

# Введение в ИИ на примере языка Prolog

## Согласование и поиск решений

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

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

11 сентября 2019 г.

# Унификация термов и поиск решений

## Основные пункты

- ▶ Познакомиться с понятием согласования термов.
- ▶ Разобраться в стратегиях, используемых интерпретатором языка Prolog для поиска ответов на вопросы.

# Унификация термов и поиск решений

## Определение

Два терма **унифицированы**, если они равны или если они содержат переменные и существует означивание этих переменных, такое, что термы становятся равными.

# Унификация термов и поиск решений

## Определение

1. Если  $term1$  и  $term2$  константы, то  $term1$  и  $term2$  унифицированы тогда и только тогда, когда они равны друг другу, т.е.  $term1$  и  $term2$  это один и тот же атом или число.
2. Если  $term1$  — это переменная, а  $term2$  — любой терм, то термы  $term1$  и  $term2$  унифицированы, и переменная  $term1$  инициализируется значением терма  $term2$ . Если  $term1$  и  $term2$  — переменные, они унифицированы и принимают значения друг друга. Говорят, что переменные  $term1$  и  $term2$  **разделяют значение**.
3. Если  $term1$  и  $term2$  составные термы, то они унифицированы тогда и только тогда, когда:
  - (a) Их функторы и арность совпадают.
  - (b) Соответствующие аргументы обоих термов унифицированы. При этом означивание переменных согласовано.
4. Два терма унифицированы тогда и только тогда, когда это следует из одного из предыдущих пунктов.

# Унификация термов и поиск решений

## Примеры унификации

### Какие из термов унифицированы?

- ▶ `john = john.`
- ▶ `42 = 42.`
- ▶ `'Строка слов' = 'Строка слов'.`
- ▶ `'mia' = mia.`
- ▶ `'42' = 42.`
- ▶ `mia = X.`
- ▶ `X = Y.`
- ▶ `kill(shoot(gun),Y) = kill(X,stab(knife)).`
- ▶ `loves(X,X) = loves(marcellus,mia).`
- ▶ `parent(A) = A.`

# Унификация термов и поиск решений

## Поиск решений

Интерпретатор языка Prolog обрабатывает условия запроса последовательно слева направо. Для каждого из условий интерпретатор ищет унифицируемый с ним факт или вывод правила, проходя базу знаний сверху вниз.

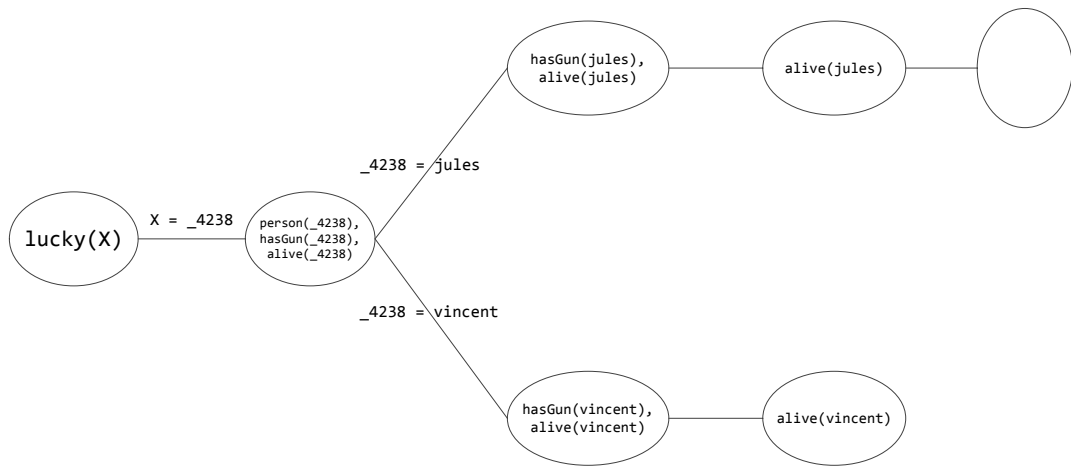
# Унификация термов и поиск решений

## Поиск решений

```
person(vincent).  
person(jules).  
hasGun(vincent).  
hasGun(jules).  
alive(jules).  
lucky(X) :- person(X),hasGun(X),alive(X).
```

# Унификация термов и поиск решений

## Дерево поиска решений





# Унификация термов и поиск решений

## Упражнения

В каких из перечисленных пар термы унифицированы? Где необходимо запишите значения переменных, необходимые для унификации.

1. `bread = bread.`
2. `'Bread' = bread.`
3. `'bread' = bread.`
4. `Bread = bread.`
5. `хлеб = котлета.`
6. `food(bread) = bread.`
7. `food(bread) = Bread.`
8. `food(Bread) = food(bread).`
9. `food(хлеб,X) = food(Y,котлета).`
10. `graph(edge(v1,v2),X,edge(Y,v4)) = graph(Z,edge(v7,v8),edge(v4,X)).`
11. `food(X) = X.`
12. `meal(food(bread),drink(beer)) = meal(food(X),Y).`
13. `meal(food(bread),X) = meal(X,drink(beer)).`

# Унификация термов и поиск решений

## Упражнения

Пусть дана следующая база знаний.

```
witch(gullveig).
```

```
god(freyja).
```

```
god(odin).
```

```
god('Vidarr').
```

```
magic(X) :- witch(X).
```

```
magic(X) :- wizard(X).
```

```
magic(X) :- god(X).
```

Каким из запросов она удовлетворяет?

1. `magic(witch).`

2. `wizard(odin).`

3. `magic('Vidarr').`

4. `magic(Vidarr).`

# Унификация термов и поиск решений

## Упражнения

Дан следующий лексикон.

`word(article,a).`

`word(article,every).`

`word(noun,criminal).`

`word(noun,'big kahuna burger').`

`word(verb,eats).`

`word(verb,likes).`

Написать запрос на получение списка всех возможных фраз данного лексикона, имеющих вид: *артикль+существительное+глагол+артикль+существительное*.

Построить дерево поиска решений для одной из построенных фраз.

# Унификация термов и поиск решений

## Упражнения

Имеется следующий список слов: *бок, акр, сомо, бокс, хата, эдда, вино, рода, окот, анод, скала, онагр, обиход, монако, дверка, дигора, удокан, маарри, бинокль, бегония*. Пусть в базе знаний слова представляются следующим образом:

`word(эдда,э,д,д,а) .`

`word(вино,в,и,н,о) .`

`word(рода,р,о,д,а) .`

`word(окот,о,к,о,т) .`

`word(анод,а,н,о,д) .`

`word(скала,с,к,а,л,а) .`

`word(онагр,о,н,а,г,р) .`

`word(обиход,о,б,и,х,о,д) .`

# Унификация термов и поиск решений

## Упражнения

Реализовать предикат `crosswd/10` =

`crosswd(V1,V2,V3,V4,V5,V6,H1,H2,H3,H4)`, который бы заполнял данными словами следующий кроссворд таким образом, чтобы слова, идущие вертикально, были первыми 6 аргументами, а слова, идущие горизонтально — последними.

v <sup>1</sup>		v <sup>3</sup>		v <sup>5</sup>	v <sup>6</sup>
h <sup>1</sup>	v <sup>2</sup>		v <sup>4</sup>		
h <sup>2</sup>					
h <sup>3</sup>					
	h <sup>4</sup>				