Fundamentals of Database Systems
COMPSCI 351

Instructor: Sebastian Link

The University of Auckland

Relational Query Languages: Calculus

Logic is the beginning of wisdom, not the end.

— Spock

> Logic is the beginning of wisdom, not the end.

— Spock

> Logic is a science prior to all others, which contains the ideas and principles underlying all sciences.

— Kurt Gödel

Logic is the beginning of wisdom, not the end.

— Spock

Logic is a science prior to all others, which contains the ideas and principles underlying all sciences.

— Kurt Gödel

Logic will get you from A to B. Imagination will take you everywhere.

— Albert Einstein

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

SQL

```
FROM    MOVIE m, DIRECTOR d, PERSON p
;
```

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

SQL

```
FROM    MOVIE m, DIRECTOR d, PERSON p
WHERE   m.title=d.title AND m.year=d.year AND
        d.id=p.id AND p.first_name='Akira' AND p.last_name='Kurosawa';
```

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

SQL

```
SELECT  m.title, m.year
FROM    MOVIE m, DIRECTOR d, PERSON p
WHERE   m.title=d.title AND m.year=d.year AND
        d.id=p.id AND p.first_name='Akira' AND p.last_name='Kurosawa';
```

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational algebra

MOVIE DIRECTOR PERSON

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational algebra

$$\textsc{Movie} \bowtie \textsc{Director} \bowtie \textsc{Person}$$

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational algebra

$$\sigma_{\text{first\_name}=\text{'Akira'}}(\textsc{Movie} \bowtie \textsc{Director} \bowtie \textsc{Person})$$

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational algebra

$$\sigma_{\text{last\_name='Kurosawa'}}(\sigma_{\text{first\_name='Akira'}}(\textsc{Movie} \bowtie \textsc{Director} \bowtie \textsc{Person}))$$

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational algebra

$$\pi_{\text{title,year}}(\sigma_{\text{last\_name='Kurosawa'}}(\sigma_{\text{first\_name='Akira'}}(\textsc{Movie} \bowtie \textsc{Director} \bowtie \textsc{Person})))$$

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Example: The Same Query in Different Languages

Consider our database schema from before

- MOVIE(title, year, country, run_time, genre), DIRECTOR(id, title, year)
- PERSON(id, first_name, last_name, year_born), ACTOR(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational calculus

$$\{ \qquad \mid \qquad \textsc{Movie}(x_{\text{title}}, x_{\text{year}}, x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}})$$

$$\textsc{Director}(x_{\text{id}}, x_{\text{title}}, x_{\text{year}}) \quad \textsc{Person}(x_{\text{id}}, \text{'Akira'}, \text{'Kurosawa'}, x_{\text{year\_born}}) \ \}$$

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational calculus

$$\{ \quad | \quad \textsc{Movie}(x_{\text{title}}, x_{\text{year}}, x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}}) \land \textsc{Director}(x_{\text{id}}, x_{\text{title}}, x_{\text{year}}) \land \textsc{Person}(x_{\text{id}}, \text{'Akira'}, \text{'Kurosawa'}, x_{\text{year\_born}}) \}$$

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational calculus

$$\{ \qquad \mid \exists x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}}, x_{\text{id}}, x_{\text{year\_born}} (\textsc{Movie}(x_{\text{title}}, x_{\text{year}}, x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}}) \wedge \textsc{Director}(x_{\text{id}}, x_{\text{title}}, x_{\text{year}}) \wedge \textsc{Person}(x_{\text{id}}, \text{'Akira'}, \text{'Kurosawa'}, x_{\text{year\_born}}))\}$$

Example: The Same Query in Different Languages

Consider our database schema from before

- $\textsc{Movie}$(title, year, country, run_time, genre), $\textsc{Director}$(id, title, year)
- $\textsc{Person}$(id, first_name, last_name, year_born), $\textsc{Actor}$(id, title, year, role)

Query in English

What are the movies directed by 'Akira Kurosawa'?

Relational calculus

$$\{(x_{\text{title}}, x_{\text{year}}) \mid \exists x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}}, x_{\text{id}}, x_{\text{year\_born}} (\textsc{Movie}(x_{\text{title}}, x_{\text{year}}, x_{\text{country}}, x_{\text{run\_time}}, x_{\text{genre}}) \wedge \textsc{Director}(x_{\text{id}}, x_{\text{title}}, x_{\text{year}}) \wedge \textsc{Person}(x_{\text{id}}, \text{'Akira'}, \text{'Kurosawa'}, x_{\text{year\_born}}))\}$$

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}=\{\text{name, movie}\}$ and $\textsc{Actor}=\{\text{name, movie, role}\}$

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:
directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}$={name, movie} and $\textsc{Actor}$={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:
directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Directors acting in some of their movies

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}=\{\text{name, movie}\}$ and $\textsc{Actor}=\{\text{name, movie, role}\}$

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:

directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Names of directors with movie in which they did not act

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}$={name, movie} and $\textsc{Actor}$={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:
directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Names of directors that do not have movies in which they did not act

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}=\{\text{name, movie}\}$ and $\textsc{Actor}=\{\text{name, movie, role}\}$

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:

directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Actors who played different roles in the same movie

Simple Queries? - Yeah, Right!

Relation Schema

$\textsc{Director}=\{\text{name, movie}\}$ and $\textsc{Actor}=\{\text{name, movie, role}\}$

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:
directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\textsc{Director}) - \pi_{\text{name}}(\textsc{Director} - \pi_{\text{name,movie}}(\textsc{Director} \bowtie \textsc{Actor}))) \bowtie$$
$$(\pi_{\text{name}}(\textsc{Actor}) - \pi_{\text{name}}((\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor})) - \sigma_{\text{role}=\text{role}'}(\textsc{Actor} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\textsc{Actor}))))$$

Name of actors who played only one role in their movies

Simple Queries? - Yeah, Right!

Relation Schema

$\text{DIRECTOR}=\{\text{name, movie}\}$ and $\text{ACTOR}=\{\text{name, movie, role}\}$

Query in English:

Which directors acted in all of their movies in precisely one role?

In Relational algebra:
directors who acted in all their movies, and in only one role in all their movies

$$(\pi_{\text{name}}(\text{DIRECTOR}) - \pi_{\text{name}}(\text{DIRECTOR} - \pi_{\text{name,movie}}(\text{DIRECTOR} \bowtie \text{ACTOR}))) \bowtie$$
$$(\pi_{\text{name}}(\text{ACTOR}) - \pi_{\text{name}}((\text{ACTOR} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\text{ACTOR})) - \sigma_{\text{role}=\text{role}'}(\text{ACTOR} \bowtie \delta_{\text{role}\mapsto\text{role}'}(\text{ACTOR}))))$$

Directors who acted in all their movies and only in one role

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
        SELECT d2.movie FROM DIRECTOR d2
        WHERE d2.name = d1.name AND NOT EXISTS(
                SELECT a1.role FROM ACTOR a1
                WHERE a1.name = d2.name AND a1.movie = d2.movie
                AND NOT EXISTS(
                        SELECT a2.role FROM ACTOR a2
                        WHERE a2.role ≠ a1.role
                           AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
        SELECT d2.movie FROM DIRECTOR d2
        WHERE d2.name = d1.name AND NOT EXISTS(
                SELECT a1.role FROM ACTOR a1
                WHERE a1.name = d2.name AND a1.movie = d2.movie
                AND NOT EXISTS(
                        SELECT a2.role FROM ACTOR a2
                        WHERE a2.role ≠ a1.role
                          AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
         SELECT d2.movie FROM DIRECTOR d2
         WHERE d2.name = d1.name AND NOT EXISTS(
                  SELECT a1.role FROM ACTOR a1
                  WHERE a1.name = d2.name AND a1.movie = d2.movie
                  AND NOT EXISTS(
                           SELECT a2.role FROM ACTOR a2
                           WHERE a2.role ≠ a1.role
                              AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
        SELECT d2.movie FROM DIRECTOR d2
        WHERE d2.name = d1.name AND NOT EXISTS(
                SELECT a1.role FROM ACTOR a1
                WHERE a1.name = d2.name AND a1.movie = d2.movie
                AND NOT EXISTS(
                        SELECT a2.role FROM ACTOR a2
                        WHERE a2.role ≠ a1.role
                          AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
        SELECT d2.movie FROM DIRECTOR d2
        WHERE d2.name = d1.name AND NOT EXISTS(
                SELECT a1.role FROM ACTOR a1
                WHERE a1.name = d2.name AND a1.movie = d2.movie
                AND NOT EXISTS(
                        SELECT a2.role FROM ACTOR a2
                        WHERE a2.role ≠ a1.role
                           AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Simple Queries? - Yeah, Right!

Relation Schema

DIRECTOR={name, movie} and ACTOR={name, movie, role}

Query in English:

Which directors acted in all of their movies in precisely one role?

In SQL: find directors who did not direct any movie in which there is no role that they played and there is no other role that they played

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
        SELECT d2.movie FROM DIRECTOR d2
        WHERE d2.name = d1.name AND NOT EXISTS(
                SELECT a1.role FROM ACTOR a1
                WHERE a1.name = d2.name AND a1.movie = d2.movie
                AND NOT EXISTS(
                        SELECT a2.role FROM ACTOR a2
                        WHERE a2.role ≠ a1.role
                           AND a2.movie = a1.movie AND a2.name= a1.name)))
```

The Need for a Declarative Query Language

Many simple natural language queries are not simple in RA or SQL

The Need for a Declarative Query Language

Many simple natural language queries are not simple in RA or SQL

Need a query language that is:

- close to natural language
  - $\hookrightarrow$ to help express queries easily
  - $\hookrightarrow$ i.e., declarative, and not operational language

The Need for a Declarative Query Language

Many simple natural language queries are not simple in RA or SQL

Need a query language that is:

- close to natural language
  - $\hookrightarrow$ to help express queries easily
  - $\hookrightarrow$ i.e., declarative, and not operational language
- is precise
  - $\hookrightarrow$ to exclude ambiguities in queries and their results

The Need for a Declarative Query Language

Many simple natural language queries are not simple in RA or SQL

Need a query language that is:

- close to natural language
  - $\hookrightarrow$ to help express queries easily
  - $\hookrightarrow$ i.e., declarative, and not operational language
- is precise
  - $\hookrightarrow$ to exclude ambiguities in queries and their results
- provides a foundation for queries in practice
  - $\hookrightarrow$ to enable real-world querying

Relational Calculus

Relational calculus is a QL based on first-order logic

Formalization of everyday language using

- connectors (not, and, or), and
- quantifiers (exists, for all)

Relational Calculus

Relational calculus is a QL based on first-order logic

Formalization of everyday language using

- connectors (not, and, or), and
- quantifiers (exists, for all)

Safe Relational Calculus

- syntactic fragment of Relational Calculus equivalent to Relational Algebra
- queries can be transformed automatically into queries resembling SQL style

Set Comprehension

Simply declare what the result should be

Give me the set of all tuples t that satisfy the (complex) conditions expressed by φ

Set Comprehension

Simply declare what the result should be

Give me the set of all tuples t that satisfy the (complex) conditions expressed by φ

Relational Calculus does just that

$$\{t \mid \varphi\} \text{ with some formula } \varphi$$

Set Comprehension

Simply declare what the result should be

Give me the set of all tuples t that satisfy the (complex) conditions expressed by φ

Relational Calculus does just that

$$\{t \mid \varphi\} \text{ with some formula } \varphi$$

Need to formalise this approach

A language for these formulae, that is,
how do these formulae look like and what do they mean?

Flavors of Relational Calculus

Two different, yet equivalent versions

Flavors of Relational Calculus

Two different, yet equivalent versions

Tuple Relational Calculus (TRC)

Variables represent tuples

Domain Relational Calculus (DRC)

Variables represent individual values in the domains

Flavors of Relational Calculus

Two different, yet equivalent versions

Tuple Relational Calculus (TRC)

Variables represent tuples

Domain Relational Calculus (DRC)

Variables represent individual values in the domains

We focus on the Domain Relational Calculus

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}} \, (\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the ‘Event’ cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}}\, (\textsc{Screening}(\text{‘Event’}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}} \, (\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}}\ (\textsc{Screening}(\text{`Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the ‘Event’ cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}} \, (\textsc{Screening}(\text{‘Event’}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Example for a query in DRC

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Cinema} = \{\text{cinema, city, address}\}$

Query in English

What are the movies that play in the 'Event' cinema?

Variables as placeholders for domain values

x_{movie} for movies, x_{date} for dates, and x_{time} for time

The query in DRC

$$\{x_{\text{movie}} \mid \exists x_{\text{date}}, x_{\text{time}}\ (\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}))\}$$

Meaning

The set of all movies x_{movie}, such that there is a date x_{date} and a time x_{time} such that the movie x_{movie} is played in the 'Event' cinema on date x_{date} at time x_{time}

Objects: Variables

Basic objects (terms): placeholders (variables) and values (constants)

Objects: Variables

Basic objects (terms): placeholders (variables) and values (constants)

Variables

Let V_i $(i \in I)$ be set of "sufficiently many" *variables*

Objects: Variables

Basic objects (terms): placeholders (variables) and values (constants)

Variables

Let V_i $(i \in I)$ be set of "sufficiently many" *variables*

Variables are placeholders

Each variable $x \in V_i$ is a placeholder for elements from domain D_i

We say D_i is the *sort* of x

Objects: Variables

Basic objects (terms): placeholders (variables) and values (constants)

Variables

Let V_i $(i \in I)$ be set of "sufficiently many" *variables*

Variables are placeholders

Each variable $x \in V_i$ is a placeholder for elements from domain D_i
We say D_i is the *sort* of x

$V = \bigcup_{i \in I} V_i$ denotes the set of all variables

Objects: Constants and Terms

Constants

Elements of the domain D_i are called *constants* of sort D_i

Objects: Constants and Terms

Constants

Elements of the domain D_i are called *constants* of sort D_i

Terms

Variables and constants together form the set of *terms of sort* D_i

Objects: Constants and Terms

Constants

Elements of the domain D_i are called *constants* of sort D_i

Terms

Variables and constants together form the set of *terms of sort* D_i

$\#t$ denotes the sort of the term t

Objects: Constants and Terms

Constants

Elements of the domain D_i are called *constants* of sort D_i

Terms

Variables and constants together form the set of *terms of sort* D_i

$\#t$ denotes the sort of the term t

Example

'Event' constant of sort `string`, x_{time} variable of sort `time`

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \ldots, n$, then the expression $R(t_1, \ldots, t_n)$ is called a *predicate formula*

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \dots, n$, then the expression $R(t_1, \dots, t_n)$ is called a *predicate formula*

$$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$$

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

Comparison Formulae

Comparing objects

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \ldots, n$, then the expression $R(t_1, \ldots, t_n)$ is called a *predicate formula*

$$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$$

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \ldots, n$, then the expression $R(t_1, \ldots, t_n)$ is called a *predicate formula*

$$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$$

Comparison Formulae

Comparing objects

For terms t and t' of the same sort $\#t$ the equation $t = t'$ is called a *comparison formula*

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \ldots, n$, then the expression $R(t_1, \ldots, t_n)$ is called a *predicate formula*

$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$

Comparison Formulae

Comparing objects

For terms t and t' of the same sort $\#t$ the equation $t = t'$ is called a *comparison formula*

$x_{\text{movie}} = \text{'The Seven Samurai'}$

Atomic Properties of Objects

Predicate Formulae

Composing objects into rows of a table

If $R \in \mathcal{S}$ and R has arity n, and t_i is a term of sort $dom(i)$ for all $i = 1, \ldots, n$, then the expression $R(t_1, \ldots, t_n)$ is called a *predicate formula*

$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$

Comparison Formulae

Comparing objects

For terms t and t' of the same sort $\#t$ the equation $t = t'$ is called a *comparison formula*

$x_{\text{movie}} = \text{'The Seven Samurai'}$

Predicate formulae and comparison formulae together form the set $\mathcal{F}_0$ of *atomic formulae* (or *atoms*)

Complex Properties of Objects

Compose complex properties out of simpler ones

Complex Properties of Objects

Compose complex properties out of simpler ones

A *formula* is either an atom or built from other formulae by the following rules:

- if φ is a formula, then this is also true for $(\neg\varphi)$ (negation)

Complex Properties of Objects

Compose complex properties out of simpler ones

A *formula* is either an atom or built from other formulae by the following rules:

- if φ is a formula, then this is also true for $(\neg\varphi)$ (negation)
- if φ and ψ are formulae, then this is also true for $(\varphi \wedge \psi)$ (conjunction)

Complex Properties of Objects

Compose complex properties out of simpler ones

A *formula* is either an atom or built from other formulae by the following rules:

- if φ is a formula, then this is also true for $(\neg\varphi)$ (negation)
- if φ and ψ are formulae, then this is also true for $(\varphi \wedge \psi)$ (conjunction)
- if φ is a formula and $x \in V$, then $(\exists x \varphi)$ (existential quantification) is a formula

Complex Properties of Objects

Compose complex properties out of simpler ones

A *formula* is either an atom or built from other formulae by the following rules:

- if φ is a formula, then this is also true for $(\neg\varphi)$ (negation)
- if φ and ψ are formulae, then this is also true for $(\varphi \wedge \psi)$ (conjunction)
- if φ is a formula and $x \in V$, then $(\exists x \varphi)$ (existential quantification) is a formula

We omit parenthesis when this does not lead to ambiguity

Complex Properties of Objects

Compose complex properties out of simpler ones

A *formula* is either an atom or built from other formulae by the following rules:

- if φ is a formula, then this is also true for $(\neg\varphi)$ (negation)
- if φ and ψ are formulae, then this is also true for $(\varphi \wedge \psi)$ (conjunction)
- if φ is a formula and $x \in V$, then $(\exists x \varphi)$ (existential quantification) is a formula

We omit parenthesis when this does not lead to ambiguity

$\mathcal{F}$ denotes the *set of all formulae*

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

'Akira Kurosawa'

Constant from domain of name for $\textsc{Director}$

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\text{'Akira Kurosawa'}, x_{\text{movie}}$

Variable over domain of movie

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Predicate formula over $\textsc{Director}$

Example of how to compose a complex formula

$\text{SCREENING} = \{\text{cinema, movie, date, time}\}$ and $\text{DIRECTOR} = \{\text{name, movie}\}$

‘Event’ $\text{DIRECTOR}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Constant from domain of cinema

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\text{'Event'}, x_{\text{movie}}$ $\textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Variable over domain of movie

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

'Event', $x_{\text{movie}}, x_{\text{date}}$ $\textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Variable over domain of date

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}} \quad \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Variable over domain of time

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\textsc{Screening}(\text{`Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})$ $\textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Predicate formula over $\textsc{Screening}$

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\textsc{Screening}(\text{'Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})$

Conjunction of two predicate formulae

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$$\exists x_{\text{time}}(\textsc{Screening}(\text{`Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

Existential quantification over domain of time

Example of how to compose a complex formula

$\textsc{Screening} = \{\text{cinema, movie, date, time}\}$ and $\textsc{Director} = \{\text{name, movie}\}$

$\exists x_{\text{date}}(\exists x_{\text{time}}(\textsc{Screening}(\text{`Event'}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}})))$

Existential quantification over domain of date

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Implication

$\varphi \Rightarrow \psi$ short for $\neg\varphi \vee \psi$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Implication

$\varphi \Rightarrow \psi$ short for $\neg\varphi \vee \psi$

Equivalence

$\varphi \Leftrightarrow \psi$ short for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Implication

$\varphi \Rightarrow \psi$ short for $\neg\varphi \vee \psi$

Equivalence

$\varphi \Leftrightarrow \psi$ short for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

Universal quantification

$\forall x(\varphi)$ short for $\neg\exists x(\neg\varphi)$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Implication

$\varphi \Rightarrow \psi$ short for $\neg\varphi \vee \psi$

Equivalence

$\varphi \Leftrightarrow \psi$ short for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

Universal quantification

$\forall x(\varphi)$ short for $\neg\exists x(\neg\varphi)$

Successive existential quantification

$\exists x_1, x_2, \ldots, x_n(\varphi)$ short for
$\exists x_1(\exists x_2(\ldots(\exists x_n(\varphi))\ldots))$

Shortcuts

Inequation

$t \neq t'$ short for $\neg(t = t')$

Disjunction

$\varphi \vee \psi$ short for $\neg(\neg\varphi \wedge \neg\psi)$

Implication

$\varphi \Rightarrow \psi$ short for $\neg\varphi \vee \psi$

Equivalence

$\varphi \Leftrightarrow \psi$ short for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

Universal quantification

$\forall x(\varphi)$ short for $\neg\exists x(\neg\varphi)$

Successive existential quantification

$\exists x_1, x_2, \ldots, x_n(\varphi)$ short for
$\exists x_1(\exists x_2(\ldots(\exists x_n(\varphi))\ldots))$

Successive universal quantification

$\forall x_1, x_2, \ldots, x_n(\varphi)$ short for
$\forall x_1(\forall x_2(\ldots(\forall x_n(\varphi))\ldots))$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\forall x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \Rightarrow \neg(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\forall x_{\mathsf{movie}}, x_{\mathsf{date}}, x_{\mathsf{time}}(\neg\textsc{Screening}(x_{\mathsf{cinema}}, x_{\mathsf{movie}}, x_{\mathsf{date}}, x_{\mathsf{time}}) \vee \neg(\textsc{Director}(\text{'Polanski'}, x_{\mathsf{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\mathsf{movie}})))$$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\neg(\neg\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \vee \neg(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}}))))$$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\neg\neg\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})\wedge$$
$$\neg\neg(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

Example for Syntactic Transformation (without Change of Meaning)

Cinemas in which every movie screened is directed neither by Polanski nor Kubrick

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge (\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

Cinemas in which there is no screening of a movie directed by Polanski or Kubrick

Free Variables

Those placeholders that describe the structure of query answers

Free Variables

Those placeholders that describe the structure of query answers

Those variables that are not bound by any quantifier

Let $fr(\varphi)$ denote the set of free variables of $\varphi \in \mathcal{F}$

- If φ is an atom, then each variable occurring in φ belongs to $fr(\varphi)$

Free Variables

Those placeholders that describe the structure of query answers

Those variables that are not bound by any quantifier

Let $fr(\varphi)$ denote the set of free variables of $\varphi \in \mathcal{F}$

- If φ is an atom, then each variable occurring in φ belongs to $fr(\varphi)$
- Negation does not change the set of free variables, i.e., $fr(\neg\varphi) = fr(\varphi)$

Free Variables

Those placeholders that describe the structure of query answers

Those variables that are not bound by any quantifier

Let $fr(\varphi)$ denote the set of free variables of $\varphi \in \mathcal{F}$

- If φ is an atom, then each variable occurring in φ belongs to $fr(\varphi)$
- Negation does not change the set of free variables, i.e., $fr(\neg\varphi) = fr(\varphi)$
- For conjunction we have $fr(\varphi \wedge \psi) = fr(\varphi) \cup fr(\psi)$

Free Variables

Those placeholders that describe the structure of query answers

Those variables that are not bound by any quantifier

Let $\mathit{fr}(\varphi)$ denote the set of free variables of $\varphi \in \mathcal{F}$

- If φ is an atom, then each variable occurring in φ belongs to $\mathit{fr}(\varphi)$
- Negation does not change the set of free variables, i.e., $\mathit{fr}(\neg\varphi) = \mathit{fr}(\varphi)$
- For conjunction we have $\mathit{fr}(\varphi \wedge \psi) = \mathit{fr}(\varphi) \cup \mathit{fr}(\psi)$
- For existential quantification we obtain $\mathit{fr}(\exists x(\varphi)) = \mathit{fr}(\varphi) - \{x\}$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})\wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)$=

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})\wedge (\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)=$

$$fr(\cdot)$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge (\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)=$

$$fr(\cdot) - \{x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)=$

$$(fr(\cdot) \cup \{x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}) - \{x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})\wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)$=

$$((\{x_{\text{movie}}\} \cup \{x_{\text{movie}}\}) \cup \{x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}) - \{x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}})\wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)=$

$$\{x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\} - \{x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}\}$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\mathsf{movie}}, x_{\mathsf{date}}, x_{\mathsf{time}}(\textsc{Screening}(x_{\mathsf{cinema}}, x_{\mathsf{movie}}, x_{\mathsf{date}}, x_{\mathsf{time}}) \wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\mathsf{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\mathsf{movie}})))$$

$fr(\varphi)=$

$$\{x_{\mathsf{cinema}}\}$$

Example for Free Variables

Compute the free variables of the following formula φ:

$$\neg\exists x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}(\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge$$
$$(\textsc{Director}(\text{'Polanski'}, x_{\text{movie}}) \vee \textsc{Director}(\text{'Kubrick'}, x_{\text{movie}})))$$

$fr(\varphi)=$

$$\{x_{\text{cinema}}\}$$

Indeed, we are looking for all the cinemas in which
no movie directed by Polanski or Kubrick is screened.

Formulae and their Evaluation

A formula describes a (complex) property of an object

Formulae and their Evaluation

A formula describes a (complex) property of an object

Formulae evaluate to either true (**T**) or false (**F**) since
a property either holds or does not hold for any object

The evaluation of any formulae over $\mathcal{S}$ is determined by

- a database instance over $\mathcal{S}$
- and a mapping of all its placeholders to constants from their domains

Interpretation of Formulae

An *interpretation* $\mathcal{I}$ of the domain relational calculus over a database schema $\mathcal{S}$ consists of

- a database *db* over $\mathcal{S}$ and

Interpretation of Formulae

An *interpretation* $\mathcal{I}$ of the domain relational calculus over a database schema $\mathcal{S}$ consists of

- a database *db* over $\mathcal{S}$ and
- a variable assignment $\tau : V \to D$ with $\tau(x) \in D_i$ for $\#x = D_i$

Interpretation of Formulae

An *interpretation* $\mathcal{I}$ of the domain relational calculus over a database schema $\mathcal{S}$ consists of

- a database *db* over $\mathcal{S}$ and
- a variable assignment $\tau : V \to D$ with $\tau(x) \in D_i$ for $\#x = D_i$

Write $\mathcal{I} = (db, \tau)$

Interpretation of Formulae

An *interpretation* $\mathcal{I}$ of the domain relational calculus over a database schema $\mathcal{S}$ consists of

- a database *db* over $\mathcal{S}$ and
- a variable assignment $\tau : V \to D$ with $\tau(x) \in D_i$ for $\#x = D_i$

Write $\mathcal{I} = (db, \tau)$

Next: define how to evaluate formulae based on a fixed interpretation $\mathcal{I} = (db, \tau)$

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{'Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{`Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Given the fixed SCREENING-relation:

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	4pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

The query $Q(x_{\text{movie}}, x_{\text{time}})$ returns all those $(x_{\text{movie}}, x_{\text{time}})$-pairs which make the formula $Q(x_{\text{movie}}, x_{\text{time}})$ true

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{'Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Given the fixed $\textsc{Screening}$-relation:

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	4pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

The query $Q(x_{\text{movie}}, x_{\text{time}})$ returns all those $(x_{\text{movie}}, x_{\text{time}})$-pairs which make the formula $Q(x_{\text{movie}}, x_{\text{time}})$ true

E.g. the variable assignments ($x_{\text{movie}} \mapsto$ 'The Godfather', $x_{\text{time}} \mapsto$ '4pm') and ($x_{\text{movie}} \mapsto$ 'The Godfather', $x_{\text{time}} \mapsto$ '8pm') make $Q(x_{\text{movie}}, x_{\text{time}})$ true, i.e., ('The Godfather','4pm') and ('The Godfather','8pm') form the answer

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{'Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Given the fixed SCREENING-relation:

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	4pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

The query $Q(x_{\text{movie}}, x_{\text{time}})$ returns all those $(x_{\text{movie}}, x_{\text{time}})$-pairs which make the formula $Q(x_{\text{movie}}, x_{\text{time}})$ true

E.g. the variable assignments $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'4pm'})$ and $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'8pm'})$ make $Q(x_{\text{movie}}, x_{\text{time}})$ true, i.e., ('The Godfather','4pm') and ('The Godfather','8pm') form the answer

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{'Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Given the fixed SCREENING-relation:

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	4pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

The query $Q(x_{\text{movie}}, x_{\text{time}})$ returns all those $(x_{\text{movie}}, x_{\text{time}})$-pairs which make the formula $Q(x_{\text{movie}}, x_{\text{time}})$ true

E.g. the variable assignments $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'4pm'})$ and $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'8pm'})$ make $Q(x_{\text{movie}}, x_{\text{time}})$ true, i.e., ('The Godfather','4pm') and ('The Godfather','8pm') form the answer

Example for an Interpretation of a Formula

Which movies are shown at which time at the *Event* cinema on *28.03.*?

$$Q(x_{\text{movie}}, x_{\text{time}}) = \textsc{Screening}(\text{'Event'}, x_{\text{movie}}, \text{'28.03.'}, x_{\text{time}})$$

Given the fixed SCREENING-relation:

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	4pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

The query $Q(x_{\text{movie}}, x_{\text{time}})$ returns all those $(x_{\text{movie}}, x_{\text{time}})$-pairs which make the formula $Q(x_{\text{movie}}, x_{\text{time}})$ true

E.g. the variable assignments $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'4pm'})$ and $(x_{\text{movie}} \mapsto \text{'The Godfather'}, x_{\text{time}} \mapsto \text{'8pm'})$ make $Q(x_{\text{movie}}, x_{\text{time}})$ true, i.e., ('The Godfather','4pm') and ('The Godfather','8pm') form the answer

How are formulae evaluated in general?

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$
- for formulae $\varphi \in \mathcal{F}$ we proceed inductively:
  - $\omega(R(t_1, \ldots, t_n)) = \mathbf{T}$ iff $R(\omega(t_1), \ldots, \omega(t_n)) \in db$

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$
- for formulae $\varphi \in \mathcal{F}$ we proceed inductively:
  - $\omega(R(t_1, \dots, t_n)) = \mathbf{T}$ iff $R(\omega(t_1), \dots, \omega(t_n)) \in db$
  - $\omega(t = t') = \mathbf{T}$ iff $\omega(t)$ and $\omega(t')$ are equal in D_i

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$
- for formulae $\varphi \in \mathcal{F}$ we proceed inductively:
  - $\omega(R(t_1, \dots, t_n)) = \mathbf{T}$ iff $R(\omega(t_1), \dots, \omega(t_n)) \in db$
  - $\omega(t = t') = \mathbf{T}$ iff $\omega(t)$ and $\omega(t')$ are equal in D_i
  - $\omega(\neg\varphi) = \mathbf{T}$ iff $\omega(\varphi) = \mathbf{F}$

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$
- for formulae $\varphi \in \mathcal{F}$ we proceed inductively:
  - $\omega(R(t_1, \dots, t_n)) = \mathbf{T}$ iff $R(\omega(t_1), \dots, \omega(t_n)) \in db$
  - $\omega(t = t') = \mathbf{T}$ iff $\omega(t)$ and $\omega(t')$ are equal in D_i
  - $\omega(\neg\varphi) = \mathbf{T}$ iff $\omega(\varphi) = \mathbf{F}$
  - $\omega(\varphi \wedge \psi) = \mathbf{T}$ iff $\omega(\varphi) = \omega(\psi) = \mathbf{T}$

The Evaluation of Formulae

Extend $\mathcal{I} = (db, \tau)$ to all terms and formulae using the notation $\omega = \omega_{(db,\tau)}$:

- for terms t we simply put $\omega(t) = \begin{cases} \tau(t) & \text{, if } t \text{ is a variable,} \\ \omega(t) = t & \text{, if } t \text{ is a constant} \end{cases}$
- for formulae $\varphi \in \mathcal{F}$ we proceed inductively:
  - $\omega(R(t_1, \ldots, t_n)) = \mathbf{T}$ iff $R(\omega(t_1), \ldots, \omega(t_n)) \in db$
  - $\omega(t = t') = \mathbf{T}$ iff $\omega(t)$ and $\omega(t')$ are equal in D_i
  - $\omega(\neg\varphi) = \mathbf{T}$ iff $\omega(\varphi) = \mathbf{F}$
  - $\omega(\varphi \wedge \psi) = \mathbf{T}$ iff $\omega(\varphi) = \omega(\psi) = \mathbf{T}$
  - $\omega(\exists x(\varphi)) = \mathbf{T}$ iff there is some replacement of x in φ by a constant $d \in \#x$ that leads to a formula ψ with $\omega(\psi) = \mathbf{T}$

Example for an Interpretation of a Formula

Which cinemas show on which day a movie directed by *Akira Kurosawa*?

$$Q(x_{\text{cinema}}, x_{\text{date}}) =$$
$$\exists x_{\text{movie}}, x_{\text{time}} \, (\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

SCREENING-relation

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

DIRECTOR-relation

name	movie
Akira Kurosawa	The Seven Samurai
F.F. Coppola	The Godfather
Christopher Nolan	Inception

Example for an Interpretation of a Formula

Which cinemas show on which day a movie directed by *Akira Kurosawa*?

$$Q(x_{\text{cinema}}, x_{\text{date}}) =$$
$$\exists x_{\text{movie}}, x_{\text{time}}\ (\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

Screening-relation

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

Director-relation

name	movie
Akira Kurosawa	The Seven Samurai
F.F. Coppola	The Godfather
Christopher Nolan	Inception

Assignment $(x_{\text{cinema}} \mapsto \text{'Event'}, x_{\text{date}} \mapsto \text{'27.03.'})$ makes $Q(x_{\text{cinema}}, x_{\text{date}})$ true

Example for an Interpretation of a Formula

Which cinemas show on which day a movie directed by *Akira Kurosawa*?

$$Q(x_{\text{cinema}}, x_{\text{date}}) =$$
$$\exists x_{\text{movie}}, x_{\text{time}}\ (\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

Screening-relation

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

Director-relation

name	movie
Akira Kurosawa	The Seven Samurai
F.F. Coppola	The Godfather
Christopher Nolan	Inception

Assignment $(x_{\text{cinema}} \mapsto \text{'Event'}, x_{\text{date}} \mapsto \text{'27.03.'})$ makes $Q(x_{\text{cinema}}, x_{\text{date}})$ true

Example for an Interpretation of a Formula

Which cinemas show on which day a movie directed by *Akira Kurosawa*?

$$Q(x_{\text{cinema}}, x_{\text{date}}) =$$
$$\exists x_{\text{movie}}, x_{\text{time}} \, (\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

Screening-relation

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

Director-relation

name	movie
Akira Kurosawa	The Seven Samurai
F.F. Coppola	The Godfather
Christopher Nolan	Inception

Assignment $(x_{\text{cinema}} \mapsto \text{'Event'}, x_{\text{date}} \mapsto \text{'27.03.'})$ makes $Q(x_{\text{cinema}}, x_{\text{date}})$ true

There are values for x_{movie} ('The Seven Samurai') and x_{time} ('8pm') such that
$\textsc{Screening}(\text{'Event'}, \text{'The Seven Samurai'}, \text{'27.03.'}, \text{'8pm'}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, \text{'The Seven Samurai'})$
evaluates to true

Example for an Interpretation of a Formula

Which cinemas show on which day a movie directed by *Akira Kurosawa*?

$$Q(x_{\text{cinema}}, x_{\text{date}}) =$$
$$\exists x_{\text{movie}}, x_{\text{time}}\ (\textsc{Screening}(x_{\text{cinema}}, x_{\text{movie}}, x_{\text{date}}, x_{\text{time}}) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, x_{\text{movie}}))$$

SCREENING-relation

cinema	movie	date	time
Event	The Seven Samurai	27.03.	8pm
Event	The Godfather	28.03.	8pm
Rialto	Inception	28.03.	8pm

DIRECTOR-relation

name	movie
Akira Kurosawa	The Seven Samurai
F.F. Coppola	The Godfather
Christopher Nolan	Inception

Assignment $(x_{\text{cinema}} \mapsto \text{'Event'}, x_{\text{date}} \mapsto \text{'27.03.'})$ makes $Q(x_{\text{cinema}}, x_{\text{date}})$ true

In fact, it is true that
('Event', 'The Seven Samurai', '27.03.', '8pm') forms a row in the SCREENING-table and
('Akira Kurosawa', 'The Seven Samurai') forms a row in the DIRECTOR-table

Domain Relational Calculus

Each query Q in the language $\mathcal{L}_{\mathrm{DRC}}$ of the *domain relational calculus* has the form

$$Q \quad = \quad \{(x_1, \ldots, x_n) \mid \varphi\}$$

with variables $x_i \in V$ and a formula $\varphi \in \mathcal{F}$ with $fr(\varphi) = \{x_1, \ldots, x_n\}$

Domain Relational Calculus

Each query Q in the language $\mathcal{L}_{\text{DRC}}$ of the *domain relational calculus* has the form

$$Q \quad = \quad \{(x_1, \ldots, x_n) \mid \varphi\}$$

with variables $x_i \in V$ and a formula $\varphi \in \mathcal{F}$ with $fr(\varphi) = \{x_1, \ldots, x_n\}$

- The input-schema is the underlying database schema: $in(Q) = \mathcal{S}$

Domain Relational Calculus

Each query Q in the language $\mathcal{L}_{\text{DRC}}$ of the *domain relational calculus* has the form

$$Q \quad = \quad \{(x_1, \ldots, x_n) \mid \varphi\}$$

with variables $x_i \in V$ and a formula $\varphi \in \mathcal{F}$ with $fr(\varphi) = \{x_1, \ldots, x_n\}$

- The input-schema is the underlying database schema: $in(Q) = \mathcal{S}$
- The output-schema is $out(Q) = \{ans(Q)\}$ with a relation schema $ans(Q)$ of arity n with $dom(i) = \#x_i$ for $i = 1, \ldots, n$

Domain Relational Calculus

Each query Q in the language $\mathcal{L}_{\mathrm{DRC}}$ of the *domain relational calculus* has the form

$$Q \quad = \quad \{(x_1, \ldots, x_n) \mid \varphi\}$$

with variables $x_i \in V$ and a formula $\varphi \in \mathcal{F}$ with $fr(\varphi) = \{x_1, \ldots, x_n\}$

- The input-schema is the underlying database schema: $in(Q) = \mathcal{S}$
- The output-schema is $out(Q) = \{ans(Q)\}$ with a relation schema $ans(Q)$ of arity n with $dom(i) = \#x_i$ for $i = 1, \ldots, n$
- The query mapping $q(Q)$ is given by

$$q(Q)(db) \quad = \quad \{ans(Q)(\tau(x_1), \ldots, \tau(x_n)) \mid \omega_{(db,\tau)}(\varphi) = \mathbf{T}\},$$

  where τ denotes variable assignments

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{ \quad | \quad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \qquad\qquad\qquad\qquad\qquad\qquad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \quad \textsc{Screening}(\qquad , m, \quad , \ ) \ \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \qquad \textsc{Screening}(\text{`Event'}, m, \quad , \quad) \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{`Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{ \qquad \mid \qquad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{‘Event’}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$

$$\}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{`Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$

$$\textsc{Cinema}(c, c', a) \qquad \textsc{Screening}(c, m, d, t) \qquad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$

$$\textsc{Cinema}(c, c', a) \qquad \textsc{Screening}(c, m, d, t) \quad \textsc{Director}(\text{'Akira Kurosawa'}, m)\ \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \quad \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{`Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{`Akira Kurosawa'}, m))\}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \land \exists m(\textsc{Screening}(c, m, d, t) \land \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{ \quad \mid \quad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{(m) \mid \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{(m) \mid \quad \textsc{Actor}(\text{'Reese Witherspoon'}, m, \quad) \quad \textsc{Actor}(\text{'Jack Nicholson'}, m, \quad)\ \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{(m) \mid \quad \textsc{Actor}(\text{'Reese Witherspoon'}, m, \quad) \vee \textsc{Actor}(\text{'Jack Nicholson'}, m, \quad)\ \}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{(m) \mid \exists r(\textsc{Actor}(\text{'Reese Witherspoon'}, m, r) \vee \textsc{Actor}(\text{'Jack Nicholson'}, m, r))\}$$

Examples for Queries in DRC

Example (Which movies are screened in the *Event* cinema?)

$$\{(m) \mid \exists d, t(\ \textsc{Screening}(\text{'Event'}, m, d, t))\}$$

Example (Where and when do we find screenings of movies directed by *Akira Kurosawa*?)

$$\{(c, c', a, d, t) \mid$$
$$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$$

Example (In which movies did *Reese Witherspoon* or *Jack Nicholson* act?)

$$\{(m) \mid \exists r(\textsc{Actor}(\text{'Reese Witherspoon'}, m, r) \vee \textsc{Actor}(\text{'Jack Nicholson'}, m, r))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{ \quad | \quad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \qquad \text{SCREENING}(\qquad , m, d, t) \ \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \quad \textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t) \;\}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{ \qquad \mid \qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t) \ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \qquad\qquad\qquad \textsc{Screening}(c, m, d, t)$$
$$\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \qquad \textsc{Screening}(c, m, d, t)$$
$$\textsc{Director}(n, m) \qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\begin{array}{ll} \{(c, c', a) \mid \textsc{Cinema}(c, c', a) & \textsc{Screening}(c, m, d, t) \\ \qquad\qquad \textsc{Director}(n, m) & \textsc{Actor}(n, m', r) \quad \} \end{array}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\begin{array}{ll} \{(c, c', a) \mid \textsc{Cinema}(c, c', a) & \textsc{Screening}(c, m, d, t) \\ \qquad\qquad \textsc{Director}(n, m) & \neg\exists m', r(\textsc{Actor}(n, m', r))\ \} \end{array}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t) \ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\begin{array}{ll} \{(c, c', a) \mid \textsc{Cinema}(c, c', a) & \qquad (\textsc{Screening}(c, m, d, t) \\ \qquad \wedge \textsc{Director}(n, m)) & \neg\exists m', r(\textsc{Actor}(n, m', r)) \ \} \end{array}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \qquad (\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r))\ \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \quad \forall m, d, t, n((\textsc{Screening}(c, m, d, t)$$
$$\wedge\ \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{\quad \mid \quad\quad\quad\quad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \qquad \textsc{Director}(n, m) \qquad\qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \qquad \textsc{Director}(n, m) \\ \textsc{Actor}(n, m', r) \qquad\qquad\qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \quad \textsc{Director}(n, m) \qquad \textsc{Director}(n, m') \qquad \textsc{Actor}(n, m', r) \qquad \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \quad \textsc{Director}(n, m) \qquad \textsc{Director}(n, m') \quad \exists r(\textsc{Actor}(n, m', r) \qquad)\ \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \quad \textsc{Director}(n, m) \qquad \textsc{Director}(n, m') \qquad \Rightarrow \exists r(\textsc{Actor}(n, m', r) \qquad)\ \}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \quad \textsc{Director}(n, m) \quad \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \quad))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t)$$
$$\wedge\ \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \qquad \textsc{Director}(n, m) \qquad \forall m'(\textsc{Director}(n, m')$$
$$\Rightarrow \exists r(\textsc{Actor}(n, m', r) \qquad \textsc{Actor}(n, m', r') \Rightarrow r = r'\ ))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t) \ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \qquad \textsc{Director}(n, m) \qquad \forall m'(\textsc{Director}(n, m') \\ \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t) \ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \land \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \quad \textsc{Director}(n, m) \ \land \forall m'(\textsc{Director}(n, m') \\ \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

More Examples for Queries in DRC

Example (Are there screenings in the *Rialto* cinema?)

$$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t)\ \}$$

Example (In which cinemas are only movies screened that are directed only by non-actors?)

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \\ \land \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Example (Which directors played exactly one role in each of their movies?)

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \\ \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Tuple Relational Calculus: Sorts and Terms

The TRC is based on a named global perspective

Tuple Relational Calculus: Sorts and Terms

The TRC is based on a named global perspective

Sorts

- Sorts are not only the domains D_i, but also relation schemata $R \subseteq \mathcal{U}$ including those in $\mathcal{S}$
- Only provide 'sufficiently large' sets V_R of variables for each of the 'relation sorts' $R \subseteq \mathcal{U}$

Tuple Relational Calculus: Sorts and Terms

The TRC is based on a named global perspective

Sorts

- Sorts are not only the domains D_i, but also relation schemata $R \subseteq \mathcal{U}$ including those in $\mathcal{S}$
- Only provide 'sufficiently large' sets V_R of variables for each of the 'relation sorts' $R \subseteq \mathcal{U}$

Terms

- terms of sort R are only the variables $t \in V_R$
- terms of sort D_i are either the constants $d \in D_i$ or have the form $t : A$, where A is an attribute in $\#t$ (again denoting the sort of t) with $dom(A) = D_i$

Tuple Relational Calculus: Atoms and Formulae

Atoms

- Either *predicate formulae* of the form $R(t)$ with $t \in V_R$ or
- *comparison formulae* of the form $t_1 = t_2$ with terms t_1, t_2 of the same sort

Tuple Relational Calculus: Atoms and Formulae

Atoms

- Either *predicate formulae* of the form $R(t)$ with $t \in V_R$ or
- *comparison formulae* of the form $t_1 = t_2$ with terms t_1, t_2 of the same sort

Formulae

- Are built inductively from the atoms as for the DRC
- Free variables are defined analogously

Tuple Relational Calculus: Interpretation and Queries

Interpretation

Needs value assignment τ to the tuple variables:
for $t \in V_R$ the value $\tau(t)$ is some R-tuple

Tuple Relational Calculus: Interpretation and Queries

Interpretation

Needs value assignment τ to the tuple variables:
for $t \in V_R$ the value $\tau(t)$ is some R-tuple

Extending interpretations

Interpret an atom $R(t)$ as true iff $\tau(t)$ is a tuple in the R-relation of the database db

Tuple Relational Calculus: Interpretation and Queries

Interpretation

Needs value assignment τ to the tuple variables:
for $t \in V_R$ the value $\tau(t)$ is some R-tuple

Extending interpretations

Interpret an atom $R(t)$ as true iff $\tau(t)$ is a tuple in the R-relation of the database *db*

Queries in the TRC

$$\{t \mid \varphi\} \text{ where } fr(\varphi) = \{t\}$$

Tuple Relational Calculus: Interpretation and Queries

Interpretation

Needs value assignment τ to the tuple variables:
for $t \in V_R$ the value $\tau(t)$ is some R-tuple

Extending interpretations

Interpret an atom $R(t)$ as true iff $\tau(t)$ is a tuple in the R-relation of the database db

Queries in the TRC

$$\{t \mid \varphi\} \text{ where } fr(\varphi) = \{t\}$$

Query mapping

$$q(Q)(db) \quad = \quad \{\tau(t) \mid \omega_{(db,\tau)}(\varphi) = \mathbf{T}\}$$

Tuple Relational Calculus: Interpretation and Queries

Interpretation

Needs value assignment τ to the tuple variables:
for $t \in V_R$ the value $\tau(t)$ is some R-tuple

Extending interpretations

Interpret an atom $R(t)$ as true iff $\tau(t)$ is a tuple in the R-relation of the database db

Queries in the TRC

$$\{t \mid \varphi\} \text{ where } fr(\varphi) = \{t\}$$

Query mapping

$$q(Q)(db) \quad = \quad \{\tau(t) \mid \omega_{(db,\tau)}(\varphi) = \mathbf{T}\}$$

DRC and TRC can shown to be equivalent

Examples of TRC queries

Which movies are screened in the *Event* cinema?

- in DRC: $\{(m) \mid \exists d, t\ (\textsc{Screening}(\text{'Event'}, m, d, t))\}$
- in TRC: $\{(s\text{:m}) \mid \exists s', s'\text{:c}, s'\text{:m}(\textsc{Screening}(s') \wedge s'\text{:c} = \text{'Event'} \wedge s'\text{:m} = s\text{:m})\}$

Examples of TRC queries

Which movies are screened in the *Event* cinema?

- in DRC: $\{(m) \mid \exists d, t\ (\textsc{Screening}(\text{'Event'}, m, d, t))\}$
- in TRC: $\{(s\text{:m}) \mid \exists s', s'\text{:c}, s'\text{:m}(\textsc{Screening}(s') \land s'\text{:c} = \text{'Event'} \land s'\text{:m} = s\text{:m})\}$

Which movie directors played exactly one role in each of their movies?

- in DRC:

$$\begin{aligned}\{(n) \mid \exists m.\textsc{Director}(n, m) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \\ \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}\end{aligned}$$

- in TRC:

$$\begin{aligned}\{(d\text{:n}) \mid \exists d', d'\text{:n}(\textsc{Director}(d') \land d\text{:n} = d'\text{:n} \land \\ \forall d'', d''\text{:n}, d''\text{:m}((\textsc{Director}(d'') \land d''\text{:n} = d'\text{:n}) \Rightarrow \\ \exists a, a\text{:n}, a\text{:m}, a\text{:r}(\textsc{Actor}(a) \land a\text{:n} = d''\text{:n} \land a\text{:m} = d''\text{:m} \land \\ \forall a', a'\text{:n}, a'\text{:m}, a'\text{:r}((\textsc{Actor}(a') \land a'\text{:n} = a\text{:n} \land a'\text{:m} = a\text{:m}) \Rightarrow a'\text{:r} = a\text{:r}))))\}\end{aligned}$$

Problematic Queries: Example 1

Intended: List cinemas that never screen 'The Godfather'

$$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$$

Problematic Queries: Example 1

Intended: List cinemas that never screen 'The Godfather'

$$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$$

produces an *infinite* answer, since $\sharp c = \texttt{string}$ and for every string c that is not an actual cinema no d and no t exist such that $(c, \text{'The Godfather'}, d, t)$ is in $\textsc{Screening}$

Problematic Queries: Example 1

Intended: List cinemas that never screen 'The Godfather'

$$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$$

produces an *infinite* answer, since $\sharp c = \texttt{string}$ and for every string c that is not an actual cinema no d and no t exist such that $(c, \text{'The Godfather'}, d, t)$ is in $\textsc{Screening}$

$$\{(c) \mid \exists c', a(\textsc{Cinema}(c, c', a)) \wedge \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$$

Problematic Queries: Example 2

Intended: List (c, d) pairs that shows cinemas which screen 'The Piano' on $d =$ '28.03.', or screen 'Inception' in the $c =$ 'Event' cinema

$$\{(c, d) \mid$$
$$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$$

Problematic Queries: Example 2

Intended: List (c, d) pairs that shows cinemas which screen 'The Piano' on $d =$ '28.03.', or screen 'Inception' in the $c =$ 'Event' cinema

$$\{(c, d) \mid \exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$$

undesired answers: if ('Rialto','The Piano','28.03.','9pm') in Screening, then ('Rialto', d) in answer for every date d, and if ('Event','Inception','26.03.','8pm') in Screening, then (c, '26.03.') in answer for every string c

Problematic Queries: Example 2

Intended: List (c, d) pairs that shows cinemas which screen ‘The Piano’ on $d =$ ’28.03.’, or screen ’Inception’ in the $c =$ ’Event’ cinema

$$\{(c, d) \mid \exists t(\textsc{Screening}(c, \text{‘The Piano’}, \text{’28.03.’}, t) \vee \textsc{Screening}(\text{‘Event’}, \text{’Inception’}, d, t))\}$$

undesired answers: if (’Rialto’,‘The Piano’,’28.03.’,’9pm’) in $\textsc{Screening}$, then (’Rialto’, d) in answer for every date d, and if (’Event’,‘Inception’,’26.03.’,’8pm’) in $\textsc{Screening}$, then (c, ’26.03.’) in answer for every string c

$$\begin{aligned}\{(c, d) \mid \exists t(\quad & (\textsc{Screening}(c, \text{‘The Piano’}, d, t) \wedge d = \text{’28.03.’}) \vee \\ & (\textsc{Screening}(c, \text{’Inception’}, d, t) \wedge c = \text{‘Event’}))\}\end{aligned}$$

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problems

- produces a finite answer which depends on the domains *dom*(date) and *dom*(time)

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problems

- produces a finite answer which depends on the domains *dom*(date) and *dom*(time)
- if *dom*(date) is infinite, then the answer relation will be empty

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problems

- produces a finite answer which depends on the domains *dom*(date) and *dom*(time)
- if *dom*(date) is infinite, then the answer relation will be empty
- if it is finite, it is possible to get a non-empty answer relation

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problems

- produces a finite answer which depends on the domains *dom*(date) and *dom*(time)
- if *dom*(date) is infinite, then the answer relation will be empty
- if it is finite, it is possible to get a non-empty answer relation

$$\{(c) \mid \exists c', a(\textsc{Cinema}(c, c', a)) \land \forall m, d, t(\textsc{Screening}(c, m, d, t) \Rightarrow \textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problematic Queries: Example 3

Intended: List cinemas that always screen 'When Harry met Sally'

$$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

Problems

- produces a finite answer which depends on the domains *dom*(date) and *dom*(time)
- if *dom*(date) is infinite, then the answer relation will be empty
- if it is finite, it is possible to get a non-empty answer relation

$$\{(c) \mid \quad \exists c', a(\textsc{Cinema}(c, c', a)) \wedge \forall m, d, t(\textsc{Screening}(c, m, d, t) \Rightarrow \textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$$

General problem: answer relation depends on the domains D_i

Solutions - Active Domains

Notation

Let *adom*(Q) denote the (finite) set of constants occurring in the query Q, and let *adom*(db) denote the (finite) sets of constants occurring in the database db

Solutions - Active Domains

Notation

Let $adom(Q)$ denote the (finite) set of constants occurring in the query Q, and let $adom(db)$ denote the (finite) sets of constants occurring in the database db

Restriction of variable assignments

$$\tau(x) \in (adom(Q) \cup adom(db)) \cap D_i \quad \text{is required for} \quad \#x = D_i$$

Solutions - Active Domains

Notation

Let $adom(Q)$ denote the (finite) set of constants occurring in the query Q, and let $adom(db)$ denote the (finite) sets of constants occurring in the database db

Restriction of variable assignments

$$\tau(x) \in (adom(Q) \cup adom(db)) \cap D_i \quad \text{is required for} \quad \#x = D_i$$

we talk of *active domain semantics*

Solutions - Domain Independence

Consider sets *adom* with

$$adom(db) \cup adom(Q) \subseteq adom \subseteq \bigcup_{i \in I} D_i$$

Solutions - Domain Independence

Consider sets *adom* with

$$adom(db) \cup adom(Q) \subseteq adom \subseteq \bigcup_{i \in I} D_i$$

- consider variable assignments restricted to $\tau(x) \in adom \cap D_i$ for $\#x = D_i$
- only allow queries for which the result does not depend on the chosen *adom*
- we talk of *domain independent semantics*

Solutions - Domain Independence

Consider sets *adom* with

$$adom(db) \cup adom(Q) \subseteq adom \subseteq \bigcup_{i \in I} D_i$$

- consider variable assignments restricted to $\tau(x) \in adom \cap D_i$ for $\#x = D_i$
- only allow queries for which the result does not depend on the chosen *adom*
- we talk of *domain independent semantics*

domain independence is an undecidable property

Solutions - Safe Relational Calculus

syntactic restrictions on the formulae used in DRC queries

Solutions - Safe Relational Calculus

syntactic restrictions on the formulae used in DRC queries

such formulae will be called *safe*

Safe Range Normal Form

Safe range normal form (*SRNF*)

Checks whether a query is safe or not

Safe Range Normal Form

Safe range normal form (*SRNF*)

Checks whether a query is safe or not

SRNF of a query Q does not change Q's meaning, it just checks if Q is safe

Safe Range Normal Form

Safe range normal form (*SRNF*)

Checks whether a query is safe or not

SRNF of a query Q does not change Q's meaning, it just checks if Q is safe

SRNF results from the application of transformation rules to formulae that preserve their meaning

Transformation Rules

Elimination of shortcuts

First eliminate shortcuts related to universal quantification $\forall$, implication $\Rightarrow$ and equivalence $\Leftrightarrow$

Transformation Rules

Elimination of shortcuts

First eliminate shortcuts related to universal quantification $\forall$, implication $\Rightarrow$ and equivalence $\Leftrightarrow$

Bounded renaming

Change the names of bounded variables, i.e., those occurring within the scope of a quantifier, in such a way that there is no variable which is both free and bound.

Transformation Rules

Elimination of shortcuts

First eliminate shortcuts related to universal quantification $\forall$, implication $\Rightarrow$ and equivalence $\Leftrightarrow$

Bounded renaming

Change the names of bounded variables, i.e., those occurring within the scope of a quantifier, in such a way that there is no variable which is both free and bound.

Shift negation

Successively replace subformulae in a way that negation only occurs in front of an existential quantifier or an atom:

- remove double negation, i.e., replace $\neg\neg\varphi$ by φ
- replace $\neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$ by $\neg\varphi_1 \vee \cdots \vee \neg\varphi_n$
- replace $\neg(\varphi_1 \vee \cdots \vee \varphi_n)$ by $\neg\varphi_1 \wedge \cdots \wedge \neg\varphi_n$

Transformation Rules

Shift disjunction

Successively apply the distribution laws for conjunction and disjunction until there is no more disjunction occurring within the scope of a conjunction.

Transformation Rules

Shift disjunction

Successively apply the distribution laws for conjunction and disjunction until there is no more disjunction occurring within the scope of a conjunction.

Omit parentheses

Omit all parentheses that are unnecessary according to associativity laws.

Example: Transformation into SRNF

Hitchcock query

$$\exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))$$

Example: Transformation into SRNF

Replace inner implication

$$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))$$

Example: Transformation into SRNF

Replace inner implication

$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\neg\textsc{Actor}(n, m', r') \vee r = r')))$

Example: Transformation into SRNF

Replace inner universal quantification

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\neg\textsc{Actor}(n,m',r') \vee r = r')))$$

Example: Transformation into SRNF

Replace inner universal quantification

$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\neg(\neg\textsc{Actor}(n, m', r') \vee r = r'))))$

Example: Transformation into SRNF

Shift negation

$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\neg(\neg\textsc{Actor}(n,m',r') \vee r = r'))))$

Example: Transformation into SRNF

Shift negation

$\exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \neg\exists r'(\neg\neg\textsc{Actor}(n, m', r') \land \neg r = r')))$

Example: Transformation into SRNF

Eliminate double negation

$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\neg\neg\textsc{Actor}(n,m',r') \wedge \neg r = r')))$

Example: Transformation into SRNF

Eliminate double negation

$$\exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \neg\exists r'(\textsc{Actor}(n,m',r') \land r \neq r')))$$

Example: Transformation into SRNF

Replace implication

$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$

Example: Transformation into SRNF

Replace implication

$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\neg\textsc{Director}(n, m') \vee \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$

Example: Transformation into SRNF

Replace universal quantification

$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\neg\textsc{Director}(n, m') \vee \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$

Example: Transformation into SRNF

Replace universal quantification

$\exists m(\textsc{Director}(n, m)) \wedge \neg\exists m'(\neg(\neg\textsc{Director}(n, m') \vee \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r'))))$

Example: Transformation into SRNF

Shift negation

$\exists m(\textsc{Director}(n, m)) \wedge \neg\exists m'(\neg(\neg\textsc{Director}(n, m') \vee \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r'))))$

Example: Transformation into SRNF

Shift negation

$\exists m(\textsc{Director}(n, m)) \land \neg\exists m'(\neg\neg\textsc{Director}(n, m') \land \neg\exists r(\textsc{Actor}(n, m', r) \land \neg\exists r'(\textsc{Actor}(n, m', r') \land r \neq r')))$

Example: Transformation into SRNF

Eliminate double negation

$\exists m(\textsc{Director}(n, m)) \land \neg\exists m'(\neg\neg\textsc{Director}(n, m') \land \neg\exists r(\textsc{Actor}(n, m', r) \land \neg\exists r'(\textsc{Actor}(n, m', r') \land r \neq r')))$

Example: Transformation into SRNF

Eliminate double negation

$$\exists m(\textsc{Director}(n,m)) \wedge \neg\exists m'(\textsc{Director}(n,m') \wedge \neg\exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\textsc{Actor}(n,m',r') \wedge r \neq r')))$$

Example: Transformation into SRNF

Hitchcock in SRNF

$\exists m(\textsc{Director}(n, m)) \wedge \neg\exists m'(\textsc{Director}(n, m') \wedge \neg\exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$

Safe Formulae

Major characteristic of formulae in SRNF

for each negated formula $\neg\varphi$,
the subformula φ is either an atom or an existentially quantified formula

Safe Formulae

Major characteristic of formulae in SRNF

for each negated formula $\neg\varphi$,
the subformula φ is either an atom or an existentially quantified formula

Roughly speaking a formula is safe iff all free variables in its SRNF are range-restricted

Range-restricted Variables

The set $rr(\varphi)$ of *range-restricted* variables is defined inductively

Range-restricted Variables

The set $rr(\varphi)$ of *range-restricted* variables is defined inductively

Atoms

$$rr(\varphi) = \begin{cases} \textit{undefined} & \text{, if } \varphi \text{ denotes } x = y \\ fr(\varphi) & \text{, otherwise} \end{cases}$$

Range-restricted Variables

The set $rr(\varphi)$ of *range-restricted* variables is defined inductively

Atoms

$$rr(\varphi) = \begin{cases} \textit{undefined} & \text{, if } \varphi \text{ denotes } x = y \\ fr(\varphi) & \text{, otherwise} \end{cases}$$

Conjunction $\varphi \equiv \varphi_1 \wedge \cdots \wedge \varphi_n$

- Let $\mathbb{F}$ be those φ_i that are not comparison formulae $x = y$ involving just two variables or negations of such formulae
- If at least one $rr(\varphi_i)$ with $\varphi_i \in \mathbb{F}$ is undefined, then $rr(\varphi)$ is undefined, too.
- Otherwise, for equivalence class $[\![x]\!]_=$ of x wrt equality in $\varphi_i \notin \mathbb{F}$ define:

$$rr(\varphi) = \bigcup_{\varphi_i \in \mathbb{F}} rr(\varphi_i) \cup \{[\![x]\!]_= \mid \text{some } \varphi_i \text{ is } x = y \text{ and } [\![x]\!]_= \cap rr(\varphi_j) \neq \emptyset \text{ for some } \varphi_j \in \mathbb{F}\}$$

Range-restricted Variables

Disjunction $\varphi \equiv \varphi_1 \vee \cdots \vee \varphi_n$

- $rr(\varphi) = \bigcap_{i=1}^{n} rr(\varphi_i)$, if all $rr(\varphi_i)$ are defined
- Otherwise let $rr(\varphi)$ be undefined

Range-restricted Variables

Disjunction $\varphi \equiv \varphi_1 \vee \cdots \vee \varphi_n$

- $rr(\varphi) = \bigcap_{i=1}^{n} rr(\varphi_i)$, if all $rr(\varphi_i)$ are defined
- Otherwise let $rr(\varphi)$ be undefined

Existential formulae

- $rr(\exists x(\psi)) = rr(\psi) - \{x\}$, if $rr(\psi)$ is defined and $x \in rr(\psi)$
- Otherwise let it be undefined

Range-restricted Variables

Disjunction $\varphi \equiv \varphi_1 \vee \cdots \vee \varphi_n$

- $rr(\varphi) = \bigcap_{i=1}^{n} rr(\varphi_i)$, if all $rr(\varphi_i)$ are defined
- Otherwise let $rr(\varphi)$ be undefined

Existential formulae

- $rr(\exists x(\psi)) = rr(\psi) - \{x\}$, if $rr(\psi)$ is defined and $x \in rr(\psi)$
- Otherwise let it be undefined

Negation

- $rr(\neg\varphi) = \emptyset$, if $rr(\varphi)$ is defined
- Otherwise let it be undefined

Safe Relational Calculus

A formula φ is called *safe* iff its SRNF ψ satisfies $rr(\psi) = fr(\varphi)$

Safe Relational Calculus

A formula φ is called *safe* iff its SRNF ψ satisfies $rr(\psi) = fr(\varphi)$

A query in DRC is *safe* iff its defining formula is safe

Our Examples Revisited

$\{(m) \mid \exists d, t\ (\textsc{Screening}(\text{'Event'}, m, d, t))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

Our Examples Revisited

$\{(m) \mid \exists d, t \ (\textsc{Screening}(\text{`Event'}, m, d, t))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

$\{(c, c', a, d, t) \mid$
$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{`Akira Kurosawa'}, m))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{c, c', a, d, t\} = fr(\varphi)$.

Our Examples Revisited

$\{(m) \mid \exists d, t\, (\textsc{Screening}(\text{'Event'}, m, d, t))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

$\{(c, c', a, d, t) \mid$
$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{c, c', a, d, t\} = fr(\varphi)$.

$\{(m) \mid \exists r(\textsc{Actor}(\text{'Reese Witherspoon'}, m, r) \vee \textsc{Actor}(\text{'Jack Nicholson'}, m, r))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

Our Examples Revisited

$\{(m) \mid \exists d, t\ (\textsc{Screening}(\text{'Event'}, m, d, t))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

$\{(c, c', a, d, t) \mid$
$\textsc{Cinema}(c, c', a) \wedge \exists m(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(\text{'Akira Kurosawa'}, m))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{c, c', a, d, t\} = fr(\varphi)$.

$\{(m) \mid \exists r(\textsc{Actor}(\text{'Reese Witherspoon'}, m, r) \vee \textsc{Actor}(\text{'Jack Nicholson'}, m, r))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \{m\} = fr(\varphi)$.

$\{() \mid \exists m, d, t(\textsc{Screening}(\text{'Rialto'}, m, d, t))\}$

The formula is already in SRNF. Here we get $rr(\varphi) = \emptyset = fr(\varphi)$.

Another Example Revisited

Original Query

$$\{(c,c',a) \mid \textsc{Cinema}(c,c',a) \wedge \forall m,d,t,n((\textsc{Screening}(c,m,d,t) \wedge \textsc{Director}(n,m)) \Rightarrow \neg\exists m',r(\textsc{Actor}(n,m',r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \\ \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \\ \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) {\color{red}\Rightarrow} \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \land \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) \Rightarrow \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \land \forall m, d, t, n(\neg(\textsc{Screening}(c, m, d, t) \land \textsc{Director}(n, m)) \lor \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n(\neg(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \vee \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \neg \exists m, d, t, n \neg(\neg(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \vee \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \textsc{Cinema}(c, c', a) \wedge \neg\exists m, d, t, n\neg(\neg(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \vee \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \neg\exists m, d, t, n\neg\neg(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \wedge \neg\neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \text{CINEMA}(c, c', a) \wedge \forall m, d, t, n((\text{SCREENING}(c, m, d, t) \wedge \text{DIRECTOR}(n, m)) \Rightarrow \neg\exists m', r(\text{ACTOR}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \text{CINEMA}(c, c', a) \wedge \neg\exists m, d, t, n \neg\neg(\text{SCREENING}(c, m, d, t) \wedge \text{DIRECTOR}(n, m)) \wedge \neg\neg\exists m', r(\text{ACTOR}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg\exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \neg\exists m, d, t, n(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \wedge \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Another Example Revisited

Original Query

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \forall m, d, t, n((\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \Rightarrow \neg \exists m', r(\textsc{Actor}(n, m', r)))\}$$

Transform into SRNF

$$\{(c, c', a) \mid \quad \textsc{Cinema}(c, c', a) \wedge \neg \exists m, d, t, n(\textsc{Screening}(c, m, d, t) \wedge \textsc{Director}(n, m)) \wedge \exists m', r(\textsc{Actor}(n, m', r)))\}$$

The query is safe because $rr(\varphi) = \{c, c', a\} = fr(\varphi)$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Another Example Revisited

Original Query

$$\{(n) \mid \quad \exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \quad \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\neg\textsc{Actor}(n,m',r') \vee r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \forall r'(\neg\textsc{Actor}(n,m',r') \lor r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'\neg(\neg\textsc{Actor}(n,m',r') \vee r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r' \neg(\neg\textsc{Actor}(n,m',r') \vee r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\neg\neg\textsc{Actor}(n,m',r') \wedge \neg r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\neg\neg\textsc{Actor}(n,m',r') \wedge \neg r = r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \neg\exists r'(\textsc{Actor}(n,m',r') \land r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\text{Director}(n, m)) \land \forall m'(\text{Director}(n, m') \Rightarrow \exists r(\text{Actor}(n, m', r) \land \forall r'(\text{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\text{Director}(n, m)) \land \forall m'(\text{Director}(n, m') \Rightarrow \exists r(\text{Actor}(n, m', r) \land \neg\exists r'(\text{Actor}(n, m', r') \land r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \wedge \forall m'(\neg\textsc{Director}(n, m') \vee \exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \\ \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \forall m'(\neg\textsc{Director}(n,m') \vee \\ \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\textsc{Actor}(n,m',r') \wedge r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \land \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \land \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \land \neg\exists m'\neg(\neg\textsc{Director}(n, m') \lor \exists r(\textsc{Actor}(n, m', r) \land \neg\exists r'(\textsc{Actor}(n, m', r') \land r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \neg\exists m' \neg(\neg\textsc{Director}(n,m') \vee \exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\textsc{Actor}(n,m',r') \wedge r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \land \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \land \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \land \neg\exists m'(\neg\neg\textsc{Director}(n,m') \land \neg\exists r(\textsc{Actor}(n,m',r) \land \neg\exists r'(\textsc{Actor}(n,m',r') \land r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \wedge \neg\exists m'(\neg\neg\textsc{Director}(n, m') \wedge \neg\exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n, m)) \wedge \forall m'(\textsc{Director}(n, m') \Rightarrow \exists r(\textsc{Actor}(n, m', r) \wedge \forall r'(\textsc{Actor}(n, m', r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n, m)) \wedge \neg\exists m'(\textsc{Director}(n, m') \wedge \neg\exists r(\textsc{Actor}(n, m', r) \wedge \neg\exists r'(\textsc{Actor}(n, m', r') \wedge r \neq r')))$$

Another Example Revisited

Original Query

$$\{(n) \mid \exists m(\textsc{Director}(n,m)) \wedge \forall m'(\textsc{Director}(n,m') \Rightarrow \exists r(\textsc{Actor}(n,m',r) \wedge \forall r'(\textsc{Actor}(n,m',r') \Rightarrow r = r')))\}$$

Transform into SRNF

$$\exists m(\textsc{Director}(n,m)) \wedge \neg\exists m'(\textsc{Director}(n,m') \wedge \neg\exists r(\textsc{Actor}(n,m',r) \wedge \neg\exists r'(\textsc{Actor}(n,m',r') \wedge r \neq r')))$$

The query is safe because $rr(\varphi) = \{n\} = fr(\varphi)$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) =$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) =$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \{c, d, t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \{c, d, t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \{c\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \{c\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) =$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) =$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

**$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$**

- Is in SRNF
- It is not safe because $rr(\varphi) = \{c, t\} \cap \{d, t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \{t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \{t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi)$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi)$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi) = \{c, d, t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg \textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

**$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$**

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg \exists d, t(\neg \textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi) = \{c, d, t\}$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

**$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$**

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi) = \emptyset$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\text{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\text{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\text{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \text{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\text{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\text{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi) = \emptyset$

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi)$ is undefined

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi)$ is undefined

Revisiting our Problematic Examples

$\{(c) \mid \forall d, t(\neg\textsc{Screening}(c, \text{'The Godfather'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\textsc{Screening}(c, \text{'The Godfather'}, d, t))$
- It is not safe because $rr(\varphi) = \emptyset \neq \{c\} = fr(\varphi)$

$\{(c, d) \mid$
$\exists t(\textsc{Screening}(c, \text{'The Piano'}, \text{'28.03.'}, t) \vee \textsc{Screening}(\text{'Event'}, \text{'Inception'}, d, t))\}$

- Is in SRNF
- It is not safe because $rr(\varphi) = \emptyset \neq \{c, d\} = fr(\varphi)$

$\{(c) \mid \forall d, t(\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))\}$

- The SRNF is $\neg\exists d, t(\neg\textsc{Screening}(c, \text{'When Harry met Sally'}, d, t))$
- The query is not safe because $rr(\varphi)$ is undefined, while $fr(\varphi) = \{c\}$

A Clever Way of Writing Difficult SQL Queries

- Formalize your query in safe relational calculus
- Transform your query into SRNF
- Transform SRNF into SQL
- Example: the SRNF of the Hitchcock-query is simple to translate into SQL
- TRC queries in SRNF are even closer to SQL syntax

Example: from TRC to SQL

Hitchcock query in TRC in SRNF

$$
\begin{aligned}
\{d{:}\mathrm{n} \mid \exists d1, d1{:}\mathrm{n}(&\textsc{Director}(d1) \wedge d{:}\mathrm{n} = d1{:}\mathrm{n} \wedge \\
&\neg\exists d2, d2{:}\mathrm{n}, d2{:}\mathrm{m}(\textsc{Director}(d2) \wedge d2{:}\mathrm{n} = d1{:}\mathrm{n} \wedge \\
&\neg\exists a1, a1{:}\mathrm{n}, a1{:}\mathrm{m}, a1{:}\mathrm{r}(\textsc{Actor}(a1) \wedge a1{:}\mathrm{n} = d2{:}\mathrm{n} \wedge a1{:}\mathrm{m} = d2{:}\mathrm{m} \wedge \\
&\neg\exists a2, a2{:}\mathrm{n}, a2{:}\mathrm{m}, a2{:}\mathrm{r}(\textsc{Actor}(a2) \wedge a2{:}\mathrm{n} = a1{:}\mathrm{n} \wedge a2{:}\mathrm{m} = a1{:}\mathrm{m} \wedge a2{:}\mathrm{r} \neq a1{:}\mathrm{r}))))\}
\end{aligned}
$$

Example: from TRC to SQL

Hitchcock query in TRC in SRNF

$$\{d\text{:n} \mid \exists d1, d1\text{:n}(\textsc{Director}(d1) \wedge d\text{:n} = d1\text{:n} \wedge$$
$$\neg\exists d2, d2\text{:n}, d2\text{:m}(\textsc{Director}(d2) \wedge d2\text{:n} = d1\text{:n} \wedge$$
$$\neg\exists a1, a1\text{:n}, a1\text{:m}, a1\text{:r}(\textsc{Actor}(a1) \wedge a1\text{:n} = d2\text{:n} \wedge a1\text{:m} = d2\text{:m} \wedge$$
$$\neg\exists a2, a2\text{:n}, a2\text{:m}, a2\text{:r}(\textsc{Actor}(a2) \wedge a2\text{:n} = a1\text{:n} \wedge a2\text{:m} = a1\text{:m} \wedge a2\text{:r} \neq a1\text{:r}))))\}$$

Transformed into SQL

```
SELECT d1.name FROM DIRECTOR d1
WHERE NOT EXISTS(
         SELECT d2.movie FROM DIRECTOR d2
         WHERE d2.name = d1.name AND NOT EXISTS(
                  SELECT a1.role FROM ACTOR a1
                  WHERE a1.name = d2.name AND a1.movie = d2.movie
                  AND NOT EXISTS(
                           SELECT a2.role FROM ACTOR a2
                           WHERE a2.role ≠ a1.role
                              AND a2.movie = a1.movie AND a2.name= a1.name)))
```

Summary

- Relational calculus offers a declarative way for specifying queries
  - $\hookrightarrow$ based on first-order logic
  - $\hookrightarrow$ close to important fragment of natural language
  - $\hookrightarrow$ two equivalent versions: Domain and Tuple Relational Calculus
- Some queries in Relational Calculus are problematic
  - $\hookrightarrow$ their evaluation depends on domains of attributes
- Safe Relational Calculus
  - $\hookrightarrow$ fragment of Relational Calculus
  - $\hookrightarrow$ domain-independent (no problems)
  - $\hookrightarrow$ equivalent to Relational Algebra
- Foundation of industry standard for query language (SQL)