DSCM031Semantic Technologies

Coursework 1: Logical Modelling

Student: Zornitsa Hristova Hristova
FN: 120168

I've chosen my microworld for modelling to be **M=My Music**.

The Group work tasks:

- The **objects** in my microworld are:
  Me, Mariya, Miya -> **persons**
  "Lonely At the Top", "Game Over", "End of Summer", "The Manifesto", "Jealous Type", "Love Me Not", "Alone" -> **songs**
  Ravyn Lenae, Doja Cat, Tame Impala, Gorillaz, Trueno, Proof, Asake, Avenged Sevenfold -> **artists**
  Afrobeats, Progressive Metal, Psychedelic Rock, Psychedelic Pop, Hip hop, Electronic, Pop, R&B, Jazz -> **genres**
  "Chill & Atmospheric Vibes", "Heavy & Energetic, Pop & Dance Mix", "Hip-Hop & Beat Driven", "Indie, Dreamy, Experimental" -> **playlists**
- Some of the **constants** in my microworld are: Me, Mariya, Miya, "Lonely At the Top", "Game Over", Pop, Jazz
- Some **Unary Predicates**:
  Person(x) -> True when *x is a person* in my microworld
  Song(x) -> True when x is a song
  Artist(x) -> True when x is an artist
  Genre(x) -> True when x is a genre
- Some **Binary Predicates**:
  Likes(Person,Song)
  Meaning: *a person likes a particular song*.
  Example: Likes(Me,"Lonely At the Top")
  Artist_of(Artist, Song)
  Meaning: *a particular artist created/performs the song*.
  Example: Artist_of(Gorillaz, "The Manifesto")
  Genre_of(Song, Genre)
  Meaning: a song belongs to a specific music genre*.*
  Example: Genre_of("Game Over"*,* Psychedelic Rock*)*
  In_playlist(Song, Playlist)
  Meaning: a song is part of a playlist*.*
  Example: **In_playlist(**"End of Summer"**,** "Indie, Dreamy, Experimental"**)**
- Attribute Function 1: **`genreOf(x)`**
  Type: 1-ary (unary) function
  Domain: Songs
  Codomain: Genres
  ### Intended meaning:
  `genreOf(x)` returns **the genre of a song**.
  Examples of the semantic interpretation in my model:

Song	genreOf(song) = Genre
Lonely At the Top	Afrobeats
Game Over	Psychedelic Rock
End of Summer	Psychedelic Pop
The Manifesto	Electronic
Jealous Type	R&B
Love Me Not	Pop
Alone	Afrobeats

Type: 1-ary function
Domain: Songs
Codomain: Artists

Intended meaning:

`artistOf(x)` returns **the artist who created the song**.

- Attribute Function 2: artistOf(x)

  Type: 1-ary function
  Domain: Songs
  Codomain: Artists

  ### Intended meaning: `artistOf(x)` returns **the artist who created the song**.

Song	artistOf(song) = Artist
Lonely At the Top	Asake
Game Over	Avenged Sevenfold
End of Summer	Tame Impala
The Manifesto	Gorillaz
Jealous Type	Ravyn Lenae
Love Me Not	Doja Cat

Facts (semantic-level facts about the microworld):
--- unary type facts
Person(me).
Person(mariya).
Person(miya).

Song(lonely_at_the_top).
Song(game_over).
Song(end_of_summer).

Artist(ravyn_lenae).
Artist(doja_cat).
Artist(tame_impala).

Genre(afrobeats).

Genre(progressive_metal).
Genre(psychedelic_rock).

artist_of(asake, lonely_at_the_top).
artist_of(trueno, game_over).
artist_of(tame_impala, end_of_summer).

--- function-based attribute facts (functional form)
genreOf(lonely_at_the_top) = afrobeats.
genreOf(game_over) = hip_hop.
genreOf(end_of_summer) = psychedelic_pop.

artistOf(lonely_at_the_top) = asake.
artistOf(game_over) = trueno.
artistOf(end_of_summer) = tame_impala.

--- some personal preferences / playlist ownership (extra facts)
likes(me, lonely_at_the_top).
likes(mariya, the_manifesto).
likes(miya, end_of_summer).

in_playlist(lonely_at_the_top, playlist_chill).
in_playlist(the_manifesto, playlist_energy).

Heuristics (4 general rules / axioms):

1. Heuristic H1 — Artist → song membership

   FOL:

   $$\forall a \forall s\ (artist\_of(a, s) \rightarrow (Artist(a) \wedge Song(s)))$$

   Rule Variant (propagation):

   $$\forall p \forall a \forall s\ (likes\_artist(p, a) \wedge artist\_of(a, s) \rightarrow likes(p, s))$$

   Text: If a person likes an artist, then they like every song by that artist.

2. Heuristic H2 — Genre preference propagation

   FOL:

   $$\forall p \forall g \forall s\ (likes\_genre(p, g) \wedge genre\_of(s, g) \rightarrow likes(p, s))$$

   Text: If a person likes a genre, they like every song that belongs to that genre.

3. Heuristic H3 — Playlist membership by genre

FOL:

$$\forall pl \forall g \forall s\ (playlist\_genre(pl, g) \wedge genre\_of(s, g) \rightarrow in\_playlist(s, pl))$$

Text: If a playlist is labeled with a genre `g`, then every song of genre `g` belongs to that playlist.

4. Heuristic H4 — Short songs → chill playlist

FOL (using a numeric-duration function `duration(Song)`):

$$\forall s\ (duration(s) < 200 \rightarrow in\_playlist(s, playlist\_chill))$$

Text: If a song is shorter than 200 seconds, it goes to the chill playlist.

Individual work tasks:

- Show the satisfiability of at least 1 heuristic in your semantic model

I'll use **Heuristic H3 (Playlist membership by genre)**:

$$\forall pl \forall g \forall s\ (playlist\_genre(pl, g) \wedge genre\_of(s, g) \rightarrow in\_playlist(s, pl))$$

I will use the following constants: Jealous type, Playlist Chill, R&B

Predicate interpretations in my microwolrd M (selected facts):

genre_of(Jealous type, R&B) is TRUE in M.

playlist_genre(Playlist Chill, R&B) is TRUE in M.

in_playlist(Jealous type, Playlist Chill) is TRUE in M.

Showing that the formula holds in MMM (satisfiability):

To check the universal formula

$$\forall pl \forall g \forall s\ (A(pl, g, s) \rightarrow B(pl, g, s))$$

we must verify that for every triple (pl, g, s) in M, the implication $A => B$

is true where:

- $A(pl, g, s)\ is\ playlist\_genre(pl, g)) \wedge genre\_of(s, g)$
- $B(pl, g, s)\ is\ in\_playlist(s, pl)$

In my model M:

Let's take the following arbitrary triple
$(pl, g, s) = (playlist\ chill, R\&B, Jealous\ Type)$:

- A is true because both $playlist\_genre(playlist\ chill, R\&B)$ and $genre\_of(Jealous\ Type, R\&B)$ are true in M
- B is true because $in\_playlist(Jealous\ Type, playlist\ chill)$ is fact in M

Therefore the implication $A => B$ is true for the triple

Now let's (pl, g, s) is such triple that either $playlist\_genre(pl, g)$ is false or $genre\_of(s, g)$ is false; then A is false, so $A => B$ is automatically true for such triples

Because every possible triple in the universe makes the implication true (either the antecedent is false or, when true, the consequent is true), the universal sentence **H3** is **true in M**.

- Demonstrate the use of at least 2 different inference rules to derive conclusions in your theory

Setup — facts and rules we will use:

Facts: likes(me, Lonely At the Top), genre_of(Lonely At the Top, afrobeats)

Heuristics: H2 and a helper rule which we will define now:
R5 (song → genre like)

$$\forall p \forall s \forall g\ ((likes(p, s) \land genre\_of(s, g)) \rightarrow likes\_genre(p, g))$$

//If a person likes a song and that song has genre g, then the person likes genre

First one: We have that (likes(Me, "Lonely At the Top") ∧ genre_of("Lonely At the Top", Afrobeats)) → likes_genre(Me, Afrobeats)

Second one:

Since we have

$$\forall p \forall g \forall s\ (likes\_genre(p, g) \land genre\_of(s, g) \rightarrow likes(p, s))$$

and likes(me, Lonely At the Top), genre_of(Lonely At the Top, afrobeats)

⇨ 'me' likes every song from afrobeates

- Convert at least 4 facts and 2 heuristics in your theory into Horn-clause theory:
  Facts I will use:

  i. **F1 (Me likes a song): likes(me, "Lonely At the Top")**

ii. **F2 (genre of "Lonely At the Top"): genre_of("Lonely At the Top", afrobeats)**
iii. **F3 (genre of "Jealous Type"): genre_of("Jealous Type", rnb)**
iv. **F4 (playlist has genre R&B): playlist_genre(playlist_chill, rnb)**
v. **F5 (song is in playlist): in_playlist("Jealous Type", playlist_chill)**

- **If a person likes a song and that song has genre G, then the person likes genre G**
- **If a person likes a genre and a song belongs to that genre, then the person likes that song**