一、OWA & CWA

Consider the database instance $\mathcal{D}_{\text{music}}$ given by:

StudioAlbum(Fantasy) StudioAlbum(The_Eight_Dimensions) StudioAlbum(Common_Jasmin_Orange)

DebutAlbum(Jay) LiveAlbum(2004_Incomparable_Concert)

EP(Hidden_Track) EP(Initial_D) SoundtrackAlbum(Secret) CompilationAlbum(Together)

Song(Herbalist_Manual) Song(Elimination)

Singer(Jay_Chou) Singer(Eason_Chan)

Composer(Jay_Chou) Lyricist(Vincent_Fang) Police(Black_Cat)

hasFriend(Jay_Chou, Vincent_Fang) hasFriend(Jay_Chou, Will_Liu)

releases(Jay_Chou, Jay) releases(Jay_Chou, Fantasy) releases(Jolin_Tsai, Together)

sings(Eason_Chan, Elimination) sings(Jolin_Tsai, Rewind) sings(Jay_Chou, Herbalist_Manual)

writesMusicFor(Jay_Chou, Elimination) writesMusicFor(Jay_Chou, Rewind) writesMusicFor(Jay_Chou, Herbalist_Manual) writesMusicFor(Ta-yu_Lo, Pearl_of_the_Orient)

writesLyricsFor(Jay_Chou, Elimination) writesLyricsFor(Vincent_Fang, Rewind)

writesLyricsFor(Vincent_Fang, Herbalist_Manual) DancesWith(Will_Liu, Herbalist_Manual)

Consider each of the following Boolean queries F (in DL notation).

- Write those Boolean queries (marked in red) in first-order logic (FOL) notation. (Note that for many queries there is no difference between DL notation and FOL notation).

(1)DL to FOL

(a)$Album(Fantasy)$

$$Album(Fantasy)$$

(f)$\neg$ StudioAlbum $\sqcup \neg$ LiveAlbum(2004_Incomparable_Concert)

$$\neg\ \text{StudioAlbum}(\text{2004\_Incomparable\_Concert})\ \vee \neg\ \text{LiveAlbum}(\text{2004\_Incomparable\_Concert})\ )$$

(h)$\exists$ hasFriend. T(Jay_Chou)

$$\exists y.\ hasFriend(Jay\_Chou, y)$$

(i)$\exists$ hasFriend. $\exists$ dancesWith.Song(Jay_Chou)

$$\exists y \exists z.\ (hasFriend(Jay\_Chou, y) \wedge dancesWith(y, z) \wedge Song(z))$$

(k)$\exists$ hasFriend.{Jay_Chou } (Vincent_Fang) 多个取值怎么表示?

$$hasFriend(Vincent\_Fang, Jay\_Chou)$$

- Query answering under closed world assumption: check for each Boolean F whether the answer to the query F given by $\mathcal{D}_{\text{music}}$ is "Yes" or "No".
- Query answering under open world assumption: check for each Boolean query F whether the certain answer to F given by $\mathcal{D}_{\text{music}}$ is "Yes", "No", or "Don't know".

(2)Boolean-CWA

(a)NO $Album(Fantasy)$
(b)Yes $StudioAlbum(The\_Eight\_Dimensions)$
(c)No $LiveAlbum(Common\_Jasmin\_Orange)$
(d)Yes $\neg LiveAlbum(Common\_Jasmin\_Orange)$
(e)Yes $\neg \text{EP}(secret)$
(f)Yes $\neg StudioAlbum \sqcup \neg LiveAlbum(2004\_Incomparable\_Concert)$

(g)Yes $\neg StudioAlbum \sqcup \neg LiveAlbum(Eason\_Chan)$

(h)Yes $\exists$ hasFriend. T(Jay_Chou)

(i)Yes $\exists hasFriend. \exists dancesWith. Song(Jay\_Chou)$
(j)No $\exists hasFriend. Composer(Jay\_Chou)$
(k)No $\exists hasFriend. \{Jay\_Chou\}(Vincent\_Fang)$
(l)No $DebutAlbum(2004\_Incomparable\_Concert)$

(m)No $Song(Rewind)$
(n)Yes $Singer(Jay\_Chou)$
(o)No $Singer(Jolin\_Tsai)$
(p)No $Lyricist(Jay\_Chou)$
(q)Yes $Composer(Jay\_Chou)$
(r)No $Composer(Ta-yu\_Lo)$
(s)No $Police(Jay\_Chou)$
(t)No $Police(Jolin\_Tsai)$
(u)Yes $\neg Singer-SongWriter \sqcup \neg Police(Vincent\_Fang)$
(v)Yes $\neg Singer-SongWriter \sqcup \neg Police(Ta-yu\_Lo)$
(w)No $Singer-SongWriter(Jay\_Chou)$
(x)No $Singer-SongWriter(Jolin\_Tsai)$
(y)Yes $\neg SongWriter(Vincent\_Fang)$
(z)Yes $\neg Dancer(Will\_Liu)$

(3)Boolean-OWA

(a)Don't know $Album(Fantasy)$
(b)Yes $StudioAlbum(The\_Eight\_Dimensions)$
(c)Don't know $LiveAlbum(Common\_Jasmin\_Orange)$
(d)Don't know $\neg LiveAlbum(Common\_Jasmin\_Orange)$
(e)Don't know $\neg \mathrm{EP}(secret)$
(f)Don't know $\neg StudioAlbum \sqcup \neg LiveAlbum(2004\_Incomparable\_Concert)$

(g)Don't know $\neg StudioAlbum \sqcup \neg LiveAlbum(Eason\_Chan)$

(h)Yes $\exists\ \mathrm{hasFriend}.\ \mathrm{T}(\mathrm{Jay\_Chou})$

(i)Yes $\exists hasFriend.\exists dancesWith.Song(Jay\_Chou)$
(j)Don't know $\exists hasFriend.Composer(Jay\_Chou)$
(k)Don't know $\exists hasFriend.\{Jay\_Chou\}(Vincent\_Fang)$
(l)Don't know $DebutAlbum(2004\_Incomparable\_Concert)$
(m)Don't know $Song(Rewind)$
(n)Yes $Singer(Jay\_Chou)$
(o)Don't know $Singer(Jolin\_Tsai)$
(p)Don't know $Lyricist(Jay\_Chou)$
(q)Yes $Composer(Jay\_Chou)$
(r)Don't know $Composer(Ta-yu\_Lo)$
(s)Don't know $Police(Jay\_Chou)$
(t)Don't know $Police(Jolin\_Tsai)$
(u)Don't know $\neg Singer-SongWriter \sqcup \neg Police(Vincent\_Fang)$
(v)Don't know $\neg Singer-SongWriter \sqcup \neg Police(Ta-yu\_Lo)$
(w)Don't know $Singer-SongWriter(Jay\_Chou)$
(x)Don't know $Singer-SongWriter(Jolin\_Tsai)$
(y)Don't know $\neg SongWriter(Vincent\_Fang)$
(z)Don't know $\neg Dancer(Will\_Liu)$

Consider the following non-Boolean queries F_i ($1 \leq i \leq 4$):

(a) $F_1(x) = \text{Singer}(x)$

(b) $F_2(x) = \neg\text{Singer}(x)$

(c) $F_3(x,y) = \text{hasFriend}(x,y)$

(d) $F_4(x) = (\text{Lyricist}(x) \vee \text{Composer}(x)) \wedge \neg\text{releases}(x, \text{Jay})$

For each query F_i, give

- for closed world assumption: answer(F_i,$\mathcal{D}_{\text{music}}$);
- for open world assumption: certanswer(F_i,$\mathcal{D}_{\text{music}}$).

(4)non-Boolean-CWA

$answer(F_1, D_{music}) = \{Jay\_Chou, Eason\_Chan\}$

$answer(F_2, D_{music}) = Ind(D_{music})/\{Jay\_Chou, Eason\_Chan\}$

$answer(F_3, D_{music}) = \{(Jay\_Chou, Vincent\_Fang), (Jay\_Chou, Will\_Liu)\}$

$answer(F_4, D_{music}) = \{Vincent\_Fang\}$

(5)non-Boolean-OWA

$certanswer(F_1, D_{music}) = \{Jay\_Chou, Eason\_Chan\}$

$certanswer(F_2, D_{music}) = \emptyset$

$certanswer(F_3, D_{music}) = \{(Jay\_Chou, Vincent\_Fang), (Jay\_Chou, Will\_Liu)\}$

$certanswer(F_4, D_{music}) = \emptyset$

二、Querying with TBox

Following Question 1, consider now the TBox $\mathcal{T}$ given as:

$$\begin{aligned}
\text{StudioAlbum} &\sqsubseteq \text{Album}\\
\text{LiveAlbum} &\sqsubseteq \text{Album}\\
\text{StudioAlbum} \sqcap \text{LiveAlbum} &\sqsubseteq \bot\\
\text{EP} \sqcap \text{LiveAlbum} \sqcap \text{SoundTrackAlbum} &\sqsubseteq \bot\\
\text{DebutAlbum} &\sqsubseteq \text{StudioAlbum}\\
\text{Album} &\equiv \exists\text{hasTrack.Song}\\
\text{Singer} &\equiv \exists\text{releases.Album} \sqcap \exists\text{sings.Song}\\
\exists\text{releases}^{-}.\text{Police} \sqcap \text{Album} &\sqsubseteq \bot\\
\text{Lyricist} &\equiv \exists\text{writesLyricsFor.Song}\\
\text{Composer} &\sqsubseteq \exists\text{writesMusicFor.Song}\\
\text{SongWriter} &\sqsubseteq \text{Lyricist} \sqcap \text{Composer}\\
\text{Singer-SongWriter} &\sqsubseteq \text{Singer} \sqcap \text{SongWriter}\\
\exists\text{sings}^{-}.\top &\sqsubseteq \text{Song}\\
\exists\text{writesLyricsFor}^{-}.\top &\sqsubseteq \text{Song}
\end{aligned}$$

- Re-consider the Boolean queries F given in Question 1. Compute the certain answers in the context of $\mathcal{D}_{\text{music}}$, and in the context of $(\mathcal{T}, \mathcal{D}_{\text{music}})$.

(1)

注意陷阱

(a)Yes $Album(Fantasy)$
(b)Yes $StudioAlbum(The\_Eight\_Dimensions)$
(c)No $LiveAlbum(Common\_Jasmin\_Orange)$
(d)Yes $\neg LiveAlbum(Common\_Jasmin\_Orange)$
(e)Don't know $\neg\text{EP}(secret)$
(f)Yes $\neg StudioAlbum \sqcup \neg LiveAlbum(2004\_Incomparable\_Concert)$

(g)Yes $\neg StudioAlbum \sqcup \neg LiveAlbum(Eason\_Chan)$

(h)Yes $\exists$ hasFriend. T(Jay_Chou)

(i)Yes $\exists hasFriend.\exists dancesWith.Song(Jay\_Chou)$
(j)Don't know $\exists hasFriend.Composer(Jay\_Chou)$
(k)Don't know $\exists hasFriend.\{Jay\_Chou\}(Vincent\_Fang)$
(l)No $DebutAlbum(2004\_Incomparable\_Concert)$
(m)Yes $Song(Rewind)$
(n)Yes $Singer(Jay\_Chou)$
(o)Don't know $Singer(Jolin\_Tsai)$
(p)Yes $Lyricist(Jay\_Chou)$
(q)Yes $Composer(Jay\_Chou)$
(r)Don't know $Composer(Ta-yu\_Lo)$
(s)No $Police(Jay\_Chou)$
(t)Don't know $Police(Jolin\_Tsai)$
(u)Don't know $\neg Singer-SongWriter \sqcup \neg Police(Vincent\_Fang)$
(v)Don't know $\neg Singer-SongWriter \sqcup \neg Police(Ta-yu\_Lo)$
(w)Don't know $Singer-SongWriter(Jay\_Chou)$

(x)Don't know $Singer - SongWriter(Jolin\_Tsai)$
(y)Don't know $\neg SongWriter(Vincent\_Fang)$
(z)Don't know $\neg Dancer(Will\_Liu)$

- The addition of the TBox $\mathcal{T}$ to the database instance $\mathcal{D}_{\text{music}}$ allows one to draw new conclusions from $\mathcal{D}_{\text{music}}$, and may render some of the data (ABox assertion) α in $\mathcal{D}_{\text{music}}$ redundant, i.e., $(\mathcal{T}, \mathcal{D} \setminus \{\alpha\}) \models \mathcal{D}$. Can you identify all such assertions α?

(2)

Find all assertions that after rendering them and they can still be reasoned by remaining assertions:

$Singer(Jay\_Chou)$

$Lyricist(Vincent\_Fang)$

$Song(Herbalist\_Manual)$

$Song(Elimination)$

三、 Computing $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ in $\mathcal{EL}$

Consider the $\mathcal{EL}$ TBox $\mathcal{T}$:

$$
\begin{aligned}
\text{Guitarist} &\sqsubseteq \exists \text{plays\_for.RockBand} \\
\text{Bassist} &\sqsubseteq \exists \text{plays\_for.RockBand} \\
\text{Drummer} &\sqsubseteq \exists \text{plays\_for.RockBand} \\
\text{RockBand} &\sqsubseteq \exists \text{managed\_by.Manager} \\
\text{Manager} &\sqsubseteq \text{Employee} \\
\text{Manager} &\sqsubseteq \exists \text{managed\_by.Manager}
\end{aligned}
$$

and the ABox $\mathcal{A}$:

$$
\begin{aligned}
\text{Guitarist(John\_Lennon)} &\quad \text{Bassist(Paul\_McCartney)} \\
\text{Drummer(Ringo\_Starr)} &\quad \text{RockBand(Beatles)} \\
\text{managed\_by(Beatles, Brian\_Epstein)} &
\end{aligned}
$$

(1)Compute $\mathcal{I}_{\mathcal{T},\mathcal{A}}$

initial assignment:初始为空的r关系也要给个初始化空集

$$
\begin{aligned}
S(d_{Guitarist}) &= \{Guitarist\} \\
S(d_{Bassist}) &= \{Bassist\} \\
S(d_{Drummer}) &= \{Drummer\} \\
S(d_{RockBand}) &= \{RockBand\} \\
S(d_{Manager}) &= \{Manager\} \\
S(d_{Employee}) &= \{Employee\} \\
R(managed\_by) &= \{(Beatles, Brain\_Epstein)\} \\
S(John\_Lennon) &= \{Guitarist\} \\
S(Paul\_McCartney) &= \{Bassist\} \\
S(Ringo\_starr) &= \{Drummer\} \\
S(Beatles) &= \{RockBand\} \\
S(Brain\_Epstein) &= \emptyset \\
R(plays\_for) &= \emptyset
\end{aligned}
$$

applications of (simpleR),(rightR):

- Update S using (simpleR):

$$S(d_{Manager}) = \{Manager, Employee\}$$

- Update R using (rightR):

$$R(plays\_for) = \{(d_{Guitarist}, d_{RockBand}), (d_{Bassist}, d_{RockBand}), (d_{Drummer}, d_{RockBand})\}$$

- Update R using (rightR):

$$R(managed\_by) = \{(Beatles, Brain\_Epstein), (d_{RockBand}, d_{Manager}), (d_{Manager}, d_{Manager})\}$$

- Update R using (rightR):

$$R(plays\_for) = \{(d_{Guitarist}, d_{RockBand}), (d_{Bassist}, d_{RockBand}), (d_{Drummer}, d_{RockBand}) \\ , (John\_Lennon, d_{RockBand}), (Paul\_McCartney, d_{RockBand}), (Ringo\_starr, d_{RockBand})\}$$

- Update R using (rightR):

$$R(managed\_by) = \{(Beatles, Brain\_Epstein), (d_{RockBand}, d_{Manager}), (d_{Manager}, d_{Manager}), (Beatles, d_{Manager})\}$$

So final assignment:

$$\begin{aligned}
S(d_{Guitarist}) &= \{Guitarist\} \\
S(d_{Bassist}) &= \{Bassist\} \\
S(d_{Drummer}) &= \{Drummer\} \\
S(d_{RockBand}) &= \{RockBand\} \\
S(d_{Manager}) &= \{Manager, Employee\} \\
S(d_{Employee}) &= \{Employee\} \\
R(managed_by) &= \{(Beatles, Brain_Epstein), (d_{RockBand}, d_{Manager}), (d_{Manager}, d_{Manager}), (Beatles, d_{Manager})\} \\
R(plays_for) &= \{(d_{Guitarist}, d_{RockBand}), (d_{Bassist}, d_{RockBand}), (d_{Drummer}, d_{RockBand}) \\
&\quad , (John_Lennon, d_{RockBand}), (Paul_McCartney, d_{RockBand}), (Ringo_starr, d_{RockBand})\} \\
S(John_Lennon) &= \{Guitarist\} \\
S(Paul_McCartney) &= \{Bassist\} \\
S(Ringo_starr) &= \{Drummer\} \\
S(Beatles) &= \{RockBand\} \\
S(Brain_Epstein) &= \emptyset
\end{aligned}$$

$\mathcal{I}_{\mathcal{T},\mathcal{A}}$:

- $\Delta^{\mathcal{I}}_{\mathcal{T},\mathcal{A}} = \{d_{Guitarist}, d_{Bassist}, d_{Drummer}, d_{RockBand}, d_{Manager}, d_{Employee}$
  $, John\_Lennon, Paul\_McCartney, Ringo\_starr, Beatles, Brain\_Epstein\}$
- $Guitarist^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{Guitarist}, John\_Lennon\}$
- $Bassist^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{Bassist}, Paul\_McCartney\}$
- $Drummer^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{Drummer}, Ringo\_starr\}$
- $RockBand^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{RockBand}, Beatles\}$
- $Manager^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{Manager}\}$
- $Employee^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{d_{Employee}, d_{Manager}\}$
- $plays\_for^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{(d_{Guitarist}, d_{RockBand}), (d_{Bassist}, d_{RockBand}), (d_{Drummer}, d_{RockBand})$
  $, (John\_Lennon, d_{RockBand}), (Paul\_McCartney, d_{RockBand}), (Ringo\_starr, d_{RockBand})\}$
- $managed\_by^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{(Beatles, Brain\_Epstein), (d_{RockBand}, d_{Manager}), (d_{Manager}, d_{Manager}), (Beatles, d_{Manager})\}$

- For $\mathcal{EL}$ concept queries, we know that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ gives the answer "Yes" iff $(\mathcal{T}, \mathcal{A})$ gives the certain answer "Yes". Check this for the following queries:
  - ∃plays_for.RockBand(John_Lennon);
  - ∃managed_by.Manager(Paul_McCartney);
  - ∃plays_for.∃managed_by.Manager(Ringo_Starr).

(2)

- Yes
- Don't know ($(\mathcal{T}, \mathcal{A})$ doesn't give the certain answer "Yes")
- Yes

- For more complex queries, $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ can give the answer "Yes" even if $(\mathcal{T}, \mathcal{A})$ does not give the certain answer "Yes". Check this for:
  - $F(x, y) = \exists z.(\text{plays\_for}(x, z) \wedge \text{plays\_for}(y, z))$.
  - $F = \exists x.\text{managed\_by}(x, x)$.

(3)

- $F(x,y)=\exists z\cdot(\text{ plays\_for }(x,z)\wedge\text{ plays\_for }(y,z))$

  Both $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ and $(\mathcal{T},\mathcal{A})$ give the certain answer:

$$certanswer(F(x,y),\mathcal{I}_{\mathcal{T},\mathcal{A}})=\{(John\_Lennon,Paul\_McCartney),(John\_Lennon,Ringo\_starr),(Paul\_McCartney,Ringo\_starr)\\,(Paul\_McCartney,John\_Lennon),(Ringo\_starr,John\_Lennon),(Ringo\_starr,Paul\_McCartney)\\,(John\_Lennon,John\_Lennon),(Paul\_McCartney,Paul\_McCartney),(Ringo\_starr,Ringo\_starr)\}$$

$$certanswer(F(x,y),(\mathcal{T},\mathcal{A}))=\{(John\_Lennon,John\_Lennon),(Paul\_McCartney,Paul\_McCartney),(Ringo\_starr,Ringo\_star$$

- $F=\exists x.\text{ managed\_by }(x,x)$

  The certain answer to **F** in $(\mathcal{T},\mathcal{A})$ is `Don't know`

  But if we modify the $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ computed in **(1)** by adding:

$$Manager^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}=\{d_{Manager},some\_manageer\}$$

  Then $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ can give the answer "Yes" enen if $(\mathcal{T},\mathcal{A})$ does not give the certain answer "Yes"

四、Conjunctive queries

Consider the following database $\mathcal{D}$ consisting of the following tables:

Person:		Enrollment:		Attendance:		Course:	
ID	Name	StudentID	Since	StudentID	CourseID	ID	Title
2001	Jay_Chou	2002	2020	2001	30000160	30000160	KR&P
2002	Jolin_Tsai	2003	2021	2002	30000160	30000180	PR&CV
2003	Stefanie_Sun	2004	2020	2002	30000170	30000170	NLP
2004	Ta-yu_Lo			2003	30000180		

- Define the finite first-order interpretation $\mathcal{I}_{\mathcal{D}}$ corresponding to $\mathcal{D}$.

(1)Interpretation

- $\Delta^{\mathcal{I}_{\mathcal{D}}}=\{2001,2002,2003,2004,30000160,30000170,30000180,2020,2021\\,Jay\_Chou,Jolin\_Tsai,Stefanie\_Sun,Ta-yu\_Lo,KR\&P,PR\&CV,NLP\}$
- $Person\_ID^{\mathcal{I}_{\mathcal{D}}}=\{2001,2002,2003,2004\}$
- $Enrollment\_StedentID^{\mathcal{I}_{\mathcal{D}}}=\{2002,2003,2004\}$
- $Attendance\_StedentID^{\mathcal{I}_{\mathcal{D}}}=\{2001,2002,2003\}$
- $Course\_ID^{\mathcal{I}_{\mathcal{D}}}=\{30000160,30000170,30000180\}$
- $Name^{\mathcal{I}_{\mathcal{D}}}=\{Jay\_Chou,Jolin\_Tsai,Stefanie\_Sun,Ta-yu\_Lo\}$
- $Since^{\mathcal{I}_{\mathcal{D}}}=\{2020,2021\}$
- $Title^{\mathcal{I}_{\mathcal{D}}}=\{KR\&P,PR\&CV,NLP\}$
- $Person\_Nmae^{\mathcal{I}_{\mathcal{D}}}=\{(2001,Jay\_Chou),(2002,Jolin\_Tsai),(2003,Stefanie\_Sun),(2004,Ta-yu\_Lo)\}$
- $Enrollment\_Since^{\mathcal{I}_{\mathcal{D}}}=\{(2002,2020),(2003,2021),(2004,2020)\}$
- $Attendence\_Course^{\mathcal{I}_{\mathcal{D}}}=\{(2001,30000160),(2002,30000160),(2002,30000170),(2003,30000180)\}$
- $Course\_Title^{\mathcal{I}_{\mathcal{D}}}=\{(30000160,KR\&P),(30000180,PR\&CV),(30000170,NLP)\}$

- Reformulate each of the following SQL queries Q into first-order queries f_Q, and identify which of them are conjunctive queries.
- Answer Q in the context of $\mathcal{D}$ and f_Q in the context of $\mathcal{I}_{\mathcal{D}}$.

(a) SELECT ∗ FROM Person

(b) SELECT Person.Name FROM Person, Attendance, Course
WHERE Person.ID = Attendance.PersonID
AND Course.ID = Attendance.CourseID
AND Course.Title = "KR&P"

(c) SELECT Person.Name FROM Person, Enrollment
WHERE Person.ID = Enrollment.PersonID
AND NOT EXISTS (
SELECT ∗ FROM Attendance
WHERE Person.ID = Attendance.PersonID)

(2)Reformulate SQL queries into first-order queries

(a)yes

$$f_Q((x,y)) = Person\_Name(x,y)$$

(b)yes

$$f_Q(x) = \exists y \exists z.\,(Person\_Name(y,x) \wedge Attendence\_Course(y,z) \wedge Course\_Title(z, KR\&P))$$

(c)no

$$f_Q(x) = \exists y \exists z.\,(Person\_Name(y,x) \wedge Enrollment\_Since(y,z)) \wedge \forall w \neg Attendence\_Course(y,w))$$

(3)Answer

Answer Q in the context of $\mathcal{D}$ and answer f_Q in the context of $I_{\mathcal{D}} are always the same$

(a)

$$f_Q((x,y)) = \{(2001, Jay\_Chou), (2002, Jolin\_Tsai), (2003, Stefanie\_Sun), (2004, Ta-yu\_Lo)\}$$

(b)

$$f_Q(x) = \{Jay\_Chou, Jolin\_Tsai\}$$

(c)

$$f_Q(x) = \{Ta-yu\_Lo\}$$

五、Conjunctive queries in different contexts

Question 5. Certain answers in different contexts
Consider the following $\mathcal{ALC}$ knowledge base $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ with:

$$\mathcal{T} := \{X \sqsubseteq Y, Y \sqsubseteq \exists r.X, X \sqsubseteq \forall r.Y, \forall r.X \sqsubseteq Y, W \equiv \neg V, \exists r.Y \sqsubseteq \neg V\}$$
$$\mathcal{A} := \{(\text{Jay\_Chou, Jolin\_Tsai}) : r, (\text{Jolin\_Tsai, Stefanie\_Sun}) : r, (\text{Stefanie\_Sun, Jay\_Chou}) : r, (\text{Jolin\_Tsai, Jolin\_Tsai}) : r, (\text{Stefanie\_Sun, Stefanie\_Sun}) : r, \text{Stefanie\_Sun} : X\}$$

4

- Compute the certain answers to the following conjunctive queries in the context of $\mathcal{A}$.
- Compute the certain answers to the following conjunctive queries in the context of $\mathcal{K}$.

(a) $r(x, y) \wedge Y(y)$

(b) $\exists y(r(x, y) \wedge Y(y))$

(c) $\exists x, y(r(x, y) \wedge r(y, x))$

(d) $\exists z, w(r(x, y) \wedge r(y, z) \wedge r(z, x) \wedge r(z, w) \wedge W(w))$

(1)ABox only

(a)$\emptyset$

(b)$\emptyset$

(c)Yes

(d)$\emptyset$

(2)ABox + TBox(Rewriting)

First we append the $ABox$ by reasoning $\{X \sqsubseteq Y, Y \sqsubseteq \exists r.X, X \sqsubseteq \forall r.Y, \forall r.X \sqsubseteq Y, W \equiv \neg V, \exists r.Y \sqsubseteq \neg V\}$:

- From $X \sqsubseteq Y$ we know: $\{Stefanie\_Sun : Y\}$
- From $X \sqsubseteq \forall r.Y$ we know: $\{Jay\_Chou : Y\}$
- From $Y \sqsubseteq \exists r.X$ we know: $\{(Jay_Chou, d_X) : r\}$这里用dx比较好
- From $W \equiv \neg V and \ \exists r.Y \sqsubseteq \neg V$ we know: $\{Stefanie\_Sun, Jolin\_Tsai\} : W$

(a)$\{(Stefanie\_Sun, Stefanie\_Sun), (Jolin\_Tsai, Stefanie\_Sun), (Stefanie\_Sun, Jay\_Chou)\}$

(b)$\{Stefanie\_Sun, Jolin\_Tsai\}$

(c)Yes

(d)

$$\{(Jay\_Chou, Jolin\_Tsai), (Jolin\_Tsai, Stefanie\_Sun), (Stefanie\_Sun, Jay\_Chou), (Jolin\_Tsai, Jolin\_Tsai), (Stefanie, Stefanie)\}$$

六、Simpleness of ABox

Consider feeding arbitrary ABoxes rather than simple ABoxes as input to the problem of ontology-mediated querying. Does this affect the data complexity results?

No, this doesn't affect the data complexity results.

The data complexity of OMQA query entailment may vary considerably in these 3 possibilities:

(i)Prove ontology-mediated querying in DL-Lite is always in AC^0 in data complexity:

Because the inputs are identical and the "yes"-inputs also coincide, we can neglect representational differences between an $ABox\mathcal{A}$ and the corresponding interpretationl $\mathcal{I}_\mathcal{A}$.

We can reduce it to entailment of their FO-rewriting $q_\mathcal{T}$.

DL-Lite are FO-reducible and FO-reducibility implies a data complexity in AC^0 for query answering, and thus in particular tractability w.rt. data complexity

Therefore ontology-mediated querying in DL-Lite is always in complexity class AC^0 (below LogSpace and PTime) in data complexity.

(ii)Prove ontology-mediated querying in $\mathcal{EL}$ remains in data complexity:

Note that the exponential size of $q_\mathcal{T}$ is irrelevant since $q_\mathcal{T}$ is fixed and not an input. By utilising the PTime data complexity of Datalog query entailment, we can derive from the rewritings that conjunctive queries entailment in $\mathcal{EL}$ is in PTime-complete regarding data complexity.

Arbitrary ABox will not change the complexity in LogSpace-reduction of path system accessibility, which suggests that ontology-mediated querying in $\mathcal{EL}$ remains P-complete.

(iii)Prove ontology-mediated querying in $\mathcal{ALC}$ remains in data complexity:

The data complexity result remains coNP-complete in $\mathcal{ALC}$ because we can still use tableau algorithm to solve the problem.

Questions like non-3-colorability can be reduced into $\mathcal{ALC}$ by the same way with simple ABoxes.

七、k-colorability

Is it possible to show that the problem of conjunctive query entailment (CQ-entailment) in $\mathcal{ALC}$ is coNP-hard w.r.t. data complexity using a reduction from non-k-colorability in graphs? What if k is fixed?

No, but it's possible when k is fixed

proof:

- If k is not fixed, the answer is not possible. This is because we have to assume that TBox and the query are constant.
- If k is fixed. Then whether it's possible or not depends on the value of k:

(i)$k = 1$:no

The graph is totally disconnected that every single vertice is isolated. So it can be identified in PTime.

So it's not possible.

(ii)$k = 2$:no

All vertices are divided into 2 different subset and any 2 of them have relatiion E iff they are in different subset.

Therefore solving this CQ-entailment is equvalent to deciding whether graph $G = (\boldsymbol{W}, \boldsymbol{E})$ is a bipartite graph.

So we gives the deciding algorithm for bipartite graph:

```
is_bipartite(G):
for (each u in W)//.............................................O(|W|)
  u.color = WHITE, u.d = INF//Record the parity of 'u.d' to decide which subset u belongs to
for (each u in W)//.............................................O(|W|)
  if (u.color== WHITE)//WHITE vertice has not been used
    u.color = GRAY, u.d = 0
    Q.enque(u)//Q is a stack
    while (!Q.empty())
       v = Q.dequeue()
       v.color = BLACK
       for (each edge (v,w) in E)//.........................O (|E|) in all circulation in total
            if((w.d+v.d)%2==0)
                return false
            else if (w.color == WHITE)
                w.color = GRAY
                w.d = v.d+1
                Q.enque(w)
return true
```

This algorithm ends in $O(|W| + |E|)$, which is PTime.

So it's not possible.

(iii)$k \geq 3$:yes

Assume $G = (\boldsymbol{W}, \boldsymbol{E})$ is given and k is fixed, construction as follows:

Construct the ABox $\mathcal{A}_G$ by taking a role name r and setting:

- $r(a, b) \in \mathcal{A}$ for all$a, b \in W$ with$(a, b) \in E$

Construct the $\mathcal{ALC}$ TBox $\mathcal{T}_C$ by taking concept names $Color_1, Color_2 \cdots, Color_k$ and $Clash$ taking the inclusions:

- $\top \sqsubseteq \bigsqcup_{i=1}^{k} Color_i$
- $Color_i \sqcap \exists r.\, Color_i \sqsubseteq Clash, \text{ for all } i$

Let $F = \exists x Clash(x)$. Then $(\mathcal{T}_C, \mathcal{A}_G) \models F$ iff, G is not k -colorable.

Because 3-colorability is NP-complete, thus k-colorability is also NP-complete.

So we can know that the problem of CQ-entailment is coNP-hard w.r.t data complexity in $\mathcal{ALC}$