

Введение в ИИ на примере языка Prolog

Списки

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

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

30 сентября 2019 г.

Понятие и вид списка в Prolog

Списки

Список в Prolog

- ▶ Списком называют конечную последовательность элементов.
- ▶ В синтаксисе языка Prolog границы списка определяются квадратными скобками [и].
- ▶ Элементы списка отделяются друг от друга запятой. Элементами списка могут быть любые термы.
- ▶ Любой непустой список можно разделить на голову и хвост (head и tail). Для записи используется специальный символ |. $[Head \mid Tail] = [one, two, three] \Rightarrow Head = one, Tail = [two, three]$. Голова списка — это элемент, хвост списка — это список.
- ▶ Пустой список не может быть разделен на голову и хвост.

Списки

Простой пример

```
[]
```

```
[first, second, third, fourth, fifth]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
```

```
[first, 2, color(cornie, black), F, fifth, F]
```

```
[first, second, [third, fourth], [fifth, color(cornie, black)]]
```

```
[[], [], car(volkswagen), F, 1, 2, [1, F, car(bmw), [1, 2, 4]], X]
```

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

► `[H|T] = [abyssian, bobtail, [bengal, birman]]`.

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

- ▶ $[H|T] = [\text{abyssian}, \text{bobtail}, [\text{bengal}, \text{birman}]]$.
- ▶ $[F,S|T] = [\text{abyssian}, \text{bobtail}, [\text{bengal}, \text{birman}]]$.

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

- ▶ `[H|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[F,S|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[_ ,S|T] = [abyssian, bobtail, [bengal, birman]]`.

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

- ▶ `[H|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[F,S|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[_ ,S|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[First,_,_,Fourth|_] = [abyssian, bobtail, [bengal, birman]]`.

Списки

Получение элементов списка

```
[abyssian, bobtail, [bengal, birman]]
```

- ▶ `[H|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[F,S|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[_ ,S|T] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[First,_,_,Fourth|_] = [abyssian, bobtail, [bengal, birman]]`.
- ▶ `[_ ,_,[_ |T]|_] = [abyssian, bobtail, [bengal, birman]]`.

Операции со списками

Проверка вхождения элемента

Предикат `member/2 = member(?Elem, ?List)` имеет два аргумента — некоторый терм и список, и принимает значение `true` в случае, когда `?Elem` содержится в списке `?List`. В противном случае предикат принимает значение `false`.

Операции со списками

Проверка вхождения элемента

Предикат `member/2 = member(?Elem, ?List)` имеет два аргумента — некоторый терм и список, и принимает значение `true` в случае, когда `?Elem` содержится в списке `?List`. В противном случае предикат принимает значение `false`.

Как можно реализовать предикат `member/2` самостоятельно?

Операции со списками

Проверка вхождения элемента

Предикат `member/2 = member(?Elem, ?List)` имеет два аргумента — некоторый терм и список, и принимает значение `true` в случае, когда `?Elem` содержится в списке `?List`. В противном случае предикат принимает значение `false`.

Как можно реализовать предикат `member/2` самостоятельно?

```
member(X, [X|_]).
```

Операции со списками

Проверка вхождения элемента

Предикат `member/2 = member(?Elem, ?List)` имеет два аргумента — некоторый терм и список, и принимает значение `true` в случае, когда `?Elem` содержится в списке `?List`. В противном случае предикат принимает значение `false`.

Как можно реализовать предикат `member/2` самостоятельно?

```
member(X, [X|T]).
```

```
member(X, [_|T]) :- member(X,T).
```

Операции со списками

Проверка вхождения элемента

Предикат `member/2 = member(?Elem, ?List)` имеет два аргумента — некоторый терм и список, и принимает значение `true` в случае, когда `?Elem` содержится в списке `?List`. В противном случае предикат принимает значение `false`.

Как можно реализовать предикат `member/2` самостоятельно?

```
member(X, [X|_]).
```

```
member(X, [_|T]) :- member(X,T).
```

```
member(X, [X|_]).
```

```
member(X, [_|T]) :- member(X,T).
```

Операции со списками

Основные операции

- ▶ Подсчет длины списка.
- ▶ Конкатенация двух списков.
- ▶ Поиск префикса, суффикса и подсписка заданного списка.
- ▶ Поиск последнего элемента списка.
- ▶ Обращение списка
- ▶ Сортировка.