



Сошников Дмитрий Валерьевич

к.ф.-м.н., доцент

<http://soshnikov.com>

Лекция 1: Введение

Логическое программирование

<https://soshnikov.com/courses/logpro/>

Обо мне

<http://soshnikov.com>



- Майкрософт, Cloud Developer Advocate
- Кандидат физ.-мат. наук
 - Распределенные интеллектуальные системы с явным представлением знаний
 - Интеллектуальная реструктуризация социальных сетей на основе онтологий
- Доцент:
 - Факультет компьютерных наук, НИУ ВШЭ
 - Кафедра Алгоритмов и технологий программирование, МФТИ
 - Кафедра Вычислительной математики и программирования МАИ
 - Студенческая лаборатория MAILabs (www.mailabs.ru)

Программирование? Really?



```
d(X,X,1) :- !.  
d(T,X,0) :- atomic(T).  
  
d(U+V,X,DU+DV) :- d(U,X,DU), d(V,X,DV).  
d(U-V,X,DU-DV) :- d(U,X,DU), d(V,X,DV).  
d(-T,X,-R) :- d(T,X,R).  
d(C*U,X,C*W) :- atomic(C), C\\=X, !, d(U,X,W).  
d(U*V,X,Vd*U+Ud*V) :- d(U,X,Ud), d(V,X,Vd).  
d(U/V,X,(Ud*V-Vd*U)/(V*V)) :- d(U,X,Ud),  
d(V,X,Vd).
```

```
?- d((x-1)/(x+1),x,R).  
R = ((1-0)*(x+1)-(1+0)*(x-1))/((x+1)*(x+1))
```




Мечта человечества

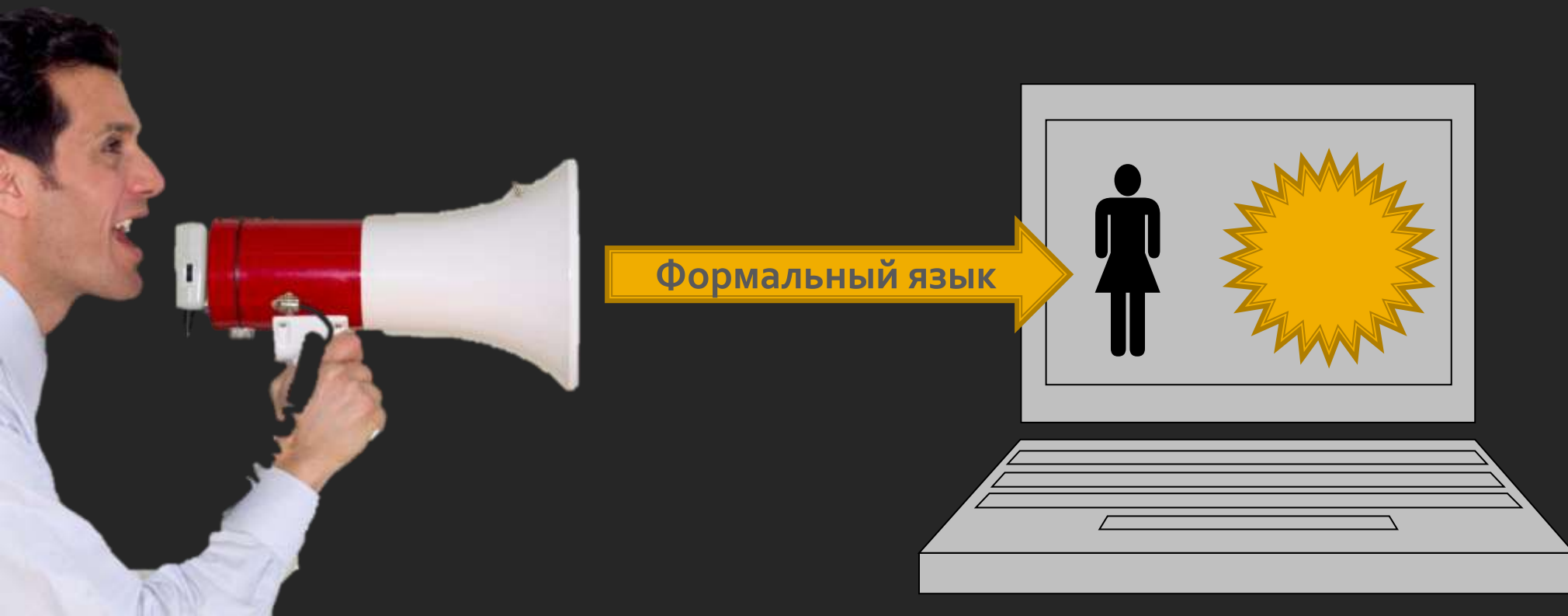


Возможно ли это?



- Тест Тьюринга – подробнее в курсе ИИ
- Проблемы:
 - Неоднозначность человеческого языка
 - При коммуникации мы полагаемся на картину мира, которая есть у нас в голове (common knowledge)

Потенциальный способ реализации



Путь к автоматическому решению задач



- Формальный язык: логика
- Метод рассуждений: логический вывод

Логика
высказываний

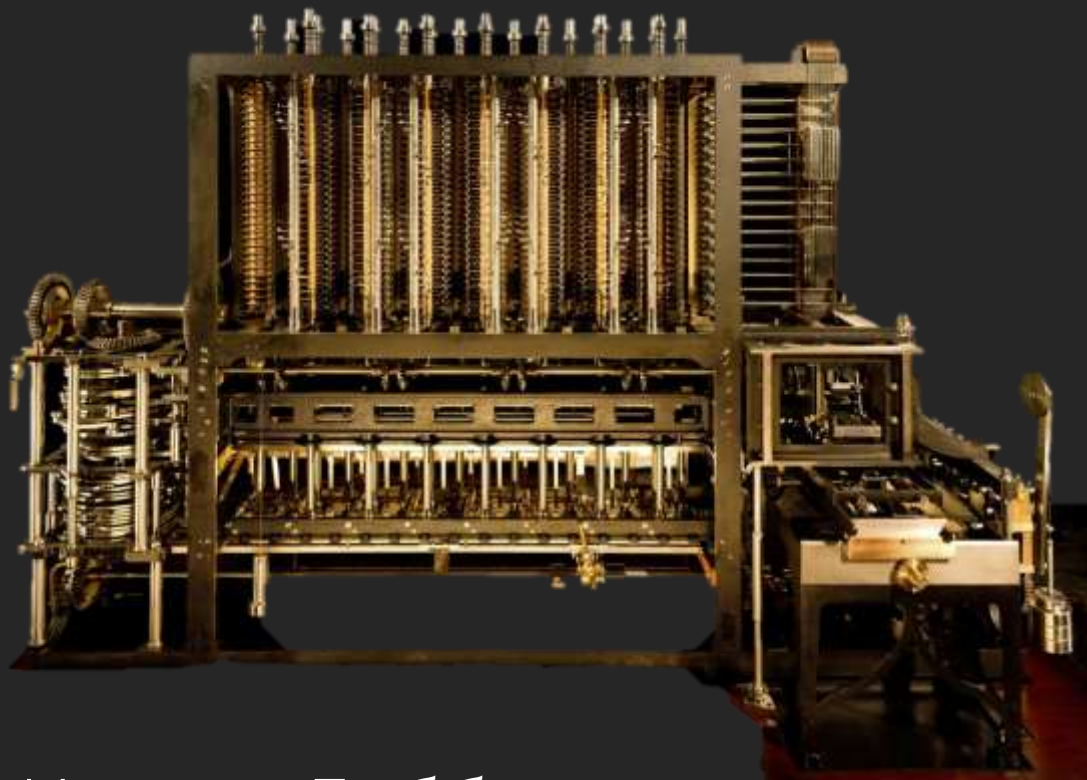
Логика
предикатов 1 п



Лог.про.

Семейство дескриптивных логик

Как появились компьютеры

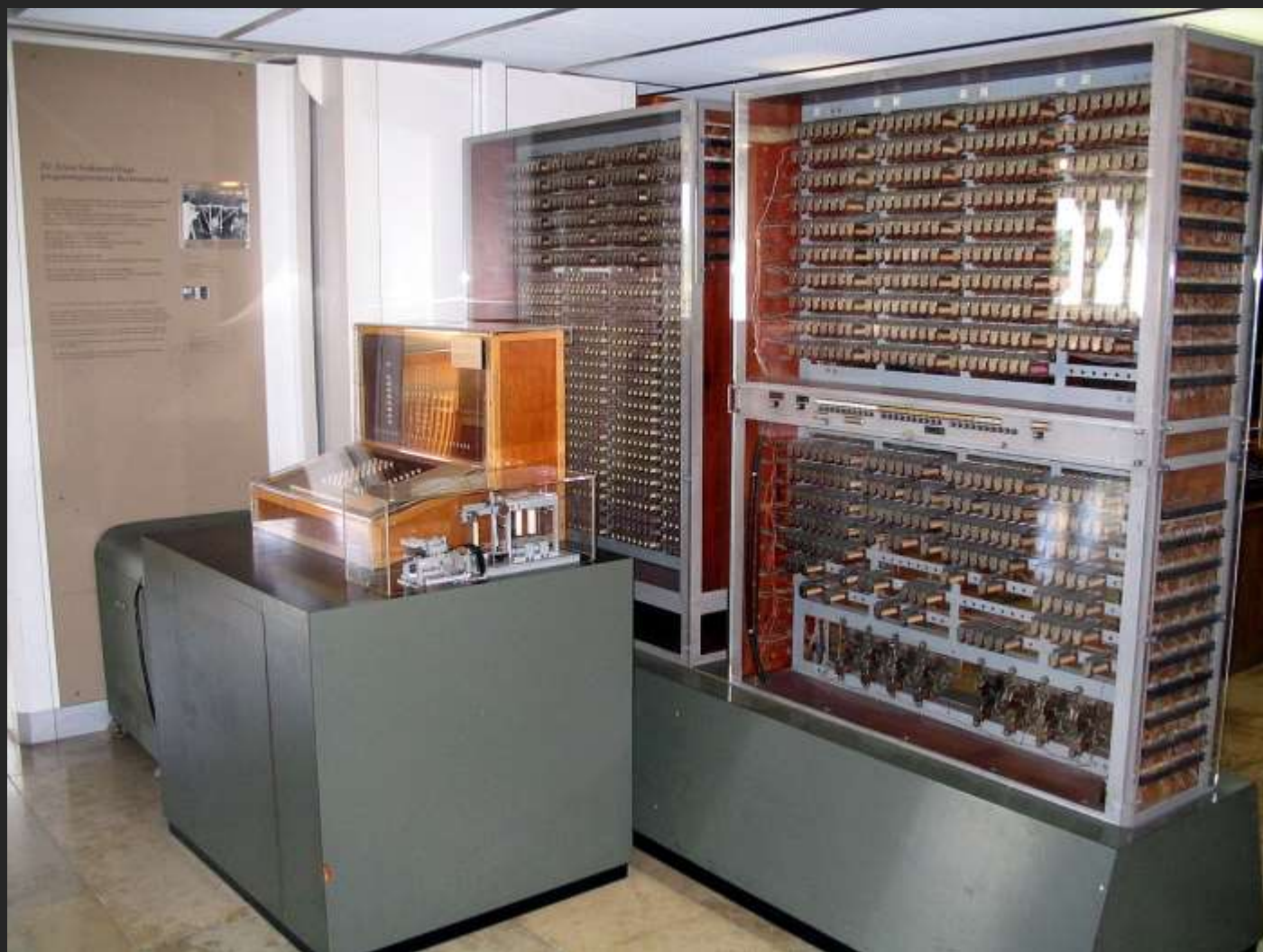


Чарльз Беббидж,
1822 г.



IBM, 1890 г.

Конрад Цузе, Z3, 1938 г.



Обратимся к истории



1954-57 г., Дж.Бэкус

- FORTRAN
- язык ассемблера
- машинные коды
- программирование переключателей

```
0000 0A 12 1F 4B C3 E0 EE F1
0008 C3 1D 23 17 F2 00 0C 0D
0010 ...
```

```
      MOV     AX, [ARG1]
      ADD     AX, [ARG2]
      MOV     [RES], AX
      JMP     NEXT
ARG1:  DB      10
ARG2:  DB      20
RES:   DB      0
NEXT:  ...
```

```
      S = 0
      DO 10 I=1,10
      S = S + I*I
10    CONTINUE
```

1950

1960

1970

1980

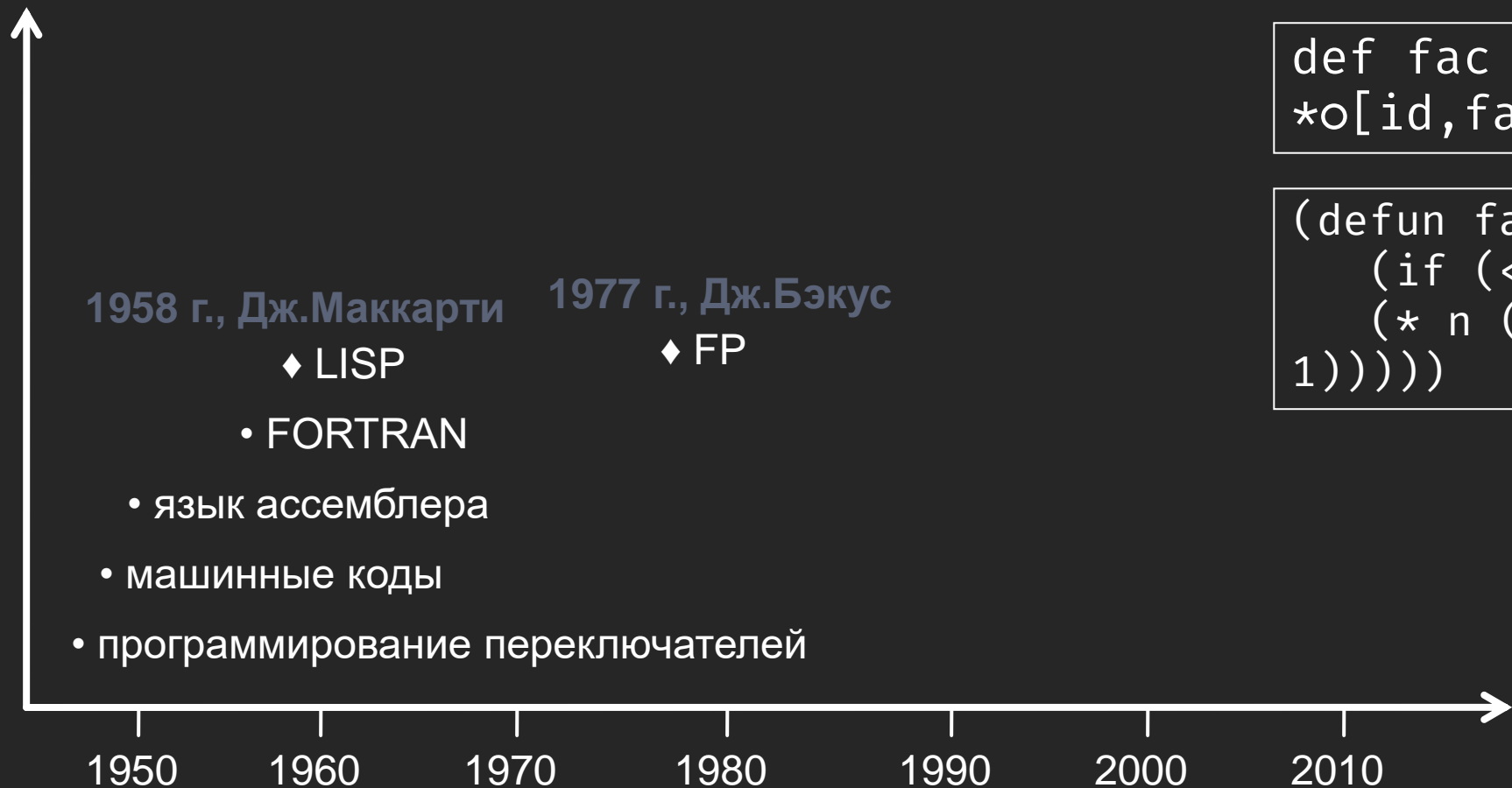
1990

2000

2010

- Первый язык программирования высокого уровня – ФОРТРАН – был создан Дж.Бэкусом, чтобы математики могли программировать на уровне формул.

Программирование для математиков

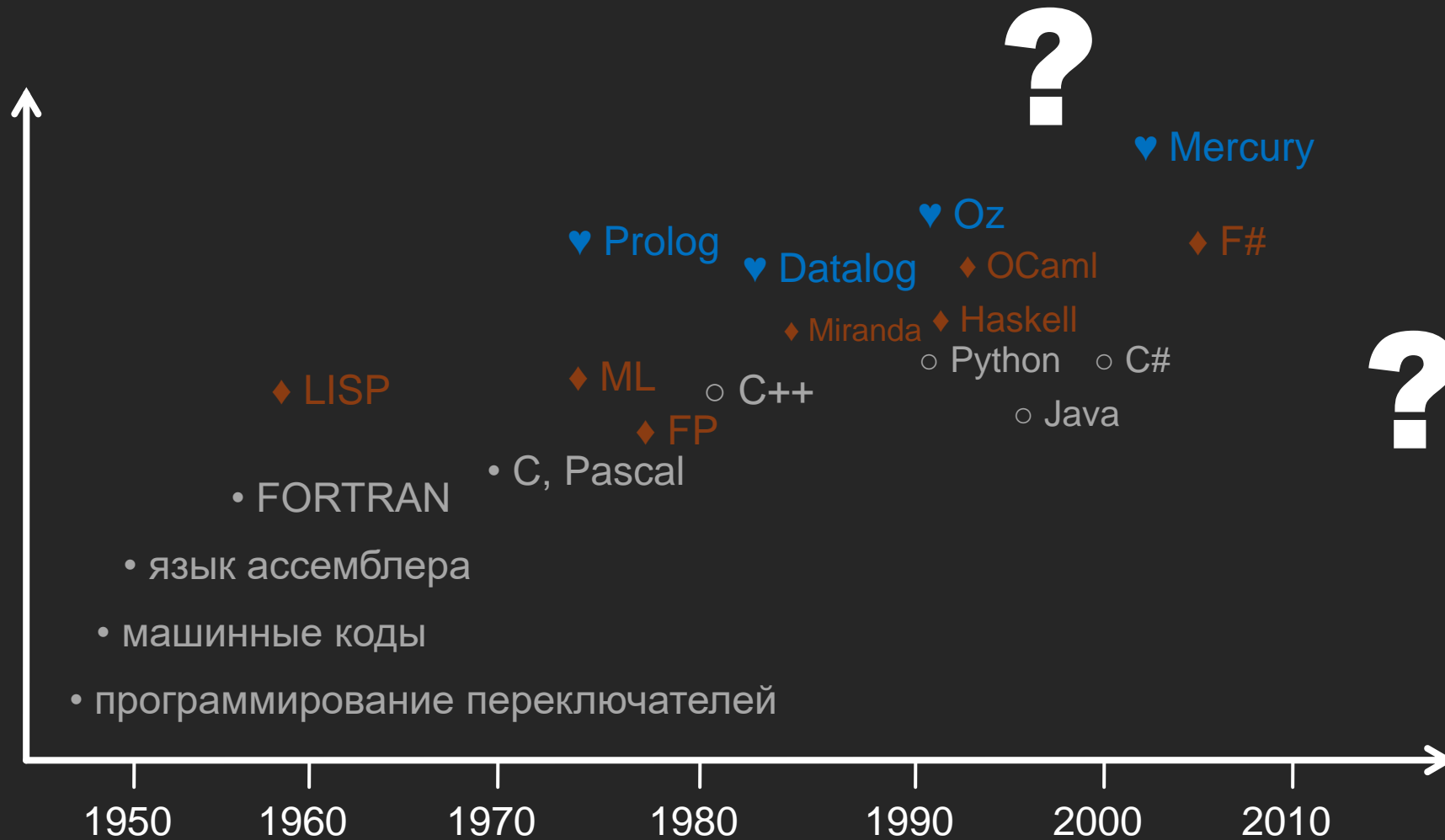


```
def fac = eq 0 → 1;  
*o[id,faco(-o [id,1])]
```

```
(defun factorial (n)  
  (if (<= n 1) 1  
      (* n (factorial (- n  
1))))))
```

- Позже Дж.Бэкус пошел дальше и предложил язык FP, в котором формулы более соответствовали математическому понятию функции

Языки логического программирования



Декларативность



```
d(X,X,1) :- !.  
d(T,X,0) :- atomic(T).  
  
d(U+V,X,DU+DV) :- d(U,X,DU), d(V,X,DV).  
d(U-V,X,DU-DV) :- d(U,X,DU), d(V,X,DV).  
d(-T,X,-R) :- d(T,X,R).  
d(C*U,X,C*W) :- atomic(C), C\\=X, !, d(U,X,W).  
d(U*V,X,Vd*U+Ud*V) :- d(U,X,Ud), d(V,X,Vd).  
d(U/V,X,(Ud*V-Vd*U)/(V*V)) :- d(U,X,Ud),  
d(V,X,Vd).
```

$(\forall U) (\forall V) (\forall X) (\forall dU) (\forall dV)$
 $(\forall pA) (\forall pB) (\forall A) (\forall B)$
 $d(U,X,DU) \wedge d(V,X,DV) \supset$
 $d(U+V,X,dU+dV)$

```
?- d((x-1)/(x+1),x,R).  
R = ((1-0)*(x+1)-(1+0)*(x-1))/((x+1)*(x+1))
```

Декларативность



```
fact(1)=1.  
fact(N)=N*fact(N-1).
```

Логическое программирование (Mercury)

```
function  
fact(x:integer):integer;  
var i, r : integer;  
begin  
    r:=1;  
    for i:=1 to x do r:=r*i;  
    fact:=r  
end;
```

*Императивное программирование
(Pascal)*

Логическое программирование



- Парадигма декларативного программирования, в которой
 - программа представляет собой описание требуемого решения в терминах определенной логики
 - решение задачи строится в процессе логического вывода по заданному описанию
- Различные разновидности логического программирования: индуктивное, в ограничениях, ...
- Подход к программированию
- Языки программирования Prolog, Datalog, Mercury, Oz, ...

Парадигмы программирования



Императивные языки = императивное программирование

- Оперируют состоянием памяти. Выполнение операторов изменяет состояние.
- Алгоритмическая модель: машина Тьюринга, процессор фон-Неймана

Функциональные языки = аппликативное программирование

- Оперируют данными. Применение функции к аргументам изменяет данные.
- Подход, ориентированный на данные.
- Алгоритмическая модель: лямбда-исчисление, комбинаторная логика

Логические языки = декларативное программирование

- Оперируют пространством поиска решений.
- Программа задаёт множество возможных переходов в пространстве поиска
- Алгоритмическая модель: логика дизъюнктов Хорна

Парадигмы программирования



Объединение роторных вычислителей
Слева направо: АНАЛИТИЧЕСКАЯ МАШИНА (США),
Z3 (Германия), КОЛОССУС (Великобритания).

Императивное (алгоритмическое)

- Машина Тьюринга, Машина фон Неймана
- Pascal, C и т.д.



Аппликативное (функциональное)

- λ-исчисление, рекурсивные функции
- F#, LISP / Scheme, ML и друзья, Haskell



Декларативное (логическое)

- Ограниченная логика предикатов 1-го порядка
- Prolog, Mercury, Oz, ...

Ситуационное (продукционное)

- Нормальные алгоритмы Маркова
- Рефал

Объектное, компонентное, многоагентное (эмерджентное)

- Синергетика, теория сложных систем

Мультипарадигмальные языки



- C# - императивный (ОО) + элементы функциональности
- F# - функциональный с элементами императивности
- Mercury – функционально-логический
- Oz
- Python
- ...

**Почему важно изучать логическое
программирование?**

Какие задачи хорошо решаются на логических языках?



- Задачи искусственного интеллекта
 - Экспертные системы
- Лингвистика, обработка естественного языка
- Задачи с неопределенностью
- Задачи, связанные с поиском решений
- Мета-программирование, построение специализированных языков

Особенности логических языков



- Отсутствие операторов присваивания и побочных эффектов
- Декларативное программирование
- Естественная математическая модель вычислений
- Заложенная в язык возможность возвратов и перебора
- Заложенные в язык возможности по представлению списков, деревьев
- Развитые возможности мета-программирования и построения проблемно-ориентированных языков

Как установить Prolog?

Варианты использования



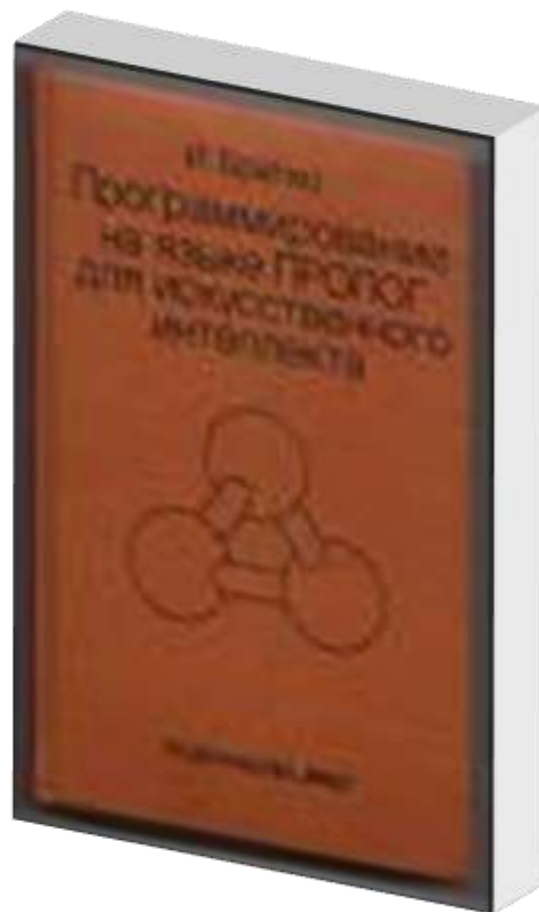
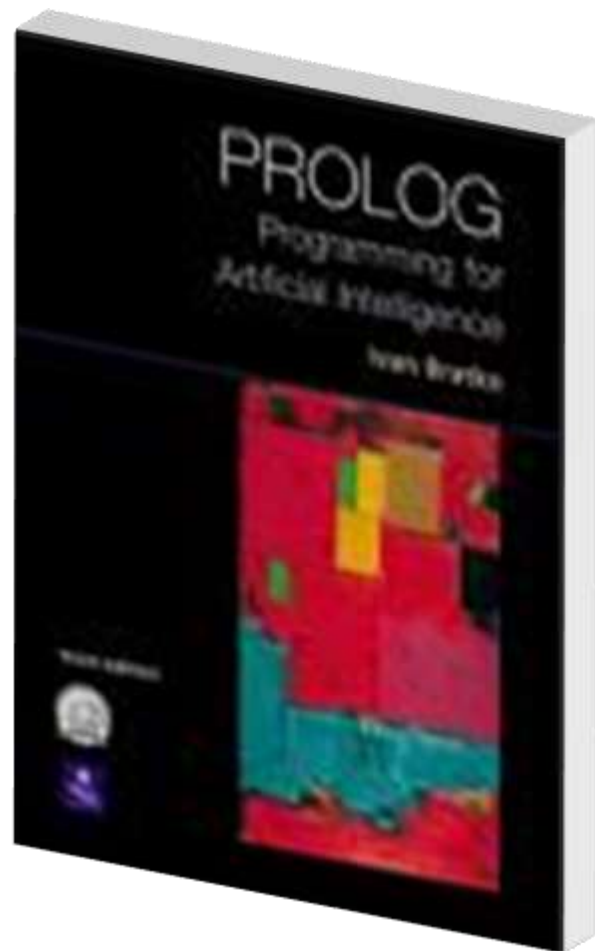
- SWI Prolog + Visual Studio Code (с VSC Prolog Extension)
 - SWI Prolog: <http://www.swi-prolog.org/>
 - VS Code: <https://code.visualstudio.com/>
 - VSC Extension: <https://marketplace.visualstudio.com/items?itemName=arthurwang.vsc-prolog>
- SWI Prolog + Jupyter Notebook Kernel
 - Start Binder from <http://github.com/shwars/logpro>
 - Jupyter Kernel: <https://github.com/targodan/jupyter-swi-prolog>
- GNU Prolog: <http://www.gprolog.org/>
- Strawberry Prolog: <http://www.dobrev.com/>

Что будем изучать



- Принципы логического программирования
- Математическая теория в основе логического программирования – логика предикатов, логический вывод, вывод типов
- Семантика языков логического программирования, вопросы реализации
- Языки логического программирования:
 - Prolog
 - Mercury

Источники



Источники



- Сошников Д.В., Парадигма логического программирования
- Братко И. Программирование на языке Пролог для искусственного интеллекта. пер. с англ. – М.: Мир, 1990.
- Bratko I. Programming in Prolog for Artificial Intelligence (3rd edition), Addison-Wesley Publishers, 2001.
- Клоксин У., Меллиш К. Программирование на языке Пролог. – М.: Мир, 1987.
- Хоггер К. Введение в логическое программирование: Пер. с англ. -М.: Мир, 1988.
- Набебин А.А. Логика и Пролог в дискретной математике. – М.: Изд-во МЭИ, 1996.
- Малпас Дж. Реляционный язык Пролог и его применение: Пер. с англ. -М.: Наука, 1990.
- Стерлинг Х., Шапиро Э. Искусство программирования на языке Пролог: Пер. с англ. – М.: Мир, 1990.

Вопросы?



- http://t.me/log_pro
- <https://soshnikov.com/courses/logpro/>