SBML2SHACL

Edoardo De Matteis
1746561

10 agosto 2020

Indice

1 Introduzione

SBML è un formato basato su XML per definire conoscenza medica e biochimica, più che di linguaggio si parla di lingua franca dal momento che si pone come tale, risolvendo il problema causato dall'eterogeneità di standard tra software in ambito biomedico.

SHACL invece è uno standard del W3C per la verifica di grafi rispetto a dei vincoli definiti, questi grafi sono rappresentati in qualsiasi formato RDF, ad esempio Turtle; Il Resource Description Network (RDF) è necessario per la codifica e la manipolazione di metadati e consente la modellazione di informazioni come risorse web, dal momento che si fa uso di logiche descrittive tali informazioni sono codificate sotto forma di triple <subject> <predicate> <object> equivalente in logica del primo ordine ad una formula ove subject e object sono due termini ground e predicate un predicato binario.

Dato un modello SHACL il linguaggio di query che permette di interrogarlo è SPARQL; RDF, SHACL e SPARQL sono standard del World Wide Web Consortium (W3C).

L'obiettivo di questo progetto, sotto la guida del Professor Tronci, è quello di convertire automaticamente codice SBML in SHACL. A tal fine il problema è stato diviso in tre fasi:

1. Selezionare un sottoinsieme di costrutti in SBML e modellarlo in SHACL.

2. Scrivere un parser che traduca una specifica SBML in SHACL.

3. Scrivere un parser che traduca una specifica SHACL in SBML.

2 Modellazione

Nella prima fase si è scelto un sottoinsieme di SBML 3.2 e la totalità dei costrutti aggiuntivi di extended SBML, nella tabella 1 vengono descritti i principali costrutti, sono state volontariamente omesse entità quali `listOf*` il cui significato è intuitivo e avrebbero solo reso la tabella meno leggibile, in ogni caso è possibile consultare ulteriormente il diagramma `diagram.png` e il file `shapes.ttl`, in qest'ultimo è scritta la definizione del modello con i vincoli che dovranno essere rispettati dai file di output del parser. Per entrambi i file si è scelto di utilizzare il formato Turtle.

Extended SBML introduce la definizione di una gerarchia, è possibile definire ad esempio l'annidamento "cellula-nucleo-dna", aggiunge inoltre la possibilità di definire modelli e poterli importare ed esportare, favorendo il riuso di codice.

Entità	**Descrizione**
	SBML 3
SBase	Classe astratta che viene definita tipo, è vero dal momento che ogni classe definisce un tipo e ogni nodo sarà sottoclasse di SBase ma poiché non esistono attributi di tipo SBase a questa classe si riserva un trattamento differente rispetto ai tipi modellati quali ID, SId, ecc.. che sono omessi nel diagramma ma sono presenti nel file Turtle.
Sbml	Ogni file SBML ha un'etichetta con tag sbml, grazie a questo nodo è possibile costruire grafi SHACL composti da multipli modelli SBML in input dato che saranno sempre radicati in questo nodo.
Model	Rappresenta il modello, se ne possono avere più di uno per come detto sopra ed è in relazione con numerosi "List", dal momento che queste sono abbastanza esplicative da qui in poi ci concentriamo solo sulla classe di interesse.

Unit	Definisce un'unità di misura definita dall'utente, sono definite nel linguaggio delle unità base (i.e. le unità del SI e altre scelte dagli sviluppatori di SBML) e queste sono usate per definire nuovi tipi. Le unità base - kind - in `shapes.ttl` sono trattate come normalissime unità di misura, in SBML non è concesso avere come attributo kind un'unità di misura che non lo sia indi per cui il nostro modello non fa alcuna differenza.
Compartment	Rappresenta un compartimento in cui raccogliere qualche oggetto, spesso Species, come ad esempio una cellula.
Species	Rappresentano specie quali ad esempio differenti tipi di proteine.
Parameter	In SBML si possono definire parametri sia locali che non, il nostro sottoinsieme di SBML non presenta ancora località, una conseguenza è che Model avrà come attributo ListOfParameters con moltiplicità [0,n] piuttosto che [0,1] perché quando incontra un parametro lo considera globale anche se appartiene ad una ListOfParameters differente.
	Extended SBML
ExternalModelDefinition	In extended SBML è possibile importare modelli differenti (ad esempio scaricati da internet) e introdurli nel proprio.
ModelDefinition	Dal momento che viene introdotta la gerarchia per farlo ci si serve di due costrutti che collaborano, questo è il primo e tramite ModelDefinition si dà una definizione del modello che poi vorrà essere usato e al quale si vorrà fare riferimento.
Submodel	L'istanza di una ModelDefinition è rappresentata da Submodel e si ha effettivamente un modello dentro ad un altro modello. Durante la fase di test non ho usato esempi da Biomodels per extended SBML perché la gerarchia non veniva rappresentata tramite Submodel ma facendo uso di un attributo "outer" in Compartment, con il Professor Tronci si è ritenuto fosse una definizione di terze parti e si è preferito attenersi allo standard W3C per la scrittura del parser.

Port	Un'istanza di Port permette di definire allo sviluppatore come ci si deve interfacciare con un Model, di norma è preferibile seguire le indicazioni dello sviluppatore.
Deletion	Non è detto che i modelli importati abbiano solo ed esclusivamente componenti desiderabili e con Deletion è possibile ignorare quelle ridondanti.
Replacement	Come sopra ma si considera una sostituzione. A causa di Replacement sono presenti più parser, il primo `parser.py` esplora il file XML come una lista e associare un Replacement ad un componente risulta estremamente macchinoso se non impossibile, in `extended_parser.py` questo problema non si presenta perché il file XML viene esplorato come un albero.
SBaseRef	Port, Deletion, Replacement e Submodel utilizzano dei riferimenti, essendo sottoclassi di SBaseRef - che similmente a SBase è astratta - li ereditano.

Tabella 1: Modellazione SHACL

Come già detto si assume a priori che i file SBML in input siano corretti, la verifica è eseguibile online.

3 Test

Nella cartella dei test sono presenti dei file XML, alcuni sono scaricati dal sito Biomodels e usano solo SBML 3 mentre altri (nella sottocartella `custom`) sono esempi forniti direttamente dal W3C. Quindi questi ultimi potranno essere usati solo per il file extended_parser.py il quale rispetto a parser.py aggiunge SBML gerarchico, il codice è più conciso, può essere esteso più facilmente ed è più veloce come si può vedere dai risultati dell'esecuzione del file `test.sh`.

Tabella 2: Performance

File	**System (s)**	**User (s)**	**Total**	**CPU**
parser.py	52.90	2445.38	42:54.54	97 %
extended_parser.py	42.48	833.54	15:02.19	97 %

4 Commenti e critiche

Durante lo sviluppo del progetto sotto direttive del Professor Enrico Tronci si è tenuto conto dei seguenti commenti e critiche.

Tabella 4: Cronologia

Data	**Commento**
22/07/20	Videochiamata con specifica del problema da parte del Professor Tronci.
27/07/20	Prima stesura di una modellazione dei costrutti in SBML ma avendo io dimenticato di modellare SBML gerarchico mi è stato fatto notare e ho risolto, il materiale consultato è stato reso disponibile dal Professore stesso.
28/07/20	Corretto il punto precedente non avevo implementato Deletions e Replacements, il professore ha inoltre consigliato una strategia di testing.
31/07/20	Dopo aver corretto le mie mancanze e eseguito dei test con risultati positivi è seguita una videochiamata con il Professor Tronci in cui mi è stato indicato di modificare il parser in maniera tale da poter ricevere in input più modelli SBML creando quindi un grafo più complesso, e di capire se il risultato di una query SPARQL rappresenti lo stesso tipo di conoscenza di un file XML/SBML, così da poter sfruttare questa corrispondenza per eseguire dei controtest. Questa corrispondenza esiste ed è usata spesso.

5 Fonti

- The systems biology markup language.
- Sparql query language for rdf, 2008.
- Sparql query results xml format (second edition), 2013.
- Rdf 1.1 concepts and abstract syntax, 2014.
- Shapes constraint language (shacl), 2017.
- Michael Hucka, Frank T. Bergmann, Claudine Chaouiya, Andreas Dräger, Stefan Hoops, Sarah M. Keating, Matthias König, Nicolas Le Novère,Chris J. Myers, Brett G. Olivier, Sven Sahle, James C. Schaff, RahumanSheriff, Lucian P. Smith, Dagmar Waltemath, Darren J. Wilkinson, andFengkai Zhang. The systems biology markup language (sbml): Languagespecification for level 3 version 2 core, 2019.
- Lucian P. Smith, Stefan Hoops, Martin Ginkel, Ion Moraru, Michael Hucka,Andrew Finney, Chris J. Myers, and Wolfram Liebermeister. Hierarchicalmodel composition, 2013.