

Исчисление предикатов

Гипотеза о физической символьной системе:

- физически ограниченная система ведет себя соответственно своим целям, приспособляясь к требованиям окружающей среды, проявляя при этом разумное в широком смысле поведение;
- физическая система проявляет разумное поведение тогда и только тогда, когда она является физической символьной системой;
- достаточность значит, что разумность может быть достигнута любой правильно организованной физической символьной системой;
- необходимость - каждый объект, проявляющий разумное поведение, должен являться физической символьной системой, т.е. разумное поведение объекта достигается путем физической реализации операций над символьными структурами.

Сформулирована в 1976г А.Ньюэллом и Г.Саймоном.

Замечание. Эта гипотеза лежит в основе создания умных машин и делает очевидными основные предположения в исследовании искусственного интеллекта. Если уровень интеллекта определяется исключительно структурой системы символов, то любая среда, которая успешно реализует правильные шаблоны и процессы, достигнет этого уровня интеллекта, независимо от того, состоит она из нейронов или логических цепей.

Исчисление высказываний

Символы: высказывания (P, Q, R, S, \dots), значения истинности (true, false), скобки - (,) и логические связки ($\vee, \wedge, \neg, \equiv, \rightarrow$).

Предложения формируются из символов по следующим правилам:

- каждый символ и значение истинности есть предложение;
- отрицание (\neg) предложения есть предложение;
- конъюнкция (\vee) или логическое И двух предложений есть предложение;
- дизъюнкция (\wedge) или логическое ИЛИ двух предложений есть предложение;
- эквивалентность (\equiv) двух предложений есть предложение;
- импликация (\rightarrow) двух предложений есть предложение.

Замечание. В импликации $P \rightarrow Q$, элементы называются предпосылкой и заключением.

Интерпретация предложения (семантика исчисления высказывания) – утверждение о правдивости предложения в некотором мире (присвоение ему значения истинности Т или F, т.е. отображение предложения на множество $\{T, F\}$).

Пример. Утверждение «если идет дождь, то земля мокрая» можно представить в виде импликации $P \rightarrow Q$, если P – высказывание «идет дождь», а Q – «земля мокрая».

P	Q	$P \wedge Q$	$P \vee Q$	$P \equiv Q$	$P \rightarrow Q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	F	T
F	F	F	F	T	T

Значение истинности предложений определяется **таблицами истинности**, которые содержат все возможные варианты значения истинности для элементарных суждений или атомарных формул, составленных из символов высказываний.

Синтаксис предикатов

Алфавит предикатов: буквы (русских или английских), цифры, скобки, знак подчеркивания.

Символы начинаются с буквы, за которой следует последовательность знаков, например, *День_недели*, *погода*, *plus*. Символы - константы, переменные, функции и предикаты.

Константы соответствуют некоторым объектам или их свойствам из описываемого мира и начинаются со строчной буквы или состоят только из цифр.

Переменные обозначают общие классы объектов или свойств и начинаются с прописной буквы. Переменная должна быть определена на некотором множестве констант, имеющих общее свойство. Например, переменная *День_недели* определена на множестве {пн, вт, ср, чт, пт, сб, вс}, т.е. может принимать любое из этих значений.

Функция - отражение одного или нескольких элементов множества (область определения), в **однозначно** определяемый элемент другого множества (область значений). Функция состоит из символа, начинающегося со **строчной** буквы, и заключенного в круглые скобки списка аргументов. Число аргументов - аридность функции.

Замена функции ее значением называется **оцениванием функции**. Например, результат оценивания функции *plus(7,минус_2)* – целое число 5.

Предикаты определяют истинность отношения между несколькими объектами в мире. Количество объектов определяет аридность предиката. Предикаты подобны функциям – в скобках после символа предиката список термов, но у всех предикатов одна область значений {Т, F}. **Термом** может быть константа, переменная, или функция.

Например, *друг(иван,петр)* или *друг(отец(X),отец(Y))* - предикаты, причем в последнем случае аргументы предиката являются функциями. Для интерпретации последнего предиката надо задать область определения и область значений функции *отец*.

Кванторы переменных

Квантор переменной ограничивает предложение, содержащее переменную на которую данный квантор указывает (связывает).

Квантор существования \exists указывает, что предложение истинно, хотя бы **для одного** значения из области определения переменной.

Например, высказывание «У Ивана есть друг» $\exists X \text{ друг}(\text{иван}, X)$, где X – множество людей.

Квантор всеобщности \forall указывает, что предложение истинно **для всех** значений из области определения переменной.

Например, высказывание «Все любят мороженное» $\forall X \text{ любить}(X, \text{мороженное})$.

Замечание. Здесь и далее используются два предиката: $\text{друг}(X, Y)$, где X, Y – конечное множество имен людей; $\text{любить}(X, Z)$ – человек по имени X любит предмет Z , где Z – конечное множество предметов.

Для указания **области действия квантора** используются круглые скобки. Например, $\forall X(p(X) \vee q(Y) \rightarrow r(X))$, указывает, что переменная X связана квантором всеобщности в предложениях $p(X)$ и $r(X)$. В исчислении предикатов все выражения должны быть связаны кванторами.

Правила вывода

Правило вывода **обосновано**, если каждое выражение, полученное в соответствии с этим правилом из множества выражений, логически следует из этого множества.

Правило отделения.

Если предложения P и $P \rightarrow Q$ истинны в некоторой интерпретации, то тогда и Q в этой интерпретации тоже истинно.

Пример. Если P - «идет дождь», Q - «земля мокрая», то утверждение «если идет дождь, то земля мокрая» - импликация $P \rightarrow Q$. При условии, что $P \equiv T$ - сейчас идет дождь, получаем набор истинных предложений: $P \rightarrow Q$, P . Применив правило отделения, получаем, что $Q \equiv T$. Можно добавить в набор новый факт - «сейчас земля мокрая».

Правило универсального инстанцирования.

Если в истинном предложении любую переменную, стоящую под квантором всеобщности, заменить соответствующим термом из области ее определения, то результирующее предложение останется истинным.

Пример. Высказывание «все люди смертны» можно представить в исчислении предикатов как, $\forall X(\text{человек}(X) \rightarrow \text{смертный}(X))$, а утверждение «Сократ - человек» как, $\text{человек}(\text{сократ})$. Переменная X в высказывании стоит под знаком квантора всеобщности. Согласно правилу универсального инстанцирования, ее можно заменить на любое значение из области определения и, применив правило отделения, получить утверждение $\text{смертный}(\text{сократ})$, т.е. «Сократ - смертен».

Абдуктивное рассуждение

Системы автоматического рассуждения, использующие логику исчисления предикатов, основываются на следующих предположениях о базе знаний:

- предложения в базе знаний адекватно и полно описывают предметную область, т.е. содержат **всю** необходимую информацию;
- предложения в базе знаний не противоречат друг другу.

Проблема при разработке экспертной системы – **абдуктивное рассуждение**, когда из $P \rightarrow Q$ и $Q \equiv T$ выводится $P \equiv T$, что логически не обосновано.

Пример.

В системе диагностики автомобиля может существовать утверждение «если двигатель не вращается, и фары не горят, то проблема в аккумуляторе или проводке», что напоминает обычное предикатное выражение, используемое в правиле отделения, но это не так. Данное высказывание (абдуктивное рассуждение) не является истинным. А вот обратное утверждение – истинно, но пользы от него, с точки зрения экспертной системы, практически нет.

Еще одна проблема при создании базы знаний о реальной предметной области на основе исчисления предикатов - значительный **объем предикатов**, описывающих как отношения между объектами, так и возможные изменения этих отношений.

Получение новых знаний

Мир Маши, Оли и Тани

Исходная база знаний

Предикаты: $\text{любит}(X, Z)$ и $\text{друг}(X, Y)$. Связь: $(\text{любит}(X, Z) \wedge \text{любит}(Y, Z)) \rightarrow \text{друг}(X, Y)$.

$X, Y = \{\text{маша}, \text{оля}, \text{таня}\}$ – переменная «люди»

$Z = \{\text{мороженное}, \text{чай}, \text{кофе}\}$ – переменная «предметы»

Исходные предикаты: $\text{любит}(\text{маша}, \text{чай})$; $\text{любит}(\text{маша}, \text{мороженное})$; $\text{любит}(\text{оля}, \text{кофе})$; $\text{любит}(\text{оля}, \text{мороженное})$; $\text{любит}(\text{таня}, \text{кофе})$; $\text{друг}(\text{оля}, \text{таня})$.

Функционирование

Вопрос. $\text{друг}(\text{маша}, \text{оля}) \equiv T$

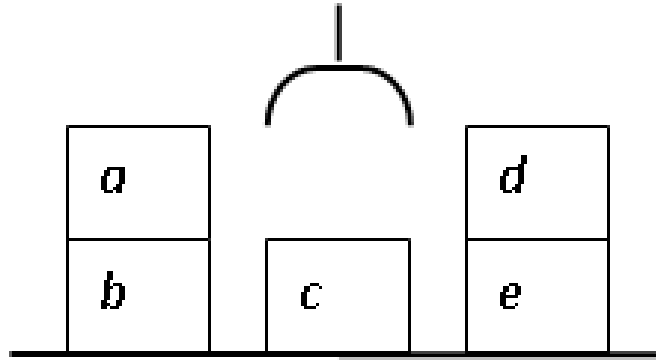
Последовательность действий:

1. Поиск в базе предиката $\text{друг}(\text{маша}, \text{оля})$. Нет
2. Поиск импликации с заключением $\text{друг}(X, Y)$. Есть
3. Определение предпосылок к найденной импликации - $\text{любит}(X, Z) \wedge \text{любит}(Y, Z)$
4. Поиск какого-либо предиката вида $\text{любит}(\text{маша}, Z)$ – $Z = \text{чай}$
5. Поиск предиката $\text{любит}(\text{оля}, \text{чай})$. Нет
6. Поиск другого предиката вида $\text{любит}(\text{маша}, Z)$ – $Z = \text{мороженное}$
7. Поиск предиката $\text{любит}(\text{оля}, \text{мороженное})$. Есть
8. Ответ $\text{друг}(\text{маша}, \text{оля})$. Да - новая информация.

Проблема полноты. Любой факт считается ложным, если он не включен в базу знаний. Например, на запрос $\text{любит}(\text{оля}, \text{чай})$ ответ будет «Нет», хотя информация о таком факте просто отсутствует в базе знаний.

Мир блоков в предикатах

«Мир блоков» - одинаковые блоки на поверхности стола и возможные действия робота, который имеет информацию о координатах расположения блоков на столе. Рука робота имеет захват, который может схватить свободный блок, т.е. блок со свободной верхней гранью, и переместить его в любую точку стола или установить на свободный блок.



Состояние блоков и их отношение можно описать следующими предикатами:

на_столе(X) – блок X находится непосредственно на столе;

на(X,Y) – блок X находится на верхней грани блока Y;

чисто(X) – верхняя грань блока X свободна, т.е. блок свободен и может быть захвачен;

захват(X) – захват удерживает блок X;

пусто() – захват свободен.

Правила, проверяющие истинность состояний:

$\forall X(\neg(\exists Y)на(Y,X) \rightarrow чисто(X))$ – если не существует блока Y, находящегося на блоке X, то блок X свободен;

$\forall X(\neg захват(X) \rightarrow пусто())$ – если в захвате нет блока, то он свободен.

Мир блоков в предикатах

Действия робота, приводящие мир к новому состоянию можно описать как операторы:

взять(X) – взять со стола блок X и держать его в захвате, причем

$\forall X(\text{взять}(X) \rightarrow (\text{захват}(X) \leftarrow (\text{пусто}() \wedge \text{чисто}(X) \wedge \text{на_столе}(X)));$

поместить(X) – поместить в некоторую точку стола блок X, находящийся в захвате,

причем $\forall X(\text{поместить}(X) \rightarrow ((\text{пусто}() \wedge \text{чисто}(X) \wedge \text{на_столе}(X) \leftarrow \text{захват}(X)));$

снять(X,Y) – снять свободный блок X, стоящий на блоке Y, причем

$(\forall X \forall Y)(\text{снять}(X,Y) \rightarrow ((\text{чисто}(Y) \wedge \text{захват}(X)) \leftarrow (\text{на}(X,Y) \wedge \text{чисто}(X) \wedge \text{пусто}()));$

поставить(X,Y) – поставить блок X, находящийся в захвате, на свободный блок Y,

причем $(\forall X \forall Y)(\text{поставить}(X,Y) \rightarrow ((\text{на}(X,Y) \wedge \text{чисто}(X) \wedge \text{пусто}()) \leftarrow (\text{чисто}(Y) \wedge \text{захват}(X)));$

Замечание. $A \rightarrow (B \leftarrow C)$ – из оператора A следует предикат B, если условие C истинно.

<i>взять</i> (X)	П: <i>пусто</i> () \wedge <i>чисто</i> (X) \wedge <i>на_столе</i> (X) Д: <i>захват</i> (X) У: <i>пусто</i> () \wedge <i>чисто</i> (X) \wedge <i>на_столе</i> (X)
<i>поместить</i> (X)	П: <i>захват</i> (X) Д: <i>пусто</i> () \wedge <i>чисто</i> (X) \wedge <i>на_столе</i> (X) У: <i>захват</i> (X)
<i>снять</i> (X,Y)	П: <i>на</i> (X,Y) \wedge <i>чисто</i> (X) \wedge <i>пусто</i> () Д: <i>чисто</i> (Y) \wedge <i>захват</i> (X) У: <i>на</i> (X,Y) \wedge <i>чисто</i> (X) \wedge <i>пусто</i> ()
<i>поставить</i> (X,Y)	П: <i>чисто</i> (Y) \wedge <i>захват</i> (X) Д: <i>на</i> (X,Y) \wedge <i>чисто</i> (X) \wedge <i>пусто</i> () У: <i>чисто</i> (Y) \wedge <i>захват</i> (X)

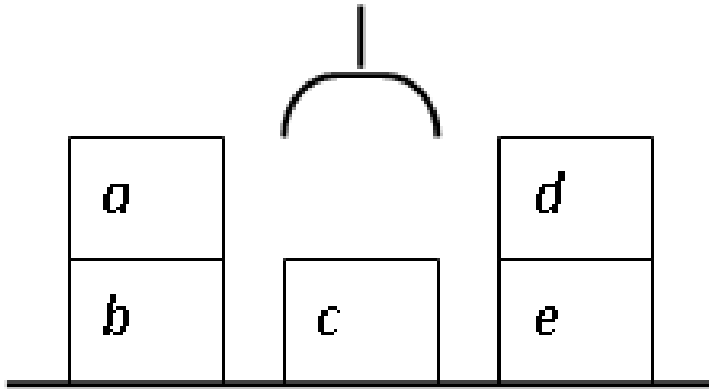
предусловия (П), которым должно удовлетворять описание для применения оператора,

дополнения (Д) к описанию состояния блоков в результате применения оператора,

удаления (У) из описания состояния блоков.

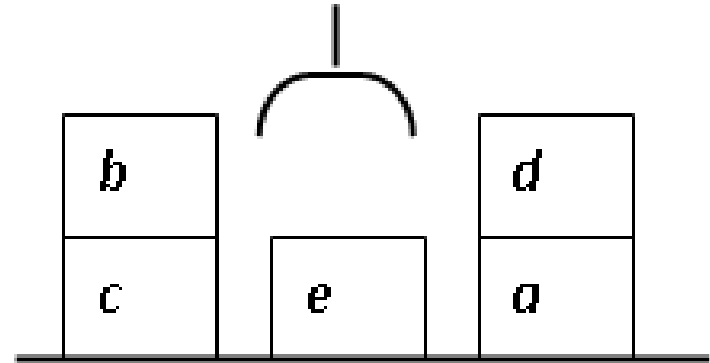
Каждый предикат из описания состояния, который не упоминается в списках П и У, остается истинным в описании нового состояния после применения оператора

Мир блоков в предикатах



Описание для состояния 1:

$\text{на_столе}(b) \wedge \text{на_столе}(c) \wedge \text{на_столе}(e) \wedge$
 $\text{на}(a,b) \wedge \text{на}(d,e) \wedge \text{чисто}(a) \wedge \text{чисто}(c) \wedge$
 $\text{чисто}(d) \wedge \text{пусто}()$.



Описание для состояния 2:

$\text{на_столе}(a) \wedge \text{на_столе}(c) \wedge \text{на_столе}(e) \wedge$
 $\text{на}(b,c) \wedge \text{на}(d,a) \wedge \text{чисто}(b) \wedge \text{чисто}(e) \wedge \text{чисто}(d) \wedge$
 $\text{пусто}()$.

Замечание. Учтено взаимное расположение блоков на столе и друг на друге. Не указано расположение блоков относительно друг друга, например, с между b и e . Это потребует создания дополнительных предикатов, описывающих состояние мира блоков.

Переход от состояния 1 к 2 можно выполнить следующей набором операций:

$\text{снять}(a,b)$; $\text{поместить}(a)$; $\text{взять}(b)$; $\text{поставить}(b,c)$; $\text{снять}(d,e)$; $\text{поставить}(d,a)$.

Замечание. Чистое исчисление предикатов не позволяет автоматически найти эту или другую последовательность операций.