# TD 02 - Logical aspect of databases

Antoine Groudiev

14th February 2024

**Exercise 1.** We suggest the following changes:

- Add a unique identifier to `Movie`

- Replace the `Title` field in `Cinema`, `Produced`, `Seen` and `Likes` by a `Movie` identifier

**Exercise 2.** Consider the following queries:

1. PSJR algebra:
$$\Pi_{\text{Name,Time}}\left(\sigma_{\text{Title=``Mad Max''}}(\text{Cinema})\right)$$
Conjunctive calculus:
$$\text{Cinema}(x_{\text{Title}}, x_{\text{Name}}, \text{``Mad Max''})$$

2. PSJR algebra:
$$\Pi_{\text{Title}}\left(\sigma_{\text{Director=``Orson Welles''}}(\text{Movie})\right)$$
Conjunctive calculus:
$$\exists x, \text{Movie}(x_{\text{Title}}, \text{``Orson Welles''}, x)$$

3. PSJR algebra:
$$\Pi_{\text{Actor}}\left(\sigma_{\text{Title=``Ran''}}\right)$$
Conjunctive calculus:
$$\exists x, \text{Movie}(\text{``Ran''}, x, x_{\text{Actor}})$$

4. PSJR algebra:

$$\Pi_{\text{Name}}\left(\Pi_{\text{Title}}(\sigma_{\text{Actor=``Signoret''}}(\text{Movie}) \underset{\text{Title=Cinema,Title}}{\bowtie} \text{Cinema})\right)$$

Conjunctive calculus:

$$\exists x_{\text{Title}}\left(\exists x\, \text{Movie}(x_{\text{Title}}, x, \text{``Signoret''})\right) \wedge \left(\exists x\, \text{Cinema}(x_{\text{Name}}, x, x_{\text{Title}})\right)$$

5. PSJR algebra:
$$\Pi_{\text{Actor}}\left(\text{Movie} \underset{\text{Movie.Actor=Producer.Producer}}{\bowtie} \text{Producer}\right)$$
Conjunctive calculus:

$$(\exists x, \exists y, \text{Movie}(x, y, x_{\text{Actor}})) \wedge (\exists x, \text{Producer}(x_{\text{Actor}}, x))$$

6. PSJR algebra:

$$\Pi_{\text{Actor}} \left( \text{Movie} \underset{\substack{\text{Movie.Actor=Producer.Producer} \\ \text{Movie.Title=Producer.Title}}}{\bowtie} \text{Producer} \right)$$

Conjunctive calculus:

$$\exists x_{\text{Title}} \, (\exists d \, \text{Movie}(x_{\text{Title}}, d, x_{\text{Actor}}) \wedge \text{Producer}(x_{\text{Actor}}, x_{\text{Title}}))$$

7. PSJR algebra:

$$\Pi_{\text{Actor}} \left( \Pi_{\text{Title}}(\sigma_{\text{Actor=``Orson Welles''}}(\text{Movie})) \underset{\text{Title=Movie.Title}}{\bowtie} \text{Movie} \right)$$

Conjunctive calculus:

$$\exists x_{\text{Title}} \, \Big( (\exists d \, \text{Movie}(x_{\text{Title}}, d, \text{``Orson Welles''})) \wedge (\exists d \, \text{Movie}(x_{\text{Title}}, d, x_{\text{Actor}})) \Big)$$

8. This query is undecidable since it is not monotone. Indeed, consider $D_0$ a database on the given schema, containing a movie

$$\text{Movie(``Citizen Kane'', ``Orson Welles'', ``Orson Welles'')}$$

and a producer

$$\text{Producer(``Orson Welles'', ``Citizen Kane'')}$$

In this database, the query "Which producers produce all the movies directed by Akira Kurosawa?" returns the set {"Orson Welles"}. Now, consider $D_1$ the database $D_0$ to which we added the following movie:

$$\text{Movie(``七人の侍'', ``Akira Kurosawa'', ``Toshiro Mifune'')}$$

In this database, the query "Which producers produce all the movies directed by Akira Kurosawa?" returns the empty set. Therefore, this query is not monotone, and cannot be expressed in PSJR algebra and conjunctive calculus.

**Exercise 3.** Consider the following queries:

1. PSJRU algebra:

$$t$$

Conjunctive calculus:

$$\forall m \, \text{Viewer}(x, m)$$

2. PSJRU algebra:

$$\Pi_{\text{Viewer}}(\text{Seen}) \setminus \Pi_{\text{Viewer}}(\text{Seen} \setminus \text{Likes})$$

Conjunctive calculus:

$$t$$

3. PSJRU algebra:

$$t$$

Conjunctive calculus:

$$t$$

4. PSJRU algebra:

$$t$$

Conjunctive calculus:

$$t$$

5. Undecidable.