Direct Mapping Relational Databases to RDFS/OWL and RDF

Juan F. Sequeda

Daniel P. Miranker

University of Texas at Austin

Direct Mapping

- Produce automatically a (putative) ontology from the relational schema
- Produce RDF as instance of the (putative) ontology
- Rules in datalog. Well defined semantics
- Can translate datalog to SQL or RIF

Relational Database Predicates

- Rel(r) = r is a relation; e.g. Rel(PERSON) holds, Rel(ID) does not
- Attr(x,r) = x is an attribute in relation r; e.g. Attr(ID,PERSON) holds
- NN(x,r) = x is an attribute (or a set of attributes) in relation r with NOT NULL constraint(s); e.g. NN(NAME,PERSON) holds
- Unq(x,r) = x is an attribute (or a set of attributes) in relation r with UNIQUE constraint; e.g. Unq({NAME},DEPT) holds
- Chk(x,r) = x is an attribute in relation r with enumerated list (CHECK IN) constraint; e.g. Chk(SESSION,SEMESTER) holds
- PK(x,r) = x is the (single or composite) primary key of relation r; e.g. PK({ONO,RNO},STUDY) holds; Also: PK(x,r) ⊔ Unq(x,r) ⊔ NN(x,r)
- FK(x,r,y,s) = x is a (single or composite) foreign key in relation r and references y in relation s; e.g. FK({ID},STUDENT,{ID},PERSON) holds
- NonFK(x,r) = x is an attribute in relation r that does not participate in any foreign key; e.g. NonFK(NAME,DEPT) holds

Ontology (RDFS/OWL) Predicates

- Class(m) = m is a class
- ObjP(p,d,r) = p is an object property with domain d and range r
- DTP(p,d,r) = p is an data type property with domain d and range r
- Inv(p,q) = when p and q are object properties, p is an inverse of q
- FP(p) = p is a functional property
- IFP(p) = p is an inverse functional property
- Crd(p,m,v) = the (max and min) cardinality of property p for class m is v
- MinC(p,m,v) = the min cardinality of property p for class m is v
- MaxC(p,m,v) = the max cardinality of property p for class m is v
- Subclass(m,n) = m is a subclass of class n

Ontology Functions

- fkey(x,r,s) = takes a set of attributes x, relations r and s, and returns the foreign key defined on attributes x in r referencing s
- type(x) = maps an attribute x to its suitable OWL recommended data type
- list(x) = maps an attribute x to a list of allowed values; applicable only to attributes with a CHECK IN constraint, i.e. Chk(x) is true

Direct Mapping RDB to RDFS/OWL

Rule 1

- BinRel(r,s,t) <- Rel(r), FK(q,r,_,t), FK(p,r,_,s), p!=q, ¬E(r), FK(z,r,_,u), fkey(z,r,u) is member of {fkey(p,r,s),fkey(q,r,t)}
- E(r) <- Attr(y,r), NonFK(y,r)

Rule 2

- Class(r) <- Rel(r), -BinRel(r,_,_)

Rule 3

- ObjP(r,s,t) <- BinRel(r,s,t), Rel(s), Rel(t), -BinRel(s,_,_), -BinRel(t,_,_)

Direct Mapping RDB to RDFS/OWL

Rule 4

- NonBinFK(x,s,y,t) <- FK(x,s,y,t), Rel(s), Rel(t), -BinRel (s,_,_), -BinRel(t,_,_)
- ObjP(x,s,t), FP(x), MinC(x',t,0) <- NonBinFK(x,s,y,t), -NN (x), -Unq(x)
- ObP(x,s,t), FP(x), Crd(x,s,1), MinC(x',t,0) <- NonBinFK (x,s,y,t), NN(x), -Unq(x)
- ObjP(x,s,t), FP(x), FP(x') <- NonBinFK(x,s,y,t), -NN(x), Unq(x)
- ObjP(x,s,t), FP(x), Crd(x,s,1), FP(x') <- NonBinFK(x,s,t), NN(x), Unq(x), -PK(x,s)

Direct Mapping RDB to RDFS/OWL

Rule 5

- DTP(x,r,type(x)), FP(x) <- NonFK(x,r)
- DTP(x,r,type(x)), FP(x), Crd(x,r,1) <- NonFK(x,r), NN(x,r)
- DTP(x,r,type(x) INTERSECTION list(x)), FP(x) <- NonFK(x,r), Chk(x,r)

Rule 6

- Subclass(r,s) <- Rel(r), Rel(s), PK(x,r), FK(x,r,_,s)

Direct Mapping RDB to RDF

- If we don’t care of the ontology
  - Student(id, name, age)
  - Triple(“student”+id,”name”,name) <- student(id, name,_)
  - Triple(“student”+id,”age”,age) <- student(id, _, age)
- However age is a Datatype property with Domain “student” and Range “string”
- There is always an ontology (it’s implicit)

Direct Mapping RDB to RDF

- Given the ontology we can create instances of the ontology → RDF
- In addition to the RDB to RDFS/OWL rules, add new rules to produce the RDF
- Triple(s,p,o) ← …

Conclusion

- Datalog as the mapping language has well-defined semantics
- Adding some extra rules you can get RDF
- Adding some extra rules you can represent the knowledge of an expert