

15.04.24. Лекция 10.

Основная часть программы: база знаний, вопрос.

Первоначальный способ фиксации

Декларативные и процедурные особенности работы Prolog

На Prolog мы имитируем интел. работу человека.

Декларативный \Rightarrow описывает предм. область в декларативной форме.

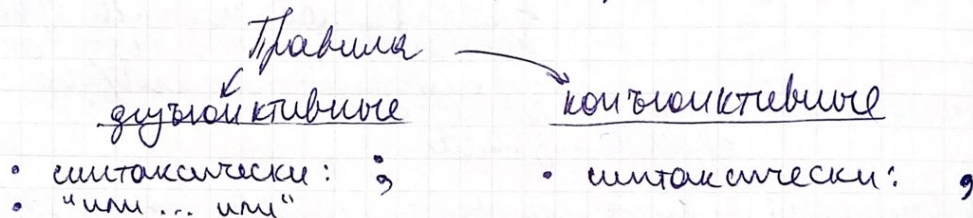
База знаний

Получает вопрос, система должна ответить на ~~вопрос~~ ^{цель, цель} ~~вопрос~~ ^{цель, цель}

Все, чем может упр. программист:

1. иметь знания, представить

Порядок может влиять на эф-ть работы системы.



! В п/р: конъюнктивные.

Для системы ~~вопрос~~ ^{цель} ~~вопрос~~ ^{цель} — цель (ответ) — найти знание, с помощью кот. она попытается ответить на вопрос.

База знаний — свое, перепи ее цели.

Система не меняет базу знаний во время работы.

Все данные правил хранятся в Д.П. (опер. памяти).

База знаний, вопрос — обещ. хранятся в памяти.

7-е прим. данные.

Результат — чем, кот. система должна преобр. — некоторая область памяти.

Система меняет уровень в теч. работы, ~~того, что~~ ^{того, что} ~~мне~~ ^{мне} ~~подобрать~~ ^{подобрать} ~~знания~~ ^{знания} ~~для~~ ^{для} ~~подбора~~ ^{подбора} ~~знаний~~ ^{знаний} ~~нужных~~ ^{нужных} ~~для~~ ^{для} ~~ответа~~ ^{ответа} ~~на~~ ^{на} ~~вопрос~~ ^{вопрос}

— для того, чтобы подобрать знание, нужное для правильного ответа на вопрос.

Процедура — иск. предположений, ^(гипотеза) ~~иск.~~ ^{иск.} одно и то же ~~знание~~ ^{знание} ~~по~~ ^{по} ~~смыслу~~ ^{смыслу} ~~про~~ ^{про} ~~одно~~ ^{одно} ~~и~~ ^и ~~то~~ ^{то} ~~же~~ ^{же}

Так только вопрос поступил в систему, он попадает в очередь.

затем система подбирает жанры.

Если алгоритм универсализации работает успешно, значит есть возможность попытаться ответить на вопрос: т.е. имеется заголовок и тело \Rightarrow поставь цель: доказать истинность тела \Rightarrow ~~до~~ редукция.

(\Rightarrow новые цели, кот. нужно доказать).
 \Rightarrow есть упрощения.

В результате утилизации получаем конкр. значение переменной (именованного) - подсчитываем. Она хранится не в базе знаний, а в отд. области памяти. Именованное переменное уникальны в рамках всего предположение (заголовок + тело). -

Если в задании были подстановочные значения переменных при этих же значениях необход. доказать истинность теор. (поэтому применение рекурсии в два этапа: терм вы-роста заменяется на тело подстановки применение к новой рекурсии)

Простой вопрос \rightarrow сравнение терма вопроса с заголовком
выбор значений имен-переменных \rightarrow замена
результаты, перестановка значений.

Редольвента "растет слева" - стек ("растет сверху"?)

У системы должна быть возможность изменить ^(воплотить в жизнь) действит.
если значение перем. передано неправильно,
система должна попытаться передать другое значение.

- задача МЧ: обеспечить работу на всех предложенных базах знаний.
(backtraining)

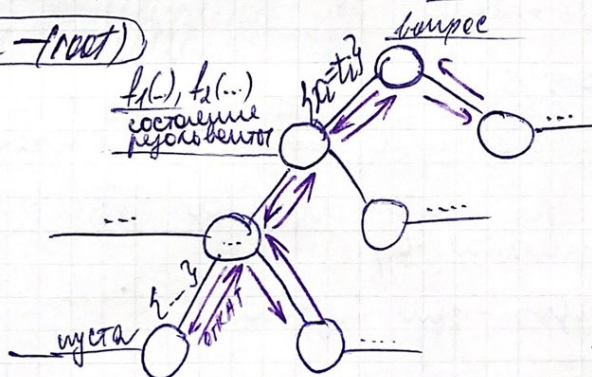
↑
σκατορ

Дерево поиска решений

• ф.б. грим. мушкет ~~то~~

Дерево поиска решений строится не системно, а по уровню.

Баня (root)



каждое верушка - опре-
делённое состояние вопроса
каждое слово -

Переход в вершинно-
порожнотное
применено пост.

Мист по-се усмихна
если речевета нуста,
(Ветина)

Будем считать, что при попытке перехода возникает пункт. ситуация?

Рано или поздно восстанавливается весь вопрос.

Если вы-и отдают нечто вост. пред. сост.
революцион., месикр. перемены и т.д.

⇒ зем. области адм. для восстановления.

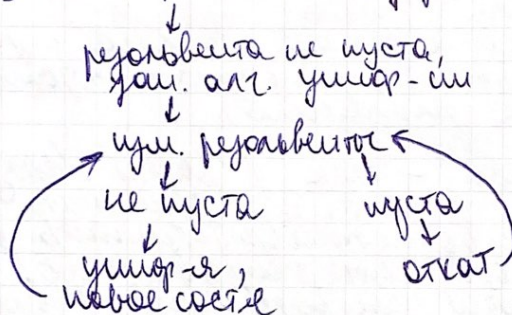
Система должна иметь признак конца работы. — пока ^{реж.} не пуста.
(комната про работу машины)

Если где-то док-ва нек. термина порождает факт, тело — пустое →
⇒ рефлексива частично освобождается.

Рефлексива пуста ⇒ однократно получить ответ "да".

Тупиковые ситуации (при попытке доказать верность рефлексива, второй эл-т ...)

Задаем вопрос → пом. в рефлексиву



1) Формальный признак остановки работы системы

2) Обеспечение не использ-я одних и тех же знаний при ответе на вопрос ("зацикливание")

- правила, кот-е уже использованы, спец. образом помечаются
- метки постепенно (после возврата) снимаются (внутри процедуры)

Рекурсия в Prolog:

в теле правил сослаться на знание

⇒ м.б. так, что предикат (знание) помечено неск-раз.

Форм. признак: все метки как-то "вынуть" из-под знания — все знание сб.

Чем меньше переходов — тем лучше

Невыполнение эф-ти работ программ.

Предикат - одно значение, формируемое в виде одной процедуры.

① предусматривать проверку соблюдения процедур и предложений в процедурах.

② использование системных предикатов.

% данных \neq тип данных

↑
выделение на передок работы систем.

Системные предикаты



- (воскл. знак)
- сигн. (отсечение)
- не имеет аргументов
- абс. значением и. функции.
- можно использовать везде, где м.б. использован составной терм.
- м.б. использован в некотором правиле.

правило
вызывает
системный

→ $r1:- a, b, !, c, d.$

$r2:- \dots$

если a - составной (составное значение)

то d - попытка д-ть b .

если b - хвост. сигн-а, попытка пересом. a .

если система д-ла a и д-ла b , то $!$ на приемном ходе - истина.

если c - сигн., то $\rightarrow d$.

если d - хвост., пересом. c .

если происх. попытка пересом. b (после c),
то запрещается использовать всю процедуру $r1$ в программе.

- можно отсечь безуспешные ветви