

## Aufgabenblatt 7

### 7.1 Aufgabe

- a) The title of all movies that have been created before 1970.

**RA**  $\pi_{Movie.title} \sigma_{Movie.year < 1970} Movie$

**TRC**  $\{t.title \mid Movie(t) \wedge t.year < 1970\}$

**DRC**  $\{title \mid \exists id, year (Movie(id, title, year) \wedge year < 1970)\}$

- b) The name of all persons who participated in an “action” movie.

**RA**

$$\pi_{name}(\pi_{name}(Person \bowtie_{Person.id=actor.id} actor) \bowtie_{actor.movie=hasGenre.movie} \sigma_{hasGenre.Genre='action'} hasGenre)$$
  
*Vereinigung*  

$$(\pi_{name}(Person \bowtie_{Person.id=director.id} director) \bowtie_{director.movie=hasGenre.movie} \sigma_{hasGenre.Genre='action'} hasGenre)$$
  

$$)$$

**TRC**

$$\{p.name \mid Person(p) \wedge (\exists a(actor(a) \wedge a.person = p.id) \wedge (\exists hg(hasGenre(hg) \wedge a.movie = hg.movie) \wedge hg.genre = 'action')) \vee (\exists d(director(d) \wedge d.person = p.id) \wedge (\exists hg(hasGenre(hg) \wedge d.movie = hg.movie) \wedge hg.genre = 'action')))\}$$

## DRC

$$\begin{aligned} & \{name| \\ & \quad (\exists id, gender, birth(Person(id, name, gender, birth)) \\ & \quad \wedge \exists person, movie, role(actor(person, movie, role) \wedge id = person) \\ & \quad \wedge \exists movie, genre(hasGenre(movie, genre) \wedge genre = ('action'))) \\ & \quad \vee \\ & \quad (\exists id, gender, birth(Person(id, name, gender, birth)) \\ & \quad \wedge \exists person, movie(director(person, movie) \wedge id = person) \\ & \quad \wedge \exists movie, genre(hasGenre(movie, genre) \wedge genre = ('action'))) \\ & \quad \} \end{aligned}$$

- c) The name of all person who only played in the movie “The mighty Oracle” but apart from that in no other movie.

## RA

$$\begin{aligned} & \pi_{name}( \\ & \quad (\pi_{id,name}(Person \bowtie_{Person.id=actor.id} actor) \\ & \quad \bowtie_{actor.movie=movie.id} \sigma_{Movie.title='The mighty Oracle'} Movie) \\ & \quad \setminus \\ & \quad (\pi_{id,name}(Person \bowtie_{Person.id=actor.id} actor) \\ & \quad \bowtie_{actor.movie=movie.id} \sigma_{\neg(Movie.title='The mighty Oracle')} Movie) \\ & \quad ) \end{aligned}$$

## TRC

$$\begin{aligned} & \{t_1.name| \\ & \quad ((Person(t_1) \wedge \exists t_2(actor(t_2)) \wedge t_1.id = t_2.person \wedge \exists t_3( \\ & \quad Movie(t_3) \wedge t_2.movie = t_3.id \wedge t_3.name = 'The mighty Oracle')))) \\ & \quad \wedge \\ & \quad (\exists t_4(Person(t_4) \wedge \exists t_5(actor(t_5)) \wedge t_4.id = t_5.person \wedge \exists t_6( \\ & \quad Movie(t_6) \wedge t_5.movie = t_6.id \wedge \neg(t_3.name = 'The mighty Oracle')))) \\ & \quad \wedge \\ & \quad \neg(t_4.id = t_1.id)) \} \end{aligned}$$

## DRC

$$\begin{aligned}
 & \{name | \\
 & \quad (\exists id, gender, birth (Person(id, name, gender, birth) \\
 & \quad \wedge \\
 & \quad \exists person, movie, role (actor(person, movie, role) \wedge id = person) \\
 & \quad \wedge \\
 & \quad (\exists mid, title, year (Movie(mid, title, year) \\
 & \quad \wedge mid = movie \wedge title = 'The mighty Oracle')))) \\
 & \quad \wedge \\
 & \quad (\exists id1, gender1, birth1 (Person(id1, name, gender1, birth1)) \\
 & \quad \wedge \\
 & \quad \exists person1, movie1, role1 (actor(person1, movie1, role1) \wedge id1 = person1)) \\
 & \quad \wedge \\
 & \quad (\exists mid1, title1, year1 (Movie(mid1, title1, year1) \\
 & \quad \wedge mid1 = movie1 \wedge \neg(title1 = 'The mighty Oracle')))) \\
 & \quad \wedge \\
 & \quad \neg(id = id1))))) \}
 \end{aligned}$$

Ansonsten Komplikationen mit oberer id. (Vorsichtig mit Klammern umgehen, da sonst gebundene Variablen weiter unten nicht mehr sichtbar sind!)

## 7.2 Aufgabe

Express following statements in **natural** language:

- a)  $\{per | \exists mov, year ($   
*Movie*(mov, "The Power of Projection", year)  $\wedge$   
*actor*(per, mov, "forest ranger")  
 $)\} \setminus \{per | \exists mov, r (actor(per, mov, r) \wedge hasGenre(mov, "comedy"))\}$   
 Zeige alle Person die in "The Power of Projektion" mitspielen und die Rolle eines "Forest Rangers" haben ohne die Personen die in einer Komödie mitspielen.
- b)  $\{p.name | Person(p) \wedge \forall a ($   
*actor*(a)  $\wedge a.person = p.id) \rightarrow ($

$$\exists hg(hasGenre(hg) \wedge a.movie = hg.movie \wedge hg.genre = "drama")$$

$$)$$

$$)\}$$

Zeige alle Namen von Personen die keine Schauspieler sind oder in einem Film mitspielen, dessen Genre Drama ist.

### 7.3 Aufgabe

- a) What are unsafe queries and how can you avoid them?  
 Unsichere Abfragen sind Abfragen die möglicherweise eine unbegrenzte Anzahl von Tupeln zurückgeben. Vermeiden lassen sich unsichere Abfragen nur durch die sogenannte "Geschlossene-Welt-Annahme". Diese Annahme begrenzt die Tupel die für Tupelvariablen ersetzt werden können auf die, die in den aktuell betrachteten Relationen auch wirklich vorkommen.
- b) What is relational completeness?  
 Es gibt drei Abfrageparadigmen: relational Algebra, tuple relational calculus (TRC) und domain relational calculus (DRC). Eine Abfrage die in einem der drei Kalküle formuliert wurde, kann auch in jedes der drei Kalküle umgewandelt werden. Diese Eigenschaft nennt man relational komplett. Jede andere Abfragesprache die in eine von den drei Kalkülen abgebildet werden kann ist ebenfalls relational komplett.
- c) What is the difference between a formula and an atom?  
 Eine Formel besteht aus mindestens einem Atom. Ein Atom ist automatisch auch eine Formel, jedoch kann eine Formel auch aus mehreren Atomen und Operatoren bestehen.