

Математическая логика и логическое программирование

mk.cs.msu.ru → Лекционные курсы
→ Математическая логика и логическое программирование (3-й поток)

Блок 46

Отрицание в логическом программировании
Допущение замкнутости мира

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2022/2023, осенний семестр

Пример (ХЛП \mathcal{P})

птица(**орёл**);

летает(**орёл**);

птица(**пингвин**);

летает(**самолёт**);

Послав подходящий запрос, можно узнать у \mathcal{P} , какие по её мнению есть летающие птицы:

?птица(X), летает(X)

На этот запрос есть ровно один правильный (и он же ровно один SLD-вычислимый) ответ: летающая птица — это орёл

{ X /**орёл**}

Если бы в телах можно было использовать не только атомы, но и их **отрицания**, то запрос о нелетающей птице выглядел бы естественно:

?птица(X), \neg летает(X)

\neg — полезная логическая связка, и хотелось бы его использовать в логических программах

А почему до сих пор было запрещено так использовать \neg ?

Пример (ХЛП \mathcal{P} и запрос \mathcal{Q})

птица(**орёл**); летает(**орёл**);
птица(**пингвин**); летает(**самолёт**);
?птица(X), \neg летает(X)

Применив любой известный метод проверки общезначимости формул логики предикатов, можно легко убедиться, что

$\not\models$ птица(**орёл**) & птица(**пингвин**) &
летает(**орёл**) & летает(**самолёт**) \rightarrow
птица(**пингвин**) & \neg летает(**пингвин**)

А значит (по **теореме о логическом следствии**),

$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{летает}(\mathbf{самолёт}) \end{array} \right\} \not\models \text{птица}(\mathbf{пингвин}) \ \& \ \neg \text{летает}(\mathbf{пингвин})$

То есть программа \mathcal{P} не считает пингвина нелетающей птицей

И летающей птицей программа \mathcal{P} пингвина тоже не считает:

$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{летает}(\mathbf{самолёт}) \end{array} \right\} \not\models \text{птица}(\mathbf{пингвин}) \ \& \ \text{летает}(\mathbf{пингвин})$

При этом \mathcal{P} уверена, что пингвин — это птица:

$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{летает}(\mathbf{самолёт}) \end{array} \right\} \models \text{птица}(\mathbf{пингвин})$

Всё-таки летает ли пингвин?

Пример (ХЛП \mathcal{P} и запрос \mathcal{Q})

птица(**орёл**);

летает(**орёл**);

птица(**пингвин**);

летает(**самолёт**);

?птица(X), \neg летает(X)

У \mathcal{P} как у системы формул есть модель, в которой утверждается, что пингвин не летает

Значит, **достоверно** заключить, что пингвин летает, \mathcal{P} не может

Но у \mathcal{P} есть и модель, в которой утверждается, что пингвин летает

Например, такой моделью \mathcal{P} является **\mathcal{H} -интерпретация**

$$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{ птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{ летает}(\mathbf{самолёт}), \text{ летает}(\mathbf{пингвин}) \end{array} \right\}$$

Значит, **достоверно** заключить, что пингвин не летает, \mathcal{P} тоже не может

Строго говоря, ответ «пингвин — это нелетающая птица» и должен быть неправильным: если невозможно доказать, что пингвин не летает, то из этого не следует, что пингвин летает

Пример (ХЛП \mathcal{P} и запрос \mathcal{Q})

птица(**орёл**);

летает(**орёл**);

птица(**пингвин**);

летает(**самолёт**);

?птица(X), \neg летает(X)

Можно заставить программиста **явно** записать факт

\neg летает(**пингвин**)

Но тогда программисту придётся аналогично перечислить всё то, что, согласно его знаниям, не умеет летать (*и это ОЧЕНЬ много фактов*)

Чтобы программист был освобождён от бремени перечисления всего того, чего нет, принято использовать **допущение замкнутости мира** (предположение о замкнутости мира; closed world assumption; **CWA**), имеющее несколько строгих формулировок с таким общим смыслом: **если утверждение невозможно доказать, то оно считается ложным**

Пример (ХЛП \mathcal{P} и запрос \mathcal{Q})

птица(**орёл**); летает(**орёл**);
птица(**пингвин**); летает(**самолёт**);
?птица(X), \neg летает(X)

Аналогичный принцип (*презумпция невиновности*) используется в суде: обвиняемый не обязан явно опровергать весь спектр гипотетически возможных фактов, которые могли бы доказать его вину, и считается невиновным, если невозможно явно доказать обратное

Если программа \mathcal{P} (*обвинитель*) не может доказать, что пингвин (*виновен в том, что*) летает, то он не летает (*невиновен*)

В этом смысле (\models_{cwa}) пингвин — нелетающая птица:

$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{ птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{ летает}(\mathbf{самолёт}) \end{array} \right\} \models_{cwa} \text{птица}(\mathbf{пингвин}) \ \& \ \neg \text{летает}(\mathbf{пингвин})$

Попробуем предоставить хотя бы один вариант строгой корректной формулировки CWA

Моделью хорновской логической программы \mathcal{P} будем называть модель системы формул $\Phi_{\mathcal{P}}$

Для семейства \mathcal{I} \mathcal{H} -интерпретаций записью $\bigcap \mathcal{I}$ будем обозначать \mathcal{H} -интерпретацию $\{A \mid \forall I \in \mathcal{I} : A \in I\}$

Лемма (о пересечении моделей ХЛП). Для любой ХЛП \mathcal{P} и любого семейства \mathcal{I} её эрбрановских моделей интерпретация $\bigcap \mathcal{I}$ также является моделью ХЛП \mathcal{P}

Доказательство.

Предположим *от противного*, что $\bigcap \mathcal{I}$ не является моделью \mathcal{P}

Тогда существует правило $\forall \dots (B_1 \& \dots \& B_k \rightarrow A) \in \Phi_{\mathcal{P}}$, такое что $\bigcap \mathcal{I} \not\models \forall \dots (B_1 \& \dots \& B_k \rightarrow A)$

По семантике \forall , существует набор предметов \tilde{d}^n , такой что $\bigcap \mathcal{I} \not\models (B_1 \& \dots \& B_k \rightarrow A)[\tilde{d}^n]$

Для технической простоты без ограничения общности положим, что в сигнатуре языка формул содержится хотя бы одна константа

Лемма о пересечении моделей ХЛП (доказательство)

$$\begin{aligned} & (\forall \dots (B_1 \& \dots \& B_k \rightarrow A)) \in \Phi_{\mathcal{P}} \\ \cap \mathcal{I} \not\models & (B_1 \& \dots \& B_k \rightarrow A)[\tilde{d}^n] \text{ для любых } \tilde{d}^n \end{aligned}$$

Тогда предметами \mathcal{H} -интерпретаций являются основные термы, и последнее соотношение означает, что существует основной пример $B_1^g \& \dots \& B_k^g \rightarrow A^g$ правила из $\Phi_{\mathcal{P}}$, такой что $\cap \mathcal{I} \not\models B_1^g \& \dots \& B_k^g \rightarrow A^g$. По семантике $\&$ и \rightarrow : $\cap \mathcal{I} \models B_1^g, \dots, \cap \mathcal{I} \models B_k^g, \cap \mathcal{I} \not\models A^g$.

По теоретико-множественному заданию \mathcal{H} -интерпретаций:

$$B_1^g \in \cap \mathcal{I}, \dots, B_k^g \in \cap \mathcal{I}, A^g \notin \cap \mathcal{I}$$

По определению $\cap \mathcal{I}$:

- ▶ существует \mathcal{H} -модель \mathcal{I} ХЛП \mathcal{P} , такая что $A^g \notin \mathcal{I}$, и
- ▶ для любой \mathcal{H} -модели \mathcal{J} ХЛП \mathcal{P} (в том числе для \mathcal{I}) верно $B_1^g \in \mathcal{J}, \dots, B_k^g \in \mathcal{J}$.

Применив теоретико-множественный способ задания \mathcal{H} -интерпретаций и семантику $\&$, \rightarrow и \forall к \mathcal{I} так же, как они применялись к $\cap \mathcal{I}$, но в обратном порядке, получим соотношение $\mathcal{I} \not\models \forall \dots (B_1 \& \dots \& B_k \rightarrow A)$. То есть \mathcal{I} не является моделью для $\Phi_{\mathcal{P}}$, что **противоречит** выбору \mathcal{I} ▼

Наименьшей моделью ХЛП \mathcal{P} будем называть \mathcal{H} -модель, наименьшую относительно теоретико-множественного включения среди всех \mathcal{H} -моделей ХЛП \mathcal{P} (то есть модель, вложенную как множество в каждую другую модель)

Теорема. Для любой ХЛП \mathcal{P} существует наименьшая модель

Доказательство.

По последней лемме, если \mathcal{I} — семейство всех \mathcal{H} -моделей ХЛП \mathcal{P} , то $\bigcap \mathcal{I}$ — \mathcal{H} -модель ХЛП \mathcal{P}

Нетрудно видеть, что эта модель является наименьшей по теоретико-множественному включению в семействе \mathcal{I} ▼

$M_{\mathcal{P}}$ — так будем обозначать наименьшую модель ХЛП \mathcal{P}

Теорема. $M_{\mathcal{P}} = B_{\mathcal{H}} \cap \{\varphi \mid \Phi_{\mathcal{P}} \models \varphi\}$

Доказательство. (по прежнему считаем, что в сигнатуре есть хотя бы одна константа)

(\supseteq): Рассмотрим основной атом A , такой что $\Phi_{\mathcal{P}} \models A$

По определению логического следования, верно $M_{\mathcal{P}} \models A$

Следовательно, $A \in M_{\mathcal{P}}$

(\subseteq): Предположим *от противного*, что существует основной атом A , такой что $A \in M_{\mathcal{P}}$ и $\Phi_{\mathcal{P}} \not\models A$

По определению логического следования, существует модель \mathcal{I} системы $\Phi_{\mathcal{P}}$, такая что $\mathcal{I} \not\models A$

По семантике \neg , верно $\mathcal{I} \models \neg A$

По теореме об \mathcal{H} -интерпретациях, существует эрбрановская интерпретация \mathcal{I}_h , такая что $\mathcal{I}_h \models \neg A$

По семантике \neg , верно $\mathcal{I}_h \not\models A$

По теоретико-множественному заданию \mathcal{H} -интерпретаций, верно $A \notin \mathcal{I}_h$

Следовательно, $A \notin M_{\mathcal{P}}$, что *противоречит* выбору A ▼

Пример

Для программы

```
птица(орёл);           летает(орёл);  
птица(пингвин);       летает(самолёт);  
обтекаемый(самолёт);  
обтекаемый(X) ← птица(X), летает(X);
```

наименьшая модель устроена так:

$$\left\{ \begin{array}{ll} \text{птица}(\mathbf{орёл}), & \text{птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), & \text{летает}(\mathbf{самолёт}), \\ \text{обтекаемый}(\mathbf{орёл}), & \text{обтекаемый}(\mathbf{самолёт}) \end{array} \right\}$$

В этой модели содержится наименьший набор основных фактов, которые могут быть истинными в модели программы

Будем считать, что формула φ **следует в допущении замкнутости мира** из системы $\Phi_{\mathcal{P}}$, отвечающей ХЛП \mathcal{P} ($\Phi_{\mathcal{P}} \models_{cwa} \varphi$), если $M_{\mathcal{P}} \models \varphi$

В частности, в таком определении действительно верно, что пингвин — нелетающая птица:

$$\left\{ \begin{array}{l} \text{птица}(\mathbf{орёл}), \text{ птица}(\mathbf{пингвин}), \\ \text{летает}(\mathbf{орёл}), \text{ летает}(\mathbf{самолёт}) \end{array} \right\} \models_{cwa} \text{птица}(\mathbf{пингвин}) \ \& \ \neg \text{летает}(\mathbf{пингвин}),$$

так как в наименьшей модели программы содержится (выполняется) атом «птица(**пингвин**)» и не содержится (не выполняется) атом «летает(**пингвин**)»