PDDL2SPARQL

User Guide documentation

Elena Jaramillo, Guangyi Chen

July 8, 2021

1 Introduction

PDDL2SPARQL is a small program which converts the text in **PDDL** forms to the corresponding **SPARQL** queries. Currently, this program aims at the relation syntax in the **PDDL** expressions, hence it could not converts the whole **PDDL** files but the relation fragments. More specifically:

1.1 Relationship Tuples

PDDL2SPARQL is capable of recognising relationship tuples in **PDDL** syntax, which can be generally divided into two scenarios:

(1). with predicate, e.g. `on ?ob ?underobject`
(2). without predicate, e.g. `holding ?object`

PDDL2SPARQL is capable of differentiating the two scenarios and convert them into the corresponding **SPARQL** relationship forms.

1.2 Logical Symbols

PDDL2SPARQL is also capable of detecting logical symbols (i.e. "`and`", "`or`", "`not`") and converting them into corresponding syntaxes in **SPARQL** ("&&", "||", "`NOT EXISTS`").

1.3 Prefixes

When relation tuples in the given **PDDL** syntax contain prefix (e.g. an url), PDDL2SPARQL is capable of listing them uniquely at the beginning of the output **SPARQL** syntax.

2 How To Use

Firstly add the `PDDL2SPARQL.jar` to your working Java project as the external JARs. Then the program can be invoked by the following codes:

```
import java.io.IOException;
import PDDL_SPARQL.*;

public class Test_PDDL2SPARQL {

   public static void main(String[] args) throws IOException {

      String InputPDDL = "Your-PDDL-Expressions";

      PDDL_SPARQL.translator(InputPDDL);

   }

}
```

The console will print the converted expressions in **SPARQL**.

3 Examples

Some examples are given as follows:

3.1 Example 1

The input is:

```
(and
(http://127.0.0.1/ontology/ontosem.owl#Accepted
    ?http://127.0.0.1/services/1.1/bookpersoncreditcardaccount_price_service.owls#_CREDITCARDACCOUNT)
(http://127.0.0.1/ontology/core-plus-office.owl#Authorized
    ?http://127.0.0.1/services/1.1/bookpersoncreditcardaccount_price_service.owls#_PERSON)
(http://127.0.0.1/ontology/Mid-level-ontology.owl#accountHolder
    ?http://127.0.0.1/services/1.1/bookpersoncreditcardaccount_price_service.owls#_PERSON
    ?http://127.0.0.1/services/1.1/bookpersoncreditcardaccount_price_service.owls#_CREDITCARDACCOUNT))
```

The output is:

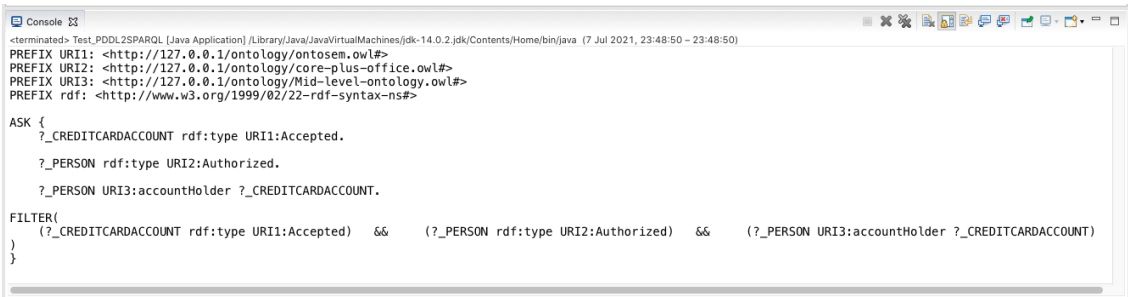

Figure 1: If PDDL2SPARQL runs successfully, then console should have outputs resembling the above (Example 1).

3.2 Example 2

The input is:

```
(http://127.0.0.1/ontology/ShoppingCart.owl#ShoppingCartRequestItems
    ?http://127.0.0.1/services/1.1/bookpersoncreditaccount__Beaservice.owls#_BOOK)
```

The output is:

3.3 Example 3

The input is:

```
(not (http://127.0.0.1/ontology/books.owl#Novel
    ?http://127.0.0.1/services/1.1/book_authorprice_service.owls#_BOOK))
```

The output is:

Figure 2: If PDDL2SPARQL runs successfully, then console should have outputs resembling the above (Example 2).

Figure 3: If PDDL2SPARQL runs successfully, then console should have outputs resembling the above (Example 3).

3.4 Example 4

The input is:

```
(and (not (http://127.0.0.1/ontology/SUMO.owl#equal
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_ADDRESS1
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_ADDRESS2))
(http://127.0.0.1/ontology/protont.owl#locatedIn
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_CITY1
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_STATE1)
(http://127.0.0.1/ontology/protont.owl#locatedIn
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_CITY2
    ?http://127.0.0.1/services/1.1/addressDistanceCalculator.owls#_ST
```

The output is:

Figure 4: If PDDL2SPARQL runs successfully, then console should have outputs resembling the above (Example 4).

3.5 Example 5

The input is:

```
(or (not (http://127.0.0.1/ontology/SUMO.owl#equal
    ?http://127.0.0.1/services/1.1/calculateDistanceInMiles.owls#_LATITUDE1
    ?http://127.0.0.1/services/1.1/calculateDistanceInMiles.owls#_LATITUDE2))
(not (http://127.0.0.1/ontology/SUMO.owl#equal
    ?http://127.0.0.1/services/1.1/calculateDistanceInMiles.owls#_LONGITUDE1
    ?http://127.0.0.1/services/1.1/calculateDistanceInMiles.owls#_LONGITUDE2)))
```

The output is:

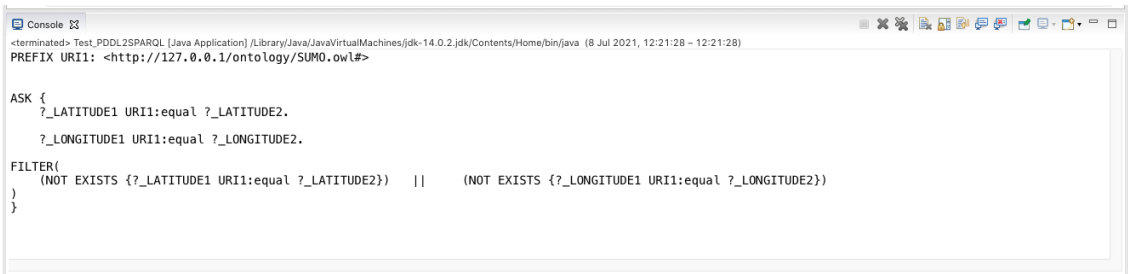

Figure 5: If PDDL2SPARQL runs successfully, then console should have outputs resembling the above (Example 5).

4 Remarks

For the time being, PDDL2SPARQL does not accept "conditional effects" ("when"), "universal/existential quantifications" ("forall"/"exists"), "derived predicates", function definitions. Which is, currently only the "and", "or", "not" syntax (as illustrated in the Examples) are supported. This point will be addressed in the future development.