

## Краткие обобщения к семинарам 1-2 (самост. работа №1)

1 Программа на Прологе состоит из множества предложений (хорновских). Каждое предложение определяет некоторое отношение, существующее между терминами. Каждый терм обозначает некоторый объект реального мира. Отношение в Прологе называется предикатом. Можно считать, что предикат – это идентификатор (имя) отношения. Логика хорновских предложений является ограниченной формой исчисления предикатов первого порядка (факты, правила и цели).

2 Предложения – это либо факты, либо правила.

Факты описывают свойства(property) объектов(термов) или отношения(relation) между объектами, о которых нам известно, что они истинны.

Правила определяют условия, при которых отношения между объектами являются истинными. Позволяют выводить одни факты из других. Синтаксически правила состоят из заголовка и тела (заголовок if тело). Правило(его заголовок) принимает значение истина только в том случае, если приняты значения истинности все предложения, составляющие его тело.

3 Для описания объектов и отношений между ними используются идентификаторы.

Идентификаторы объекта и отношения начинаются со строчной буквы, за которой может следовать любое число символов- буквы верхнего и нижнего регистров, цифры, знаки подчеркивания.

**!!! Нельзя!** -пробелы, знак минус, слеш.

4 Предикат- это предикатный символ (имя) и последовательность аргументов, заключенная в круглые скобки. Количество аргументов предиката определяет его арность (размерность). Предикат определяет, является ли отношение истинным или ложным в зависимости от того, имеет место данное отношение или нет. Предикат- это индикатор отношения, могущий принять только два значения –true или false. Не путать предикат с хорновским предложением!! Хорновское предложение – это логическая импликация (правило). При определенных условиях может быть фактом или целью. Все предложения программы должны быть построены в соответствии с синтаксисом языка, и тогда они называются правильно построенными формулами (ППФ)

5 Процедура – это последовательность предложений, описывающая один и тот же предикат.

Все предложения, относящиеся к одной и той же процедуре, имеют одинаковое имя предиката и одну и ту же арность. Эти предложения должны следовать друг за другом.

6 Объекты, связанные определенным отношением, в Прологе называются аргументами. Аргументы могут быть константами или переменными.

Переменная – позволяет описать общие факты и правила и задать общие вопросы. Роль переменной в предложении определяется способом ее квантификации. Если на месте термина в формуле записана переменная, то это означает, что вместо переменной может стоять более, чем один элемент из области интерпретации. Такой ППФ нельзя приписать значение истинности, так как неясно, какие значения может принять переменная. Перед тем как присвоить ППФ значение истинности, данную переменную необходимо явно квантифицировать. (ППФ –правильно построенная формула).

Имя переменной пишется с заглавной буквы.

Пролог не имеет оператора присваивания. Переменная получает свои значения в результате сопоставления с константами в фактах и правилах. До получения значения переменная свободна. После получения значения переменная считается связанной. Переменная остается связанной после присвоения ей значения только то время, которое

необходимо для получения решения в соответствии с поставленной целью, затем Пролог освобождает переменную и пытается найти другое решение, если оно существует.

7. Для игнорирования ненужных значений используется анонимная переменная(сопоставима с любыми данными) Никогда не принимает какого-либо значения. Синтаксически начинается со знака подчеркивания.

8. Цель(goal) – простая и составная Может быть конъюнкцией (A and B) или дизъюнкцией(A or B). Запятая понимается как конъюнкция. Точка с запятой понимается как дизъюнкция.

Цель может быть внутренней и внешней. В случае использования внутренней цели необходимо использовать стандартный предикат write() для вывода результата на экран монитора.

9. Информация о фактах, которые не являются истинными и об отношениях, которые не соблюдаются, называется отрицательной информацией. Такая информация в явной форме в программе не хранится. Просто считается, что вся информация, которая отсутствует в программе, ложна.

Встроенный предикат not имеет один аргумент. Этим аргументом является подцель, значение истинности которой (после выполнения подцели) заменяется на противоположное.

Предикат not будет успешным, если не может быть доказана истинность данной подцели. *Это приводит к предотвращению связывания внутри not несвязанных переменных.*

Свободные переменные в предикате not не разрешены, потому что для связывания свободных переменных в подцели подцель должна унифицироваться с каким-либо предложением и выполняться, то есть, если вы используете внутри not переменную, то она должна быть означена до своего использования.

### Квантификация переменных в подцели с предикатом not

? not(sister(ann,X)) если Ann не является сестрой X, то предикат sister –неуспешен, а not(sister...)- истина.

В подцели sister(ann,X) переменная X квантифицирована экзистенциально (существует ли некоторый X, для которого Ann является сестрой?)

Переменная X в подцели с отрицанием not(sister(...)) квантифицирована универсально (является ли Ann не сестрой X? То есть для любого X является ли Ann не сестрой X? Для проверки этого положения требуется отыскать только одно значение X, при котором подцель с отрицанием потерпит неудачу.

Квантификация переменных всегда изменяется на противоположную, если фраза подвергается отрицанию.

### Замечание

Существует так называемое предположение о замкнутости мира: если предложение P не представлено в программе, то считается, что представлено отрицание P. Это означает, что компилятор не в состоянии отличить неизвестную фразу от доказуемо неистинной фразы. Множество предложений программы называется миром. Этот мир – замкнутый, так как компилятор считает, что в нем содержатся все возможные знания. Фраза входит в состав мира тогда и только тогда, когда она присутствует в программе.

По умолчанию компилятор руководствуется предположением о замкнутости мира. Если мы хотим воспользоваться предположением об открытости мира, то мы должны считать фразу, отсутствующую в программе ни истинной, ни ложной.(то есть ее значение может быть истиной, ложью или неизвестным). Для работы с открытым миром надо выразить это желание явно, и тогда программа сможет предпринять какие-то особые действия, например, обратиться к альтернативному источнику знаний.

10 Программу можно расширить, добавляя в нее новые утверждения.

