Lista 1

Weronika Jakimowicz

29.02.2024

Zadanie 1.
Relacja P(A, B) zawiera p krotek, a relacja S(B, C) zawiera s krotek. Nic nie wiadomo na temat kluczy relacji. Dla każdego z poniższych wyrażeń wylicz (w zależności od p i s) jaka może być minimalna i maksymalna liczba zwracanych krotek.

a) $\mathsf{P} \cup \rho_{\mathsf{S(A,B)}}\mathsf{S}$

b) $\pi_{\mathsf{A,C}}(\mathsf{P} \bowtie \mathsf{S})$

c) $\pi_{\mathsf{B}}(\mathsf{P}) \setminus (\pi_{\mathsf{B}}(\mathsf{P}) \setminus \pi_{\mathsf{B}}(\mathsf{S}))$

d) $(\mathsf{S} \bowtie \mathsf{S}) \bowtie \mathsf{S}$

e) $\sigma_{\mathsf{A<B}}(\mathsf{P}) \cup \sigma_{\mathsf{A>B}}(\mathsf{P})$

Rozwiązanie.
a) $\mathsf{P} \cup \rho_{\mathsf{S(A,B)}}\mathsf{S}$

Operacja $\rho_{\mathsf{S(A,B)}}\mathsf{S}$ bierze całą relację S(B, C) i podmienia nazwy jej kolumn na A, B, resztę pozostawiając bez zmian.

min = max(s, k)

W takim razie, najmniej krotek w sumie mnogościowej P i S(A, B) będzie jeśli cała jedna baza ma te same informacje co druga, np. gdy mamy takie, troszkę bezsensowne, bazy:

P	
A=nr	*B=imię*
1	Weles
2	Kycia

S	
B=imię	*C=właściciel*
2	Kycia

W takim razie najmniejsza ilość krotek to max(s, k).

max = s + p

Najwięcej krotek będzie, gdy bazy po przemianowaniu będą całkowicie rozłączne, np.:

P	
A=owoc	*B=ilość*
banan	1
winogran	7
jabłka	3

S	
B=ilość	*C=zakup*
5	kartofli

b) $\pi_{A,C}(P \bowtie S)$

min = 0

Jeśli w B nie ma wspólnych wpisów, to P ⋈ S będzie pusty, czyli najmniejsza ilość krotek w wyniku to 0:

P	
A=nr	*B=imię*
1	Weles
2	Kycia

S	
B=imię	*C=właściciel*
Stefan	Ania

P ⋈ S		
A=nr	*B=imię*	*C=właściciel*
-	-	-

max = p · s

Jeśli B ma tylko jeden element i w P i w S, to wtedy ⋈ zachowuje się jak produkt kartezjański

P	
A=nr	*B=imię*
1	Weles
2	Weles

S	
B=imię	*C=właściciel*
Weles	Kycia
Weles	Mirek
Weles	Ronia

P ⋈ S		
A=nr	*B=imię*	*C=właściciel*
1	Weles	Kycia
1	Weles	Ronia
1	Weles	Mirek
2	Weles	Kycia
2	Weles	Mirek
2	Weles	Ronia

i wtedy krotek jest tyle, ile elementów w większej relacji, czyli s · p.

c) $\pi_B(P) \setminus (\pi_B(P) \setminus \pi_B(S))$

min = 0

Tutaj minimum to 0, jeśli zapisy w kolumnie B relacji S są całkiem rozłączne z zapisami w kolumnie B relacji P:

P	
A=nr	*B=imię*
1	Weles
2	Kycia

S	
B=imię	*C=właściciel*
Doruś	Halina
Dziunia	Grażyna
Stefan	Agata

wtedy $\pi_B(P) \setminus \pi_B(S) = \pi_B(P)$.

max = p

Natomiast najwięcej krotek jakie możemy dostać to p, jeśli wyrazy w kolumnie B tablicy S są takie same jak wyrazy z kolumny B tablicy P i w dodatku każda krotka z P jest unikalna:

P	
A=nr	*B=imię*
1	Weles
2	Kycia

S	
B=imię	*C=właściciel*
Weles	Halina
Kycia	Grażyna

d) $(S \bowtie S) \bowtie S$

min = max = s

Tutaj zawsze będzie s krotek, bo $S \bowtie S = S$.

e) $\sigma_{A<B}(P) \cup \sigma A > B(P)$

min = 0

Tutaj najmniejsza ilość krotek w wyniku jest wtedy, gdy A = B zawsze w tabeli P, np.

P	
A=nr w dzienniku	*B=ocena*
1	1
2	2
3	3

ponieważ $\sigma_{A>B}(P)$ oraz $\sigma_{A<B}(P)$ są zbiorami pustymi, to ich suma też jest pusta.

max = p

Natomiast, jeśli nie ma krotek, gdzie elementy są takie same, to dostaniemy p sztuk krotek:

P	
A=nr w dzienniku	*B=ocena*
1	2
2	3
3	6

Zadanie 2.
Czy operator różnicy \ da się wyrazić za pomocę algebry relacji z operatorami $\pi, \sigma, \rho, \times, \cup$? Przyjmijmy, że warunki F są formułami zbudowanymi przy użyciu koniunkcji, alternatywy oraz zawierają wyłącznie atomy postaci $Atr_1 = const$ lub $Atr_1 = Atr_2$, gdzie Atr_1, Atr_2 są atrybutami, a const stałą odpowiedniego typu. Czy odpowiedź na pytanie zmieni się, jeśli w warunkach dopuścimy negację? *Wskazówka: poszukaj pewnej charakterystycznej cechy, którą mają wszystkie zapytania wyrażane za pomocą $\pi, \sigma, \rho, \times, \cup$, a której nie musi mieć zapytanie wyrażone z użyciem* \.

Rozwiązanie.
Dla operatorów, które nie są \, jeśli $A \subseteq B$, to op $A \subseteq$ op B. Stąd, nie da się zapisać \ przy użyciu

pozostałych, nie ważne czy mamy negację czy nie mamy.

Zadanie 3.
X, Y i Z są relacjami zawierającymi pojedynczą kolumnę o nazwie A. STudent ma napisać wyrażenie algebry relacji wyliczające wartość $X \cap (Y \cup Z)$ nie używając operatorów sumy i przekroju relacji. W bazie danych rozwiązań zadań z poprzednich edycji kursu znalazł następujące wyrażenie:

$$\pi_A(\sigma_{A=A_Y \vee A=A_Z}(X \times \rho_{Y(A_Y)} Y \times \rho_{Z(A_Z)} Z))$$

Czy powinien użyć tego rozwiązania? Jeśli zapytanie jest poprawne, to uzasadnij to, jeśli nie, to zastanów się czy i jak można je poprawić.

Rozwiązanie.
Zacznijmy od napisania, kiedy $x \in X \cap (Y \cup Z)$:

$$x \in X \cap (Y \cup Z) \iff x \in X \wedge (x \in Y \vee x \in Z).$$

W rozwiązaniu w bazie zaczynamy od krotek (x, y, z), gdzie $x \in X$, $y \in Y$ i $z \in Z$. Potem wybieramy z nich te elementy, dla których $y \in X$ lub $z \in X$. Po rzutowaniu na pierwszą współrzędną dostajemy więc elementy $x \in X$, dla których $x \in Y$ lub $x \in Z$, czyli jest to rozwiązanie poprawne.

Zadanie 4.

Rozwiązanie.

```
pi movies.name, directors.last_name, movies_genres.genre (
  sigma (movies.id = movies_genres.movie_id and movies.year <
     1960) (
        movies x
        sigma (movies_genres.movie_id = movies_directors.
           movie_id) (
                movies_genres x
                (
                        sigma (
                                movies_directors.director_id =
                                    directors.id) (
                                        movies_directors x
                                           directors)
                        )
                )
        )
)
```

```
pi actors.first_name, actors.last_name (
  sigma (directors.last_name != 'Tarantino' and directors.
     first_name != 'Quentin') (
    (sigma movies_directors.movie_id = roles.movie_id (
```

```
            (sigma directors.id = movies_directors.director_id
                (
          directors x movies_directors
          )
        ) x
            (sigma roles.actor_id = actors.id (
          roles x actors
          )
        )
      )
    )
  )
)
```

```
pi actors.first_name, actors.last_name (sigma actors.id =
   actors.ii (
                 actors x (
                         (pi ii (rho ii <- id actors)) -
                         pi roles.actor_id (
                                 sigma roles.role != roles.r
                                 (
                                 sigma roles.actor_id = roles.
                                    act_id (
                                         roles x
                                         (rho act_id <-
                                            actor_id, mv_id <-
                                            movie_id, r <- role
                                             roles)
                                 )
                                 )
                         )
                 )
        )
)
```

```
pi movies.name
  (
    sigma movies.id = movies_genres.mv_id ( movies x
    ( sigma movies_genres.genre = 'Drama' and movies_genres.g
       = 'Sci-Fi' ( sigma
      movies_genres.mv_id = movies_genres.movie_id (
         movies_genres x
        (rho mv_id <- movie_id, g <- genre movies_genres)
      )
      )
    )
  )
  )
```

```
pi actors.last_name ( sigma roles.ac_id = actors.id (actors x
(
sigma roles.ac_id = roles.actor_id and roles.role = roles.r
   and roles.mv_id != roles.movie_id (
        roles x
        (rho r <- role, ac_id <- actor_id, mv_id <- movie_id
           roles)
)
)
)
)
```

```
pi directors.last_name (
sigma directors.id = movies_directors.director_id (
directors x
(
pi movies_directors.director_id movies_directors -
pi movies_directors.director_id (
        sigma movies_genres.genre = 'Horror'
        (
                sigma movies_directors.movie_id =
                   movies_genres.movie_id (movies_directors x
                   movies_genres)
        )
)
)
)
)
```

Tutaj krzyczy błąd, ale nie ma reżyserów, którzy nakręcili film bez kobiet i to chyba krzyczy na iloczyn kartezjański z pustą listą?

```
pi directors.last_name (sigma directors.id = movies_directors.
   director_id ((pi movies_directors.director_id
   movies_directors) -
(pi movies_directors.director_id
(sigma actors.gender = 'F' and actors.id = roles.actor_id (
actors x
(sigma movies_directors.movie_id = roles.movie_id (
   movies_directors x roles))
))
) x directors)
)
```

Zadanie 5.

Rozwiązanie.

a) tutaj problemem jest, że czasem aktor gra w więcej niż jednym filmie, jak to się dzieje z Michaelem

Naprawa:

```
tau genders desc
gamma actors.first_name;
count(acotrs.gender) -> genders (pi first_name, id, gender (
   actors join actors.id=roles.actor_id roles))
```

b)

```
gamma directors.last_name; count(roles.actor_id) -> nr
  (pi directors.last_name, directors.id, roles.actor_id
  (roles join roles.movie_id=movies_directors.movie_id
    (directors join directors.id=movies_directors.director_id
       movies_directors)
  )
  )
```