Математическая логика и логическое программирование

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

Блок 45

Отрицание в логическом программировании
Допущение замкнутости мира

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

ВМК МГУ, 2025, сентябрь–декабрь

Пример (ХЛП $\mathcal{P}$)

$$\text{птица}(\textbf{орёл}); \qquad \text{летает}(\textbf{орёл});$$
$$\text{птица}(\textbf{пингвин}); \qquad \text{летает}(\textbf{самолёт});$$

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

$$?\text{птица}(\mathtt{X}), \text{летает}(\mathtt{X})$$

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

$$\{\mathtt{X}/\textbf{орёл}\}$$

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

$$?\text{птица}(\mathtt{X}), \neg\text{летает}(\mathtt{X})$$

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

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

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

$$\text{птица}(\textbf{орёл}); \qquad \text{летает}(\textbf{орёл});$$
$$\text{птица}(\textbf{пингвин}); \qquad \text{летает}(\textbf{самолёт});$$
$$?\text{птица}(\mathtt{X}), \neg\text{летает}(\mathtt{X})$$

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

$$\not\models \quad \text{птица}(\textbf{орёл}) \,\&\, \text{птица}(\textbf{пингвин}) \,\&\, \text{летает}(\textbf{орёл}) \,\&\, \text{летает}(\textbf{самолёт}) \rightarrow \text{птица}(\textbf{пингвин}) \,\&\, \neg\text{летает}(\textbf{пингвин})$$

А значит (по теореме о логическом следствии),

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

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

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

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

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

$$\left\{\begin{matrix}\text{птица}(\textbf{орёл}), \ \text{птица}(\textbf{пингвин}), \\ \text{летает}(\textbf{орёл}), \text{летает}(\textbf{самолёт})\end{matrix}\right\} \models \text{птица}(\textbf{пингвин})$$

Всё-таки летает ли пингвин?

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

$$\begin{array}{ll} \text{птица}(\textbf{орёл}); & \text{летает}(\textbf{орёл}); \\ \text{птица}(\textbf{пингвин}); & \text{летает}(\textbf{самолёт}); \end{array}$$

$$?\text{птица}(\mathtt{X}), \neg\text{летает}(\mathtt{X})$$

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

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

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 $\mathcal{Q}$)

$$\text{птица}(\textbf{орёл}); \qquad \text{летает}(\textbf{орёл});$$
$$\text{птица}(\textbf{пингвин}); \qquad \text{летает}(\textbf{самолёт});$$
$$?\text{птица}(\mathtt{X}), \neg\text{летает}(\mathtt{X})$$

Можно заставить программиста **явно** записать факт

$$\neg\text{летает}(\textbf{пингвин})$$

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 $\mathcal{Q}$)

$$\text{птица}(\textbf{орёл}); \qquad \text{летает}(\textbf{орёл});$$
$$\text{птица}(\textbf{пингвин}); \qquad \text{летает}(\textbf{самолёт});$$
$$?\text{птица}(\mathtt{X}), \neg\text{летает}(\mathtt{X})$$

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 доказать вину

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:

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

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 $\mathfrak{I}$ $\mathcal{H}$-интерпретаций записью $\cap\mathfrak{I}$ будем обозначать $\mathcal{H}$-интерпретацию $\{A \mid \forall \mathcal{I} \in \mathfrak{I} : A \in I\}$

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

Доказательство.

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

Тогда существует правило $(B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)^{\forall} \in \Phi_{\mathcal{P}}$, такое что $\cap\mathfrak{I} \not\models (B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)^{\forall}$

По семантике $\forall$, существует набор предметов $\widetilde{d}^n$, такой что $\cap\mathfrak{I} \not\models (B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)[\widetilde{d}^n]$

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

Лемма о пересечении моделей ХЛП (доказательство)

$$(B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)^\forall \in \Phi_{\mathcal{P}}$$
$$\cap\mathfrak{I} \not\models (B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)[\widetilde{d}^n] \text{ для любых } \widetilde{d}^n$$

Тогда предметами $\mathcal{H}$-интерпретаций являются основные термы, и последнее соотношение означает, что существует основной пример $B_1^g \mathbin{\&} \ldots \mathbin{\&} B_k^g \to A^g$ правила из $\Phi_{\mathcal{P}}$, такой что $\cap\mathfrak{I} \not\models B_1^g \mathbin{\&} \ldots \mathbin{\&} B_k^g \to A^g$

По семантике $\&$ и $\to$: $\cap\mathfrak{I} \models B_1^g$, ..., $\cap\mathfrak{I} \models B_k^g$, $\cap\mathfrak{I} \not\models A^g$

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

$$B_1^g \in \cap\mathfrak{I}, \quad \ldots, \quad B_k^g \in \cap\mathfrak{I}, \quad A^g \notin \cap\mathfrak{I}$$

По определению $\cap\mathfrak{I}$:

- 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 $B_1^g \in \mathcal{J}$, ..., $B_k^g \in \mathcal{J}$

Применив теоретико-множественный способ задания $\mathcal{H}$-интерпретаций и семантику $\&$, $\to$ и $\forall$ к $\mathcal{I}$ так же, как они применялись к $\cap\mathfrak{I}$, но в обратном порядке, получим соотношение $\mathcal{I} \not\models (B_1 \mathbin{\&} \ldots \mathbin{\&} B_k \to A)^\forall$

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 $\mathfrak{I}$ — семейство всех $\mathcal{H}$-моделей ХЛП $\mathcal{P}$, то $\cap\mathfrak{I}$ — $\mathcal{H}$-модель ХЛП $\mathcal{P}$

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

$M_{\mathcal{P}}$ — 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 $\mathcal{I} \models \neg A$

Значит, верно и $\mathcal{I} \models \Phi_{\mathcal{P}} \cup \{\neg A\}$

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

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

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

Из $A \notin \mathcal{I}_h$ и $\mathcal{I}_h \models \Phi_{\mathcal{P}}$ следует $A \notin M_{\mathcal{P}}$, что *противоречит* выбору A ▼

Пример

Для программы

$$\begin{array}{ll} \text{птица}(\textbf{орёл}); & \text{летает}(\textbf{орёл}); \\ \text{птица}(\textbf{пингвин}); & \text{летает}(\textbf{самолёт}); \\ \multicolumn{2}{l}{\text{обтекаемый}(\textbf{самолёт});} \\ \multicolumn{2}{l}{\text{обтекаемый}(\mathtt{X}) \leftarrow \text{птица}(\mathtt{X}), \text{летает}(\mathtt{X});} \end{array}$$

наименьшая модель устроена так:

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

В этой модели содержится наименьший набор основных фактов, которые могут быть истинными согласно написанному в программе

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:

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

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