Остановка

Для любой таблицы $\boldsymbol{S_0}$,

не существует бесконечной последовательности

$$\boldsymbol{S_0, S_1, S_2, \ldots}$$

т.ч. $\boldsymbol{S_{i+1}}$ получена из $\boldsymbol{S_i}$

применением одного из правил вывода

Остановка

Для любой таблицы S_0,

не существует бесконечной последовательности

$$S_0, S_1, S_2, \ldots$$

т.ч. S_{i+1} получена из S_i

применением одного из правил вывода

Proof: Никакое правило кроме $\longrightarrow_\forall$ не применяется дважды к одному узлу

$\longrightarrow_\forall$ никогда не применяется к x боле чем

число непосредственных потоков x (т.е, y т.ч. $(x, y) : R$),

что ограничено размером концепта

Правила добавляют узлы $z : D$ т.ч. D подконцепт C

Корректность

Если начав с $S_0 = \{x\colon C\}$
и применяя правила вывода
можно построить **полную непротиворечивую** таблицу S_n,
то C реализуем

S_n задает интерпретацию $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$:

- $\Delta^{\mathcal{I}}$ содержит все индивиды из S_n
- для $x \in \Delta^{\mathcal{I}}$ и имени концепта A,
  $$x \in A^{\mathcal{I}} \quad \text{т. и т.т., когда} \quad x\colon A \in S_n$$
- для $x, y \in \Delta^{\mathcal{I}}$ и имени роли R,
  $$(x, y) \in R^{\mathcal{I}} \quad \text{т. и т.т., когда} \quad (x, y)\colon R \in S_n$$

Легко убедиться, что C выполняется в $\mathcal{I}$, т.е., $C^{\mathcal{I}} \neq \emptyset$

Полнота

Выполнимость $\boldsymbol{S}$: отображение в интерпретацию (доска).

- Предположим, что $\boldsymbol{S}$ выполнима и

$$\boldsymbol{S} \longrightarrow_{\sqcap} \boldsymbol{S}', \quad \boldsymbol{S} \longrightarrow_{\forall} \boldsymbol{S}' \quad \text{или} \quad \boldsymbol{S} \longrightarrow_{\exists} \boldsymbol{S}'.$$

  Тогда $\boldsymbol{S}'$ также выполнима

- Если

$$\boldsymbol{S} \longrightarrow_{\sqcup} \boldsymbol{S}' \quad \text{и} \quad \boldsymbol{S} \longrightarrow_{\sqcup} \boldsymbol{S}''$$

  то (по крайней мере одна из) $\boldsymbol{S}'$ и $\boldsymbol{S}''$ выполнима

Т.о., начав с выполнимой таблицы, всегда можно построить полную и непротиворечивую таблицу.

Вычислительная сложность

- Рассматривая ветви по одной, можно реализовать в полиномиальной памяти
- PSPACE-полная задача

Сведение задачи истинности QBF

$$(Q_1 X_1) \dots (Q_n X_n)(G_1 \land \dots \land G_m)$$

Например

$$\forall X_1 \exists X_2 \forall X_3 ((X_1 \lor \neg X_2 \lor X_3) \land (\neg X_1 \lor X_2 \lor \neg X_3))$$

$$\begin{aligned}
&(\exists R.A \sqcap \exists R.\neg A) \sqcap \forall R.(\exists R.\top \sqcap \forall R.(\exists R.A \sqcap \exists R.\neg A)) \sqcap \\
&\quad \forall R.(A \sqcup \forall R.(A \sqcup \forall R.A)) \sqcap \\
&\qquad \forall R.(\neg A \sqcup \forall R.(\neg A \sqcup \forall R.\neg A))
\end{aligned}$$

TBox и импликации концептов

$\mathcal{T}$ — TBox и $C \sqsubseteq D$ — импликация концептов.

- $\mathcal{T}$ **влечет** $C \sqsubseteq D$ т. и т.т., когда каждая модель $\mathcal{T}$ является моделью $C \sqsubseteq D$.
- C **реализуется совместно с** $\mathcal{T}$ т. и т.т., когда существует интерпретация $\mathcal{I}$, модель $\mathcal{T}$, т.ч. $C^{\mathcal{I}} \neq \emptyset$.
- $\mathcal{T}$ **выполним** если существует модель $\mathcal{T}$.

Имеет место следующее:

- $\mathcal{T} \models C \sqsubseteq D$ т. и т.т., когда $C \sqcap \neg D$ не реализуется совместно с $\mathcal{T}$.
- $\mathcal{T}$ выполним т. и т.т., когда A реализуется совместно с $\mathcal{T}$ (A новое имя).

Ациклические $\mathcal{ALC}$-терминологии

$\mathcal{ALC}$ терминология это конечное множество определения вида

$$A \equiv C, \quad A \sqsubseteq C$$

таких что никакое имя концепта не определяется более одного раза. и в которой нет (явных или неявных) циклических определений.

Ациклические $\mathcal{ALC}$-терминологии

$\mathcal{ALC}$ терминология это конечное множество определения вида

$$A \equiv C, \quad A \sqsubseteq C$$

таких что никакое имя концепта не определяется более одного раза. и в которой нет (явных или неявных) циклических определений.

НУО, все определения имеют вид

$$A \equiv C$$

Ациклические $\mathcal{ALC}$-терминологии

$\mathcal{ALC}$ терминология это конечное множество определения вида

$$A \equiv C, \quad A \sqsubseteq C$$

таких что никакое имя концепта не определяется более одного раза. и в которой нет (явных или неявных) циклических определений.

НУО, все определения имеют вид

$$A \equiv C$$

Заменим

$$A \sqsubseteq C$$

$$A \equiv C \sqcap A',$$

где A' — новое имя концепта

Развёртка

$\mathcal{T}$ — ациклическая $\mathcal{ALC}$-терминология

$\mathsf{Unfold}_{\mathcal{T}}(A) = A,$	Если A не определен в $\mathcal{T}$
$\mathsf{Unfold}_{\mathcal{T}}(A) = \mathsf{Unfold}_{\mathcal{T}}(D),$	Если $A \equiv D \in \mathcal{T}$
$\mathsf{Unfold}_{\mathcal{T}}(C_1 \sqcap C_2) = \mathsf{Unfold}_{\mathcal{T}}(C_1) \sqcap \mathsf{Unfold}_{\mathcal{T}}(C_2),$	
$\mathsf{Unfold}_{\mathcal{T}}(\neg C) = \neg\mathsf{Unfold}_{\mathcal{T}}(C),$	
$\mathsf{Unfold}_{\mathcal{T}}(\exists r.C) = \exists r.\mathsf{Unfold}_{\mathcal{T}}(C),$	
$\ldots$	

$$\mathrm{T} \models \mathrm{C} \sqsubseteq \mathrm{D} \quad \text{т. и т.т., когда} \quad \emptyset \models \mathsf{Unfold}_{\mathrm{T}}(\mathrm{C}) \sqsubseteq \mathsf{Unfold}_{\mathrm{T}}(\mathrm{D})$$

т. и т.т., когда концепт

$$\mathsf{Unfold}_{\mathrm{T}}(\mathrm{C}) \sqcap \neg\mathsf{Unfold}_{\mathrm{T}}(\mathrm{D})$$

не реализуем

Вычислительная сложность

- $\mathsf{Unfold}_{\mathbf{T}}(\mathbf{C})$ может иметь экспоненциальный размер:

$$
\begin{array}{lcl}
C & \equiv & \exists r.C_1 \sqcap \exists s.C_1 \\
C_1 & \equiv & \exists r.C_2 \sqcap \exists s.C_2 \\
C_2 & \equiv & \exists r.C_3 \sqcap \exists s.C_3 \\
 & \ldots & \\
C_{n-1} & \equiv & \exists r.C_n \sqcap \exists s.C_n
\end{array}
$$

- Полиномиальная глубина
- PSPACE-алгоритм
  - Подставлять определение когда необходимо

Логический анализ

В любой интерпретации $\mathcal{I}$ для любых $\boldsymbol{E}$ и $\boldsymbol{F}$,

$\mathcal{I} \models \boldsymbol{E} \sqsubseteq \boldsymbol{F}$ т. т.т., когда $\mathcal{I} \models \top \sqsubseteq \neg \boldsymbol{E} \sqcup \boldsymbol{F}$

Т.о., можно предположить, что $\boldsymbol{T}$ содержит только аксиомы вида $\top \sqsubseteq \boldsymbol{D}$

Логический анализ

$$S \;\longrightarrow_U\; S \cup \{\, x\colon D \,\}$$

если (a) $\top \sqsubseteq D \in T$

(b) $x \in S$

(c) $x\colon D \notin S$

Добавление этого правила может привести к тому, что построение таблицы **не закончится**

Пример

$$\mathcal{T} = \{ \top \sqsubseteq \exists R.C \}$$

$$\begin{aligned}
&S_0 && = \{ x_0 \colon \top \} \\
&S_0 \rightarrow_U S_1 && = S_0 \cup \{ x_0 \colon \exists R.C \} \\
&S_1 \rightarrow_\exists S_2 && = S_1 \cup \{ (x_0, x_1) \colon R, \quad x_1 \colon C \} \\
&S_2 \rightarrow_U S_3 && = S_2 \cup \{ x_1 \colon \exists R.C \} \\
&S_3 \rightarrow_\exists S_4 && = S_3 \cup \{ (x_1, x_2) \colon R, \quad x_2 \colon C \} \\
&S_4 \rightarrow_U S_5 && = S_4 \cup \{ x_2 \colon \exists R.C \} \\
&\qquad \ldots && \ldots
\end{aligned}$$

Процесс порождает **бесконечную модель** но можно легко построить и конечную.

Можно модифицировать правило $\rightarrow_\exists$ таким образом, что

полученный алгоритм всегда **заканчивает работу**

(используя *блокировку*)

Блокировка

- Узел x **заблокирован** узлом y если

$$\{C \mid x : C \in S_i\} \subseteq \{D \mid y : D \in S_i\}$$

- Правила вывода применяются только к **не заблокированным** узлам.
- Ветви в таблице могут быть экспоненциальной длины
- EXPTIME-полная задача

О сложности

Реализуемость концептов относительно $\mathcal{ALC}$-теорий ExpTime-полна.

- Нет гарантии, что существующие (или будущие) реализации закончат работу.
- Тем не менее, существуют системы (FACT, PELLET, RACER) успешно работающие на практике.

Экспрессивные Дескрипционные Логики

Расширения $\mathcal{ALC}$

Qualified number restrictions: если C концепт а r роль, то

$$(\leq n\ r.C), \quad (\geq n\ r.C)$$

являются концептами.

Интерпретация задается

- $(\leq n\ r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \ |\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}| \leq n\ \}$
- $(\geq n\ r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \ |\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}| \geq n\}$

Примеры

- $(\geq 3$ **hasChild.Male**$)$ класс объектов у которых по крайней мере трое детей-самцов.
- $(\leq 2$ **hasChild.Male**$)$ класс объектов у которых не более двух детей-самцов.

Расширения $\mathcal{ALC}$

Обратные роли: если r имя роли, то r^- это роль, обратная к r. Интерпретация задается как

- $(r^-)^\mathcal{I} = \{(y,x) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (x,y) \in r^\mathcal{I}\}$.

r^- может входить везде, куда может входить r.

Примеры

- $\exists$**has_child**$^-$**.Gardener** класс объектов, у кого родитель садовник.
- $(\geq 3$**parent**$^-$**.Gardener**$)$ класс объектов, у кого по крайней мере трое детей-садовников.

Замечание. логический анализ в $\mathcal{EL}$ с обратными ролями ExpTime-труден (полон).

Расширения $\mathcal{ALC}$

Транзитивные роли: Декларация *transitive*(r) означает, что отношение r является транзитивны

- $\mathcal{I} \models$ *transitive*(r) т. и т.т., когда $r^{\mathcal{I}}$ транзитивно, т.е., для всех x, y, $z \in \Delta^{\mathcal{I}}$ т.ч. $(x, y) \in r^{\mathcal{I}}$ и $(y, z) \in r^{\mathcal{I}}$ мы имеем $(x, z) \in r^{\mathcal{I}}$.

Пример

- Роль “is part of” часто объявляется транзитивной

Иерархия ролей: декларация $r \sqsubseteq s$ означает, что отношение r включено в s. Т.о.,

- $\mathcal{I} \models r \sqsubseteq s$ т. и т.т. $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

Пример

- hasSon $\sqsubseteq$ hasChild

Расширения $\mathcal{ALC}$

Классы, содержащие в точности один объект (синглетоны). Для выражения этого свойства, в $\mathcal{ALC}$ вводятся номиналы.

Номиналы: a, b — имена индивидов. Индивиды ассоциированы с элементами домена.

Интерпретация $\mathcal{I}$ расширяется на элементы: $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, $b^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, и т. д.

Если a — индивид, то $\{a\}$ — номинал. $\{a_1, \ldots, a_n\}$ — множество номиналов.

Интерпретация $\mathcal{I}$:

- $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$;
- $\{a_1, \ldots, a_n\}^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$.

Расширения $\mathcal{ALC}$

В $\mathcal{ALCO}$ выражения $\{a\}$ и $\{a_1, \dots, a_n\}$ используются как концепты.

Пример:

- $\exists$**citizen_of.**$\{$**France**$\}$ (гражданин Франции).
- $\exists$**citizen_of.**$\{$**France**, **Ireland**$\}$ (гражданин Франции или Ирландии).
- $\exists$**has_colour.**$\{$**Green**$\}$ (зеленые объекты).
- $\exists$**student_of.**$\{$**Liverpool_University**$\}$ (студенты ливерпульского университета).
- Можно определить **Colour** как набор цветов

$$\textbf{Colour} \equiv \{\textbf{red}, \textbf{yellow}, \dots, \textbf{green}\}$$

и писать

$$\top \sqsubseteq \forall \textbf{has\_colour}.\textbf{Colour}.$$

Выразительная дескрипционная логика $\mathcal{SHOIQ}$

Расширение $\mathcal{ALC}$

- qualified number restrictions,
- обратные роли
- иерархии ролей
- транзитивные роли
- номиналы

называется $\mathcal{SHOIQ}$. Лежит в основе языка Web-онтологий OWL-DL.