Алгоритм логического анализа

Полиномиальный алгоритм, разрешающий $C \sqsubseteq_T D$

Алгоритм логического анализа

Полиномиальный алгоритм, разрешающий $C \sqsubseteq_T D$

Алгоритм решает для данных имен концептов A и B верно ли $A \sqsubseteq_T B$.

Алгоритм логического анализа

Полиномиальный алгоритм, разрешающий $C \sqsubseteq_T D$

Алгоритм решает для данных имен концептов A и B верно ли $A \sqsubseteq_T B$.

Этого достаточно

- $C \sqsubseteq_T D$
- $A \sqsubseteq_{T'} B$, где A и B не входят в C, D, T и TBox

$$T' = T \cup \{A \equiv C, B \equiv D\}$$

Нормальная форма

$\mathcal{EL}$-TBox находится в *нормальной форме* если он состоит только из импликаций вида

(sform) $A \sqsubseteq B$;

(cform) $A_1 \sqcap A_2 \sqsubseteq B$; имена концептов;

(rform) $A \sqsubseteq \exists r.B$;

(lform) $\exists r.A \sqsubseteq B$.

где A A_1, A_2, и B — имена концептов

Нормальная форма

$\mathcal{EL}$-TBox находится в *нормальной форме* если он состоит только из импликаций вида

(sform) $A \sqsubseteq B$;

(cform) $A_1 \sqcap A_2 \sqsubseteq B$; имена концептов;

(rform) $A \sqsubseteq \exists r.B$;

(lform) $\exists r.A \sqsubseteq B$.

где A A_1, A_2, и B — имена концептов

Произвольный $\mathcal{EL}$-Box T может быть приведен к нормальной форме за полиномиальное время так, что для всех имен концептов A, B из T:

$$A \sqsubseteq_T B \quad \Leftrightarrow \quad A \sqsubseteq_{T'} B.$$

Лемма

Определение. $C[D/E]$ есть результат замены всех вхождений концепта D в C на E

$$\mathsf{hasPart}.(\mathsf{Arm} \sqcap \mathsf{Leg})[\mathsf{Arm} \sqcap \mathsf{Leg}/\mathsf{Tail} \sqcap \mathsf{H\_Leg}] = \mathsf{hasPart}.(\mathsf{Tail} \sqcap \mathsf{H\_Leg}),$$
$$\mathsf{hasPart}.(\mathsf{Arm} \sqcap \mathsf{Leg})[\mathsf{Arm}/\mathsf{Tail}] = \mathsf{hasPart}.(\mathsf{Tail} \sqcap \mathsf{Leg})$$

Лемма. $\alpha = C \sqsubseteq D \in T$, T – TBox. Для

- $T' = T \setminus \alpha \cup \{C[E/X] \sqsubseteq D, E \sqsubseteq X\}$ и
- $T' = T \setminus \alpha \cup \{C \sqsubseteq D[E/X], X \sqsubseteq E\}$,

где X – новое имя концепта, и всех имен концептовA, B из T:

$$A \sqsubseteq_T B \quad \Leftrightarrow \quad A \sqsubseteq_{T'} B.$$

Приведение к нормальной форме

Применять следующие правила пока возможно

- заменить $C_1 \equiv C_2$ на $C_1 \sqsubseteq C_2$ и $C_2 \sqsubseteq C_1$;
- заменить $C \sqsubseteq C_1 \sqcap C_2$ на $C \sqsubseteq C_1$ и $C \sqsubseteq C_2$;
- если $\exists r.C$ входит в T и C сложный концепт, заменить все вхождения C на новое имя X_C и добавить $X_C \sqsubseteq C$ и $C \sqsubseteq X_C$ к TBox.
- Если $A_1 \sqcap \dots \sqcap A_n \sqcap \exists r_1.B_1 \sqcap \dots \sqcap \exists r_m.B_m \sqsubseteq C$ входит в T, заменить на

$$A_1 \sqcap \dots \sqcap A_n \sqcap X \sqsubseteq C \quad \text{и} \quad \exists r_1.B_1 \sqcap \dots \sqcap \exists r_m.B_m \sqsubseteq X$$

  гда X — новое имя концепта.

- если $\exists r_1.B_1 \sqcap \dots \sqcap \exists r_m.B_m \sqsubseteq \exists r.B$ входит в T, заменить на

$$\exists r_1.B_1 \sqcap \dots \sqcap \exists r_m.B_m \sqsubseteq X \quad \text{и} \quad X \sqsubseteq \exists r.B$$

  где X — новое имя концепта.

Пример

$\mathcal{T}$:

$$\begin{aligned} A_0 &\sqsubseteq B \sqcap \exists r.B' \\ A_1 \sqcap \exists r.B &\sqsubseteq A_2 \end{aligned}$$

Шаг 1

$$\begin{aligned} A_0 &\sqsubseteq B \\ A_0 &\sqsubseteq \exists r.B' \\ A_1 \sqcap \exists r.B &\sqsubseteq A_2 \end{aligned}$$

Шаг 2

$$\begin{aligned} A_0 &\sqsubseteq B \\ A_0 &\sqsubseteq \exists r.B' \\ A_1 \sqcap X &\sqsubseteq A_2 \\ \exists r.B &\sqsubseteq X \end{aligned}$$

MED в нормальной форме

$$\text{Pericardium} \sqsubseteq \text{Tissue}$$

$$\text{Pericardium} \sqsubseteq Y$$

$$\text{Pericarditis} \sqsubseteq \text{Inflammation}$$

$$\text{Pericarditis} \sqsubseteq \exists \text{has\_loc}.\text{Pericardium}$$

$$\text{Inflammation} \sqsubseteq \text{Disease}$$

$$\text{Inflammation} \sqsubseteq \exists \text{acts\_on}.\text{Tissue}$$

$$\text{Disease} \sqcap X \sqsubseteq \text{Heartdisease}$$

$$\text{Disease} \sqcap X \sqsubseteq \text{NeedsTreatment}$$

$$\exists \text{has\_loc}.Y \sqsubseteq X \quad \exists \text{cont\_in}.\text{Heart} \sqsubseteq Y \quad Y \sqsubseteq \exists \text{cont\_in}.\text{Heart}$$

Идея алгоритма разрешающего $A \sqsubseteq_T B$

По данному $\boldsymbol{T}$ в нормальной форме, строим функции

- $\boldsymbol{S}$ отображает каждое имя концепта $\boldsymbol{A}$, входящее в $\boldsymbol{T}$, в множество имен концептов;
- $\boldsymbol{R}$ отображает каждое имя роли $\boldsymbol{r}$, входящее в $\boldsymbol{T}$, в множество пар $(\boldsymbol{B_1}, \boldsymbol{B_2})$ имен концептов.

$\boldsymbol{A} \sqsubseteq_{\boldsymbol{T}} \boldsymbol{B}$ т. и т.т., когда $\boldsymbol{B} \in \boldsymbol{S}(\boldsymbol{A})$. Интуитивно, строится интерпретация $\boldsymbol{\mathcal{I}}$ с

- $\boldsymbol{\Delta}^{\boldsymbol{\mathcal{I}}}$ — множество имен концептов в $\boldsymbol{T}$.
- $\boldsymbol{A}^{\boldsymbol{\mathcal{I}}}$ множество всех $\boldsymbol{B}$ таких что $\boldsymbol{A} \in \boldsymbol{S}(\boldsymbol{B})$;
- $\boldsymbol{r}^{\boldsymbol{\mathcal{I}}}$ множество всех $(\boldsymbol{A}, \boldsymbol{B}) \in \boldsymbol{R}(\boldsymbol{r})$.

$\boldsymbol{\mathcal{I}}$ является моделью $\boldsymbol{T}$ и $\boldsymbol{A} \sqsubseteq_{\boldsymbol{T}} \boldsymbol{B}$ т. и т.т., когда $\boldsymbol{A} \in \boldsymbol{B}^{\boldsymbol{\mathcal{I}}}$.

Алгоритм

Input: $\boldsymbol{T}$ в нормальной форме.

Инициализировать: $\boldsymbol{S(A) = \{A, \top\}}$ и $\boldsymbol{R(r) = \emptyset}$ для всех $\boldsymbol{A}$ и $\boldsymbol{r}$ в $\boldsymbol{T}$.

Применять следующие правила пока это возможно:

(simpleR) If $\boldsymbol{A' \in S(A)}$ and $\boldsymbol{A' \sqsubseteq B \in T}$ and $\boldsymbol{B \not\in S(A)}$, then

$$\boldsymbol{S(A) := S(A) \cup \{B\}.}$$

(conjR) If $\boldsymbol{A_1, A_2 \in S(A)}$ and $\boldsymbol{A_1 \sqcap A_2 \sqsubseteq B \in T}$ and $\boldsymbol{B \not\in S(A)}$, then

$$\boldsymbol{S(A) := S(A) \cup \{B\}.}$$

(rightR) If $\boldsymbol{A' \in S(A)}$ and $\boldsymbol{A' \sqsubseteq \exists r.B \in T}$ and $\boldsymbol{(A, B) \not\in R(r)}$, then

$$\boldsymbol{R(r) := R(r) \cup \{(A, B)\}.}$$

(leftR) If $\boldsymbol{(A, B) \in R(r)}$ and $\boldsymbol{B' \in S(B)}$ and $\boldsymbol{\exists r.B' \sqsubseteq A' \in T}$ and $\boldsymbol{A' \not\in S(A)}$, then

$$\boldsymbol{S(A) := S(A) \cup \{A'\}.}$$

Пример

$$\begin{array}{rrcl} (1) & A_0 & \sqsubseteq & \exists r.B \\ (2) & B & \sqsubseteq & E \\ (3) & \exists r.E & \sqsubseteq & A_1 \end{array}$$

Initialise: $S(A_0) = \{A_0\}$, $S(A_1) = \{A_1\}$, $S(B) = \{B\}$, $S(E) = \{E\}$, $R(r) = \emptyset$.

- (rightR) и (1): $R(r) = \{(A_0, B)\}$;
- (simpleR) и (2): $S(B) = \{B, E\}$;
- (leftR) и (3): $S(A_0) = \{A_0, A_1\}$;
- Дальше правила не применяются

Т.о., $R(r) = \{(A_0, B)\}$, $S(B) = \{B, E\}$, $S(A_0) = \{A_0, A_1\}$.

Следовательно, $A_0 \sqsubseteq_T A_1$.

Фрагмент MED

$$\begin{aligned}
\textbf{Pericardium (Pm)} &\sqsubseteq Y \\
\textbf{Pericarditis (Ps)} &\sqsubseteq \textbf{Inflammation (Inf)} \\
\textbf{Ps} &\sqsubseteq \exists \textbf{has_loc.Pm} \\
\textbf{Inf} &\sqsubseteq \textbf{Disease (Dis)} \\
\textbf{Disease} \sqcap X &\sqsubseteq \textbf{NeedsTreatment} \\
\exists \textbf{has_loc.} Y &\sqsubseteq X
\end{aligned}$$

Неполная трасса алгоритма (показывающая, что **Ps** $\sqsubseteq_{\textbf{MED}}$ **NeedsTreatment**):

- (simpleR): $S(\textbf{Pm}) = \{Y, \textbf{Pm}\}$, $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}\}$;
- (rightR): $R(\textbf{has\_loc}) = \{(\textbf{Ps}, \textbf{Pm})\}$,
- (leftR): $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}, X\}$
- (conjR): $S(\textbf{Ps}) = \{\textbf{Inf}, \textbf{Ps}, \textbf{Dis}, X, \textbf{NeedsTreatment}\}$

Время работы

На каждом шаге алгоритм добавляет хотя бы одно имя концепта в какой-то $S(C)$ или пару имен концептов в какой-то $R(r)$.

$\Longrightarrow$ Останавливается за полиномиальное время.

Корректность алгоритма

T в нормальной форме, S, R — выход алгоритма.

Теорема. Для всех имен концептов A, B в T: если $B \in S(A)$, то $A \sqsubseteq_T B$.

$S_0, S_1, \ldots$ и $R_0, R_1, \ldots$ отображения, построенные на i-той итерации алгоритма.

Для всех i, произвольной интерпретации $\mathcal{I} : \mathcal{I} \models T$ и любого $x \in A^{\mathcal{I}}$

- Если $B \in S_i(A)$, то $x \in B^{\mathcal{I}}$
- Если $(A, B) \in R_i(r)$, то существует $y \in \Delta^{\mathcal{I}}$ т.ч. $(x, y) \in r^{\mathcal{I}}$ и $y \in B^{\mathcal{I}}$

Доказательство корректности (на доске)

- Если $B \in S_i(A)$, то $x \in B^{\mathcal{I}}$
- Если $(A, B) \in R_i(r)$, то существует $y \in \Delta^{\mathcal{I}}$ т.ч. $(x, y) \in r^{\mathcal{I}}$ и $y \in B^{\mathcal{I}}$

(simpleR) If $A' \in S(A)$ and $A' \sqsubseteq B \in T$ and $B \notin S(A)$, then

$$S(A) := S(A) \cup \{B\}.$$

(conjR) If $A_1, A_2 \in S(A)$ and $A_1 \sqcap A_2 \sqsubseteq B \in T$ and $B \notin S(A)$, then

$$S(A) := S(A) \cup \{B\}.$$

(rightR) If $A' \in S(A)$ and $A' \sqsubseteq \exists r.B \in T$ and $(A, B) \notin R(r)$, then

$$R(r) := R(r) \cup \{(A, B)\}.$$

(leftR) If $(A, B) \in R(r)$ and $B' \in S(B)$ and $\exists r.B' \sqsubseteq A' \in T$ and $A' \notin S(A)$, then

$$S(A) := S(A) \cup \{A'\}.$$

Полнота алгоритма

T в нормальной форме, S, R — выход алгоритма.

Теорема. Для всех имен концептов A, B в T: если $A \sqsubseteq_T B$, то $B \in S(A)$.

Допустим это не так, $B \notin S(A)$. Определим интерпретацию $\mathcal{I}$

- $\Delta^{\mathcal{I}} = \{A \mid A \in CN(T)\}$
- $A^{\mathcal{I}} = \{B \mid A \in S(B)\}$;
- $r^{\mathcal{I}} = \{(A, B) \in R(r)\}$.

Тогда

- $\mathcal{I} \not\models A \sqsubseteq B$
- $\mathcal{I}$ выполняет T

(Для любого имени концепта A из T и $\mathcal{EL}$-концепта C: $A \sqsubseteq_T C \quad \Leftrightarrow \quad A \in C^{\mathcal{I}}$.)

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)
- Только “позитивная”, детерминированная информация

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

  неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)
- Только “позитивная”, детерминированная информация
  - Слоны **бывают** **серыми**. А **розовые**?

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

  неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)
- Только “позитивная”, детерминированная информация
  - Слоны **бывают** **серыми**. А **розовые**?
  - Слоны **только** серые

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)
- Только “позитивная”, детерминированная информация
  - Слоны **бывают** **серыми**. А **розовые**?
  - Слоны **только** серые
  - Самцы **не могут быть** самками

$\mathcal{EL}$: основные положения

- Простейшая дескрипционная логика

  неполный набор пропозициональных связок

- Полиномиальный алгоритм логического анализа
- Хорошо подходит для описания больших и очень больших терминологий
  - SNOMED CT
  - GO
  - NCI (почти)
- Только “позитивная”, детерминированная информация
  - Слоны **бывают** **серыми**. А **розовые**?
  - Слоны **только** серые
  - Самцы **не могут быть** самками
  - Слоны **делятся на** индийских и африканских

Дескрипционная логика $\mathcal{ALC}$: терминологическая часть

База знаний (KB)

TBox (терминология, схема)

$$\text{Man} \equiv \text{Human} \sqcap \text{Male}$$
$$\text{HappyFather} \equiv \text{Man} \sqcap \exists \text{hasChild}$$
...

ABox (assertion box, данные)

$$\text{john}: \text{Man}$$
$$(\text{john}, \text{mary}): \text{hasChild}$$
...

Система анализа

Интерфейс