A Preliminary Report on Answering Complex Queries related to Drug Discovery using Answer Set Programming

Olivier Bodenreider[1], Zeynep H. Çoban[2], Mahir C. Doğanay[3]
Esra Erdem[4], and Hilal Koşucu[5]

[1] National Library of Medicine, National Institutes of Health, USA
[2] Department of Biostatistics, Harvard School of Public Health, USA
[3] Dept. of Mathematics and Computing Science, University of Groningen, The Netherlands
[4] Faculty of Engineering and Natural Sciences, Sabancı University, Turkey
[5] Department of Computer Science, University of Toronto, Canada

Abstract. We introduce a new method for integrating relevant parts of knowledge extracted from biomedical ontologies and answering complex queries related to drug safety and discovery, using Semantic Web technologies and answer set programming. The applicability of this method is illustrated in detail on some parts of existing biomedical ontologies. Its effectiveness is demonstrated by computing an answer to a real-world biomedical query that requires the integration of NCBI Entrez Gene and the Gene Ontology.

1 Introduction

Improvements in Web technologies have brought about various forms of data, and thus WWW has been a huge and easy-to-reach source of knowledge. Particularly recent advances in health and life sciences (e.g., human genome project) have led to generation of a large amount of data. In order to facilitate access to its desired parts, such a big mass of data has been stored in structured forms (like databases or ontologies). For instance, some data/information about drugs is being stored in ontologies, like DRUGBANK and PHARMGKB, available on WWW; and the genes targeted by the drug Epinephrine can be found by searching such a drug ontology using the keyword "Epinephrine."

On the other hand, storing heterogeneous data independent from each other and at different locations has made it difficult to automate high-level reasoning about the stored data. For instance, it is possible to find an answer to the query "What are the genes targeted both by Epinephrine and by Isoproterenol?" only after several steps: considering that a drug (and also a gene) might have been stored in different ontologies under different names, first for each drug a list of genes targeted by that drug could be found, and next these two lists of genes are compared to identify the common ones, by comparing these two lists of genes. Such complex queries, which require appropriate integration of knowledge stored in different places and in various forms, can be answered by current Web technologies most of the time only by some direction/reasoning of humans. This slows down vital research, like drug discovery, that requires comparative data analysis and high-level reasoning and decision making.

Motivated by these challenges, this paper studies the problem of integrating various data sources to be able to perform high-level reasoning tasks, including answering

complex queries using both Semantic Web technologies and Answer Set Programming (ASP) [1–4]. The idea is to build a rule layer using ASP over ontologies described with some Semantic Web technologies. The rule layer not only provides rules to link parts of the ontologies but also provides some background knowledge to be able to perform various reasoning tasks, such as query answering.

That most of the information about biomedical ontologies are actually defaults and that most biomedical ontologies contain incomplete knowledge motivated us to use a nonmonotonic formalism to build a rule layer over ontologies. That experts might want to express preferences as well as constraints while querying the knowledge stored in ontologies to be able to discover new knowledge, and that ASP provides an expressive language to express them and efficient solvers, like DLVHEX[6] [5] built over DLV,[7] to reason about them motivated us to use ASP as such a nonmonotonic formalism.

2 Three Ontologies

To experiment with our ASP approach to integrating biomedical ontologies and reasoning about them, and to illustrate its applicability, we have developed three ontologies, namely a gene ontology, a disease ontology, and a drug ontology. We have built these ontologies from existing knowledge from various data sources available on the Web. These ontologies are written in RDF(S). To develop our disease ontology, first we selected a set of diseases. The names (and their synonyms) of each disease are taken from PHARMGKB database.[8] Information about the symptoms of these diseases is obtained from the Medical Symptoms and Signs of Disease web page.[9] Information about the genes related to each disease are also extracted from PHARMGKB. Each disease is classified in some category relative to the information available at the Genes and Diseases web page.[10] Some components of the disease ontology is shown in Table 1. We have prepared the other two ontologies in a similar way, using PHARMGKB, UNIPROT,[11] GENE ONTOLOGY (GO),[12] GENENETWORK database,[13] DRUGBANK,[14] and the Medical Symptoms and Signs of Disease web page.

3 Integrating Knowledge Extracted from Different Ontologies

DLVHEX provides constructs to import external theories that may be in different formats. For instance, consider as an external theory our drug ontology described in RDF. All triples from this theory can be exported using the external predicate `&rdf`:

[6] `http://con.fusion.at/dlvhex/`
[7] `http://www.dbai.tuwien.ac.at/proj/dlv/`
[8] `http://www.pharmgkb.org/.`
[9] `http://www.medicinenet.com/symptoms_and_signs/article.htm.`
[10] `http://www.ncbi.nlm.nih.gov/disease/.`
[11] `http://www.ebi.uniprot.org/index.shtml.`
[12] `http://www.geneontology.org.`
[13] `http://humgen.med.uu.nl/~lude/genenetwork/.`
[14] `http://redpoll.pharmacy.ualberta.ca/drugbank/.`

Table 1. The disease "Asthma" described in our disease ontology

has_name	Asthma
has_synonyms	Bronchia, Bronchial Asthma
has_symptoms	Coughing, Wheezing, Chest tightness, Shortness of breath, Faster breathing
related_genes	ABCC1, ADA, ADAM33, ADCY9, AD0RA1, ADRB1, ADRB2, ALOX5, COMT, CRHR1
treatedBy_drugs	Isoproterenol, Flunisolide, Salbutamol

```
triple_drug(X,Y,Z) :- &rdf["URI for Drug Ontology"](X,Y,Z).
```

Not all triples may be relevant to the query asked by the user. For instance, if one asks for the names of drugs listed in the ontology, then only the triples that describe the names of drugs are sufficient to answer this query. The names of drugs, out of all properties about drugs described in `drug.rdf`, can be extracted by the following rule:

```
drug_name(A) :- triple_drug(_,"drugproperties:name",A).
```

If the query were about gene-gene interactions, then we could extract the relevant part of the gene ontology by the rules

```
gene_gene(G1,G2) :- triple_gene(X,"geneproperties:name",G1),
   triple_gene(X,"geneproperties:related_genes",B),
   triple_gene(B,Z,Y), Z!="rdf:type",
   triple_gene(Y,"geneproperties:name",G2).
```

Once necessary parts of ontologies are extracted from ontologies, one can define further concepts to integrate these knowledge. For instance, once we extract the gene-gene interactions, we can obtain all chains of gene-gene interactions for a gene targeted by a drug, by defining the transitive closure of `gene_gene`:

```
tc_gene_gene(X,Y) :- gene_gene(X,Y).
tc_gene_gene(X,Y) :- gene_gene(X,Z), tc_gene_gene(Z,Y).
```

Now let us relate this information to a gene G targeted by a drug D by finding every gene $G1$ that is related to G by means of a chain of interactions:

```
drugTargetedGene_interacts_gene(D,G,G1) :-
   drug_targets(D,G), tc_gene_gene(G,G1).
```

4 Answering Complex Queries using DLVHEX

With the help of Devrim Gözüaçık (a medical doctor and a molecular biologist), we have identified a set of meaningful queries about drugs, genes, diseases, towards drug safety and discovery. We present here only three of them:

Q6 What are the sideeffects that are shared by all the drugs that treat a disease D?

Q12 Is there a drug that has no toxicity information?

Q14 Does a drug R alleviate at least 1 symptom of a disease D and have at most 2 symptoms of D as side effects?

We integrate relevant parts of ontologies, and formulate these queries as follows.

Q6 What are the sideeffects that are shared by all the drugs that treat a disease D?

For the disease `Asthma`, this query can be formulated as follows:

```
answer :- sideeffect(S), common_sideeffect("Asthma",S).
:- not answer.
```

Here `common_sideeffect` is defined as follows:

```
-common_sideeffect(D,S) :- not drug_sideeffect(R,S),
   drug_disease(R,D), sideeffect(S).

common_sideeffect(D,S) :- not -common_sideeffect(D,S),
   sideeffect(S), disease_name(D).
```

Here is a part of the answer DLVHEX finds to the query above:

```
flushing dizziness headache
```

Q12 Is there a drug that has no toxicity information?

To answer this query, we define a new concept of "unknown" toxicity:

```
unknown_toxicity_drug(X) :- drug_synonym(R,X),
   not drug_istoxic(R), not -drug_istoxic(R).
```

where `drug_istoxic(R)` describes that the drug `R` is toxic, and `-drug_istoxic(R)` describes that the drug `R` is not toxic:

```
drug_istoxic(R) :- triple_drug(X,"drugproperties:name",R),
   triple_drug(X,"drugproperties:is_toxic","yes").
drug_istoxic(R) :- drug_synonym(R,R1), drug_istoxic(R1).

-drug_istoxic(R) :- triple_drug(X,"drugproperties:name",R),
   triple_drug(X,"drugproperties:is_toxic","no").
-drug_istoxic(R) :- drug_synonym(R,R1), -drug_istoxic(R1).
```

For the query

```
:- not unknown_toxicity_drug("Isoproterenol").
```

DLVHEX returns an answer set; therefore the answer to the query above is positive.

Q14 Does a drug R alleviate at least 1 symptom of a disease D and have at most 2 symptoms of D as side effects?

To answer this query we define a new concept:

```
a_drug_disease_relation(R,D) :-
   disease_name(D), drug_name(R),
   1 <= #count{S:drug_symptom(R,S),disease_symptom(D,S)},
   #count{S:drug_sideeffect(R,S),disease_symptom(D,S)}<=2.
```

For the query

```
:- not a_drug_disease_relation("Isoproterenol",
                               "Substance Related Disorders").
```

DLVHEX returns no answer set; therefore the answer to the query above is negative.

5 From *Glycosyltransferase* to *Congenital Muscular Dystrophy*

To investigate the effectiveness of our approach to answering real-world queries, we have considered a slight modification of the complex query studied in [6]:

> Find all the genes annotated with the molecular function *glycosyltransferase* or any of its descendants and associated with any form of *congenital muscular dystrophy*.

and tried to reproduce the same results. In the query of [6] the GO ID for glycosyltransferase is given. The query above requires integration of NCBI Entrez Gene (EG) and the Gene Ontology (GO).

To find an answer to this query, we have used the RDF version of GO that is released on February 6, 2008; it contains 416700 RDF triples. We have used an RDF version of EG that contains 673180 RDF triples.

The computation of an answer consists of two parts: extracting relevant knowledge from each ontology and integrating them. We have extracted from GO the molecular function *glycosyltransferase* and its descendants by the rules

```
mf_isa(Y) :- triple_go(Y,"go:name",YN),
   &strstr[YN,"glycosyltransferase"].
mf_isa(Y) :- triple_go(Y,"go:synonym",YN),
   &strstr[YN,"glycosyltransferase"].

mf_isa(X) :- triple_go(X,"go:is_a",Y), mf_isa(Y).
mf_isa(X) :- triple_go(X,"go:synonym",XN),
   triple_go(Z,"go:name",XN), triple_go(Z,"go:is_a",Y), mf_isa(Y).
```

The first two rules extract the molecular functions whose names or synonyms contain the string "glycosyltransferase". The last two rules extract the descendants of these molecular functions, considering their synonyms.

Similarly, we have extracted from EG the diseases with any form of *congenital muscular dystrophy*, by the rules

```
gene_disease(Y,D) :- triple_eg(Y,"eg:has_OMIM_record",Z),
   triple_eg(Z,"eg:has_textual_description",D),
   &strstr[D,"congenital"], &strstr[D,"muscular"],
   &strstr[D,"dystrophy"].
```

After that we have integrated the extracted knowledge by the rules

```
gene_mf_disease(Y,XI,D) :- gene_disease(Y,D),
   triple_eg(Y,"eg:has_GeneOntology_annotation",X),
   mf_isa(XI), triple_eg(X,"eg:has_GO_ID",XI).
```

and computed the following answer (the same as in [6]) to the query:

```
gene_mf_disease("http://www.ncbi.nlm.nih.gov/dtd/NCBI_Entrezgene.
  dtd/9215", "http://www.geneontology.org/go#GO:0008375",
  "Muscular dystrophy, congenital, type 1D")
```

DLVHEX extracts relevant knowledge from the ontologies, integrates them, and computes the answer above in 9 minutes, on a machine with Intel Centrino 1.8GHz CPU and 1 GB of RAM running on Windows XP.

6 Conclusion

We have studied integrating relevant parts of knowledge extracted from biomedical ontologies, and answering complex queries related to drug safety and discovery, using Semantic Web technologies and Answer Set Programming (ASP). We have illustrated the applicability of this method on some ontologies extracted from existing biomedical ontologies, and its effectiveness by computing an answer to a real-world biomedical query that requires the integration of NCBI Entrez Gene and the Gene Ontology. We have also compared our approach with the existing Semantic Web technologies that support representing and answering queries. We have observed about these technologies that, due to lack of support for rules or for some concepts (e.g., transitive closure, negation as failure, cardinality constraints), some queries can not be represented concisely and some queries can not be represented at all. In this sense, the ASP-approach provides a more expressive formalism to represent rules, concepts, constraints, and queries.

Acknowledgments

Devrim Gözüaçık helped us identify some of the complex queries. Thomas Krennwallner and Roman Schindlauer helped us with installing/using DLVHEX. RACER Systems provided us a free, educational version of RACERPRO,[15] to be used in connection with DLVHEX. Anonymous reviewers provided useful comments on an earlier draft. This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM).

References

1. Lifschitz, V.: Action languages, answer sets and planning. In: The Logic Programming Paradigm: a 25-Year Perspective. Springer (1999)
2. Marek, V., Truszczyński, M.: Stable models and an alternative logic programming paradigm. In: The Logic Programming Paradigm: a 25-Year Perspective. Springer (1999)
3. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. Annals of Mathematics and Artificial Intelligence **25** (1999)
4. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
5. Eiter, T., G.Ianni, R.Schindlauer, H.Tompits: Effective integration of declarative rules with external evaluations for Semantic-Web reasoning. In: Proc. of ESWC. (2006)
6. Sahoo, S.S., Zeng, K., Bodenreider, O., Sheth, A.: From "glycosyltransferase" to "congenital muscular dystrophy": Integrating knowledge from NCBI Entrez Gene and the Gene Ontology. In: Proc. of Medinfo. (2007)

[15] `http://www.racer-systems.com/` .