Συστήματα και Τεχνολογίες Γνώσης

2η Γραπτή Εργασία

Όνομα: Ευστράτιος Καραπαναγιώτης
AM: 03115177

16 Ιουνίου 2020

Ερώτημα 1

Αρχικά μετατρέπουμε τις έννοιες C_1, C_2 της περιγραφικής λογικής FL_0 σε κανονική μορφή. Έχουμε διαδοχικά:

- $C_1 \equiv D \sqcap E \sqcap \forall r.(A \sqcap \forall r.E \sqcap B \sqcap (A \sqcap B) \sqcap \forall r.\forall s.D) \equiv D \sqcap E \sqcap \forall r.(A \sqcap B \sqcap \forall r.(E \sqcap \forall s.D))$
- $C_2 \equiv E \sqcap \forall r.(\forall r.(E \sqcap B) \sqcap \forall r.\forall s.(D \sqcap A)) \equiv E \sqcap \forall r.(\forall r.(E \sqcap B \sqcap \forall s.(D \sqcap A)) \equiv E \sqcap \forall r.(\forall r.(E \sqcap B \sqcap \forall s.D \sqcap \forall s.A))$

Στη συνέχεια εφαρμόζουμε τον αλγόριθμο δομικής υπαγωγής για τη γλώσσα FL_0.

1. **Extract**(C_1, C_2) $\begin{cases} NC = \{D, E\} \\ ND = \{E\} \\ RC = \{r\} \\ RD = \{r\} \end{cases}$
2. 1^{st} **Condition** Διατρέχουμε τα στοιχεία του ND και ελέγχουμε αν ανήκουν και στο σύνολο NC. Το μοναδικό στοιχείο το οποίο υπάρχει είναι το E, το οποίο είναι μέλος και στο σύνολο NC. Επομένως, ικανοποιείται η πρώτη συνθήκη.
3. 2^{nd} **Condition** Στο σημείο αυτό για κάθε στοιχείο-ρόλο στο σύνολο RD απογυμνώνουμε τη σύνθετη έννοια με την οποία είναι συνδεδεμένη ο ρόλος και για τις δύο αρχικές έννοιες. Για παράδειγμα: $\forall r \in ND$; *do* $X := strip(C_1, r); Y := strip(C_2, r)$; *done*. Οι έννοιες X, Y είναι οι σύνθετες έννοιες που ήταν συνδεδεμένες με τον ρόλο r για να κατασκευάσουν μια έννοια. Στην περίπτωσή μας, $X \equiv A \sqcap B \sqcap \forall r.(E \sqcap \forall s.D)$ και $Y \equiv \forall r.(E \sqcap B \sqcap \forall s.D \sqcap \forall s.A)$. Στη συνέχεια εκτελούμε για τις έννοιες X, Y έλεγχο με τον ίδιο αλγόριθμο για την υπαγωή της X στην Y.
4. **Extract**(X, Y) $\begin{cases} NC = \{A, B\} \\ ND = \{\} \\ RC = \{r\} \\ RD = \{r\} \end{cases}$
5. 1^{st} **Condition** Δεν υπάρχει κάποιο στοιχείο στο σύνολο ND να διατρέξουμε οπότε η πρώτη συνθήκη ικανοποιείται.
6. 2^{nd} **Condition** Παίρνουμε $X' \equiv E \sqcap \forall s.D$ και $Y' \equiv E \sqcap B \sqcap \forall s.D \sqcap \forall s.A$. Εφαρμόζουμε, ξανά, αναδρομικά τον αλγόριθμο απόδειξης της δομικής υπαγωγής στα X' και Y'.
7. **Extract**(X', Y') $\begin{cases} NC = \{E\} \\ ND = \{E, B\} \\ RC = \{s\} \\ RD = \{s\} \end{cases}$
8. 1^{st} **Condition** Αυτη τη φορά βλέπουμε ότι δεν ικανοποιείται η συνθήκη επειδή παρόλο που το E ανήκει και στο σύνολο NC, το B **δεν** ανήκει. Έτσι, δεν ισχύει $X' \sqsubseteq Y'$.
9. **Επιστροφή** Επιστρέφεται στην προηγούμενη κλήση το αποτέλεσμα NO το οποίο με τη σειρά του πυροδοτεί την επιστροφή ενός NO στην πρώτη αναδρομική κλήση.

Επομένως, καταλήγουμε με βάση τον αλγόριθμο δομικής υπαγωγής πως **δεν** ισχύει η υπαγωγή $C_1 \sqsubseteq C_2$.

Ερώτημα 2

1. **ΚΜΑ και απαλοιφή TBox**
   Παρατηρούμε ότι για η σχέση στο TBox δεν είναι απλή όπως και η πρόταση που προσπαθούμε να αποδείξουμε ότι ισχύει. Για το λόγο αυτό χρησιμοποιούμε και για τις δύο προτάσεις την απαλοιφή με χρήση της τεχνικής της εσωτερίκευσης.

   $\neg A \sqsubseteq \exists R.B \rightarrow (A \sqcup \exists R.B)(a)$
   $\forall R.\neg B \sqsubseteq A \rightarrow (\exists R.B \sqcup A)(a)$

   Έχουμε λοιπόν ως αρχικό ταμπλό το $S_0 = \{\{(A \sqcup \exists R.B)(a), (\exists R.B \sqcup A)(a)\}\}$. Βλέπουμε ότι αφορούν την ίδια έννοια και επομένως αφού τα ABoxes είναι σύνολα ισχυρισμών λόγω του ορισμού των συνόλων τα επαναλαμβανόμενα στοιχεία παραλείπονται. Έτσι, $S_0 = \{\{(A \sqcup \exists R.B)(a)\}\}$.

   Επέκταση Ταμπλό

   (α΄) Κανόνας $K_{\sqcup}$:

   $S_1 = \{\{(A \sqcup \exists R.B)(a), A(a)\}, \{(A \sqcup \exists R.B)(a), (\exists R.B)(a)\}\}$

   (β΄) Κανόνας $K_{\exists}$ και εσωτερίκευση:

   $S_2 = \{\{(A \sqcup \exists R.B)(a), A(a)\}, \{(A \sqcup \exists R.B)(a), (\exists R.B)(a), R(a, x_1), B(x_1), (A \sqcup \exists R.B)(x_1)\}\}$

   (γ΄) Κανόνας $K_{\sqcup}$:

   $S_3 = \{\{(A \sqcup \exists R.B)(a), A(a)\}, \{(A \sqcup \exists R.B)(a), (\exists R.B)(a), R(a, x_1), B(x_1), (A \sqcup \exists R.B)(x_1), A(x_1)\},$
   $\{(A \sqcup \exists R.B)(a), (\exists R.B)(a), R(a, x_1), B(x_1), (A \sqcup \exists R.B)(x_1), (\exists R.B)(x_1)\}\}$

   (δ΄) Κανόνας $K_{\exists}$ και εσωτερίκευση:

   $S_4 = \{\{(A \sqcup \exists R.B)(a), A(a)\},$
   $\{(A \sqcup \exists R.B)(a), (\exists R.B)(a), R(a, x_1), B(x_1), (A \sqcup \exists R.B)(x_1), A(x_1)\},$
   $\{(A \sqcup \exists R.B)(a), (\exists R.B)(a), R(a, x_1), B(x_1), (A \sqcup \exists R.B)(x_1), (\exists R.B)(x_1), R(x_1, x_2), B(x_2), (A \sqcup \exists R.B)(x_2)\}\}$

   (ε΄) Blocking: ο αλγόριθμος παρατηρεί ότι το πλήθος των νέων παιδών με την εφαρμογή του προηγούμενου βήματος είναι τρία. Το ίδιο ισχύει και όταν εφαρμόστηκε ο κανόνας $K_{\exists}$ σε συνδιασμό με την ιδιότητα της εσωτερίκευσης για το x_1. Επομένως, σταματάει η επέκταση του ταμπλό και θεωρείται πλήρες.

   Εύρεση ABox το οποίο δεν περιέχει αντιφάσεις
   Κατευθείαν βλέπουμε ότι το πρώτο ABox του ταμπλό S_4 δεν παρουσιάζει αντιφάσεις. Επομένως, ισχύει η πρόταση $\neg A \sqsubseteq \exists R.B$ για το TBox $\forall R.\neg B \sqsubseteq A$.

2. **ΚΜΑ και απαλοιφή TBox**
   Παρατηρούμε ότι το TBox δεν είναι ακυκλικό καθώς $u(A, C) = 1$ και $u(C, A) = 1$. Οπότε $u'(A, A) = 2 \neq 0$. Για το λόγο αυτό θα χρησιμοποιηθεί ξανά η διαδικασία της εσωτερίκευσης για την απαλοιφή του TBox. $\begin{cases} 1.\ A \sqsubseteq C \Rightarrow (\neg A \sqcup C)(x) \\ 2.\ C \sqsubseteq \exists R.D \Rightarrow (\neg C \sqcup \exists R.D)(x) \\ 3.\ D \sqsubseteq \neg A \Rightarrow (\neg D \sqcup \neg A)(x) \\ 4.\ C \sqsubseteq \forall R.A \Rightarrow (\neg C \sqcup \forall R.A)(x) \end{cases}$

   Έτσι, το ταμπλό αρχικοποιείται στο στιγμιότυπο: $S_0 = \{\{A(a), R(a, b), (\neg A \sqcup C)(a), (\neg A \sqcup C)(b), (\neg C \sqcup \exists R.D)(a),$
   $(\neg C \sqcup \exists R.D)(b), (\neg D \sqcup \neg A)(a), (\neg D \sqcup \neg A)(b), (\neg C \sqcup \forall R.A)(a), (\neg C \sqcup \forall R.A)(b)\}\}$

   Ονομάζουμε το πρώτο ABox ως A_0 για να γίνει πιο κατανοητή και εκφραστική η διαδικασία.

   Επέκταση Ταμπλό

   (α΄) Κανόνας $K_{\sqcup}$ για την έννοια (1) και για τα δύο άτομα a, b:

   $S_1 = \{A_0 \cup \{\neg A(a), \neg A(b)\}, A_0 \cup \{\neg A(a), C(b)\}, A_0 \cup \{C(a), \neg A(b)\}, A_0 \cup \{C(a), C(b)\}\}$.

   Πλέον φαίνεται ότι ικανοποιούνται οι συνθήκες για την έννοια (3) οπότε δεν μπορεί να επεκτείνει κάτι. Επίσης αποφεύγουμε την επέκταση του ταμπλό μέσω της 2 καθώς θα προσθέσει επιπλέον άτομο και οπότε λόγω της ιδιότητας της εσωτερίκευσης θα έχουμε μεγαλύτερα ταμπλό και όχι κάποια σημαντική νέα πληροφορία.

(β′) Κανόνας $K_{\sqcup}$ για την έννοια (4) και για το άτομο a:

$$S_2 = \{A_0 \cup \{\neg A(a), \neg A(b), \neg C(a)\}, A_0 \cup \{\neg A(a), \neg A(b), (\forall R.A)(a)\}, A_0 \cup \{\neg A(a), C(b), C(a)\}, A_0 \cup \{\neg A(a), C(b), (\forall R.A)(a)\}, A_0 \cup \{C(a), \neg A(b), \neg C(a)\}, A_0 \cup \{C(a), \neg A(b), (\forall R.A)(a)\}, A_0 \cup \{C(a), C(b), \neg C(a)\}, A_0 \cup \{C(a), C(b), (\forall R.A)(a)\}\}$$

(γ′) Κανόνας $K_{\forall}$:

$$S_3 = \{A_0 \cup \{\neg A(a), \neg A(b), \neg C(a)\}, A_0 \cup \{\neg A(a), \neg A(b), (\forall R.A)(a), A(b)\}, A_0 \cup \{\neg A(a), C(b), C(a)\}, A_0 \cup \{\neg A(a), C(b), (\forall R.A)(a), A(b)\}, A_0 \cup \{C(a), \neg A(b), \neg C(a)\}, A_0 \cup \{C(a), \neg A(b), (\forall R.A)(a), A(b)\}, A_0 \cup \{C(a), C(b), \neg C(a)\}, A_0 \cup \{C(a), C(b), (\forall R.A)(a), A(b)\}\}$$

(δ′) Κανόνας $K_{\sqcup}$ για την έννοια (4) και για το άτομο b:

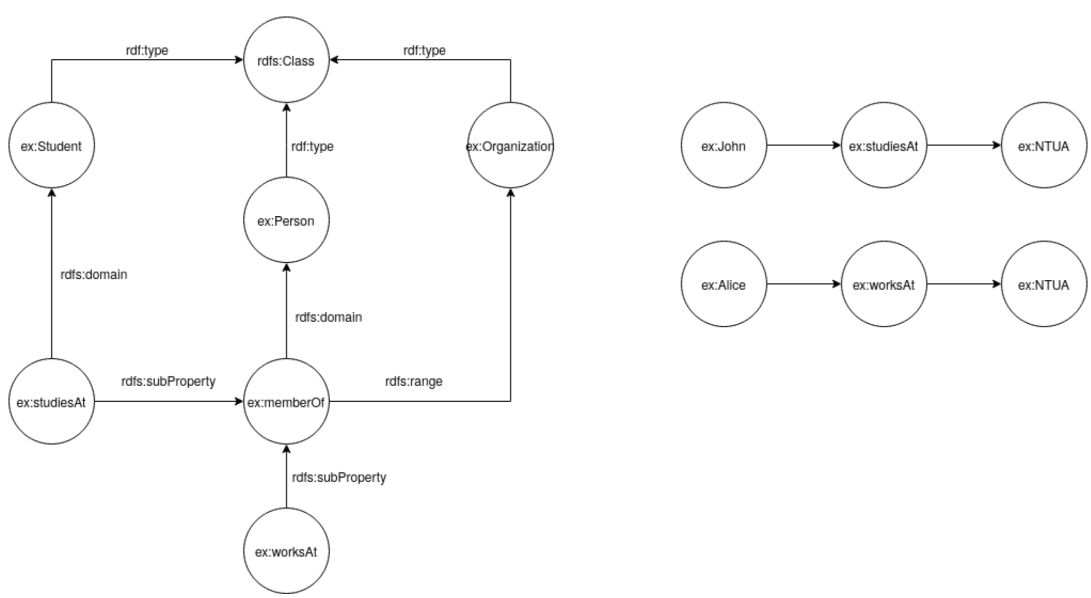
$$S_4 = \{ \left\{ \begin{array}{l}
1.\ A_0 \cup \{\neg A(a), \neg A(b), \neg C(a), \neg C(b)\}, \\
2.\ A_0 \cup \{\neg A(a), \neg A(b), \neg C(a), (\forall R.A)(b)\}, \\
3.\ A_0 \cup \{\neg A(a), \neg A(b), (\forall R.A)(a), A(b), \neg C(b)\}, \\
4.\ A_0 \cup \{\neg A(a), \neg A(b), (\forall R.A)(a), A(b), (\forall R.A)(b)\}, \\
5.\ A_0 \cup \{\neg A(a), C(b), C(a), \neg C(b)\}, \\
6.\ A_0 \cup \{\neg A(a), C(b), C(a), (\forall R.A)(b)\}, \\
7.\ A_0 \cup \{\neg A(a), C(b), (\forall R.A)(a), A(b), \neg C(b)\}, \\
8.\ A_0 \cup \{\neg A(a), C(b), (\forall R.A)(a), A(b), (\forall R.A)(b)\}, \\
9.\ A_0 \cup \{C(a), \neg A(b), \neg C(a), \neg C(b)\}, \\
10.\ A_0 \cup \{C(a), \neg A(b), \neg C(a), (\forall R.A)(b)\}, \\
11.\ A_0 \cup \{C(a), \neg A(b), (\forall R.A)(a), A(b), \neg C(b)\}, \\
12.\ A_0 \cup \{C(a), \neg A(b), (\forall R.A)(a), A(b), (\forall R.A)(b)\}, \\
13.\ A_0 \cup \{C(a), C(b), \neg C(a), \neg C(b)\}, \\
14.\ A_0 \cup \{C(a), C(b), \neg C(a), (\forall R.A)(b)\}, \\
15.\ A_0 \cup \{C(a), C(b), (\forall R.A)(a), A(b), \neg C(b)\}, \\
16.\ A_0 \cup \{C(a), C(b), (\forall R.A)(a), A(b), (\forall R.A)(b)\}
\end{array} \right. \}$$

(ε′) Δεν υπάρχει άλλη επέκταση (κάποιος κανόνας που μπορεί να εφαρμοστεί) για το ταμπλό, οπότε η παραπάνω μορφή είναι η τελική.

Εύρεση ABox το οποίο δεν περιέχει αντιφάσεις
Όλα τα ABox, από το 1 εώς το 15 περιέχουν αντιφάσεις. Το μοναδικό το οποίο δεν περιέχει αντιφάσεις είναι το 16ο ABox. Επομένως, το ABox $A = \{A(a), R(a, b)\}$ είναι ικανοποιήσιμο για το TBox που δίνεται.

Ερώτημα 3

1. Σε πρώτη φάση παρουσιάζεται ο συνολικός RDFS γράφος στην εικόνα 1. Με βάση αυτόν θα δοθεί η απάντηση για τον RDFS reasoner.



Σχήμα 1: Ο RDFS γράφος της βάσης γνώσης

Για την πρώτη πρόταση γνωρίζει ότι το ex:studiesAt είναι υπορόλος της ex:memberOf μέσω της σχέσης ex:subProperty. Η ex:subProperty δέχεται ως ορίσματα όμως δύο σχέσεις(ρόλους). Η ex:studiesAt όμως είναι το ένα όρισμά της και έτσι συνάγει ότι:
ex:studiesAt a rdfs:Property.

Για τη δεύτερη πρόταση γνωρίζει ότι ο ex:John σπουδάζει(ex:studiesAt) στο ΕΜΠ (ex:NTUA). Για το ρόλο ex:studiesAt γνωρίζει ότι έχει ως πεδίο ορισμού το ex:Student. Επομένως, το ex:John είναι μέλος της έννοιας ex:Student. Η έννοια ex:Student είναι όμως υποέννοια της ex:Person. Προκύπτει λοιπόν ότι το ex:John είναι μέλος (υπάγεται) της έννοιας ex:Person και συνάγει ότι:
ex:John a ex:Person.

2. Το ερώτημα SPARQL διατυπώνεται ως εξής:

   SELECT ?organization ?name
   WHERE
   {
   ?organization rdf:type ex:Organization
   ?name ex:memberOf ?organization
   }

   Θα πρέπει να σημειωθεί ότι το παραπάνω SPARQL ερώτημα επιστρέφει οποιοδήποτε όνομα αντικειμένου είναι μέλος του οργανισμού ?organization. Επιτρέπει δηλαδή εκτός από ανθρώπους και άλλα άτομα. Για παράδειγμα ακόμα και μια εταιρεία μπορεί να συμμετέχει σε έναν οργανισμό. Επίσης, δεν διακρίνει αυτούς που εργάζονται από τους υπόλοιπους. Έτσι, κάποιος μπορεί να είναι μέλος (π.χ. μέτοχος) αλλά να μην εργάζεται για τον οργανισμό αυτόν.

Ερώτημα 4

1.
   - ex:Place owl:disjointWith ex:Person. → DisjointClasses(ex:Place, ex:Person).
   - ex:livesIn rdfs:domain ex:Person → ObjectPropertyDomain(ex:livesin, ex:Person)
   - ex:livesIn rdfs:range ex:Place → ObjectPropertyRange(ex:livesIn, ex:Place)
   - ex:John ex:livesIn ex:Athens → ObjectPropertyAssertion(ex:livesIn, ex:John, ex:Athens)
   - ex:Athens a _:b1 → ClassAssertion(_:b1, ex:John)
   - _:b1 owl:complementOf ex:Person → equivalentClasses(_:b1, complementOf(ex:Person))

2. **ΚΜΑ και απαλοιφή TBox**
   Αρχικά γίνεται απεικόνιση της βάσης γνώσης στα αντίστοιχα σώματα ABox και TBox. Επομένως, $ABox = \{livesIn(John, Athens)\}$ και $TBox = \{\exists livesIn \sqsubseteq Person, \exists livesIn^{-} \sqsubseteq Place, Person \sqcap Place \sqsubseteq \bot\}$. Οι δύο τελευταίες τριάδες εκφράζονται ως $b_1(Athens)$ και $b_1 \equiv \neg Person$. Κατά τα γνωστά, θα πρέπει να ενσωματωθεί το TBox στο ABox προκειμένου να μπορεί να ξεκινήσει η εκτέλεση του αλγορίθμου tableau. Επειδή, οι δηλώσεις στο TBox δεν είναι απλής μορφής χρησιμοποιείται η μέθοδος της εσωτερίκευσης.

   $$\text{Δηλαδή} \begin{cases} 1.\ \exists livesIn \sqsubseteq Person \Rightarrow ((\forall livesIn.\bot) \sqcup Person)(a) \\ 2.\ \exists livesIn^{-} \sqsubseteq Place \Rightarrow ((\forall livesIn^{-}.\bot) \sqcup Place)(a) \\ 3.\ Person \sqcap Place \sqsubseteq \bot \Rightarrow (\neg Person \sqcup \neg Place)(a) \end{cases}$$

   Προκύπτει λοιπόν το πρώτο στιγμιότυπο του ταμπλό $S_0 = \{\{livesIn(John, Athens), ((\forall livesIn.\bot) \sqcup Person)(a), ((\forall livesIn^{-}.\bot) \sqcup Place)(a), (\neg Person \sqcup \neg Place)(a), \neg Person(Athens)\}\}$, όπου χρησιμοποιήθηκε απευθείας η μέθοδος ξεδιπλώματος για τα $b_1(Athens)$ και $b_1 \equiv \neg Perosn$.

   Επέκταση Ταμπλό
   Χρησιμοποιείται ο συμβολισμός A_0 για να δηλωθεί το αρχικό ABox που κατασκευάστηκε για τον αλγόριθμο.

   (α′) Κανόνας $K_{\sqcup}$ για την (1):

   $$S_1 = \{A_0 \cup \{(\forall livesIn.\bot)(a)\}, A_0 \cup \{Person(a)\}\}$$

   (β′) Κανόνας $K_{\sqcup}$ για την (2):

   $$S_2 = \{A_0 \cup \{(\forall livesIn.\bot)(a), (\forall livesIn^{-}.\bot)(a)\}, A_0 \cup \{(\forall livesIn.\bot)(a), Place(a)\}, A_0 \cup \{Person(a), (\forall livesIn.\bot)(a)\}, A_0 \cup \{Person(a), Place(a)\}\}$$

   (γ′) Κανόνας $K_{\sqcup}$ για την (3):

$$S_2 = \{ \begin{cases} 1.\ A_0 \cup \{(\forall livesIn.\bot)(a), (\forall livesIn^{-}.\bot)(a), \neg Person(a)\}, \\ 2.\ A_0 \cup \{(\forall livesIn.\bot)(a), (\forall livesIn^{-}.\bot)(a), \neg Place(a)\}, \\ 3.\ A_0 \cup \{(\forall livesIn.\bot)(a), Place(a), \neg Person(a)\}, \\ 4.\ A_0 \cup \{(\forall livesIn.\bot)(a), Place(a), \neg Place(a)\}, \\ 5.\ A_0 \cup \{Person(a), (\forall livesIn.\bot)(a), \neg Person(a)\}, \\ 6.\ A_0 \cup \{Person(a), (\forall livesIn.\bot)(a), \neg Place(a)\}, \\ 7.\ A_0 \cup \{Person(a), Place(a), \neg Person(a)\} \\ 8.\ A_0 \cup \{Person(a), Place(a), \neg Place(a)\} \end{cases} \}$$

(δ΄) Δεν υπάρχει άλλη επέκταση (κάποιος κανόνας που μπορεί να εφαρμοστεί) για το ταμπλό, οπότε η παραπάνω μορφή είναι η τελική.

Εύρεση ABox το οποίο δεν περιέχει αντιφάσεις

Το πρώτο κι ολας ABox δεν παρουσιάζει αντιφάσεις και έτσι ο OWL reasoner αποδεικνύει ότι οι δύο τελυταίες προτάσεις είναι συμπέρασμα των προηγούμενων.