

# Relazione

## Esercitazione 8

FCA

### Introduzione

La seguente esercitazione ha riguardato la Formal Concept Analysis come metodologia per fare Ontology Learning che rappresenta il processo di creazione automatica di un'ontologia a partire dai testi e dai documenti che si hanno a disposizione.

È stato preso in considerazione e costruito un dominio riguardante le diverse tipologie di veicoli (concetti) e le loro caratteristiche (features).

Le caratteristiche sono modellate come funzioni a valore booleano che individuano la presenza o l'assenza di una proprietà in ogni concetto.

Da questa matrice (redatta sotto forma di file *csv*) viene costruito il contesto formale, una rappresentazione tassonomica dei concetti e delle features a forma di grafo, i cui nodi sono i concetti latenti e gli archi sono relazioni iperonimiche. I concetti della matrice vengono fusi in questi nuovi concetti latenti e ne viene ricavata una struttura tassonomica.

Ai fini di questa esercitazione è stata utilizzata la libreria **Context** e **graphviz**

Vehicle	Engine	Windows	Wheels	Wings	Door	Seat	Tires	Lights	Brakes	Air Conditioning
bicycle			x			x			x	
raft (zattera)						x			x	
car	x	x	x		x	x	x	x	x	x
airplane	x	x	x	x	x	x	x	x	x	x
bus	x	x	x		x	x	x	x	x	x
skateboard			x							
scooter (monopattino)	x		x				x	x	x	
motorcycle	x		x			x	x	x	x	
truck	x	x	x		x	x	x	x	x	x
tractor	x	x	x		x	x	x	x	x	
subway	x	x	x		x	x		x	x	x
aerial tramway (funivia)	x	x			x	x				
helicopter	x	x		x	x	x		x		x
tram	x	x	x		x	x		x	x	x
carriage (carrozza)		x	x		x	x				
boat	x	x			x	x				
train	x	x	x		x	x		x	x	x
van (furgone)	x	x	x		x	x	x	x	x	x
ship	x	x			x	x		x	x	x
rollerblades			x						x	

## Struttura del codice ed implementazione

Per lo svolgimento di questa esercitazione l'intero codice è presente nel notebook **main.ipynb**

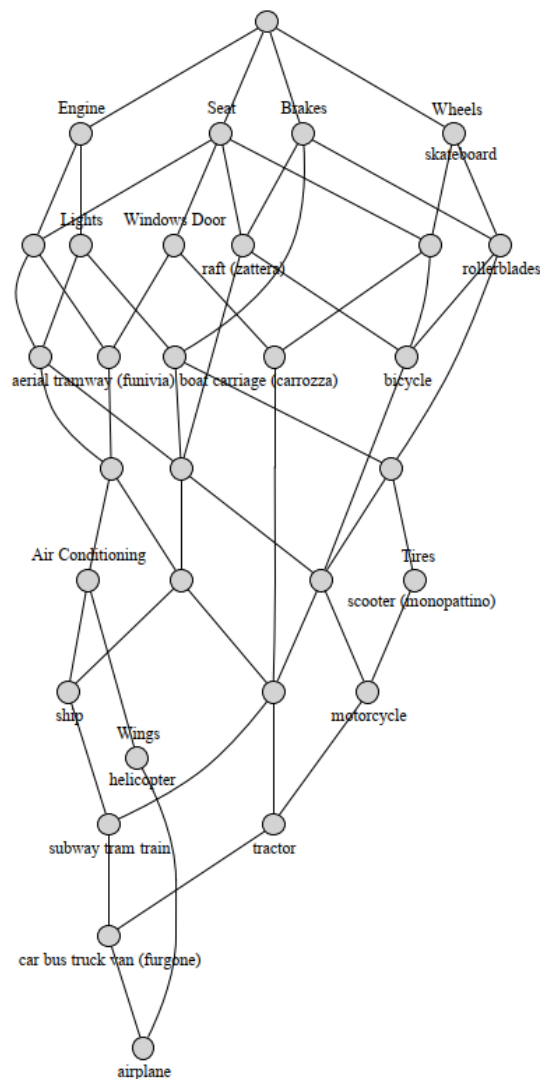
Al suo interno, tramite la libreria `concepts` ed il modulo `Context`, viene letto e parsificato il documento creato ("*fca\_data.csv*") grazie al metodo `Context.fromfile`.

```
c = Context.fromfile('./resources/fca_data.csv', frmat='csv')
```

Successivamente viene definito il lattice e visualizzato grazie all'ausilio della libreria `graphviz` attraverso la seguente linea di codice

```
c.lattice.graphviz(view=True)
```

## Risultati



Di seguito alcune `print` riguardanti le funzionalità offerte dalla libreria utilizzata

#### *common properties*

```
print(c.intension(['car', 'bicycle']))  
  
> ('Wheels', 'Seat', 'Brakes')  
  
print(c.intension(['ship', 'skateboard']))  
  
> ()
```

#### *formal context*

```
for extent, intent in c.lattice:  
    print('%r %r' % (extent, intent))  
  
> ('airplane',) ('Engine', 'Windows', 'Wheels', 'Wings', 'Door', 'Seat', 'Tires', 'Lights', 'Brakes', 'Air Conditioning')  
  
('airplane', 'helicopter') ('Engine', 'Windows', 'Wings', 'Door', 'Seat', 'Lights', 'Air Conditioning')  
  
('car', 'airplane', 'bus', 'truck', 'van (furgone)') ('Engine', 'Windows', 'Wheels', 'Door', 'Seat', 'Tires', 'Lights', 'Brakes', 'Air Conditioning')  
  
('car', 'airplane', 'bus', 'truck', 'tractor', 'van (furgone)') ('Engine', 'Windows', 'Wheels', 'Door', 'Seat', 'Tires', 'Lights', 'Brakes')  
  
('car', 'airplane', 'bus', 'motorcycle', 'truck', 'tractor', 'van (furgone)') ('Engine', 'Wheels', 'Seat', 'Tires', 'Lights', 'Brakes')  
  
('car', 'airplane', 'bus', 'scooter (monopattino)', 'motorcycle', 'truck', 'tractor', 'van (furgone)') ('Engine', 'Wheels', 'Tires', 'Lights', 'Brakes')  
  
[ . . . .]
```