

Введение в ИИ на примере языка Prolog

Ресурсия в Prolog

<https://github.com/Inscriptor/IntroductionToAI/tree/master/pdf>

Федеральное государственное автономное образовательное учреждение высшего образования
«Новосибирский национальный исследовательский государственный университет»

25 сентября 2019 г.

Определения и примеры

Рекурсия в Prolog

Простой пример

переваривает(Хищник,Добыча) :- съел(Хищник,Добыча) .

переваривает(Хищник,Добыча) :- съел(Хищник,ДругойХищник) ,
переваривает(ДругойХищник,Добыча) .

съел(комар,кровь) .

съел(лягушка, комар) .

съел(цапля, лягушка) .

Уровни понимания программ

Декларативный и процедурный

Декларативный смысл Prolog-программы касается только *отношений*, определенных в программе. Другими словами, декларативный смысл определяет, что должно быть результатом работы программы с точки зрения математической логики.

Процедурный смысл программы определяет еще и как этот результат был получен, т.е. как отношения реально обрабатываются пролог-системой.

Правила описания рекурсивных предикатов

База и шаг рекурсии

База рекурсии

переваривает(Хищник,Добыча) :- съел(Хищник,Добыча) .

переваривает(Хищник,Добыча) :- съел(Хищник,ДругойХищник) ,
переваривает(ДругойХищник,Добыча) .

съел(комар,кровь) .

съел(лягушка, комар) .

съел(цапля, лягушка) .

Правила описания рекурсивных предикатов

База и шаг рекурсии

Шаг рекурсии

переваривает(Хищник,Добыча) :- съел(Хищник,Добыча) .

переваривает(Хищник,Добыча) :- съел(Хищник,ДругойХищник) ,
переваривает(ДругойХищник,Добыча) .

съел(комар,кровь) .

съел(лягушка, комар) .

съел(цапля, лягушка) .

Правила описания рекурсивных предикатов

descending.pl

```
child(adam,cain).  
child(adam,abel).  
child(adam,seth).  
child(cain,enoch).  
child(enoch,irad).  
child(irad,mehujael).  
child(seth,enos).  
child(enos,kenan).  
child(kenan,mahalalel).
```

Правила описания рекурсивных предикатов

descending.pl

```
child(adam,cain).  
child(adam,abel).  
child(adam,seth).  
child(cain,enoah).  
child(enoah,irad).  
child(irad,mehujael).  
child(seth,enos).  
child(enos,kenan).  
child(kenan,mahalalel).  
descend(A,D) :- child(A,D).  
descend(A,D) :- child(A,C), descend(C,D).
```


Правила описания рекурсивных предикатов

arithmetic.pl

```
num(0) .
```

```
num(s(N)) :- num(N) .
```

Правила описания рекурсивных предикатов

arithmetic.pl

```
num(0).
```

```
num(s(N)) :- num(N).
```

```
plus(0,N,N).
```

```
plus(s(M),N,s(Sum)) :- plus(M,N,Sum).
```

Рекурсия и хвостовая рекурсия

Численная арифметика: вычисление факториала числа

$f(0,1).$

$f(X,Y) :- X > 0, X1 \text{ is } X-1, f(X1,Y1), Y \text{ is } Y1 * X.$

Рекурсия и хвостовая рекурсия

Численная арифметика: вычисление факториала числа

```
f(0,1).
```

```
f(X,Y) :- X>0,X1 is X-1,f(X1,Y1),Y is Y1*X.
```

```
f(N,N,F):-write(F),!.
```

```
f(N,N1,F1):- N2 is N1+1,F2 is F1*N2,f(N,N2,F2).
```

```
factorial(N):- f(N,1,1).
```

Последовательность целей и остановка программ

Возможные источники ошибок

```
child(adam,cain).  
child(adam,abel).  
child(adam,seth).  
child(cain,enoah).  
child(enoah,irad).  
child(irad,mehujael).  
child(seth,enos).  
child(enos,kenan).  
child(kenan,mahalalel).
```

```
descend(A,D) :- child(A,D).
```

```
descend(A,D) :- child(A,C), descend(C,D).
```

Последовательность целей и остановка программ

Возможные источники ошибок

```
child(adam,cain).  
child(adam,abel).  
child(adam,seth).  
child(cain,enoah).  
child(enoah,irad).  
child(irad,mehujael).  
child(seth,enos).  
child(enos,kenan).  
child(kenan,mahalalel).
```

```
descend(A,D) :- descend(C,D), child(A,C).
```

```
descend(A,D) :- child(A,D).
```

Последовательность целей и остановка программ

Возможные источники ошибок

```
child(adam,cain).
```

```
child(adam,abel).
```

```
child(adam,seth).
```

```
child(cain,enoah).
```

```
child(enoah,irad).
```

```
child(irad,mehujael).
```

```
child(seth,enos).
```

```
child(enos,kenan).
```

```
child(kenan,mahalalel).
```

```
descend(A,D) :- child(A,D).
```

```
descend(A,D) :- descend(C,D), child(A,C).
```

Последовательность целей и остановка программ

Возможные источники ошибок

```
child(adam,cain).  
child(adam,abel).  
child(adam,seth).  
child(cain,enoch).  
child(enoch,irad).  
child(irad,mehujael).  
child(seth,enos).  
child(enos,kenan).  
child(kenan,mahalalel).
```

```
descend(A,D) :- child(A,C), descend(C,D).  
descend(A,D) :- child(A,D).
```


Последовательность целей и остановка программ

Возможные источники ошибок

```
num(0).
```

```
num(s(N)) :- num(N).
```

```
num(s(N)) :- num(N).
```

```
num(0).
```

Упражнения

Упражнения

- ▶ Скачайте или перепишите листинг программы `descending.pl`. Рассмотрите каждый случай рекурсии (см. слайды ранее). Какие запросы работают, какие нет? Почему?
- ▶ Скачайте или перепишите листинг программы `arithmetic.pl`. Реализуйте следующие операции над числами, определенными в программе:
 1. **Умножение**: реализуйте предикат `times/3`, такой, чтобы последним аргументом было произведение двух первых.
 2. **Возведение в степень**: реализуйте предикат `exp(N,X,R)`, такой, чтобы $R = X^N$.
 3. **Сравнение**: реализуйте предикаты `gt/2`, `lt/2`, истинные тогда, когда первый аргумент больше (меньше), чем второй.
 4. **Равенство**: реализуйте предикат `eq/2`, истинный тогда и только тогда, когда его аргументы равны.
 5. **Максимум и минимум двух чисел**: реализуйте предикаты `max/3`, `min/3`, где третий аргумент — это максимум (минимум) из двух первых.

Упражнения

- ▶ Реализуйте программу для вычисления факториала числа (в обычной численной арифметике). Сравните время работы и количество операций в зависимости от значения входных параметров.
- ▶ Реализуйте программу для вычисления чисел Фибоначчи. Реализуйте предикат `getFibN/2 = getFibN(N, Fn)`, такой, что F_n — N -е число Фибоначчи.
- ▶ Используя предыдущую программу, реализуйте предикат `getNearestFibonacci/3`, возвращающий по заданному числу номер ближайшего к нему числа Фибоначчи и его номер.