

История

1899г. Готлоб Фреге – Begriffsschrift, исчисление понятий

$f f f a$



1920г. Шейнфинкель Моисей - О строительных камнях
математической логики

Хаскелл Брукс Карри





Алонзо Чёрч



Пусть K — множество всех множеств, которые не содержат себя в качестве своего элемента. Содержит ли K само себя в качестве элемента? Если да, то, по определению K , оно не должно быть элементом K — противоречие. Если нет — то, по определению K , оно должно быть элементом K — вновь противоречие.

Вряд ли что-нибудь может быть более нежелательным для ученого, чем обнаружить, что основания едва завершенной работы рухнули. Письмо, полученное мной от Бертрана Рассела, поставило меня именно в такое положение...

Джон Маккарти, 1958
LISt Processing



```
(define fac (n)
  (if (eq n (quote 0))
      (quote 1)
      (mul n (fac (sub n (quote 1))))))
```

Lots of Infuriating, Stupid Parenthesis



Питер Ландин

1964 – SECD

1966 – ISWIM (if you see what I mean)

$\text{rec fac}(n) = (n = 0) \rightarrow 1 ; n \text{ fac}(n-1)$



1963 - Джон Алан Робинсон, принцип резолюции
→ автоматическое доказание теорем



1971 Алан Колмероз – пролог

PROgrammation en LOGique

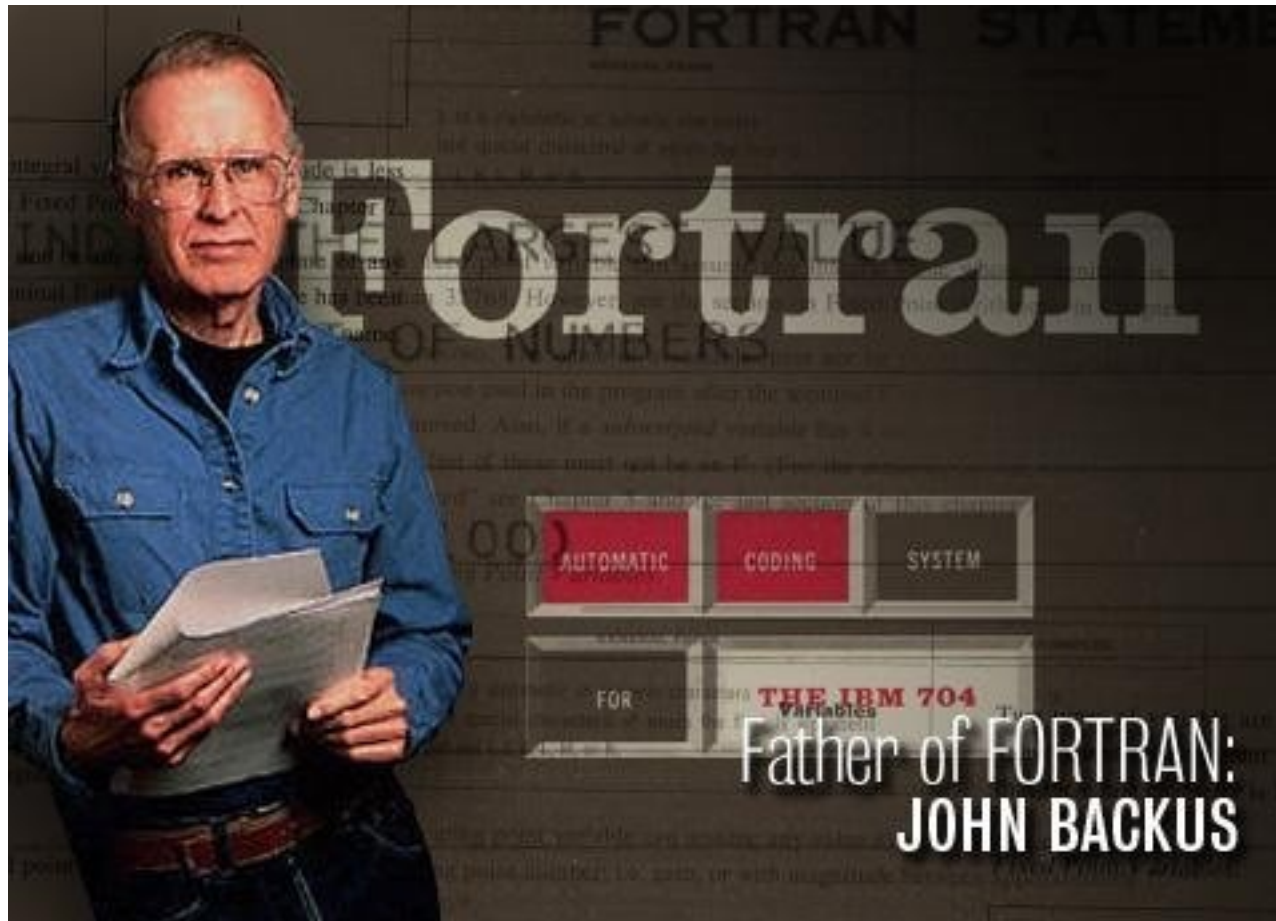
```
fac(0, 1).
```

```
fac(N, M):- sub(N, 1, N1), fac(N1,M1), mul(N, M1, M).
```

1976 Девид Уоррен - Edinburgh Prolog, WAM

1976 Петер Хендерсон и Джеймс Моррис LispKit, лень

```
fac = scanl (*) 1 [1..]  
fac n = facs !! n
```



1977 Джон Бекус, премия Тьюринга за участие в создании Алгола и Фортрана, но!
ФВП

```
def fac = eq o [id , 0] -> 1 ; x o [id , fac o - o [id , 1]]
```

1977, Эдинбург, университет

ML, metalanguage – полиморфная система типов
Робин Милнер

Вопрос: типизация?

Роджер Хиндли, 69



Ещё и императивный!
Standard ML, CAML и чисто функциональный Lazy ML

```
fun fac(n) = if n = 0 then 1 else n * fac(n-1);
```


Норе, алгебраические типы данных,
сопоставление с образцом

```
dec fact : num -> num;  
--- fact 0 <= 1;  
--- fact n <= n*fact(n-1);
```

```
data List a = Nil  
            | Cons a (List a)
```

```
data Bool = False | True
```

1985, Дэвид Тёрнер – Miranda, всё вместе!

функции высшего порядка,
ленивые вычисления,
алгебраические типы данных,
параметрический полиморфизм,
сопоставление с образцом



А может, и не он

Miranda TM

$\text{fac } n = 1, \text{ if } n = 0$
 $= n * \text{fac } (n - 1), \text{ otherwise}$

сопоставления

$\text{fac } 0 = 1$
 $\text{fac } n = n * \text{fac } (n-1)$

ZF

$\text{fac } n = \text{product } [1..n]$

1987г. FPCA '87 (Functional Programming Languages and Computer Architecture), Оперон

1991г. Haskell 1.0

1999г. The Haskell 98 Report

Ongebru'87



