SPARQL pro chemoinformatiky

Vedoucí týmu: Jakub Galgonek <jakub.galgonek@marge.uochb.cas.cz>
Členové týmu: Tomáš Hurt, Vendula Michlíková, Petr Onderka, Jan Schwarz
Termín dokončení: únor 2015

V poslední době se začínají sémantické technologie uplatňovat také v oblasti databází malých molekul. Příkladem může být projekt Open PHACTS[1] nebo RDF databáze Evropského bioinformatického institutu[2]. Rovněž probíhají práce na vytvoření společných ontologií[3], které mohou být sdíleny různými databázemi. Použití těchto technologií umožní lepší interoperabilitu mezi různými zdroji dat. Rovněž umožní flexibilnější vyhledávání v takto přístupných datech. V našem projektu se chceme zaměřit především na uživatelské hledisko takovéhoto systému a zvýšit jeho využitelnost v chemoinformatice.

Klíčovou funkcí každé databáze malých molekul je schopnost hledat sloučeniny podle jejich chemické struktury. To umožňují tzv. *chemické cartridge* (např. OrChem[4] nebo GGA Software Services Bingo[5]), které se instalují jako doplněk ke klasickým relačním databázím. Pomocí uložených procedur je pak možné provádět hledání sloučenin podle strukturní podobnosti. Bohužel jazyk SPARQL[6], který se používá k zápisu sémantických dotazů, volání uložených procedur neumožňuje.

Prvním velkým cílem projektu je tudíž rozšířit jazyk SPARQL tak, aby umožňoval také podobnostní vyhledávání v datech, tedy obecně aby podporoval volání uložených procedur. Toto rozšíření by mělo být co nejvíce transparentní - nemělo by jinak ovlivnit syntaxi jazyka a pokud možno jen minimálně jeho sémantiku. Ze syntaktického hlediska se nabízí možnost zapsat volání procedury pomocí anonymního uzlu například takto:

```
?RESULT ex:procedure_name [ ex:param1 "value1", ex:param1  "value2" ].
```

Z hlediska sémantiky je však třeba takovýto speciální vzor vyhodnotit zcela jinak.

Je požadováno, aby veškeré vyhodnocování probíhalo pokud možno uvnitř databáze. Cílem projektu je tedy vyvinout překladač, který vstupní dotaz v rozšířeném jazyce SPARQL přeloží do jazyka SQL (který interně umožňuje použít některé konstrukce jazyka SPARQL), jenž bude vyhodnocen na cílové databázi.

Součástí projektu je také požadavek na napsání rozsáhlé testovací sady, která bude (polo)automaticky testovat korektnost a úplnost implementace jazyka SPARQL 1.1 ve zvoleném uložišti. U zjištěných nepodporovaných konstrukcí musí překladač zajistit jejich překlad do podporovaných konstrukcí.

---

[1] http://www.openphacts.org
[2] http://www.ebi.ac.uk/rdf/services
[3] http://purl.obolibrary.org/obo/
[4] http://orchem.sourceforge.net/
[5] http://ggasoftware.com/opensource/bingo
[6] http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

Dalším podstatným problémem při využití jazyka SPARQL je „nedostatečná“ sémantická kontrola dotazů. Jazyk SPARQL umožňuje pokládat dotazy, které nemají vzhledem k použité ontologii dat dobrý smysl. Jazyk SPARQL totiž dotazy vůči ontologii dat vůbec nekontroluje. Pokud uživatel například použije chybný identifikátor (například kvůli překlepu), výsledkem bude validní dotaz. Uživatel v takových případech dostane pouze prázdnou odpověď bez žádného varování, což mu opravu jeho dotazu velmi znesnadňuje. Druhým velkým cílem projektu je tedy navrhnout kontrolu dotazu, která určí, zda daný dotaz může mít při dané ontologii řešení. Například následující dotaz má řešení pouze tehdy, pokud obor hodnot predikátu ex:pred1 a definiční obor predikátu ex:pred2 nejsou disjunktní:

```
SELECT ?X ?Y ?Z WHERE
{
    ?X ex: pred1 ?Y.
    ?Y ex:pred2 ?Z.
}
```

Zvolený příklad ukazuje pouze jednu z možných základních chyb. Plná implementace bude muset podporovat bloky, regulární cesty (property chain) a ostatní konstrukce.

Kontrola by měla odhalovat i částečně „nesmyslné“ dotazy. Například:

```
SELECT ?X WHERE
{
    {?X ex:pred1 ?V1.}
    UNION
    {?X ex:pred2 ?V2.}
    ?X ex:pred3 ?V3.
}
```

Pokud nejsou definiční obory predikátů `ex:pred1 a ex:pred3` disjunktní, pak má dotaz smysl (tj. výsledek nemusí být nutně prázdný). Pokud však navíc definiční obory predikátů `ex:pred2 a ex:pred3` disjunktní jsou, pak použití predikátu `ex:pred2` nedává v tomto případě dobrý smysl, což ale může kontrola odhalit, až když kontroluje řádek s predikátem `ex:pred3.`

Projekt bude vyvíjen v jazyce Java, pro uložení dat se počítá hlavně s uložištěm Virtuoso[7] (případně jiným podle aktuálních potřeb).

[7] http://virtuoso.openlinksw.com/