

Теоретическая часть

1. Что собой представляет программа на языке пролог?

Программа на Prolog представляет собой базу знаний и вопрос.

- **База знаний** состоит из предложений – фактов и правил, – используя которые программа выдает ответ на вопрос. Каждое предложение должно заканчиваться точкой.
 - **Правило** имеет вид: $A :- B_1, \dots, B_n$, где A – заголовок правила (составной терм, который содержит знание); B_1, \dots, B_n – тело правила (составные термы, которые содержат условия истинности этого знания), символ “:–” – это специальный символ-разделитель.
 - **Факт** – это частный случай правила – предложение, в котором отсутствует тело (то есть тело пустое).
- **Вопрос** – это частный случай правила – предложение, которое состоит только из тела. Используется, чтобы определить, выполняется ли некоторое отношение между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Базис пролога – матлогика. Используется символьная обработка, декларативная методология.

2. Какова структура программы на Prolog?

Программа на Prolog состоит из следующих разделов, каждый из которых начинается со своего заголовка.

- директивы компилятора — зарезервированные символьные константы,
- **CONSTANTS** — раздел описания констант,
- **DOMAINS** — раздел описания доменов,
- **DATABASE** — раздел описания предикатов внутренней базы данных,

- PREDICATES — раздел описания предикатов,
- CLAUSES — раздел описания предложений базы знаний,
- GOAL — раздел описания внутренней цели (вопроса).

В программе не обязательно должны быть все разделы.

3. Как реализуется программа на Prolog? Как формируются результаты работы программы?

Ответ: Программа на Prolog представляет собой базу знаний и вопрос. База знаний состоит из предложений — фактов и правил, которые задают истинные знания. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Вопрос рассматривается системой как цель: найти возможность, исходя из базы знаний, ответить «Да» на поставленный вопрос. Вариантов ответить «Да» на может быть несколько. При поиске ответа рассматриваются альтернативные варианты и находятся все возможные решения (методом проб и ошибок) - множества значений переменных, при которых на поставленный вопрос можно ответить - «Да».

Для выполнения логического вывода используется механизм унификации, встроенный в систему. Унификация — операция, которая позволяет формализовать процесс логического вывода. С практической точки зрения - это основной вычислительный шаг, с помощью которого происходят:

- двунаправленная передача параметров процедурам,
- неразрушающее присваивание,
- проверка условий (доказательство).

В процессе работы система выполняет большое число унификаций. Попытка "увидеть одинаковость"— сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией (неудачей). В последнем случае включается механизм отката к предыдущему шагу.

4. Что такое терм?

Терм - основной элемент языка Prolog. Терм – это:

1. константа (используется для обозначения объекта предметной области):
 - число (целое, вещественное),
 - символьный атом – комбинация символов латинского алфавита, цифр и ' _ ' (символа подчеркивания), начинающаяся со строчной буквы), используется для обозначения конкретного объекта предметной области или для обозначения конкретного отношения,
 - строка – последовательность символов, заключенных в кавычки;
2. переменная:
 - именованная – комбинация символов латинского алфавита, цифр и ' _ ', начинающаяся с прописной буквы или символа подчеркивания, может связываться с различными объектами (конкретизироваться),
 - анонимная - обозначается символом ' _ ', не может быть связана со значением;
3. составной терм – средство фиксации информации о том, что между объектами существует определенная связь, синтаксически представляется так: $f(t_1, t_2, \dots, t_m)$, где f - функтор (символьная константа, обозначающая имя отношения между объектами), t_1, t_2, \dots, t_m – термы (в том числе и составные), являющиеся аргументами (арность – число аргументов).

5. Что такое предикат в матлогике (математике)?

Предикат в математической логике – это (логическая) функция со множеством значений 0, 1 (истина/ложь), определенная на некотором множестве параметров. Предикат называю n -арным, если он определен на n -ой декартовой степени множества M . Таким образом, каждый набор параметров характеризуется либо как «истинный», либо как «ложный».

6. Что описывает предикат в Prolog?

Процедура – совокупность правил, описывающих определенное отношение (заголовки имеют одно и то же имя и одинаковую арность). Предикат – отно-

шение, определяемое процедурой. Таким образом, предикат в Prolog описывает отношение между аргументами процедуры.

7. Назовите виды предложений в программе и приведите примеры таких предложений из вашей программы. Какие предложения являются основными, а какие - не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?

В Prolog есть два типа предложений:

- **Правило** имеет вид: $A : - B_1, \dots, B_n$, где A – заголовок правила (составной терм, который содержит знание); B_1, \dots, B_n – тело правила (составные термы, которые содержат условия истинности этого знания), символ “ : - ” – это специальный символ-разделитель;
- **Факт** – это частный случай правила – в нем отсутствует тело (то есть тело пустое).

TODO Примеры из программы:

- факт: `car("Mikhail", "Cadillac", "Black", "500000")`.
- правило: `car_by_phone(Phone, Surname, Model, Cost) :- man(Surname, Phone, _), car(Name, Model, _, Cost)`.

Основными называются предложения, не содержащие переменных. Они предназначены для описания отношений, формирования базы знаний.

Предложения, содержащие переменные называются неосновными. Они предназначены для поиска ответа в базе знаний.

8. Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей - абстрактной форме: содержащее или не содержащее переменных?

Переменные предназначены для передачи информации «во времени (за конечное число шагов получить результат) и пространстве (передача значений через параметры)», для повышения уровня абстракции. Они бывают именованными и анонимными.

Именованная переменная представляет собой комбинацию символов латинского алфавита, цифр и ' _ ', начинающуюся с прописной буквы или символа подчеркивания. В процессе выполнения программы именованные переменные могут связываться с различными объектами – конкретизироваться. Именованная переменная является уникальной в рамках предложения. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

Анонимная переменная обозначается символом ' _ '. Она не может быть связана со значением. Любая анонимная переменная уникальна.

В момент фиксации система не знает, какой объект представляет переменная. Во многих языках последовательность действий при работе с переменными такая: задать значение переменной, а затем работать с самой переменной. В прологе же особый способ работы с переменными – значение переменной не задается, система сама подбирает такое(-ие) значение(-я), чтобы условие было истинным.

Предложение БЗ, содержащее переменные, сформулировано в более общей форме, так как переменные не имеют значения и могут конкретизироваться различными объектами в ходе работы системы.

TODO??

9. Что такое подстановка?

Подстановка - это функция, действующая из множества X переменных в множество T термов программы, (при этом каждой переменной $X_i \in X$ ставится в соответствие терм $t_i \in T$).

Пусть дан терм: (X_1, X_2, \dots, X_n) . Подстановка - множество пар вида: $\{X_i = t_i\}$, где X_i - переменная, а t_i - терм.

10. Что такое пример терма? Как и когда строится? Как Вы думаете система строит и хранит термы?

Пусть $\omega : X_1 = t_1, X_2 = t_2, \dots, X_n = t_n$ - подстановка. Тогда результат применения подстановки к терму обозначается $A\omega$. Применение подстановки заключается в замене каждого вхождения переменной x_i на соответствующий терм. Терм B называется примером терма A , если существует такая подстановка ω , что $B = A\omega$.

В процессе выполнения программы система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопроса, строя под-

становки и примеры термов (вопроса и формулировки знания), используя базу знаний. Построение и подстановки производится путём конкретизации переменных. Сами термы хранятся в стеке.

Лекция 1

Конъюнкция, дизъюнкция, отрицание – базовые функции матлогики. Предикат – логическая функция.

Базис пролога – матлогика.

Предикат – логическая функция

Блоки: База знаний – clauses. Запрос разработки – goal. Запросы могут быть конъюнкцией или дизъюнкцией, но нам будут запрещать их использовать

Терм – константа, переменная или составное тело.

В прологе используется символьная обработка. Декларативная методология. Мы описываем систему знаний из предметной области. Потом задаем вопрос, но хотим получить не только да/нет, но и как (как побочный эффект)? Не запрещено использовать символы.

Константы – символьные атомы обозначение объекта/процесса предметной области – комбинация латинских символов, начинающаяся с маленькой буквы.

А переменная – тоже комбинация символов. Начинается с большой латинской или с нижнего подчеркивания – именованные переменные. Есть также анонимные переменные, которые обозначаются одинаково _

!!!Зачем нужны переменные – для повышения уровня абстракции

Составные термы – зафиксировать информацию о том, что между какими-то объектами есть связь. $f(t_1, t_2, \dots, t_n)$. f – главный функтор – имя отношения между двумя объектами, символьный атом (потому что могут быть внутри еще), t – терм.

`student(ivanov, mgtu)` – константы `student(X, mgtu)` – группа студентов из `mgtu`

`student(ivanov, mgtu)` и `student(ivanov)` – для системы разные запросы.

В момент фиксации система не знает, что такое X

Первые аргументы считаются как объекты одной природы, вторые – другой. Только мы определяем смысл.

Чем больше переменных, тем выше уровень абстракции.

База знаний состоит из фактов (без переменных – основные, ост – неосн)
Правило $A:-B_1, B, \dots, B_k$ A – заголовок правила (в заголовке формулируются знания о том, что... а тело B все задает) Заголовок – фиксация знания о том, что между аргументами есть истинная связь `student(X, mgtu):-докум(X, att), выше(ball, 296)`

!!!Особенный способ работы с переменными. В других языках задаем, потом работаем. Здесь не задаем значение, система сама подбирает значение переменной, чтобы условие было истинным. Цель системы – ответить да

!!!Переменные нужны для передачи данных во времени и пространстве. Во

времени – через несколько шагов получаем, в пространстве – через параметры, переменные.

запрограммированный метод резолюции позволяет делать поиск ответа на вопрос.

Лекция 2

что такое декларативно?

в базе знаний нет порядка.

Вопрос тоже мб без переменных -> основные и неосновные (исп переменных – повышение уровня абстракции)

Переменные.

Именованные, когда исп, не имеют значение, подбирается. Но в какой момент она какое значение принимает. Методом проб и ошибок. Цель – подтвердить истинность вопроса с помощью бз.

Именованные - передача значения во времени и пространстве (для этого в какой-то момент переменная дб конкретизирована каким-то значением. Но это может быть ошибочным, есть механизм отказа, реконкретизировать переменную). Установление значения для переменной не связано с понятием типа (по указателям). Анонимные переменные система не конкретизирует значениями.

Алгоритм унификации - единственный алгоритм доказательства. Многократно запускается (в какой момент?)

про переменные – ...

Каждое утверждение программы – ...

<3

Вопросы простые. Пока просто факты.

именованная переменная входит в факты и правила с квантором всеобщности, а в вопрос с квантором существования.

Уникальность: именованная переменная уникальна в рамках одного предложения. Анонимная переменная уникальна всегда.

вообще мы знаем об объектах, набор характеристик.

Заголовок – составной терм (главный функтор и аргументы в скобочках) – фиксирует знание о том что между аргументами существует связь, имя которой является главным функтором. Известный объект - символами, неизвестный - переменная.

Заголовок – правило, тело – условие.

Факт - чс правила, пустое тело.

В момент фиксации знания (если еще оно с условием, переменными) – условная истина.

процедурные и декларативные особенности пролога

Чтобы подобрать знание (которое в заголовке). Чтобы ответить на вопрос, надо подобрать знание – надо сравнивать вопрос с заголовками. Порядок фор-

мально установлен сверху вниз. Система формально (используя только конструкцию - симв .. и порядок) подбирает знание. И то, и то – составной терм. Сравнивает по 2 сосавных терма по формальному признаку.

Если есть переменная и в вопросе, и в формулировке знания (а если еще и на одной позиции)

подстановки и примеры терма

Без переменных – основные.

Неосновные термы:

$A(X_1, \dots, X_n)$, X_i – переменные.

Подстановкой называется множество пар вида $x_i=t_i$, где x_i – переменная, t_i – терм, не содержащий переменных (t_i – значение для переменной x_i).

Чтобы подбирать значения переменных нужно построить подстановку. Принято обозначать *teta*

$= X_1 = t_1, \dots, X_n = T_n$ – применение подстановки к терму

Применение подстановки заключается в замене каждого вхождения переменной X_i на соотвв вхождение t_i . A_{teta}

Терм В явл примером терма А, если суц такая подстановка тета, что $B=A_{teta}$

Терм С называется общим примером термов А и В, если существуют такие подстановки тета1 и тета2, что $C=A_{teta1}$ и $C = B_{teta2}$.

$A = \text{plus}(1, 2, z) \text{ plus}(X, Y, 3)$

В первых лабах только факты, потом факты и правила, условие истинности разделено запятыми.

Запрещается использовать сложные вопросы – один терм.

$\text{father}(X, Y), \text{father}(Y, Z)$

Процедура – группу знаний объединяют в общее знание. Главный функтор у всех одинаковый. Природа объектов одинакова (не про память).

Процедура – совокупность правил, заголовки которых имеют одинаковые главные функторы, одинаковое число аргументов, обозначающих объекты одной природы.

Это одно знание, которое мб зафиксировано через несколько. Структура знания описывается в разделе предикатов. Не связано с понятием типа или памятью

0.1 Простейшие правила логического вывода

один составной терм

Правила вывода – утверждения о взаимосвязи между допущениями, которые с позиции исчисления предикатов верны всегда.

4 варианта факты основные (квантор существования), вопрос основной – правило-совпадение факты основные, вопрос неосновной – правило – обобщение факта факты неосновные (квантор всеобщности), вопрос основной – правило конкретизации факта факты неосновные, вопрос неосновной – система должна построить пример терма-вопроса и терма-знания (подобрать соот подстановки) общий пример строится в 2 шага — сначала конкретизация правила, а потом правило обобщения.

унификация терма

подбор знания

комп по императивному принципу. опять унификация.

унификация - операция, которая позволяет формализовать процесс логического вывода. Это основной вычислительный шаг, с помощью которого происходит

1. Двухнаправленная передача параметров процедурам (знание в неск предложений) 2. Неразрушающее присваивание (конкретизация) 3. Проверка условий.

Унифицировать (понять, что это знание подходит для доказательства этого вопроса) два терма. Два терма про одно и то же (= – принудительный (явный) запуск унификации)

$T1=T2$

Два терма унифицируются по след правилам

1. Если $T1$ и $T2$ – константы: только если они совпадают 2. Если $T1$ – неконкретизированная переменная, а $T2$ – константа или составной терм, не содержащий в качестве аргумента $T1$: унификация успешна, а $T1$ конкретизируется значением $T2$ 3. Если $T1$ и $T2$ – неконкретизированные (не имеющие значения) переменные: унификация всегда успешна, причем $T1$ и $T2$ становятся сцепленными двумя именами (указателями) одного и того же объекта. Если одна из переменных или один из термов конкретизируется значением, то второй моментально тоже конкретизируется им же 4. Если $T1$ и $T2$ – составные термы (например вопрос и заголовок): успешно унифицируются если а) у $T1$ и $T2$ одинаковые

главные функторы б) T_1 и T_2 имеют равные арности в) успешно унифицируется каждая пара их соотв компонент

Алгоритм унификации