Ampliació de la Ontologia pizza.owl

SID - Grup 03

Enric Segarra

Marc Font

Pablo Calomardo

08/03/2025

1 SuperMarioPizza

En aquesta secció, hem ampliat l'ontologia pizza.owl amb una nova classe anomenada `SuperMarioPizza` i el corresponent ingredient `TurtleTopping`.

1.1 Classes afegides

- **SuperMarioPizza**: Una pizza temàtica basada en el videojoc Super Mario Bros. Aquesta pizza es caracteritza per tenir bolets i tortugues com a ingredients. La classe es crea dins de `NamedPizza`.
- **TurtleTopping**: Un nou tipus de topping que representa les tortugues. La classe es crea dins de `FishTopping` com a subclasse.

1.2 Axiomes en Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `SuperMarioPizza` són:

$$\texttt{SuperMarioPizza} \sqsubseteq \texttt{Pizza} \tag{1}$$
$$\texttt{SuperMarioPizza} \sqsubseteq \exists \texttt{hasTopping.MushroomTopping} \tag{2}$$
$$\texttt{SuperMarioPizza} \sqsubseteq \exists \texttt{hasTopping.TurtleTopping} \tag{3}$$

`TurtleTopping` és una subclasse de `FishTopping`:

$$\texttt{TurtleTopping} \sqsubseteq \texttt{FishTopping} \tag{4}$$

1.3 Individus d'exemple

Per demostrar l'aplicació d'aquestes extensions, hem creat l'individu `PizzaMarioInferida`.
El raonador HermiT infereix correctament que `PizzaMarioInferida` és de tipus `SuperMarioPizza` basant-se en els axiomes definits. Això demostra que les condicions necessàries i suficients per a la classe `SuperMarioPizza` funcionen correctament.

2 Afegir Pizzes

En aquesta secció, s'ha ampliat l'ontologia pizza.owl amb diverses pizzes, cada una amb les seves condicions.

2.1 NamedPizzaInstancia3onMarisco

Per afegir aquesta pizza, s'ha creat la classe `NamedPizzaInstancia3onMarisco` i, per a que complís amb la condició de portar marisc.

2.1.1 Axiomes en la Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `NamedPizzaInstancia3onMarisco` són:

$$\texttt{NamedPizzaInstancia3onMarisco} \sqsubseteq \texttt{Pizza} \tag{5}$$
$$\texttt{NamedPizzaInstancia3onMarisco} \sqsubseteq \exists \texttt{hasTopping.MixedSeafoodTopping} \tag{6}$$
$$\texttt{NamedPizzaInstancia3onMarisco} \sqsubseteq \exists \texttt{hasTopping.PrawnsTopping} \tag{7}$$

2.2 PizzaDeMarisco

Per afegir aquesta pizza, s'ha creat la classe `PizzaDeMarisco`.

2.2.1 Axiomes en la Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `PizzaDeMarisco` són:

$$\texttt{PizzaDeMarisco} \sqsubseteq \texttt{Pizza} \tag{8}$$
$$\texttt{PizzaDeMarisco} \sqsubseteq \forall \texttt{hasTopping.}(\texttt{MixedSeafoodTopping} \sqcup \texttt{PrawnsTopping}) \tag{9}$$

2.2.2 Individus d'Exemple

S'ha creat una instancia de Pizza, `PizzaMarsicoInstancia` amb el toping `Gambitas` (del domini de `PrawnsTopping`). El reasoner correctament la infereix com a `PizzaDeMarisco`

2.3 PizzaEclectica

Per afegir aquesta pizza, s'ha creat la classe `PizzaEclectica`.

2.3.1 Axiomes en la Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `PizzaEclectica` són:

$$\texttt{PizzaEclectica} \sqsubseteq \texttt{Pizza} \tag{10}$$
$$\texttt{PizzaEclectica} \sqsubseteq \exists \texttt{hasTopping.min}\ 10\ \texttt{PizzaTopping} \tag{11}$$

On l'axioma $\exists$`hasTopping.min` 10 `PizzaTopping` representa que una `PizzaEclectica` ha de tenir com a mínim 10 toppings.

2.3.2 Individus d'Exemple

S'ha creat una instancia de Pizza, `PizzaEclecticaInstancia` amb 10 toppings diferents. El reasoner correctament la infereix com a `PizzaEclectica`.

2.4 PizzaDeOferta

Per afegir aquesta pizza, s'ha creat la classe `PizzaDeOferta`.

2.4.1 Axiomes en la Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `PizzaDeOferta` són:

$$\texttt{PizzaDeOferta} \sqsubseteq \texttt{Pizza} \tag{12}$$
$$\texttt{PizzaDeOferta} \sqsubseteq \exists \texttt{hasTopping.max}\ 2\ \texttt{PizzaTopping} \tag{13}$$

On l'axioma $\exists$`hasTopping.max` 2 `PizzaTopping` representa que una `PizzaDeOferta` ha de tenir com a màxim 2 toppings diferents.

2.4.2 Individus d'Exemple

S'ha creat una instancia de Pizza, `PizzaOfertaInstancia` amb 2 toppings diferents. El reasoner correctament la infereix com a `PizzaDeOferta`.

2.5 PizzaTriqueso

Per afegir aquesta pizza, s'ha creat la classe `PizzaTriqueso`.

2.5.1 Axiomes en la Lògica Descriptiva

Els axiomes de la classe són els següents

$$\texttt{PizzaTriqueso} \sqsubseteq \texttt{Pizza} \tag{14}$$
$$\texttt{PizzaTriqueso} \sqsubseteq= 3\texttt{hasTopping.CheeseTopping} \tag{15}$$

D'aquesta manera es garanteix que tota pizza amb exactament 3 toppings `CheeseTopping` és una `PizzaTriqueso`

2.5.2 Individus d'exemple

Per demostrar que la lògica és correcta, s'ha creat la instància `PizzaTriquesoInstancia`, amb exactament tres toppings de la categoria `CheeseTopping`. El reasoner l'ha inferit correctament com a una `PizzaTriquesoInstancia`.

2.6 PizzaEscandinava

Per afegir aquesta pizza, s'ha creat la classe `PizzaEscandinava`.

2.6.1 Axiomes en la Lògica Descriptiva

Els axiomes afegits a l'ontologia per definir la `PizzaEscandinava` són:

$$\texttt{PizzaEscandinava} \sqsubseteq \texttt{Pizza} \tag{16}$$
$$\texttt{PizzaEscandinava} \sqsubseteq (\exists\texttt{hasCountryOfOrigin.Denmark} \sqcup \exists \tag{17}$$
$$\texttt{hasCountryOfOrigin.Norway} \sqcup \exists \tag{18}$$
$$\texttt{hasCountryOfOrigin.Sweden}) \tag{19}$$

2.6.2 Individus d'exemple

Per demostrar l'aplicació d'aquesta extensió, es pot crear un individu `PizzaEscandinavaInstance` amb els següents axiomes:

$$\texttt{PizzaEscandinavaInstance} : \texttt{Pizza} \tag{20}$$
$$(\texttt{PizzaEscandinavaInstance}, \texttt{Norway}) : \texttt{hasCountryOfOrigin} \tag{21}$$

El raonador inferirà automàticament que `PizzaEscandinavaInstance` és de tipus `PizzaEscandinava` basant-se en el seu país d'origen.

3 Propietat hasCreator

En aquesta secció, hem ampliat l'ontologia pizza.owl amb una nova propietat anomenada `hasCreator` que permet assignar un creador a una pizza amb nom.

3.1 Propietat afegida

- **hasCreator**: Una propietat funcional que relaciona una `NamedPizza` amb la persona que l'ha creat. Aquesta propietat és important per documentar l'origen i la història de cada tipus de pizza.

3.2 Característiques de la propietat

La propietat `hasCreator` té les següents característiques:

- **Funcional**: Cada pizza només pot tenir un únic creador.
- **Domini**: Pizzes amb nom (`NamedPizza`).
- **Rang**: Persones (`Persona`).

3.3 Axiomes en Lògica Descriptiva

Els axiomes que defineixen la propietat `hasCreator` són:

$$\texttt{hasCreator} \sqsubseteq \texttt{ObjectProperty} \tag{22}$$

$$\texttt{Functional}(\texttt{hasCreator}) \tag{23}$$

$$\tag{24}$$

I per restringir el domini i el rang:

$$\exists \texttt{hasCreator}.\top \sqsubseteq \forall \texttt{hasCreator}.\texttt{NamedPizza} \tag{25}$$

$$\top \sqsubseteq \forall \texttt{hasCreator}^{-}.\texttt{Persona} \tag{26}$$

3.4 Individus d'exemple

Per demostrar l'ús d'aquesta propietat, hem creat l'associació entre la pizza `NamedPizzaInstancia1` i el seu creador `Paco`:

$$(\texttt{NamedPizzaInstancia1}, \texttt{Paco}) : \texttt{hasCreator} \tag{27}$$

On `NamedPizzaInstancia1` és un individu de tipus `NamedPizza` i `Paco` és un individu de tipus `Persona`.

4 Propietat influencedBy

En aquesta secció, hem ampliat l'ontologia pizza.owl amb una nova propietat anomenada `influencedBy` que permet representar les relacions d'influència entre pizzes amb nom.

4.1 Propietat afegida

- **influencedBy**: Una propietat transitiva que relaciona una pizza amb una altra pizza que l'ha influenciada. Aquesta propietat permet construir una xarxa de relacions històriques i culinàries entre les diferents pizzes.

4.2 Característiques de la propietat

La propietat `influencedBy` té les següents característiques:

- **Transitiva**: Si la pizza A està influenciada per la pizza B, i la pizza B està influenciada per la pizza C, llavors la pizza A també està influenciada per la pizza C.
- **Domini i Rang**: Tant el domini com el rang són pizzes amb nom (`NamedPizza`).

4.3 Axiomes en Lògica Descriptiva

Els axiomes que defineixen la propietat `influencedBy` són:

$$\texttt{influencedBy} \sqsubseteq \texttt{ObjectProperty} \tag{28}$$

$$\texttt{Transitive}(\texttt{influencedBy}) \tag{29}$$

I per restringir el domini i el rang:

$$\exists \texttt{influencedBy}.\top \sqsubseteq \exists \texttt{influencedBy}.\texttt{NamedPizza} \tag{30}$$

$$\top \sqsubseteq \exists \texttt{influencedBy}^{-}.\texttt{NamedPizza} \tag{31}$$

4.4 Individus d'exemple

Per demostrar l'ús d'aquesta propietat, hem creat l'associació entre `NamedPizzaInstancia2` i `NamedPizzaInstancia1`:

$$(\texttt{NamedPizzaInstancia2}, \texttt{NamedPizzaInstancia1}) : \texttt{influencedBy} \tag{32}$$

On `NamedPizzaInstancia2` és una instància de la classe `NamedPizza` i `NamedPizzaInstancia1` és una instància de la classe `NamedPizza`. Aquesta relació indica que la NamedPizzaInstancia2 ha estat influenciada per la NamedPizzaInstancia1 en la seva creació o desenvolupament.

4.5 Inferències

Gràcies a la transitivitat d'aquesta propietat, si afegim més relacions d'influència, podem inferir automàticament relacions indirectes. Per exemple, si definim:

$$(\texttt{NamedPizzaInstancia3}, \texttt{NamedPizzaInstancia2}) : \texttt{influencedBy} \tag{33}$$

El raonador inferirà automàticament:

$$(\texttt{NamedPizzaInstancia3}, \texttt{NamedPizzaInstancia1}) : \texttt{influencedBy} \tag{34}$$

Això ens permet construir una xarxa complexa de relacions històriques i d'influència culinària entre les diferents variants de pizza.

5 QuantumPizza

En aquesta secció, hem ampliat l'ontologia pizza.owl amb una nova classe anomenada `QuantumPizza` i una nova propietat simètrica i irreflexiva anomenada `DontCombine`.

5.1 Classes i propietats afegides

- **QuantumPizza**: Una classe equivalent que representa una pizza que conté simultàniament ingredients que no combinen bé.
- **DontCombine**: Una propietat simètrica i irreflexiva que relaciona dos toppings que no combinen bé quan es posen junts a una pizza.

5.2 Característiques de la propietat DontCombine

La propietat `DontCombine` té les següents característiques:

- **Simètrica**: Si l'ingredient A no combina bé amb l'ingredient B, llavors l'ingredient B tampoc combina bé amb l'ingredient A.
- **Irreflexiva**: Un ingredient no pot no combinar bé amb si mateix.
- **Domini i Rang**: Tant el domini com el rang són ingredients de pizza (`PizzaTopping`).

5.3 Axiomes en Lògica Descriptiva

Els axiomes que defineixen la propietat `DontCombine` són:

$$\texttt{DontCombine} \sqsubseteq \texttt{ObjectProperty Symmetric(DontCombine) Irreflexive(DontCombine)} \tag{35}$$

Els axiomes que defineixen la classe `QuantumPizza` són:

$$\texttt{QuantumPizza} \equiv \texttt{Pizza} \sqcap \quad (\exists \texttt{hasTopping.PizzaTopping} \sqcap (\texttt{DontCombine some (IsToppingOf Self)})) \tag{36}$$

Aquest axioma expressa que una `QuantumPizza` és equivalent a una pizza que té algun topping que no combina bé amb un altre topping que també té la mateixa pizza.

6 Repartiment

Les tasques del problema s'han distribuït a parts iguals per tots els membres del grup. Tots i cadascun d'ells han treballat de forma professional i cooperativa, generant així un àmbit de treball fructífer durant el temps que s'ha estat fent la pràctica.