Синтаксическая локальность

$$\mathcal{C}_{\Sigma}^{\perp} ::= A^{\perp} \mid (\neg C^{\top}) \mid (C \sqcap C^{\perp}) \mid (\exists R^{\perp}.C) \mid (\exists R.C^{\perp}) \mid (\geq nR^{\perp}.C) \mid (\geq nR.C^{\perp});$$

$$\mathcal{C}_{\Sigma}^{\top} ::= (\neg C^{\perp}) \mid (C_1^{\top} \sqcap C_2^{\perp}),$$

где $A^{\perp}, R^{\perp} \notin \Sigma$.

Аксиома **синтаксические локальна** относительно Σ если она имеет вид

- $R^{\perp} \sqsubseteq R$
- **Trans**$(R^{\perp})$
- $C^{\perp} \sqsubseteq C$
- $C \sqsubseteq C^{\top}$

Синтаксическая локальность

$$\mathcal{C}_{\Sigma}^{\bot} ::= A^{\bot} \mid (\neg C^{\top}) \mid (C \sqcap C^{\bot}) \mid (\exists R^{\bot}.C) \mid (\exists R.C^{\bot}) \mid (\geq n R^{\bot}.C) \mid (\geq n R.C^{\bot});$$

$$\mathcal{C}_{\Sigma}^{\top} ::= (\neg C^{\bot}) \mid (C_1^{\top} \sqcap C_2^{\bot}),$$

где $A^{\bot}, R^{\bot} \notin \Sigma$.

Аксиома **синтаксические локальна** относительно Σ если она имеет вид

- $R^{\bot} \sqsubseteq R$
- **Trans**$(R^{\bot})$
- $C^{\bot} \sqsubseteq C$
- $C \sqsubseteq C^{\top}$

Теорема. Если теория синтаксически локальна, она семантически локальна.

Синтаксическая локальность

$$\mathcal{C}_{\Sigma}^{\perp} ::= A^{\perp} \mid (\neg C^{\top}) \mid (C \sqcap C^{\perp}) \mid (\exists R^{\perp}.C) \mid (\exists R.C^{\perp}) \mid (\geq nR^{\perp}.C) \mid (\geq nR.C^{\perp});$$

$$\mathcal{C}_{\Sigma}^{\top} ::= (\neg C^{\perp}) \mid (C_1^{\top} \sqcap C_2^{\perp}),$$

где $A^{\perp}, R^{\perp} \notin \Sigma$.

Аксиома **синтаксические локальна** относительно Σ если она имеет вид

- $R^{\perp} \sqsubseteq R$
- **Trans**$(R^{\perp})$
- $C^{\perp} \sqsubseteq C$
- $C \sqsubseteq C^{\top}$

Теорема. Если теория синтаксически локальна, она семантически локальна.

Обратное неверно

Алгоритм выделения модуля для Σ

$\mathcal{M} := \emptyset$, $\mathcal{Q} := \mathcal{T}$.

while $Q \neq \emptyset$ **do**

$\alpha =$ select_axiom$(\mathcal{Q})$;

if locality_test$(\alpha, \Sigma \cup$ **sig**$(\mathcal{M}))$ **then**

$\mathcal{Q} := \mathcal{Q} \setminus \{\alpha\}$

else

$\mathcal{M} := \mathcal{M} \cup \{\alpha\}$

$\mathcal{Q} := \mathcal{T} \setminus \mathcal{M}$

end if

end while

return $\mathcal{M}$

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$
- $\mathcal{T} \setminus \mathcal{M}$ синтаксически локальна для Σ;

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$
- $\mathcal{T} \setminus \mathcal{M}$ синтаксически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ семантически локальна для Σ;

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$
- $\mathcal{T} \setminus \mathcal{M}$ синтаксически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ семантически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$
- $\mathcal{T} \setminus \mathcal{M}$ синтаксически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ семантически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;
- $\mathcal{T}$ является модельно консервативным расширением $\mathcal{M}$;

Свойства модулей

$\mathcal{T}$ — теория, Σ — сигнатура, $\mathcal{M} \subseteq \mathcal{T}$ — модуль.

- $\Sigma \subseteq \mathbf{sig}(\mathcal{M})$
- $\mathcal{T} \setminus \mathcal{M}$ синтаксически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ семантически локальна для Σ;
- $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;
- $\mathcal{T}$ является модельно консервативным расширением $\mathcal{M}$;
- $\mathcal{T}$ является консервативным расширением $\mathcal{M}$ относительно $\mathbf{sig}(\mathcal{M})$.

Ациклические $\mathcal{EL}$-терминологии

Для ациклических $\mathcal{EL}$-терминологий, возможно найти **минимальный** модуль $\mathcal{M} \subseteq \mathcal{T}$ т.ч.

- $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;
- $\mathcal{T}$ является консервативным расширением $\mathcal{M}$ относительно **sig**$(\mathcal{M})$.

Если $\mathcal{T}$ не содержит тривиальных аксиом

$$\boldsymbol{A} \sqsubseteq \top \quad \boldsymbol{A} \equiv \top$$

эти понятия совпадают.

Ациклические $\mathcal{EL}$-терминологии

Для ациклических $\mathcal{EL}$-терминологий, возможно найти **минимальный** модуль $\mathcal{M} \subseteq \mathcal{T}$ т.ч.

- $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;
- $\mathcal{T}$ является консервативным расширением $\mathcal{M}$ относительно $\mathbf{sig}(\mathcal{M})$.

Если $\mathcal{T}$ не содержит тривиальных аксиом

$$\boldsymbol{A} \sqsubseteq \top \quad \boldsymbol{A} \equiv \top$$

эти понятия совпадают.

Если любую **одноэлементную** интерпретацию символов Σ можно расширить до интерпретации $\mathcal{T} \setminus \mathcal{M}$, то $\mathcal{T} \setminus \mathcal{M}$ является семантической тавтологией для Σ;

Элементы данных и ABox

Реляционные базы данных

Мы рассматриваем реляционные базы данных (RDB) как наборы утверждений вида

$$\boldsymbol{A(a), \quad R(a,b),}$$

где $\boldsymbol{A}$ это свойство (имя концепта), а $\boldsymbol{R}$ — бинарное отношение (имя роли), а $\boldsymbol{a}$, $\boldsymbol{b}$ имена индивидов.

- $\boldsymbol{A(a)}$ утверждает, что $\boldsymbol{a}$ это пример $\boldsymbol{A}$;
- $\boldsymbol{R(a,b)}$ утверждает, что $\boldsymbol{(a,b)}$ это пример $\boldsymbol{R}$.

Пример

Например, рассмотрим следующие

- имена концептов **University**, **BritishUnivty**, **Student**,
- имена ролей **registered_at**, **student_at**.
- имена индивидов **LU**, **MU**, **CMU**, **Tim**, **Tom**, и **Rob** .

Рассмотрим теперь RDB ID

- $\textbf{University}(\textbf{LU}), \quad \textbf{University}(\textbf{MU}), \quad \textbf{University}(\textbf{CMU});$
- $\textbf{BritishUnivty}(\textbf{LU}), \quad \textbf{BritishUnivty}(\textbf{MU});$
- $\textbf{Student}(\textbf{Tim}), \quad \textbf{Student}(\textbf{Tom});$
- $\textbf{registered\_at}(\textbf{Tim}, \textbf{LU}), \quad \textbf{registered\_at}(\textbf{Tom}, \textbf{MU});$
- $\textbf{student\_at}(\textbf{Tim}, \textbf{LU}), \quad \textbf{student\_at}(\textbf{Rob}, \textbf{CMU}).$

Запросы к базам данных

Рассмотрим следующие запросы и ответы к ним

- Найти все университеты.

  В синтаксисе логики первого порядка,

  $$q(x) = \textbf{University}(x).$$

  Ответ: **LU**, **MU**, **CMU**.

- Найти все пары (a, b) т.ч. a зарегистрирован в b.

  $$q(x, y) = \textbf{registered\_at}(x, y).$$

  Ответ: (**Tim**, **LU**), (**Tom**, **MU**);

- Найти всех a, которые где-то зарегистрированы

  $$q(x) = \exists y.\ \textbf{registered\_at}(x, y).$$

  Ответ: **Tim**, **Tom**.

University(LU)
University(MU)
University(CMU)
BritishUnivty(LU)
BritishUnivty(MU)
Student(Tim)
Student(Tom)
registered_at(Tim, LU)
registered_at(Tom, MU)
student_at(Tim, LU)
student_at(Rob, CMU).

Запросы к базам данных

- Найти все небританские университеты

$$q(x) = \textbf{University}(x) \wedge \neg\textbf{BritishUnivty}(x).$$

Ответ: **CMU**;

- Найти все пары (a, b) т.ч. a зарегистрирован в b но a не является студентом в b.

$$q(x, y) = \textbf{registered\_at}(x, y) \wedge \neg\textbf{student\_at}(x, y).$$

Ответ: $(\textbf{Tom}, \textbf{MU})$;

- В каждом ли университете есть студенты?

$$q = \forall y.\ (\textbf{University}(y) \rightarrow \exists x.\ \textbf{student\_at}(x, y)).$$

ответ: Нет

$\textbf{University}(\textbf{LU})$
$\textbf{University}(\textbf{MU})$
$\textbf{University}(\textbf{CMU})$
$\textbf{BritishUnivty}(\textbf{LU})$
$\textbf{BritishUnivty}(\textbf{MU})$
$\textbf{Student}(\textbf{Tim})$
$\textbf{Student}(\textbf{Tom})$
$\textbf{registered\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{registered\_at}(\textbf{Tom}, \textbf{MU})$
$\textbf{student\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{student\_at}(\textbf{Rob}, \textbf{CMU}).$

Запросы к базам данных

- Верно ли, что CMU является университетом, но не британским университетом?

$$q = \textbf{University}(\textbf{CMU}) \wedge \neg\textbf{BritishUnivty}(\textbf{CMU})$$

Ответ: Да

- Правда ли, что **LU** — университет?

$$q = \textbf{University}(\textbf{LU}).$$

Ответ: Да.

- Сколько небританских университетов?

Ответ: **в точности один**

University(LU)
University(MU)
University(CMU)
BritishUnivty(LU)
BritishUnivty(MU)
Student(Tim)
Student(Tom)
registered_at(Tim, LU)
registered_at(Tom, MU)
student_at(Tim, LU)
student_at(Rob, CMU).

Формальное определение запроса к БД

RDB D задает интерпретацию $\mathcal{I}_D$:

- $\Delta^{\mathcal{I}}$ множество имен индивидов в D;
- $d \in A^{\mathcal{I}}$ т. и т.т., когда $A(d) \in D$;
- $(d_1, d_2) \in R^{\mathcal{I}}$ т. и т.т., когда $R(d_1, d_2) \in D$.

Для данных RDB D и запроса $q = q(x_1, \ldots, x_n)$
answer(D, q) это множество $(a_1, \ldots, a_n)$ т.ч. $\mathcal{I}_D \models q(a_1, \ldots, a_n)$.

Для запроса q без свободных переменных, мы говорим что

- ответ, задаваемый D, "Да", если $\mathcal{I}_D \models q$;
- ответ, задаваемый D, "Нет" если $\mathcal{I}_D \not\models q$ (или $\mathcal{I}_D \models \neg q$).

Итого

- Каждая RDB D описывает **в точности одну модель**;
- Если $A(a) \notin D$, то a НЕ является примером A в $\mathcal{I}_D$ и ответ на запрос $A(a)$ будет "Нет".
- Если $R(a, b) \notin D$, то (a, b) НЕ является примером R в $\mathcal{I}_D$ и ответом на запрос $R(a, b)$ будет "Нет".
- Т.о., предполагается **полное знание** о примерах концептов и ролей.
- Значит, при вычислении ответов на запросы принимается **предположение о замкнутости мира**. Если что-то не описано в RDB, это неверно.

Сложность запросов к RDBs

Рассмотрим, для простоты, запросы без свободных переменных. Есть два способа измерить вычислительную сложность запросов к RDBs:

- Сложность относительно данных: зафиксировать запрос и оценивать время/память необходимые для вычисления ответа на запрос как функцию от размера данных.
- Комбинированная сложность: оценивать время/память необходимы для вычисления ответа на запрос как функцию от размера данных и размера запроса.

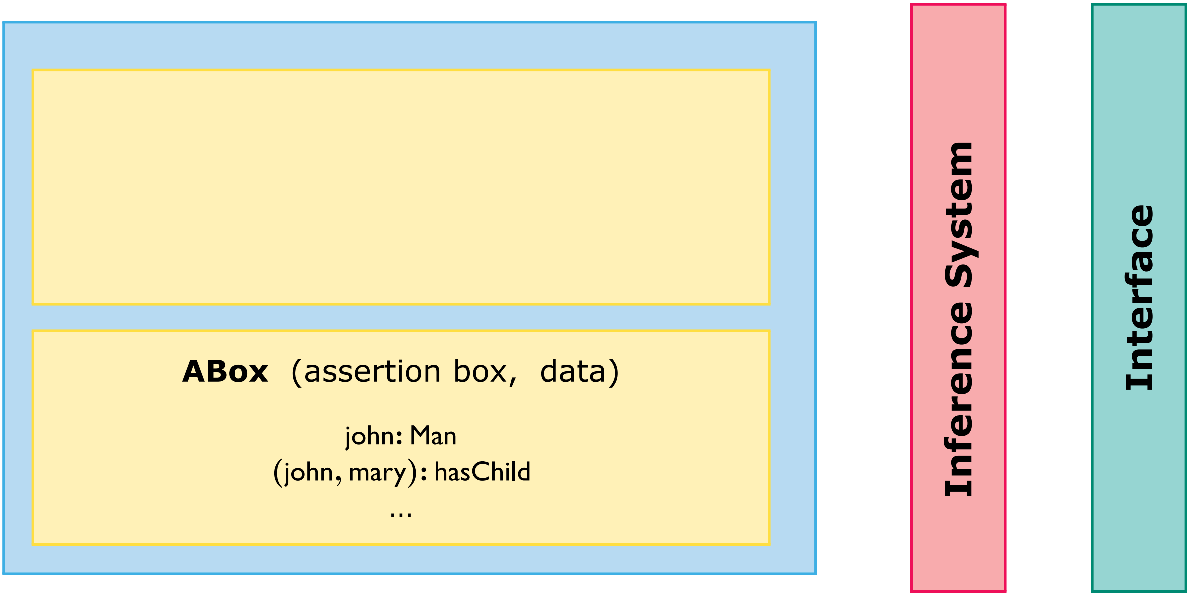
Сложность относительно данных считается более подходящей так как размер запросов обычно существенно меньше размера данных, представленных в БД.

SQL:

Сложность относительно данных LogSpace-полна,

комбинированная сложность — PSpace-полна.

ABox: Данные и предположение об открытости мире



ABox

ABox (assertion box) это конечное множество утверждений вида

$$\boldsymbol{a} : \boldsymbol{A}, \quad (\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{R},$$

где $\boldsymbol{A}$ это имя концепта, $\boldsymbol{R}$ — имя роли, а $\boldsymbol{a}$, $\boldsymbol{b}$ — имена индивидов.

- $\boldsymbol{a} : \boldsymbol{A}$ утверждает, что $\boldsymbol{a}$ это пример $\boldsymbol{A}$;
- $(\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{R}$ утверждает, что $(\boldsymbol{a}, \boldsymbol{b})$ это пример $\boldsymbol{R}$.

- Отказ от предположения о закрытости мира
- ABox описывает то, **что мы знаем**, но мы можем чего-то не знать!

Открытый мир

- Каждый ABox описывает **класс** интерпретаций совместных с ABox;
- Несмотря на то, что $\boldsymbol{A(a)} \notin \boldsymbol{\mathcal{A}}$, $\boldsymbol{a}$ все равно может быть примером $\boldsymbol{A}$ в какой-то интерпретации $\boldsymbol{\mathcal{A}}$. Значит, ответом на запрос $\neg \boldsymbol{A(a)}$ будет "не знаю".
- Аналогично, несмотря на то, что $\boldsymbol{R(a,b)} \notin \boldsymbol{\mathcal{A}}$, $\boldsymbol{(a,b)}$ может быть примером $\boldsymbol{R}$ в какой-то интерпретации $\boldsymbol{\mathcal{A}}$. Ответ на запрос $\neg \boldsymbol{R(a,b)}$ "не знаю".
- Предполагается **неполное знание** о примерах концептов и ролей
- **Гипотеза об открытости мира**.

Пример

Используем ту же RDB **ID** что и до этого, но обозначаем $\mathrm{ID}_{\boldsymbol{A}}$, чтобы отличать от ID:

- $\mathbf{LU} : \mathbf{University}, \quad \mathbf{MU} : \mathbf{University}, \quad \mathbf{CMU} : \mathbf{University};$
- $\mathbf{LU} : \mathbf{BritishUnivty}, \quad \mathbf{MU} : \mathbf{BritishUnivty};$
- $\mathbf{Tim} : \mathbf{Student}, \quad \mathbf{Tom} : \mathbf{Student};$
- $(\mathbf{Tim}, \mathbf{LU}) : \mathbf{registered\_at}, \quad (\mathbf{Tom}, \mathbf{MU}) : \mathbf{registered\_at};$
- $(\mathbf{Tim}, \mathbf{LU}) : \mathbf{student\_at}, \quad (\mathbf{Rob}, \mathbf{CMU}) : \mathbf{student\_at}.$

Запросы к ABox

Рассмотрим следующие запросы и *точные ответы*, данные при помощи $\mathrm{ID}_{\boldsymbol{A}}$

- Найти все университеты (более точно, все объекты, про которые сказано, что они являются университетами).

$$q(x) = \textbf{University}(x).$$

Ответ: $\textbf{LU}, \textbf{MU}, \textbf{CMU}$ (такой же как и с помощью ID).

- Найти все пары $(\boldsymbol{a}, \boldsymbol{b})$ т. ч. $\boldsymbol{a}$ зарегистрирован в $\boldsymbol{b}$ (более точно, все пары $(\boldsymbol{a}, \boldsymbol{b})$ про которые сказано, что $\boldsymbol{a}$ зарегистрирован в $\boldsymbol{b}$).

$$q(x, y) = \textbf{registered\_at}(x, y).$$

Ответ: $(\textbf{Tim}, \textbf{LU}), (\textbf{Tom}, \textbf{MU})$ (также как ID);

University(**LU**)
University(**MU**)
University(**CMU**)
BritishUnivty(**LU**)
BritishUnivty(**MU**)
Student(**Tim**)
Student(**Tom**)
registered_at(**Tim**, **LU**)
registered_at(**Tom**, **MU**)
student_at(**Tim**, **LU**)
student_at(**Rob**, **CMU**).

Запросы к ABox

- найти все a, которые где-то зарегистрированы (точнее, все a про которые сказано, что они где-то зарегистрированы)

$$q(x) = \exists y.\ \textbf{registered\_at}(x, y).$$

Ответ: **Tim, Tom** (также как ID).

- Найти все университеты, которые не являются британскими университетами (более точно, найти все объекты, про которые сказано, что они не являются британскими университетами)

$$q(x) = \textbf{University}(x) \wedge \neg\textbf{BritishUnivty}(x).$$

Ответ: пустой список.

University(**LU**)
University(**MU**)
University(**CMU**)
BritishUnivty(**LU**)
BritishUnivty(**MU**)
Student(**Tim**)
Student(**Tom**)
registered_at(**Tim**, **LU**)
registered_at(**Tom**, **MU**)
student_at(**Tim**, **LU**)
student_at(**Rob**, **CMU**).

Запросы к ABox

- Найти все пары (a, b) т.ч. a зарегистрирован в b и a не является студентом в b (более точно, найти все пары (a, b) про которые сказано, что a зарегистрирован в b и a не является студентом в b).

$$\mathrm{q}(\mathrm{x},\mathrm{y}) = \textbf{registered\_at}(\mathrm{x},\mathrm{y}) \wedge \neg\textbf{student\_at}(\mathrm{x},\mathrm{y}).$$

Ответ: пустой список.

- В каждом ли университете есть студент?

$$\mathrm{q} = \forall \mathrm{y}.\ (\textbf{University}(\mathrm{y}) \to \exists \mathrm{x}.\ \textbf{student\_at}(\mathrm{x},\mathrm{y}))$$

Ответ: не знаю.

$\textbf{University}(\textbf{LU})$
$\textbf{University}(\textbf{MU})$
$\textbf{University}(\textbf{CMU})$
$\textbf{BritishUnivty}(\textbf{LU})$
$\textbf{BritishUnivty}(\textbf{MU})$
$\textbf{Student}(\textbf{Tim})$
$\textbf{Student}(\textbf{Tom})$
$\textbf{registered\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{registered\_at}(\textbf{Tom}, \textbf{MU})$
$\textbf{student\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{student\_at}(\textbf{Rob}, \textbf{CMU}).$

Запросы к ABox

- Верно ли, что CMU это университет, но не британский университет?

$$q = \textbf{University}(\textbf{CMU}) \wedge \neg\textbf{BritishUnivty}(\textbf{CMU}).$$

  Ответ: не знаю.

- Верно ли, что **LU** это университет?

$$q = \textbf{University}(\textbf{LU}).$$

  Ответ: Да.

$\textbf{University}(\textbf{LU})$
$\textbf{University}(\textbf{MU})$
$\textbf{University}(\textbf{CMU})$
$\textbf{BritishUnivty}(\textbf{LU})$
$\textbf{BritishUnivty}(\textbf{MU})$
$\textbf{Student}(\textbf{Tim})$
$\textbf{Student}(\textbf{Tom})$
$\textbf{registered\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{registered\_at}(\textbf{Tom}, \textbf{MU})$
$\textbf{student\_at}(\textbf{Tim}, \textbf{LU})$
$\textbf{student\_at}(\textbf{Rob}, \textbf{CMU}).$

Формальное определение запросов к ABox

Интерпретация $\mathcal{I}$ является моделью для ABox $\mathcal{A}$ если:

- все имена индивидов a из $\mathcal{A}$ интерпретируются как $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
- $a^{\mathcal{I}} \in A^{\mathcal{I}}$ когда $a : A \in \mathcal{A}$;
- $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ когди $R(a, b) \in \mathcal{A}$.

Для данного ABox $\mathcal{A}$ и запроса $q = q(x_1, \ldots, x_n)$, точным ответом на q относительно $\mathcal{A}$,

$$\textbf{CertAnswer}(\mathcal{A}, q),$$

являются все $(a_1, \ldots, a_n)$ т.ч. $\mathcal{I} \models q(a_1, \ldots, a_n)$, для всех моделей $\mathcal{I}$ ABox $\mathcal{A}$.

Для запроса q без свободных переменных

- ответом по отношению к $\mathcal{A}$ будет ″да″ если $\mathcal{I} \models q$ для всех моделей $\mathcal{I}$ of $\mathcal{A}$;
- ответом по отношению к $\mathcal{A}$ будет ″нет″ если $\mathcal{I} \not\models q$ для всех моделей $\mathcal{I}$ of $\mathcal{A}$.
- Иначе, ответом будет ″не знаю″.