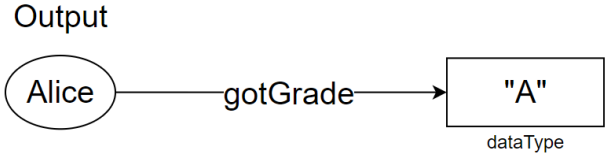
Input: $p rdf:type owl:ObjectProperty .
Output: $p1 rdf:type owl:DataProperty .

Examples

Artificial





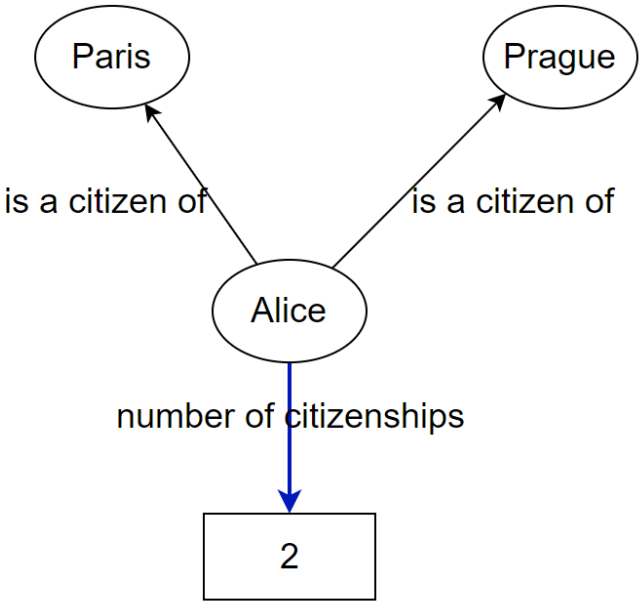


Pattern 6 Link Counting Property

Creating a new dataproperty that counts the same objected property, which lead from the same object.

Examples

Artificial



Pattern 7 Role Relationship Shortcutting

Role modeling has several variants ranging from fully blown to quite degenerated. The chosen transformation is thus just one member of such a TP family. Most variants of role pattern can be found in D3.2, p. 15:

- Role holder classes (a)
- Generic “role holding” property (b)
- Roles implicitly present through relationships which establish them (c)
- Ternary “participation” role, linking the role holder, role and the subject of participation (d)
    - That one should be refined, though. This way it is not clearly prescribed whether the role holder is the person or the course. There should also preferably be some generic properties rather than those specific to the given classes (Person, Course)
- Yet another advanced option is to have a hierarchy of role classes instead of a flat set of roles. (new option (e))
- Even more complex is e.g. the role modeling by Sunagawa et al. (distinguishing role concepts, natural concepts and role holders)

Cf. the JWS paper from 2016 on role model transformation, Section 4:

- The source pattern can be either
    - “class-centric role modelling” - corresponds to a combination of (a) and (b)
    - “property-centric role modelling” - corresponds to (b) alone
- The AgentRole pattern (defining the Role class and hasRole + isRoleOf properties) is also a part of the source, but not yet integrated
- The target pattern for both the class-centric and property-centric input is either cca model (e) or (b).

The currently chosen approach is the transformation from a modified version of (d), called (d’), to the combination of (a)+(c).

Role transformation (d’) to (a+c)

Abox structure:

- Input: $x $rhp $y ; $rp $z ; $rop $w . $y a $c . $w a $d . $z a $r . $x a $P .
- Output: $y a $e ; $p $w . $e rdfs:subClassOf $c .

(Note: rhp = role holder property; rp = role property; rop = role object property; r = role class; P = participation class)

Tbox structure:

-

Features: contractive; single; monotonous

Semantic links: $z → $e { $z, $d } → $p

Examples

Artificial, Abox

Input: :participation123 v:roleHolder :person45 ; v:role :reviewer ; v:inRelationTo :paper67 . :person45 a v:Person . :paper67 a v:Paper . :reviewer a v:Role . :participation123 a v:Participation .
Output: :person45 a :Reviewer ; :reviewedPaper :paper67 .
:Reviewer rdfs:subClassOf v:Person .

Output, TBox:
:reviewedPaper rdfs:domain :Reviewer .
:reviewedPaper rdfs:range :Paper .
:Reviewer rdfs:subClassOf :Person .

Features

Contractive - size-neutral - expansive
Whether a replacement transformation decreases or increases the number of Abox triples. (Obviously, a merely additive transformation would always be “expansive”.)
Single - single multi-input - multiple
Whether the transformation is assumed on a single instance of detected OP, or on multiple (unspecified number of) instances as one transformation, or on multiple (unspecified number of) instances but each as a separate transformation
Monotonous - non-monotonous
Whether the TP is likely to be used in a monotonous manner, i.e., just adding new triples, or also non-monotonously, i.e., replacing the original structures