



Knowledge-Based Systems

Laboratory activity 2021

Car Configurator Ontology

Students: Dragotescu Bogdan, Rusu Andrei

Assoc. Prof.dr. eng. Adrian Groza
Adrian.Groza@cs.utcluj.ro

Contents

1	Competency questions and use cases	3
1.1	Competency questions	3
1.2	Use Cases	3
2	TBox	4
3	ABox	6
4	Roles	10
5	Rules	11
6	Ontology Design Patterns	12
6.1	Set	12
6.2	ViewInheritance	12
6.3	Parameter	13
6.4	PartOf	13
7	Racer Java API	14
7.1	Obtaining the JAR	14
7.2	Establishing a connection	14
7.3	Infering compatibility	15
7.4	Results	15
8	FuzzyDL	16
8.1	Introduction	16
8.2	Steps	16
8.3	Experimental Results	16
8.3.1	Code	16
9	Queries	18
9.1	Racer Queries	18
9.2	NRQL Queries	19
9.3	Query Results	19
10	Appendix	20
.1	Racer axioms	20
.2	Ontology evaluation	23
.3	FuzzyDL axioms	24
.4	Java Code	24

Chapter 1

Competency questions and use cases

1.1 Competency questions

- Is item X compatible with model Y?
- Is item X compatible only with model Y?
- What models are compatible with item X?
- Is item X_1 compatible with item X_2 ?
- Is item X standard or optional?
- Is model Y an electric vehicle?
- What interior trim options are there for model Y?
- Does item X_1 include item X_2 ?
- Is item X an interior or exterior item?
- Is model Y available with large wheels?
- Is a glass roof available for model Y?

1.2 Use Cases

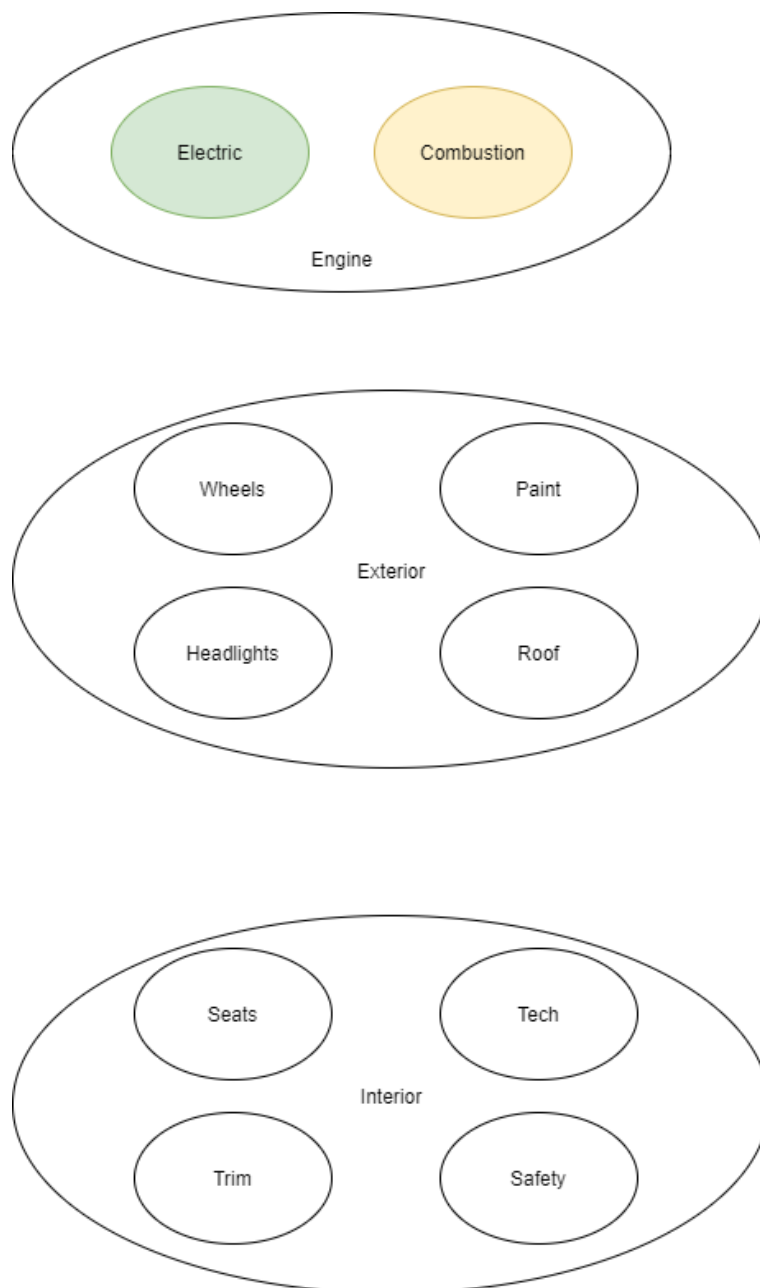
The main use cases are:

- Finding out which configuration items are compatible with the desired vehicle
- Checking the compatibility of the items between themselves

Chapter 2

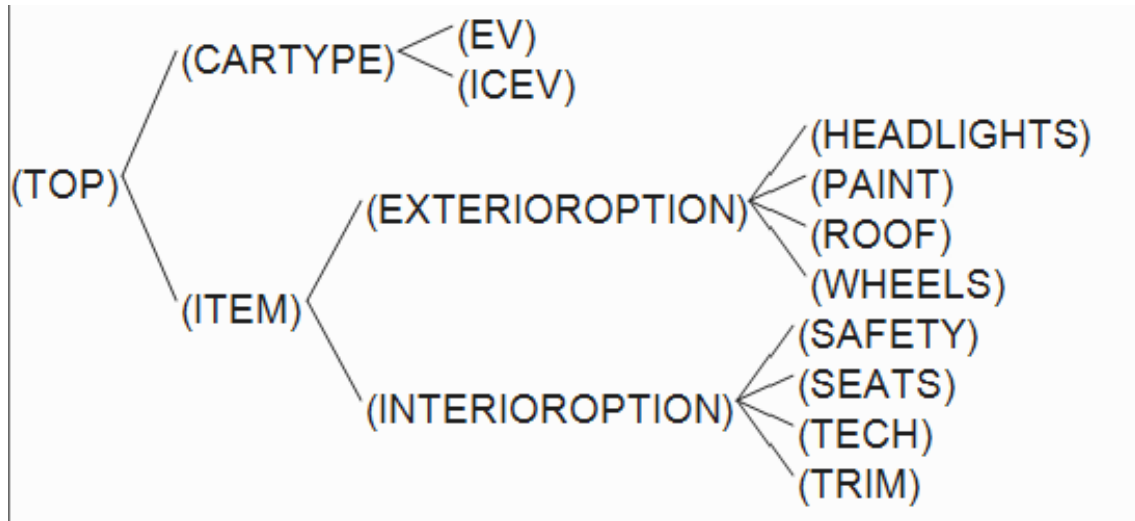
TBox

The Car Configuration ontology is used for checking the compatibility of configuration items for a desired vehicle. Therefore, the options have to be divided into categories, and rules established for selecting items from them, so as to fit together. Below is the preliminary classification of car configuration items:



We have split the vehicles into two categories: electric and with combustion engines. The optional items are then split into Exterior and Interior items, where you would find your usual expected items. Using this classification, we will be able to defines rules that specify that you may only choose one item from Wheels, but more then one item from Tech, for example.

Below is an extract from the Taxonomy page of Racer:



Chapter 3

ABox

We will briefly show the structure and knowledge present in the knowledge base.

The categories from Chapter 2 each have their own instances, which can be seen below. We shall start with the interior items, which can be seen below:

```
; --- WHEELS ---
(instance Wheels19 Wheels)
(instance Wheels20 Wheels)
(instance Wheels21 Wheels)

; --- PAINT ---
(instance MetallicPaint Paint)
(instance PearlescentPaint Paint)

; --- HEADLIGHTS ---
(instance LEDLights Headlights)
(instance MatrixLEDLights Headlights)

; --- ROOF ---
(instance CarbonRoof Roof)
(instance GlassRoof Roof)
(instance PanoramicRoof Roof)
```

Some items have prices attached since we will use this information in the upcoming chapter about FuzzyDL.

The same goes for the exterior options, which have been defined as seen below:

```

; --- ROOF ---
(instance CarbonRoof Roof)
(instance GlassRoof Roof)
(instance PanoramicRoof Roof)

; --- SEATS ---
(instance HeatedSeats Seats)
(instance HeatedSeats (equal isOptional 1))
(instance HeatedSeats (equal hasPrice 500))

(instance RegularSeats Seats)
(instance RegularSeats (equal isOptional 0))
(instance RegularSeats (equal hasPrice 0))

(instance SportSeats Seats)

; --- TECH ---
(instance ElectricBoot Tech)
(instance Camera Tech)
(instance ACC Tech)
(instance FullTechPack Tech)
(instance StarterTechPack Tech)

; --- TRIM ---
(instance MetalTrim Trim)
(instance WoodTrim Trim)
(instance RegularLeather Trim)
(instance NappaLeather Trim)

; --- SAFETY ---
(instance BlindSpotMonitor Safety)
(instance BlindSpotMonitor (equal isOptional 1))
(instance BlindSpotMonitor (equal hasPrice 300))

(instance FullSafetySystem Safety)
(instance FullSafetySystem (equal isOptional 1))
(instance FullSafetySystem (equal hasPrice 900))

(instance ParkingSensors Safety)
(instance ParkingAssistant Safety)

```

There are two vehicle models, *Taycan* and *Macan*:

```

(DESCRIBE-INDIVIDUAL1 TAYCAN DEFAULT)
(TAYCAN
 :ASSERTIONS
 ((TAYCAN EV))
 :ROLE-FILLERS
 ((ISCOMPATIBLECARITEM
  (WOODTRIM
   LEDLIGHTS
   GLASSROOF
   PARKINGASSISTANT
   BLINDSPOTMONITOR
   PARKINGSSENSORS
   STARTERTECHPACK
   CAMERA
   ACC
   ELECTRICBOOT
   HEATEDSEATS
   REGULARSEATS
   PEARLESCENTPAINT
   WHEELS21
   WHEELS20
   PANORAMICROOF
   FULLTECHPACK
   MATRIXLEDLIGHTS
   NAPPALETHER
   FULLSAFETYSYSTEM
   SPORTSEATS) ) )
 :TOLD-ATTRIBUTE-FILLERS
 NIL
 :TOLD-DATATYPE-FILLERS
 NIL
 :ANNOTATION-DATATYPE-PROPERTY-FILLERS
 NIL
 :ANNOTATION-PROPERTY-FILLERS
 NIL
 :DIRECT-TYPES
 :TO-BE-COMPUTED)

```

```

(DESCRIBE-INDIVIDUAL1 MACAN DEFAULT)
(MACAN
 :ASSERTIONS
 ((MACAN ICEV))
 :ROLE-FILLERS
 ((ISCOMPATIBLECARITEM
  (REGULARSEATS
   PARKINGSSENSORS
   METALTRIM
   ELECTRICBOOT
   METALLICPAINT
   WHEELS20
   WHEELS19
   CARBONROOF
   STARTERTECHPACK
   LEDLIGHTS
   REGULARLEATHER
   PARKINGASSISTANT
   HEATEDSEATS) ) )
 :TOLD-ATTRIBUTE-FILLERS
 NIL
 :TOLD-DATATYPE-FILLERS
 NIL
 :ANNOTATION-DATATYPE-PROPERTY-FILLERS
 NIL
 :ANNOTATION-PROPERTY-FILLERS
 NIL
 :DIRECT-TYPES
 :TO-BE-COMPUTED)

```

The compatibilities between cars and configuration items have been defined below. We only defined model-to-item compatibility with the topmost item and let Racer infer item-item compatibility with lower-level items. We also used the Racer Java API for extra computational work regarding setting item-to-item compatibilities.


```

; --- PACKS & INCLUSIONS ---

; Seats packs
(related RegularSeats HeatedSeats isIncluded)
(related HeatedSeats SportSeats isIncluded)

; Light packs
(related LEDLights MatrixLEDLights isIncluded)

; Roof packs
(related GlassRoof PanoramicRoof isIncluded)

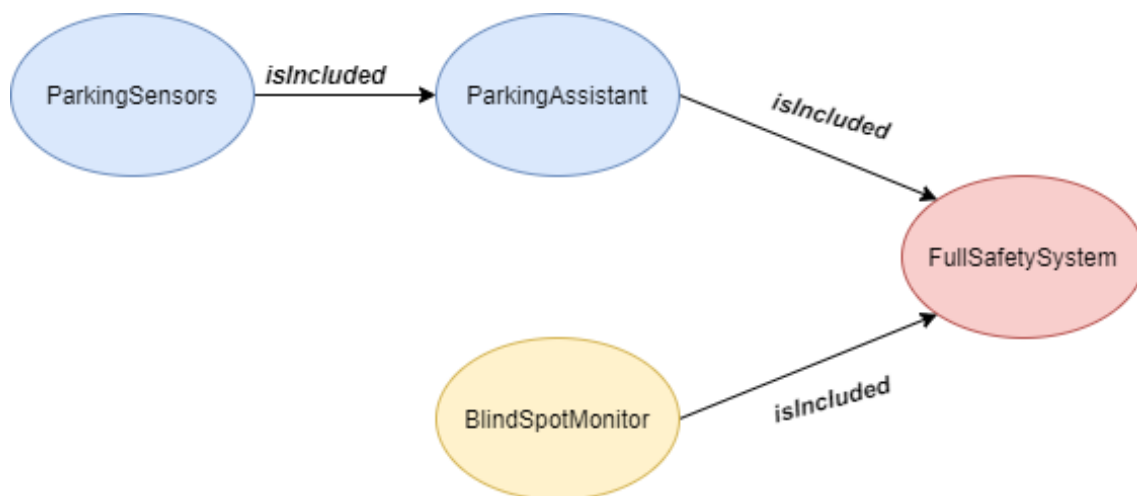
; Tech packs
(related ElectricBoot StarterTechPack isIncluded)
(related StarterTechPack FullTechPack isIncluded)
(related Camera FullTechPack isIncluded)
(related ACC FullTechPack isIncluded)

; Trim packs
(related MetalTrim RegularLeather isIncluded)
(related WoodTrim NappaLeather isIncluded)

; Safety Pack
(related ParkingSensors ParkingAssistant isIncluded)
(related ParkingAssistant FullSafetySystem isIncluded)
(related BlindSpotMonitor FullSafetySystem isIncluded)

```

As can be seen from the above screenshot, almost each category of interior and exterior items has been grouped into equipment packs, using the *isIncluded* role. Below is an example of how the *Safety* pack is organized:



Due to the later-defined rules we have created, when a vehicle is set as compatible to an item, it is automatically compatible with all the items that are included in it, similar to picking a root in a tree and considering all its children as compatible, as in the diagram shown above.

Chapter 4

Roles

The roles we have defined are the following:

- `isCompatibleCarItem`
- `isCompatibleItemItem`
- `isIncluded`

The role *isCompatibleCarItem* defines which equipment items are compatible with a specified model, while the role *isCompatibleItemItem* defines which items are compatible between each other. For example, two different sets of wheels are incompatible, since only one type of wheel can be selected in a car configuration.

The role *isIncluded* specifies which items are included in other items, for example, the blind spot monitoring system is included in the complete safety system.

Additionally, we have defined two additional attributes for each equipment item: *price* and *optional*, which show the price of an item and whether it is given as standard or it is an option. Standard equipment costs nothing.

Below is the formal declaration of the roles from above:

```
(define-primitive-role isCompatibleCarItem :domain CarType :range Item)
(define-primitive-role isCompatibleItemItem :domain Item :range Item :symmetric t)
(define-primitive-role isIncluded :domain Item :range Item :asymmetric t)
```

Chapter 5

Rules

We have defined a rule that propagates the compatibility of a items to a vehicle model, when an item includes another item:

```
(define-rule (?car ?item2 isCompatibleCarItem)
  (and (?car CarType) (?item1 Item) (?item2 Item) (not (same-as ?item1 ?item2))
    (?car ?item1 isCompatibleCarItem) (?item2 ?item1 isIncluded)))
```

Meaning that vehicle `?car` and item `?item2` are deemed compatible if there is another item `?item1` with which `?car` is compatible and which also includes `?item2`.

For example, if a vehicle is compatible with the `FullSafetySystem` item, which includes the `BlindSpotMonitor` item, it then means that the vehicle is also compatible with the `BlindSpotMonitor`, however not at the same time.

In order to infer which items are compatible with other items, we decided to use the Racer Java API, since defining a set of rules that would produce this result in racer syntax proved too difficult.

We defined two items that are compatible as being from different sets. For example, a glass roof is compatible with 19' wheels, since they do not interfere with each other. However, 19' wheels are not compatible with 20' wheels, since a vehicle can only have one type of item from the `Wheels` category. Code is included in later chapters.

It is also possible to count the instances of a set and, for example, say that only one item of some categories can be selected, but more from another category, like `Safety` for example. However, we removed the need for this extra verification by combining the items in item packs, that are like upgrades, with each pack containing the features of the previous pack, plus additional ones.

For example, the item `HeatedSeats` includes the item `RegularSeats`, which are both in the `Seats` category, meaning that only one can be selected as a configuration. The same goes for the `FullSafetySystem` item, which includes the `BlindSpotMonitor` item and are both part of the `Safety` category. This way we can ensure that only one item from each category is selected.

Additionally, we have defined an extra rule that is applied between items with inclusions. It implements the transitivity property of the inclusion of items, which could not be defined in the rule definition of the `isIncluded` role, due to the fact that inclusion is assymetrical:

```
(define-rule (?item1 ?item3 isIncluded)
  (and (?item1 Item) (?item2 Item) (?item3 Item) (not (same-as ?item1 ?item2)) (not
    (same-as ?item2 ?item3)) (not (same-as ?item1 ?item3))
    (?item1 ?item2 isIncluded) (?item2 ?item3 isIncluded)))
```

Chapter 6

Ontology Design Patterns

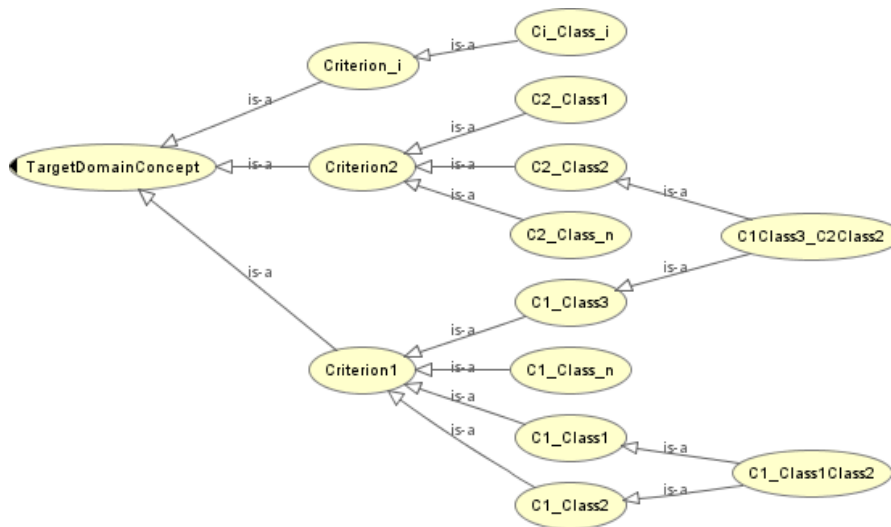
6.1 Set

This design pattern is used to model sets of elements, meaning those collections of elements that cannot have a duplicate.

In the case of our project, all the concepts defined in Chapter 3 are concepts, since no category is allowed to have duplicates. This includes the categories *Wheels*, *Paint*, *Headlights*, *Roof* which are all *Exterior* items, as well as *Seats*, *Tech*, *Trim*, *Safety*, which are *Interior* items. The *Engine* concept, with its two subcategories also follows this pattern, since there is only one vehicle with a specific name.

6.2 ViewInheritance

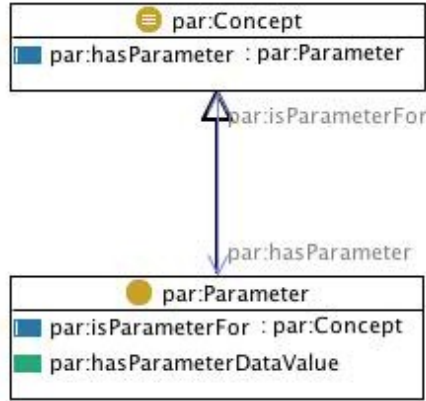
ViewInheritance is similar to the inheritance that we are familiar with from Object-Oriented Programming languages. There are ontology domain concepts that are difficult to represent due to the complexities in their definition and the presence of multiple alternative criteria to classify their abstractions, similar to how in OOP languages there are abstract classes that define a concept, but have no instances or implementations.



In our ontology, all the item categories (*Interior* and *Exterior* items) follow this pattern. There are no individuals pertaining to the *Interior* item concept, since it represents an abstract concept. In order to "instantiate" an individual, one would need to choose between the pre-defined categories that represent an actual type of *Interior* item, such as *Seats*, for example.

6.3 Parameter

The Parameter ODP is used to model various parameters used for the concepts in the ontology. In Racer ontologies, parameters are represented as *domain attributes*, which are values that pertain to a certain instance of a concept, and only that instance



We defined two domain attributes, *hasPrice* and *isOptional*, which will be useful later when comparing the cost of items or their availability. They are both integer types and the *optional* attribute has either a 0 or a 1 as value.

```

(define-concrete-domain-attribute hasPrice :type integer)
(define-concrete-domain-attribute isOptional :type integer)
  
```

6.4 PartOf

The PartOf ODP is used to represent entities and they component parts.



In the context of our project, this design pattern is followed in the equipment packs that can be selected for configuration.

For example, the *FullSafetySystem* item includes the *BlindSpotMonitor* item and the *ParkingAssist* items. In other terms, the larger item is made up of smaller pieces that are part of it.

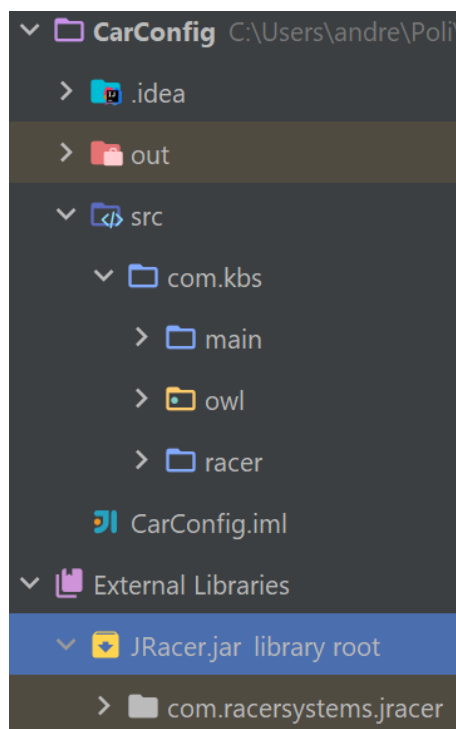
Chapter 7

Racer Java API

We have used the *Racer Java API* to perform tasks that were too difficult to perform in plain Racer syntax. Our main objective was to infer the compatibility of items with other items, based on the existing relations (compatibility, inclusion) with vehicles and other items.

7.1 Obtaining the JAR

We downloaded the .JAR file from the Moodle page, and added it to a project in IntelliJ IDEA. The project structure can be seen below



7.2 Establishing a connection

In order to establish a connection to the Racer server, it has to first be running in the background (otherwise an exception will be thrown).

Once the Racer server starts, we can connect to it from the JRacer library using the following syntax:

```
String ip = "127.0.0.1";
int port = 8088;
RacerClient racer = new RacerClient(ip, port);
try {
    racer.openConnection();
    System.out.println(racer.sendRaw( command: "(racer-read-file " + input + ")"));
}
```

Once the connection has been established, we can use this Java program and JRacer to interact with the Racer files and the server, almost totally removing the need for the RacerPorter program. We can read Racer files, perform queries and add infer knowledge from the ontology.

7.3 Inferring compatibility

The main task we wanted to achieve using the Java API was to determine which items were compatible with other items, a task which proved too difficult to perform in Racer, without a major compromise to the rest of the ontology.

Therefore, we took a simple approach. We gathered all the individuals, except the vehicles, and compared the categories they are in using Racer's *most-specific-instantiators* function. This allowed us to dynamically add relations into the Racer ontology and infer the compatibilities between the items. The code is shown below:

```
// obtain all individuals
String answer = racer.sendRaw( command: "(concept-instances *top*)");
String[] result = answer.replaceAll( regex: "[()]", replacement: "").split( regex: " ");

// if their categories are disjoint, they are compatible
for (String item1 : result) {
    for (String item2 : result) {
        if (!item1.equals(item2) &&
            !item1.equals("MACAN") &&
            !item1.equals("TAYCAN") &&
            !item2.equals("MACAN") &&
            !item2.equals("TAYCAN")) {
            boolean disjoint = racer.returnBoolean(racer.sendRaw( command: "(evaluate (concept-disjoint-p (car (car (most-specific-instantiators '"
                + item1 + " (current-abox)))) (car (car (most-specific-instantiators '"
                + item2 + " (current-abox)))) (current-ibox))))");
            if (disjoint) {
                racer.sendRaw( command: "(related " + item1 + " " + item2 + " ISCOMPATIBLEITEMITEM)");
            }
        }
    }
}
```

7.4 Results

As can be seen from the text below, the java program correctly added the correct item compatibilities. Result truncated due to large size.

$(RELATED-INDIVIDUALS\ ISCOMPATIBLEITEMITEM) \rightarrow ((BLINDSPOTMONITORWHEEL$

Chapter 8

FuzzyDL

8.1 Introduction

We created the text file "FuzzyCar.txt" based on our ontology. We defined some concepts related to price ranges. We made two instances P1 and P2 (Part1, Part2 respectively) using the two price ranges define previously for the parts. Using interogations we checked how expensive can those parts be considered as.

8.2 Steps

- *We created the file "FuzzyCar.txt" which contains the concepts, intances and interogations in Fuzzy DL. Its code has been put into the Code section.*
- *We downloaded the <http://www.umbertostraccia.it/cs/software/fuzzyDL/fuzzyDL.html>*
- *We accessed the folder in which the .jar and "FuzzyCar.txt" reside*
- *We opened the cmd terminal by typing cmd into the file path area (Windows) and ran the following command "java -jar FuzzyDL.jar FuzzyCar.txt"*

8.3 Experimental Results

```
E:\An 4\Sem II\KBS\Proj\FuzzyDLWindows\FuzzyDL>java -jar FuzzyDL.jar FuzzyCar.txt
Is P1 instance of ExpensivePart ? >= 0.0
Is P2 instance of ExpensivePart ? >= 0.416667
```

8.3.1 Code

```
(define-modifier very linear-modifier(0.8))

(define-fuzzy-concept Part1PriceRange crisp(0,10000,80,1500))
(define-fuzzy-concept Part2PriceRange crisp(0,10000,6000,10000))

(define-fuzzy-concept Expensive right-shoulder(0,10000,4000,10000))

(define-concept ExpensivePart (and Part (some Price (very Expensive))))
```


*(instance P1 (**and** Part (**some** Price Part1PriceRange)) 1)*
*(instance P2 (**and** Part (**some** Price Part2PriceRange)) 1)*

(min-instance? P1 ExpensivePart)
(min-instance? P2 ExpensivePart)

Chapter 9

Queries

9.1 Racer Queries

```
(abox-consistent?) ; --> T
(tbox-coherent?) ; --> T
(tbox-cyclic?) ; --> NIL

; large answer
(related-individuals isCompatibleCarItem)
(related-individuals isCompatibleItemItem)
(related-individuals isIncluded)
(individual-fillers Macan isCompatibleCarItem)

; manually defined
(concept-disjoint? Wheels Tech) ; --> T
(individuals-related? BlindSpotMonitor FullAutopilot isIncluded)
(evaluate (> (retrieve-individual-told-attribute-value 'HeatedSeats 'hasPrice (current-abox))
(retrieve-individual-told-attribute-value 'RegularSeats 'hasPrice (current-abox))))

; deduced
(individuals-related? Taycan BlindSpotMonitor isCompatibleCarItem)
(individuals-related? Taycan RegularSeats isCompatibleCarItem)
(individuals-related? BlindSpotMonitor inch19 isCompatibleItemItem)

(realize-abox)
(retrieve-individual-told-attribute-value 'HeatedSeats 'hasPrice (current-abox))
```

9.2 NRQL Queries

```
; NRQL
(get-nrql-version)
(enable-nrql-warnings)
(defquery is-interior (?x) (or (?x Seats) (?x Safety) (?x Tech) (?x Trim)))
(defquery is-included (?x ?y) (?x ?y is-included))

; all interior items
(retrieve (?x) (?x is-interior))

; items included in other items
(retrieve (?x ?y) (?x ?y is-included))

; interior trim for Taycan
(defquery trim-options (?car ?trim) (and (?car CarType) (?trim Trim) (?car ?trim is-compatible-car-item)))
(retrieve (?trim) (Taycan ?trim trim-options))

; models compatible with an item
(defquery compat-cars (?car ?item) (and (?car CarType) (?item Item) (?car ?item is-compatible-car-item)))
(retrieve (?car) (?car RegularSeats compat-cars))

(all-queries)
```

9.3 Query Results

```
Racer Message (STDOUT):(ABOX-CONSISTENT?) --> T
(TBOX-COHERENT?) --> T
(TBOX-CYCLIC?) --> NIL
(RELATED-INDIVIDUALS ISCOMPATIBLECARITEM) --> ((MACAN REGULARSEATS) (TAYCAN BLINDSPOTMONITOR) (TAYCAN METAL) (TAYCAN FULLAUTOPILOT))
(RELATED-INDIVIDUALS ISCOMPATIBLEITEM) --> ((BLINDSPOTMONITOR INCH19) (BLINDSPOTMONITOR INCH20) (BLINDSPOTMONITOR ELECTRICBOOT) (BLINDSPOTMONITOR WOOD) (BLINDSPOTMONITOR METAL) (BLINDSPOTMONITOR HEATEDSEATS) (BLINDSPOTMONITOR ISINCLUDED) --> ((BLINDSPOTMONITOR FULLAUTOPILOT) (REGULARSEATS HEATEDSEATS))
(INDIVIDUAL-FILLERS MACAN ISCOMPATIBLECARITEM) --> (REGULARSEATS)
(CONCEPT-DISJOINT? WHEELS TECH) --> T
(INDIVIDUALS-RELATED? BLINDSPOTMONITOR FULLAUTOPILOT ISINCLUDED) --> T
(EVALUATE (> (RETRIEVE-INDIVIDUAL-TOLD-ATTRIBUTE-VALUE (QUOTE HEATEDSEATS) (QUOTE HASPRICE) (CURRENT-ABOX)) (RETRIEVE-INDIVIDUAL-TOLD-ATTRIBUTE-VALUE (QUOTE REGULARSEATS) (QUOTE HASPRICE) (CURRENT-ABOX)))) --> T
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(INDIVIDUALS-RELATED? TAYCAN BLINDSPOTMONITOR ISCOMPATIBLECARITEM) --> T
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(INDIVIDUALS-RELATED? TAYCAN REGULARSEATS ISCOMPATIBLECARITEM) --> NIL
(INDIVIDUALS-RELATED? BLINDSPOTMONITOR INCH19 ISCOMPATIBLEITEM) --> T
Error: (CURRENT-ABOX) fell through ETYPESCASE expression.
Wanted one of (SYMBOL RACER::ABOX).
```

```
(GET-NRQL-VERSION) --> \"2.0.0\"
(ENABLE-NRQL-WARNINGS) --> :OKAY-WARNINGS-ENABLED
(DEFQUERY IS-INTERIOR (?X) (OR (?X SEATS) (?X SAFETY) (?X TECH) (?X TRIM))) --> IS-INTERIOR
(DEFQUERY IS-INCLUDED (?X ?Y) (?X ?Y IS-INCLUDED)) --> IS-INCLUDED
(RETRIEVE (?X) (?X IS-INTERIOR)) --> (((?X HEATEDSEATS)) ((?X REGULARSEATS)) ((?X BLINDSPOTMONITOR)) ((?X FULLAUTOPILOT)) ((?X ELECTRICBOOT)) ((?X METAL)) ((?X WOOD)))
(RETRIEVE (?X ?Y) (?X ?Y IS-INCLUDED)) --> (((?X REGULARSEATS) (?Y HEATEDSEATS)) ((?X BLINDSPOTMONITOR) (?Y FULLAUTOPILOT)))
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(DEFQUERY TRIM-OPTIONS (?CAR ?TRIM) (AND (?CAR CARTYPE) (?TRIM TRIM) (?CAR ?TRIM ISCOMPATIBLECARITEM))) --> TRIM-OPTIONS
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(RETRIEVE (?TRIM) (TAYCAN ?TRIM TRIM-OPTIONS)) --> (((?TRIM METAL)))
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(DEFQUERY COMPAT-CARS (?CAR ?ITEM) (AND (?CAR CARTYPE) (?ITEM ITEM) (?CAR ?ITEM ISCOMPATIBLECARITEM))) --> COMPAT-CARS
Role ISCOMPATIBLECARITEM exists in TBox DEFAULT. Assuming you are referring to the role ISCOMPATIBLECARITEM (nrql Warning).
(RETRIEVE (?CAR) (?CAR REGULARSEATS COMPAT-CARS)) --> (((?CAR MACAN)))
(ALL-QUERIES) --> (:QUERY-16 :QUERY-14 :QUERY-12 :QUERY-11)
```

Chapter 10

Appendix

.1 Racer axioms

```
1 (full-reset)
2
3 ; --- CONCEPTS ---
4 ; Electric and Internal Combustion are different
5 (implies
6   (or EV ICEV)
7   CarType)
8 (disjoint EV ICEV)
9
10 ; These are Exterior options
11 (implies
12   (or Wheels Paint Headlights Roof)
13   ExteriorOption)
14 (disjoint Wheels Paint Headlights Roof)
15
16 ; These are Interior options
17 (implies
18   (or Seats Tech Trim Safety)
19   InteriorOption)
20 (disjoint Seats Tech Trim Safety)
21
22 ; all are items
23 (implies
24   (or ExteriorOption InteriorOption)
25   Item)
26
27 ; all top categories are different
28 (disjoint CarType ExteriorOption InteriorOption)
29
30
31 ; --- ATTRIBUTES ---
32 ; Attributes of a car Model
33 (define-concrete-domain-attribute name :type string)
34
35 ; Attributes of each part
36 (define-concrete-domain-attribute hasPrice :type integer)
37 (define-concrete-domain-attribute isOptional :type integer)
38
39 ; --- ROLES ---
40 (define-primitive-role isCompatibleCarItem :domain CarType :range Item)
41 (define-primitive-role isCompatibleItemItem :domain Item :range Item :
   symmetric t)
```

```

42 (define-primitive-role isIncluded :domain Item :range Item :asymmetric t)
43
44
45 ; --- RULES ---
46 ; Is a car is compatible with an item that includes another item,
47 ; the car is also compatible with the second item
48 (define-rule (?car ?item2 isCompatibleCarItem)
49   (and (?car CarType) (?item1 Item) (?item2 Item) (not (same-as ?item1 ?
   item2)))
50   (?car ?item1 isCompatibleCarItem) (?item2 ?item1 isIncluded))
51 )
52
53 ; Is an item includes another item, that in turn includes another item,
54 ; the first and last items are compatible
55 (define-rule (?item1 ?item3 isIncluded)
56   (and (?item1 Item) (?item2 Item) (?item3 Item) (not (same-as ?item1 ?
   item2))) (not (same-as ?item2 ?item3)) (not (same-as ?item1 ?item3))
57   (?item1 ?item2 isIncluded) (?item2 ?item3 isIncluded))
58 )
59
60 ; --- MODELS ---
61 (instance Taycan EV)
62 (instance Macan ICEV)
63
64 ; --- WHEELS ---
65 (instance Wheels19 Wheels)
66 (instance Wheels20 Wheels)
67 (instance Wheels21 Wheels)
68
69 ; --- PAINT ---
70 (instance MetallicPaint Paint)
71 (instance PearlescentPaint Paint)
72
73 ; --- HEADLIGHTS ---
74 (instance LEDLights Headlights)
75 (instance MatrixLEDLights Headlights)
76
77 ; --- ROOF ---
78 (instance CarbonRoof Roof)
79 (instance GlassRoof Roof)
80 (instance PanoramicRoof Roof)
81
82 ; --- SEATS ---
83 (instance HeatedSeats Seats)
84 (instance HeatedSeats (equal isOptional 1))
85 (instance HeatedSeats (equal hasPrice 500))
86
87 (instance RegularSeats Seats)
88 (instance RegularSeats (equal isOptional 0))
89 (instance RegularSeats (equal hasPrice 0))
90
91 (instance SportSeats Seats)
92
93 ; --- TECH ---
94 (instance ElectricBoot Tech)
95 (instance Camera Tech)
96 (instance ACC Tech)
97 (instance FullTechPack Tech)
98 (instance StarterTechPack Tech)
99

```

```

100 ; --- TRIM ---
101 (instance MetalTrim Trim)
102 (instance WoodTrim Trim)
103 (instance RegularLeather Trim)
104 (instance NappaLeather Trim)
105
106 ; --- SAFETY ---
107 (instance BlindSpotMonitor Safety)
108 (instance BlindSpotMonitor (equal isOptional 1))
109 (instance BlindSpotMonitor (equal hasPrice 300))
110
111 (instance FullSafetySystem Safety)
112 (instance FullSafetySystem (equal isOptional 1))
113 (instance FullSafetySystem (equal hasPrice 900))
114
115 (instance ParkingSensors Safety)
116 (instance ParkingAssistant Safety)
117
118 ; --- PACKS & INCLUSIONS ---
119
120 ; Seats packs
121 (related RegularSeats HeatedSeats isIncluded)
122 (related HeatedSeats SportSeats isIncluded)
123
124 ; Light packs
125 (related LEDLights MatrixLEDLights isIncluded)
126
127 ; Roof packs
128 (related GlassRoof PanoramicRoof isIncluded)
129
130 ; Tech packs
131 (related ElectricBoot StarterTechPack isIncluded)
132 (related StarterTechPack FullTechPack isIncluded)
133 (related Camera FullTechPack isIncluded)
134 (related ACC FullTechPack isIncluded)
135
136 ; Trim packs
137 (related MetalTrim RegularLeather isIncluded)
138 (related WoodTrim NappaLeather isIncluded)
139
140 ; Safety Pack
141 (related ParkingSensors ParkingAssistant isIncluded)
142 (related ParkingAssistant FullSafetySystem isIncluded)
143 (related BlindSpotMonitor FullSafetySystem isIncluded)
144
145 ; --- COMPATIBILITIES ---
146 ; we only define model-item compatibility with the topmost item
147 ; and infer item-item compatibility with lower items
148 (related Taycan SportSeats isCompatibleCarItem)
149 (related Taycan FullSafetySystem isCompatibleCarItem)
150 (related Taycan NappaLeather isCompatibleCarItem)
151 (related Taycan MatrixLEDLights isCompatibleCarItem)
152 (related Taycan FullTechPack isCompatibleCarItem)
153 (related Taycan PanoramicRoof isCompatibleCarItem)
154 (related Taycan Wheels20 isCompatibleCarItem)
155 (related Taycan Wheels21 isCompatibleCarItem)
156 (related Taycan PearlescentPaint isCompatibleCarItem)
157
158 (related Macan HeatedSeats isCompatibleCarItem)
159 (related Macan ParkingAssistant isCompatibleCarItem)

```

```

160 (related Macan RegularLeather isCompatibleCarItem)
161 (related Macan LEDLights isCompatibleCarItem)
162 (related Macan StarterTechPack isCompatibleCarItem)
163 (related Macan CarbonRoof isCompatibleCarItem)
164 (related Macan Wheels19 isCompatibleCarItem)
165 (related Macan Wheels20 isCompatibleCarItem)
166 (related Macan MetallicPaint isCompatibleCarItem)
167
168 (execute-all-rules)
169 (reexecute-all-rules)

```

.2 Ontology evaluation

```

1 (abox-consistent?)
2 (tbox-cyclic?)
3 (tbox-coherent?)
4
5 (realize-abox)
6 (classify-tbox)
7
8 (evaluate (length (all-individuals)))
9 (evaluate (length (all-atomic-concepts)))
10 (evaluate (length (all-roles)))
11 (evaluate (length (all-rules)))
12
13 (all-concept-assertions)
14 (all-role-assertions)
15 (all-constraints)
16
17 (describe-tbox)
18 (describe-abox)
19
20 (taxonomy)
21
22 (get-tbox-language)
23 (get-abox-language)
24
25
26 ; large answer
27 (related-individuals isCompatibleCarItem)
28 (related-individuals isCompatibleItemItem)
29 (related-individuals isIncluded)
30 (individual-fillers Taycan isCompatibleCarItem)
31
32 ; manually defined
33 (concept-disjoint? Wheels Tech) ; \rightarrow T
34 (individuals-related? ParkingSensors FullSafetySystem isIncluded)
35
36 ; minilisp
37 (evaluate (> (retrieve-individual-told-attribute-value 'HeatedSeats '
    hasPrice (current-abox))
38 (retrieve-individual-told-attribute-value 'RegularSeats 'hasPrice (current-
    abox))))
39
40 ; deduced
41 (individuals-related? Taycan BlindSpotMonitor isCompatibleCarItem)
42 (individuals-related? Taycan ParkingSensors isCompatibleCarItem)
43 (individuals-related? BlindSpotMonitor Wheels19 isCompatibleItemItem)

```

```

44
45 (retrieve-individual-told-attribute-value 'HeatedSeats 'isOptional (current-
    abox))
46
47 ; NRQL
48 (get-nrql-version)
49 (enable-nrql-warnings)
50 (defquery is-interior (?x) (or (?x Seats) (?x Safety) (?x Tech) (?x Trim)))
51 (defquery is-included (?x ?y) (?x ?y isIncluded))
52
53 ; all interior items
54 (retrieve (?x) (?x is-interior))
55
56 ; items included in other items
57 (retrieve (?x ?y) (?x ?y is-included))
58
59 ; interior trim for Taycan
60 (defquery trim-options (?car ?trim) (and (?car CarType) (?trim Trim) (?car ?
    trim isCompatibleCarItem)))
61 (retrieve (?trim) (Taycan ?trim trim-options))
62
63 ; models compatible with an item
64 (defquery compat-cars (?car ?item) (and (?car CarType) (?item Item) (?car ?
    item isCompatibleCarItem)))
65 (retrieve (?car) (?car SportSeats compat-cars))
66
67 (all-queries)

```

.3 FuzzyDL axioms

```

1 (define-modifier very linear-modifier(0.8))
2
3 (define-fuzzy-concept Part1PriceRange crisp(0,10000,80,1500))
4 (define-fuzzy-concept Part2PriceRange crisp(0,10000,6000,10000))
5
6 (define-fuzzy-concept Expensive right-shoulder(0,10000,4000,10000))
7
8 (define-concept ExpensivePart (and Part (some Price (very Expensive))))
9
10
11 (instance P1 (and Part (some Price Part1PriceRange)) 1)
12 (instance P2 (and Part (some Price Part2PriceRange)) 1)
13
14 (min-instance? P1 ExpensivePart)
15 (min-instance? P2 ExpensivePart)

```

.4 Java Code

```

1 package com.kbs.main;
2
3 import com.racersystems.jracer.RacerClient;
4
5 import java.io.BufferedReader;
6 import java.io.DataOutputStream;
7 import java.io.InputStreamReader;
8 import java.net.HttpURLConnection;

```



```

9  import java.net.URL;
10 import java.net.URLEncoder;
11 import java.nio.charset.StandardCharsets;
12 import java.util.HashMap;
13 import java.util.Map;
14
15 public class Main {
16
17     private static String getParamsString(Map<String, String> params) {
18         StringBuilder result = new StringBuilder();
19
20         for (Map.Entry<String, String> entry : params.entrySet()) {
21             result.append(URLEncoder.encode(entry.getKey(), StandardCharsets
22 .UTF_8));
23             result.append("=");
24             result.append(URLEncoder.encode(entry.getValue(),
25 StandardCharsets.UTF_8));
26             result.append("&");
27         }
28
29         String resultString = result.toString();
30         return resultString.length() > 0
31             ? resultString.substring(0, resultString.length() - 1)
32             : resultString;
33     }
34
35     public static void main(String[] args) {
36
37         String input = "\"C:/Users/andre/Poli/KBS/Project/CarConfig/src/com/
38 kbs/racer/CarOntology.racer\"";
39         String queries = "\"C:/Users/andre/Poli/KBS/Project/CarConfig/src/
40 com/kbs/racer/OntologyEvaluation.racer\"";
41
42         try {
43             URL url = new URL("http://example.com");
44             HttpURLConnection con = (HttpURLConnection) url.openConnection()
45 ;
46             con.setRequestMethod("GET");
47
48             Map<String, String> parameters = new HashMap<>();
49             parameters.put("param1", "val");
50             con.setDoOutput(true);
51             DataOutputStream out = new DataOutputStream(con.getOutputStream
52 ());
53             out.writeBytes(getParamsString(parameters));
54             out.flush();
55             out.close();
56
57             int status = con.getResponseCode();
58             System.out.println(status);
59             if (status <= 299) {
60                 BufferedReader in = new BufferedReader(new InputStreamReader
61 (con.getInputStream()));
62                 StringBuffer content = new StringBuffer();
63                 String inputLine;
64                 while ((inputLine = in.readLine()) != null) {
65                     content.append(inputLine);
66                 }
67                 in.close();

```

```

62         System.out.println(content);
63     }
64
65     con.disconnect();
66
67
68     } catch (Exception e) {
69         e.printStackTrace();
70     }
71
72     String ip = "127.0.0.1";
73     int port = 8088;
74     RacerClient racer = new RacerClient(ip, port);
75     try {
76         racer.openConnection();
77         System.out.println(racer.sendRaw("(racer-read-file " + input + "
78 )"));
79
80         // obtain all individuals
81         String answer = racer.sendRaw("(concept-instances *top*)");
82         String[] result = answer.replaceAll("[()]", "").split(" ");
83
84         // if their categories are disjoint, they are compatible
85         for (String item1 : result) {
86             for (String item2 : result) {
87                 if (!item1.equals(item2) &&
88                     !item1.equals("MACAN") &&
89                     !item1.equals("TAYCAN") &&
90                     !item2.equals("MACAN") &&
91                     !item2.equals("TAYCAN")) {
92                     boolean disjoint = racer.returnBoolean(racer.sendRaw
93 ("(evaluate (concept-disjoint-p (car (car (most-specific-instantiators '"
94 + item1 + " (current-abox)))) (car (car (
95 most-specific-instantiators '"
96 + item2 + " (current-abox)))) (current-tbox)
97 ))"));
98
99                     if (disjoint) {
100                         racer.sendRaw("(related " + item1 + " " + item2
101 + " ISCOMPATIBLEITEMITEM)");
102                     }
103                 }
104             }
105         }
106
107         System.out.println(racer.sendRaw("(racer-read-file " + queries +
108 ")"));
109
110         racer.closeConnection();
111
112     } catch (Exception e) {
113         e.printStackTrace();
114     }
115 }

```