Τμήμα ΜΗΥΠ

ΠΜΣ Υπολογιστική Δεδομένων και Αποφάσεων (ΥΔΑ)

Συλλογισμοί με Περιγραφικές Λογικές (Reasoning with Description Logics)

Ι. Χατζηλυγερούδης

Αρχιτεκτονική Συστήματος ΠΛ

Βάση Γνώσης

TBox

$Woman \equiv Person \cap Female$

$Man \equiv Person \cap \neg Female$

…

Abox

Man(BOB)

hasChild(BOB, MARY)

$\neg$Doctor(MARY)

…

Σύστημα Εξαγωγής Συμπερασμάτων

Διεπαφή Χρήστη

Συλλογισμοί στις ΠΛ

- Ικανοποιησιμότητα εννοιών

  Αν μια έννοια C είναι ικανοποιήσιμη ως προς ένα TBox T (δηλ. δεν δημιουργεί σύγκρουση)
- Σύνοψη εννοιών

  Αν μια έννοια C συνοψίζει μια έννοια D ($C \sqsubseteq D$) ως προς ένα TBox T
- Ισοδυναμία εννοιών

  Αν δύο έννοιες είναι ισοδύναμες ($C \equiv D$) ως προς ένα TBox T

Συλλογισμοί στις ΠΛ

- Ασυμβατότητα εννοιών

  Αν μια έννοια C είναι ασύμβατη (ξένη) με μια έννοια D ως προς ένα TBox T
- Έλεγχος στιγμιοτύπων

  Αν μια οντότητα *a* είναι στιγμιότυπο της έννοιας C ως προς ένα TBox T και ένα ABox A

Τα επόμενα αναφέρονται στην ΠΛ $\mathcal{ALC}$

Ικανοποιησιμότητα

- Όλοι οι προηγούμενοι συλλογισμοί ισοδυναμούν με κάποιο έλεγχο ικανοποιησιμότητας:
  - Σύνοψη εννοιών:

    $C \sqsubseteq D$ ανν $C \sqcap \neg D$ είναι μη ικανοποιήσιμη

    (η C συνοψίζεται από την D ή η D συνοψίζει (subsumes) την C)
  - Ισοδυναμία εννοιών:

    $C \equiv D$ ανν $C \sqsubseteq D$ και $D \sqsubseteq C$, δηλ.

    ανν $(C \sqcap \neg D) \sqcup (\neg C \sqcap D)$ είναι μη ικανοποιήσιμη
  - Ασυμβατότητα εννοιών:

    C ξένη με D ανν $C \sqcap D$ είναι μη ικανοποιήσιμη
  - Έλεγχος στιγμιοτύπου:

    a είναι στιγμιότυπο της C ανν $\mathcal{A} \cup \{a: \neg C\}$ είναι μη ικανοποιήσιμο ($\mathcal{A}$ είναι το ABox)

Μέθοδος Συλλογισμού Tableau

- Είναι μέθοδος ελέγχου ικανοποιησιμότητας εννοιών
- Διαδικασία
  1. Μετατροπή μιας έννοιας σε Αρνητική Κανονική Μορφή-ΑΚΜ (Negation Normal Form-NNF)
  2. Εφαρμογή Κανόνων Πληρότητας (Completion Rules) με αυθαίρετη σειρά μέχρις ότου:
     - ✓ συναντήσει περίπτωση σύγκρουσης ή
     - ✓ δεν υπάρχει άλλος εφαρμόσιμος κανόνας
  3. Η έννοια είναι ικανοποιήσιμη ανν προκύψει ένα tableau πλήρες (complete) και ελεύθερο συγκρούσεων (clash-free), δηλ. δεν περιέχει το $\perp$ ούτε κάποιο ζεύγος $\{\neg C, C\}$.

Αρνητική Κανονική Μορφή (ΑΚΜ)

- Όλες οι αρνήσεις μετακινούνται σε επίπεδο ονομάτων των εννοιών
- Κανόνες μετασχηματισμού ΑΚΜ (από διαφάνειες Zakharyaschev)

$$\begin{array}{rcll}
\neg\top & \equiv & \bot & \\
\neg\bot & \equiv & \top & \\
\neg\neg C & \equiv & C & \\
\neg(C \sqcap D) & \equiv & \neg C \sqcup \neg D & \text{(De Morgan's law)} \\
\neg(C \sqcup D) & \equiv & \neg C \sqcap \neg D & \text{(De Morgan's law)} \\
\neg\forall R.C & \equiv & \exists R.\neg C & \\
\neg\exists R.C & \equiv & \forall R.\neg C &
\end{array}$$

Αρνητική Κανονική Μορφή (ΑΚΜ)

Μετασχηματίστε την έννοια

$$\neg\exists R.(A \sqcap \neg B) \sqcup \neg\forall R.(\neg A \sqcup \neg B)$$

Σε ισοδύναμη έννοια σε ΑΚΜ (NNF)

$\neg\exists R.(A \sqcap \neg B) \sqcup \neg\forall R.(\neg A \sqcup \neg B) \equiv$ (use $\neg\exists R.D \equiv \forall R.\neg D$)

$\forall R.\neg(A \sqcap \neg B) \sqcup \neg\forall R.(\neg A \sqcup \neg B) \equiv$ (use $\neg(A \sqcap D) \equiv \neg A \sqcup \neg D$)

$\forall R.(\neg A \sqcup \neg\neg B) \sqcup \neg\forall R.(\neg A \sqcup \neg B) \equiv$ (use $\neg\neg B \equiv B$)

$\forall R.(\neg A \sqcup B) \sqcup \neg\forall R.(\neg A \sqcup \neg B) \equiv$ (use $\neg\forall R.D \equiv \exists R.\neg D$)

$\forall R.(\neg A \sqcup B) \sqcup \exists R.\neg(\neg A \sqcup \neg B) \equiv$ (use $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$)

$\forall R.(\neg A \sqcup B) \sqcup \exists R.(\neg\neg A \sqcap \neg\neg B) \equiv$ (use $\neg\neg C \equiv C$)

$\forall R.(\neg A \sqcup B) \sqcup \exists R.(A \sqcap B)$

(από διαφάνειες Zakharyaschev)

Ορισμοί

- Περιορισμός (Constraint): Έκφραση της μορφής x: C ή (x, y): R
- Σύστημα περιορισμών (Constraint system): Ένα μη-κενό πεπερασμένο σύνολο περιορισμών S
- Κανόνες πληρότητας (Completion rules): Ένας μετασχηματισμός S → S′, όπου S′ είναι ένα σύστημα περιορισμών που περιέχει το S
- Πλήρες σύστημα: Το S είναι πλήρες εάν κανένας κανόνας πληρότητας δεν μπορεί να εφαρμοστεί στο S
- Σύγκρουση (Clash): Το S περιέχει μια σύγκρουση εάν $\{x: A, x: \neg A\} \subseteq S$, όπου A είναι όνομα έννοιας

Κανόνες Πληρότητας (Completion Rules)

$$S \rightarrow_{\sqcap} S \cup \{ x: C, \; x: D \}$$

if (a) $x: C \sqcap D$ is in S

(b) $x: C$ and $x: D$ are not both in S

(τομής)

$$S \rightarrow_{\sqcup} S \cup \{ x: E \}$$

if (a) $x: C \sqcup D$ is in S

(b) neither $x: C$ nor $x: D$ is in S

(c) $E = C$ or $E = D$ **(branching!)**

(ένωσης)

(από διαφάνειες Zakharyaschev)

Κανόνες Πληρότητας (Completion Rules)

$$S \rightarrow_{\forall} S \cup \{ y : C \}$$

if (a) $x : \forall R.C$ is in S
(b) $(x, y) : R$ is in S
(c) $y : C$ is not in S

(Καθολικότητας)

$$S \rightarrow_{\exists} S \cup \{ (x, y) : R,\ y : C \}$$

if (a) $x : \exists R.C$ is in S
(b) there is no z such that both $(x, z) : R$ and $z : C$ are in S
(c) y is a fresh individual

(Υπαρξιακότητας)

(από διαφάνειες Zakharyaschev)

$$S \to_{\sqcap} S \cup \{\, x\colon C,\ x\colon D \,\}$$

if (a) $x\colon C \sqcap D$ is in S
(b) $x\colon C$ and $x\colon D$ are not both in S

Παραδείγματα

$$S \to_{\sqcup} S \cup \{\, x\colon E \,\}$$

if (a) $x\colon C \sqcup D$ is in S
(b) neither $x\colon C$ nor $x\colon D$ is in S
(c) $E = C$ or $E = D$ **(branching!)**

(από διαφάνειες Zakharyaschev)

Tableau Algorithm: example

Let $\boxed{\text{Woman} \equiv \text{Person} \sqcap \text{Female}}$, $\boxed{\text{Mother} \equiv \text{Parent} \sqcap \text{Female}}$ and $\boxed{\text{Parent} \equiv \text{Person} \sqcap \exists\text{hasChild.Person}}$

Does the concept $\boxed{\text{Woman}}$ subsume the concept $\boxed{\text{Mother}}$?
i.e., is the concept $\boxed{\neg\text{Woman} \sqcap \text{Mother}}$ satisfiable?

$$
\begin{aligned}
S_0 \quad &= \{\, x\colon (\neg\text{Person} \sqcup \neg\text{Female}) \sqcap ((\text{Person} \sqcap \exists\text{hasChild.Person}) \sqcap \text{Female}) \,\} \\
S_0 \to_{\sqcap} S_1 &= S_0 \cup \{\, x\colon \neg\text{Person} \sqcup \neg\text{Female},\ x\colon (\text{Person} \sqcap \exists\text{hasChild.Person}) \sqcap \text{Female} \,\} \\
S_1 \to_{\sqcap} S_2 &= S_1 \cup \{\, x\colon \text{Person} \sqcap \exists\text{hasChild.Person},\ x\colon \text{Female} \,\} \\
S_2 \to_{\sqcap} S_3 &= S_2 \cup \{\, x\colon \text{Person},\ x\colon \exists\text{hasChild.Person} \,\} \\
S_3 \to_{\sqcup} S_{4.1} &= S_3 \cup \{\, x\colon \neg\text{Person} \,\} \qquad \textbf{clash} \\
S_3 \to_{\sqcup} S_{4.2} &= S_3 \cup \{\, x\colon \neg\text{Female} \,\} \qquad \textbf{clash}
\end{aligned}
$$

Thus the concept $\boxed{\neg\text{Woman} \sqcap \text{Mother}}$ is unsatisfiable,
and so $\boxed{\text{Woman}}$ subsumes $\boxed{\text{Mother}}$

Παραδείγματα

(το προηγούμενο υπό μορφή δέντρου)

$\{x:(\neg Person \cup \neg Female) \cap ((Person \cap \exists hasChild.Person) \cap Female)\}$ S_0

$\rightarrow\cap$ (κανόνας τομής)

$\{x:\neg Person \cup \neg Female,\ x:(Person \cap \exists hasChild.Person) \cap Female\}$ S_1

$\rightarrow\cap$

$\{x:\neg Person \cup \neg Female,\ x:Person \cap \exists hasChild.Person,\ x:Female\}$ S_2

$\rightarrow\cap$

$\{x:\neg Person \cup \neg Female,\ x:Person,\ x:\exists hasChild.Person,\ x:Female\}$ S_3

$\rightarrow\cup$ (κανόνας ένωσης)

$\{x:\neg Person,\ x:Person,\ x:\exists hasChild.Person,\ x:Female\}$ $S_{4.1}$

(σύγκρουση-clash)

$\{x: \neg Female,\ x:Person,\ x:\exists hasChild.Person,\ x:Female\}$ $S_{4.2}$

(σύγκρουση-clash)

Παραδείγματα

(το προηγούμενο υπό μορφή πιο αφηρημένου δέντρου)

$$
\begin{array}{c}
S_0 \\
\big\downarrow \; \rightarrow\cap \\
S_1 \\
\big\downarrow \; \rightarrow\cap \\
S_2 \\
\big\downarrow \; \rightarrow\cap \\
S_3 \\
\swarrow \qquad \searrow \; \rightarrow\cup
\end{array}
$$

(σύγκρουση-clash) $S_{4.1}$ $S_{4.2}$ (σύγκρουση-clash)

Παραδείγματα

Reasoning with ABoxes: example
(από διαφάνειες Zakharyaschev)

Given: Sam is a person living in Germany. Sam drinks beer and Deuchars. A Bavarian is a person who lives in Germany, drinks beer and only beer.

Q: Is Sam a Bavarian?

ABox $\mathcal{A}$

$$\begin{array}{l} sam : Person \\ sam : \exists livesIn.Germany \\ sam : \exists drinks.Beer \\ (sam, deuchars) : drinks \end{array}$$

TBox $\mathcal{T}$

$$Bavarian \equiv Person \sqcap \exists livesIn.Germany \sqcap \exists drinks.Beer \sqcap \forall drinks.Beer$$

Is $\boxed{sam}$ an instance of $\boxed{Bavarian}$?

1. Reduction to ABox consistency:

Sam is an instance of Bavarian iff $\mathcal{A} \cup \{ sam : \neg Bavarian \}$ is unsatisfiable

2. NNF of ¬Bavarian:

$$\neg Person \sqcup \forall livesIn.\neg Germany \sqcup \forall drinks.\neg Beer \sqcup \exists drinks.\neg Beer$$

Παραδείγματα

$S \rightarrow_{\forall} S \cup \{ y\colon C \}$

if (a) $x\colon \forall R.C$ is in S
(b) $(x,y)\colon R$ is in S
(c) $y\colon C$ is not in S

$S \rightarrow_{\exists} S \cup \{ (x,y)\colon R,\ y\colon C \}$

if (a) $x\colon \exists R.C$ is in S
(b) there is no z such that both $(x,z)\colon R$ and $z\colon C$ are in S
(c) y is a fresh individual

Reasoning with ABoxes: example (cont.)

$$S_0 = \{ \text{sam}\colon \text{Person},\ \text{sam}\colon \exists\text{livesIn.Germany},\ \text{sam}\colon \exists\text{drinks.Beer},\ (\text{sam}, \text{deuchars})\colon \text{drinks},\ \text{sam}\colon \neg\text{Person} \sqcup \forall\text{livesIn.}\neg\text{Germany} \sqcup \forall\text{drinks.}\neg\text{Beer} \sqcup \exists\text{drinks.}\neg\text{Beer} \}$$

$S_0 \rightarrow_{\sqcup} S_{1.1} = S_0 \cup \{ \text{sam}\colon \neg\text{Person} \}$ **clash**

$S_0 \rightarrow_{\sqcup} S_{1.2} = S_0 \cup \{ \text{sam}\colon \forall\text{livesIn.}\neg\text{Germany} \}$

$S_{1.2} \rightarrow_{\exists} S_{2.2} = S_{1.2} \cup \{ (\text{sam}, x)\colon \text{livesIn},\ x\colon \text{Germany} \}$

$S_{2.2} \rightarrow_{\forall} S_{3.2} = S_{2.2} \cup \{ x\colon \neg\text{Germany} \}$ **clash**

$S_0 \rightarrow_{\sqcup} S_{1.3} = S_0 \cup \{ \text{sam}\colon \forall\text{drinks.}\neg\text{Beer} \}$

$S_{1.3} \rightarrow_{\exists} S_{2.3} = S_{1.3} \cup \{ (\text{sam}, x)\colon \text{drinks},\ x\colon \text{Beer} \}$

$S_{2.3} \rightarrow_{\forall} S_{3.3} = S_{2.3} \cup \{ x\colon \neg\text{Beer} \}$ **clash**

$S_0 \rightarrow_{\sqcup} S_{1.4} = S_0 \cup \{ \text{sam}\colon \exists\text{drinks.}\neg\text{Beer} \}$ (...see the next slide)

(από διαφάνειες Zakharyaschev)

Παραδείγματα

Reasoning with ABoxes: example (cont.)

$$S_0 = \{\ \text{sam}: \text{Person},\ \text{sam}: \exists \text{livesIn.Germany},\ \text{sam}: \exists \text{drinks.Beer},\ (\text{sam}, \text{deuchars}): \text{drinks},\ \text{sam}: \neg\text{Person} \sqcup \forall \text{livesIn}.\neg\text{Germany} \sqcup \forall \text{drinks}.\neg\text{Beer} \sqcup \exists \text{drinks}.\neg\text{Beer}\ \}$$

$$S_0 \to_{\sqcup} S_{1.4} = S_0 \cup \{\ \text{sam}: \exists \text{drinks}.\neg\text{Beer}\ \}$$

$$S_{1.4} \to_{\exists} S_{2.4} = S_{1.4} \cup \{\ (\text{sam}, x): \text{drinks},\ x: \neg\text{Beer}\ \}$$

$$S_{2.4} \to_{\exists} S_{3.4} = S_{2.4} \cup \{\ (\text{sam}, y): \text{drinks},\ y: \text{Beer}\ \}$$

$$S_{3.4} \to_{\exists} S_{4.4} = S_{3.4} \cup \{\ (\text{sam}, z): \text{livesIn},\ z: \text{Germany}\ \}$$

$S_{4.4}$ is a complete clash-free constraint system. Therefore,

$$\mathcal{A} \cup \{\text{sam}: \neg\text{Bavarian}\}$$

is satisfiable and Sam is **not an instance** of Bavarian.

Indeed, the interpretation which is obtained on the fourth branch on the one hand is a model of $\mathcal{A}$; on the other hand it includes the pair of constraints (sam, x): drinks and $x: \neg$Beer, which contradicts the definition of a Bavarian ('drinks only beer').

Note that nothing would change if we added $\boxed{\text{deuchars}: \text{Beer}}$ to the ABox.

(από διαφάνειες Zakharyaschev)

Tableau

- Μια διαδικασία αποφάσεων για λογικές αναπαραστάσεις.
- Μια διαδικασία αποδείξεων για προτάσεις ΚΛΠΤ.
- Ονομάζεται επίσης *δέντρο αληθείας*.
- Κάθε κόμβος περιέχει μια υπο-πρόταση της αρχικής πρότασης.
- Είναι ένα tableau που ικανοποιεί την ιδιότητα της υπο-πρότασης.

Tableau

- Για να δείξουμε ότι μια πρόταση φ είναι έγκυρη, προσπαθούμε να βρούμε μια ερμηνεία που την κάνει ψευδή (η οποία κάνει την άρνησή της ¬φ αληθή).
- Το σύνολο {¬φ } τοποθετείται στη ρίζα του δέντρου.
- Το δέντρο δημιουργείται έτσι ώστε τα φύλλα να αποτελούνται από σύνολα υπο-προτάσεων της φ σύμφωνα με συγκεκριμένους κανόνες tableau.
- Ένας κόμβος-φύλλο ορίζεται ως κλειστό αν περιέχει μια πρόταση και την άρνησή της.
- Αν όλα τα φύλλα είναι κλειστά τότε η ¬φ είναι μη ικανοποιήσιμη και γι’ αυτό η φ είναι έγκυρη, αλλιώς η ¬φ είναι ικανοποιήσιμη και γι’ αυτό η φ δεν είναι έγκυρη.

Tableau Rules

Κανόνες Tableau:

- Replace $\phi \wedge \psi$ with ϕ, ψ
- Replace $\neg(\phi \vee \psi)$ with $\neg\phi, \neg\psi$
- Replace $\neg(\phi \rightarrow \psi)$ with $\phi, \neg\psi$
- Replace $(\phi \rightarrow \psi)$ with $\phi \vee \psi$
- Replace $\neg\neg\phi$ with ϕ .
- Replace ¬ false with true.
- Replace ¬ true with false.
- Replace $\phi <-> \psi$ with $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$

Παράδειγμα Tableau

- Δείξτε ότι η (p ^ (p → q)) → q είναι έγκυρη

$$\{\neg((p \wedge (p \to q)) \to q)\}$$

$$\{(p \wedge (p \to q)), \neg q\}$$

$$\{p, (p \to q), \neg q\}$$

$$\{p, \neg p, \neg q\} \qquad \{p, q, \neg q\}$$

$$\times \qquad\qquad \times$$

- Και τα δύο φύλλα είναι κλειστά, επομένως η ((p ^ (p → q)) → q) είναι μη ικανοποιήσιμη και γι’ αυτό η (p ^ (p → q)) → q είναι έγκυρη.

Θεώρημα Tableau

- **Ορθότητα (Soundness):** Αν μια πρόταση φ προτασιακής λογικής έχει μια απόδειξη tableaux τότε η φ είναι ταυτολογία.

- **Πληρότητα (Completeness)** Αν μια πρόταση φ προτασιακής λογικής είναι ταυτολογία τότε έχει μια απόδειξη tableaux .