

从芯片定制到语言定制
程序设计语言的系统化定制及其支撑环境
(Ongoing Work)

胡振江

北京大学计算机科学技术系

2020年10月24日

如何使程序运行速度提高1倍？

```
-- Type annotation (optional)
fib :: Int -> Integer

-- With self-referencing data
fib n = fibs !! n
      where fibs = 0 : scanl (+) 1 fibs
              -- 0,1,1,2,3,5,...

-- Same, coded directly
fib n = fibs !! n
      where fibs = 0 : 1 : next fibs
              next (a : ts@(b:_)) = (a+b) : next ts

-- Similar idea, using zipWith
fib n = fibs !! n
      where fibs = 0 : 1 : zipWith (+) fibs (tail fibs)

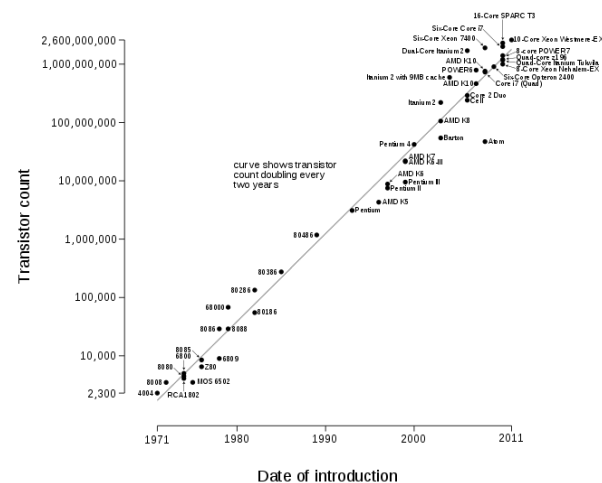
-- Using a generator function
fib n = fibs (0,1) !! n
      where fibs (a,b) = a : fibs (b,a+b)
```



通用计算机

摩尔定律：等18个月

Microprocessor transistor counts 1971-2011 & Moore's law



如何使程序运行速度提高1倍？

受益例：函数式程序设计走出了象牙塔

30年（1990-2020）速度提升百万倍！



おまけ：函数式语言是一个很好的硬件描述语言

FP



VLSI

No.	论文信息
1994	
7	孙永强 胡振江 袁昕 . FP——VLSI自动综合系统. 软件学报, 1994, 5 (01): 26-32
1992	
6	夏心杰 孙永强 胡振江 . 脉动阵列的自动综合?FP方法. 软件学报, 1992, 3 (03): 1-9
5	夏心杰 孙永强 胡振江 . 脉动阵列的自动综合—FP方法. , 1992, 3 (03): 1-9
4	胡振江 孙永强 . 并行算法的FP描述及其脉动化的判定. 软件学报, 1992, 3 (03): 9-17
1990	
3	胡振江 孙永强 夏心杰 . <u>函数式语言对数字电路的描述,综合及模拟</u> . 计算机学报, 1990, 13 (06): 441-449
2	胡振江 孙永强 . 通过程序变换对数字系统进行优化. 软件学报, 1990, 1 (02): 1-12
1	白英彩 胡振江 杨骏 . 用Transputer实现FP的研究. 小型微型计算机系统, 1990, 11 (10): 1-10



(cdblp中的数据)



MIT论文：通用计算机时代即将终结



Neil Thompson

@ProfNeilT

フォローする

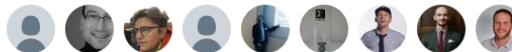


My new working paper with Svenja Spanuth shows how the economic engine that made computers so **general-purpose is failing** and may reverse itself, **fragmenting computing**. This will divide applications into those still in the fast lane and those left behind.

ssrn.com/abstract=32877...

10:57 - 2018年12月4日

4件のリツイート 9件のいいね

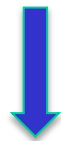


4



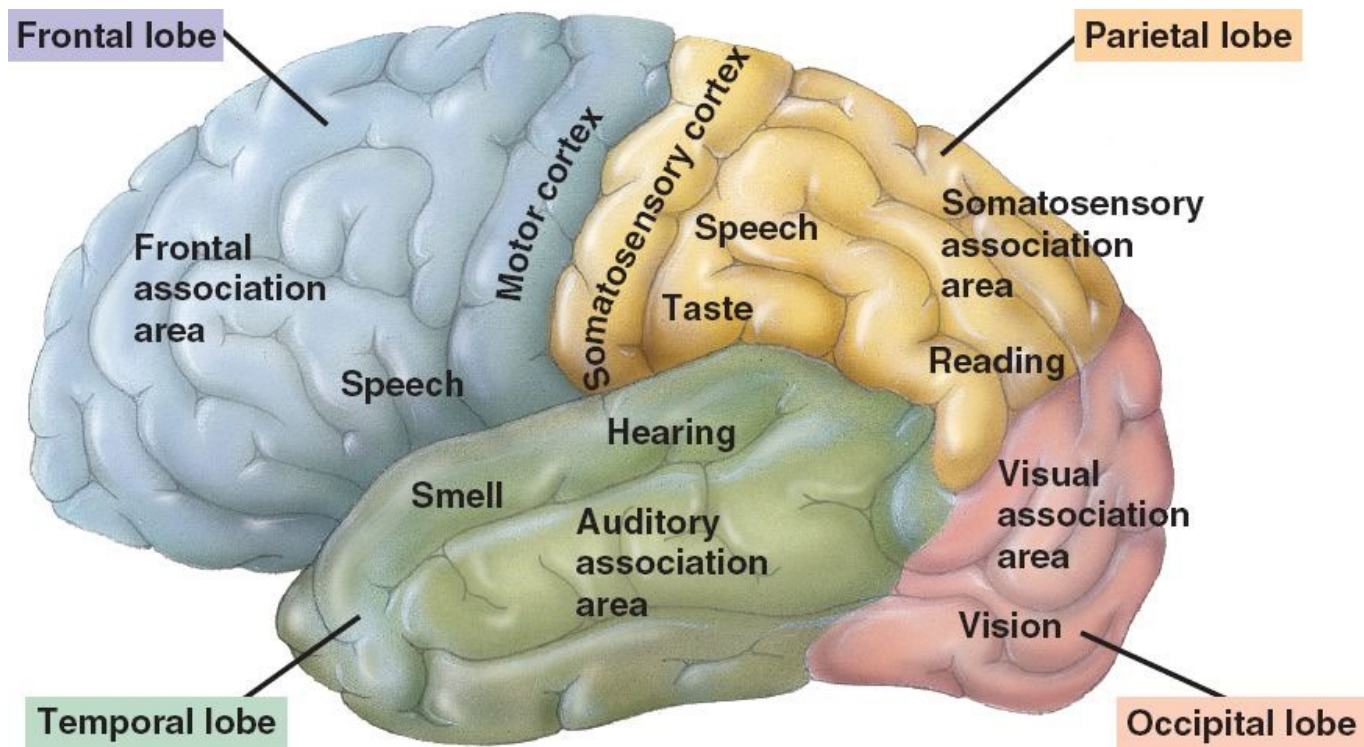
9

通用芯片



专用芯片

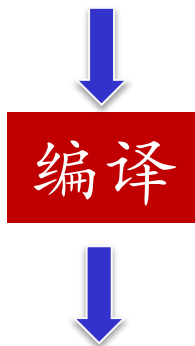
人脑：Fragmented Computation



[From web]

计算的变化 必将带来 程序设计语言的变化

高级通用程序设计语言



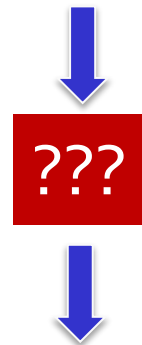
编译

低级通用程序设计语言



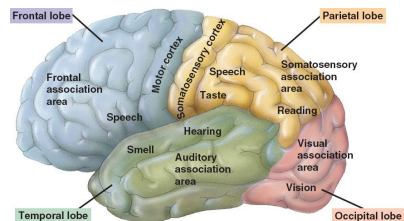
通用计算

高级??程序设计语言



???

低级??程序设计语言



Fragmented 计算

新型编译器

应用领域

P^1

DSL¹

医疗领域

P^2

DSL²

营养领域

P^3

DSL³

药学领域

P^4

DSL⁴

大数据处理领域

...



计算领域

SDX

P_1

DSL₁

特制AI芯片

P_2

DSL₂

数据中心

P_3

DSL₃

传感器

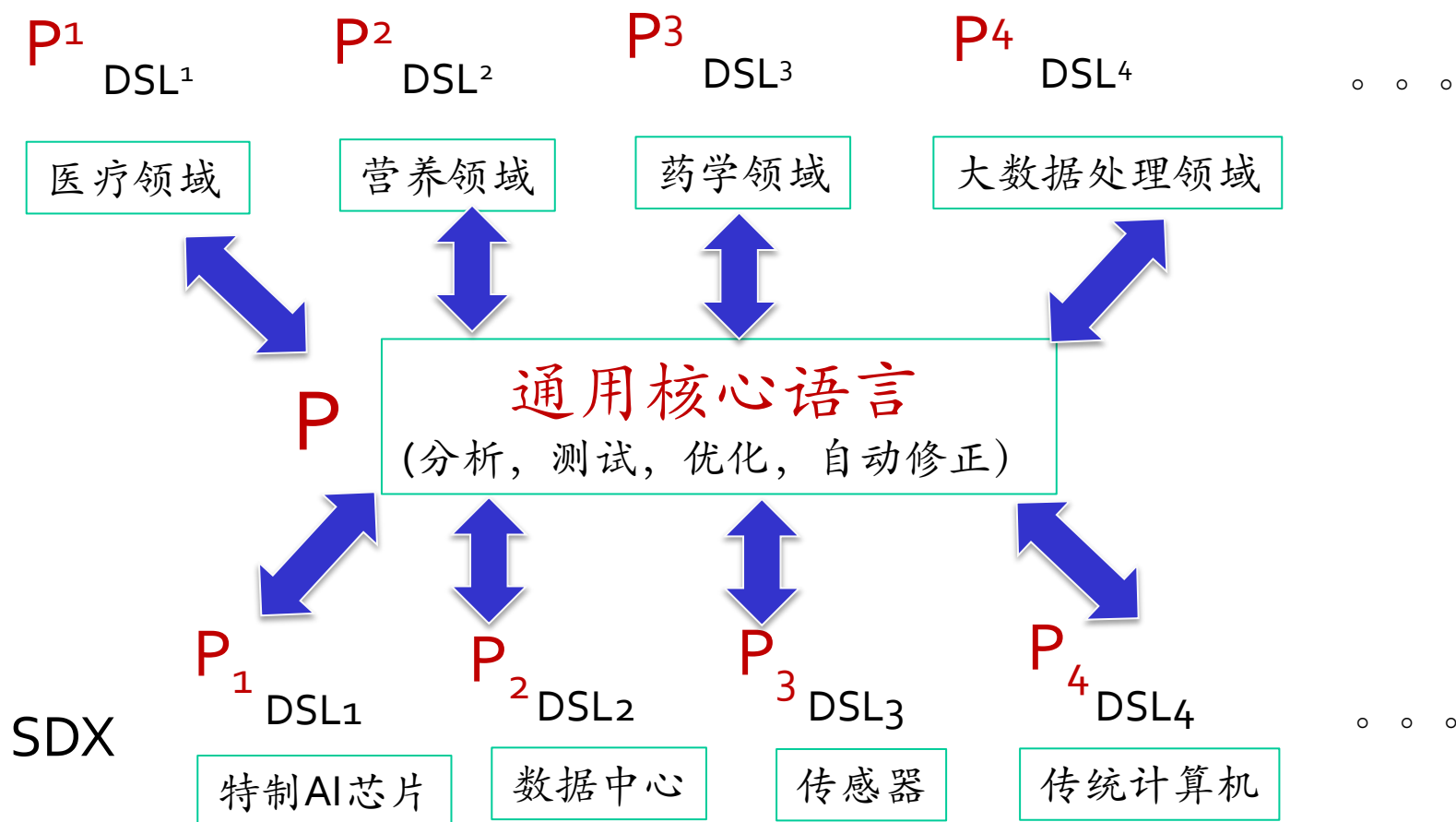
P_4

DSL₄

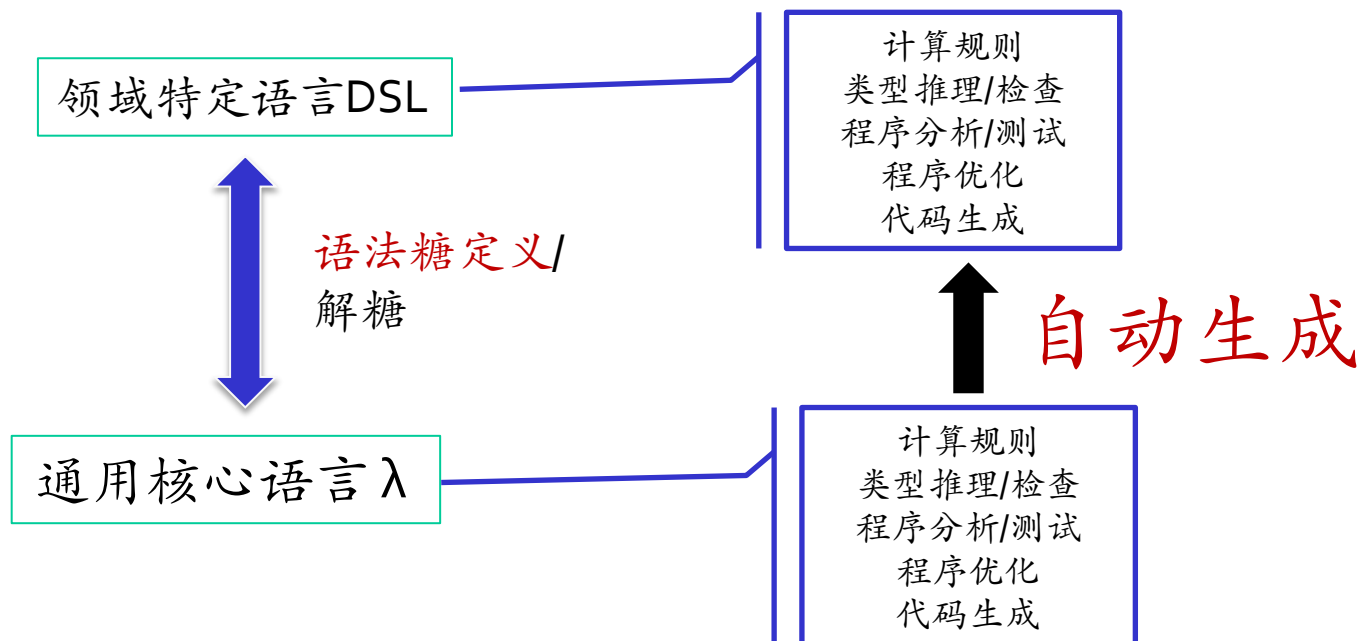
传统计算机

...

解决方法：通用核心语言+语言定制+双向变换

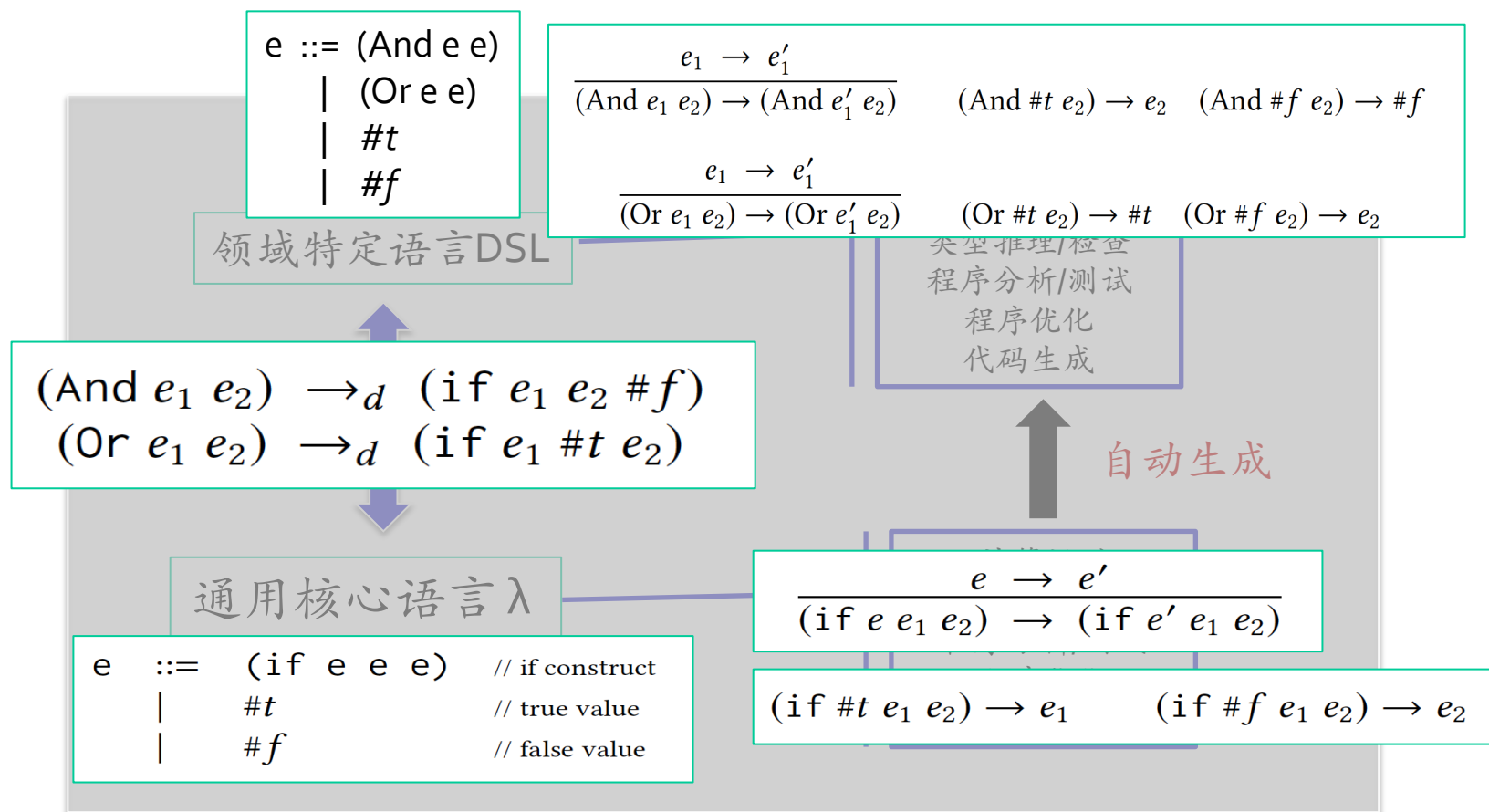


程序设计语言的系统化定制



这个“自动生成”和传统的EDSL有着本质的区别！

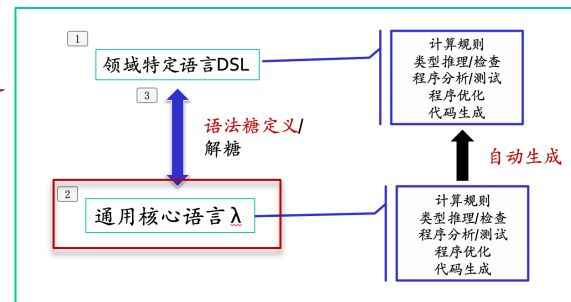
程序设计语言的系统化定制



程序设计语言的系统化定制的课题

- 通用核心语言

- 易于表达各种计算
- 已有成熟的解释，分析，优化，代码生成的理论基础

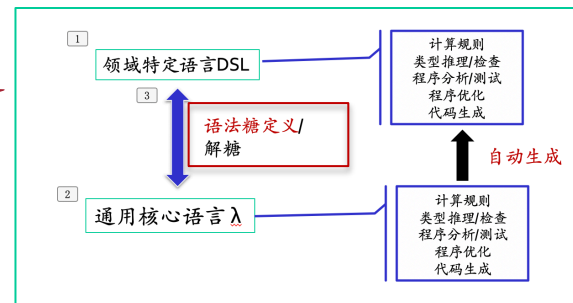


```
CoreExp ::= (CoreExp CoreExp ...) // apply
          | (lambda (x ...) CoreExp) // call-by-value
          | (lambdaN (x ...) CoreExp) // call-by-need
          | (if CoreExp CoreExp CoreExp) // condition
          | (let x CoreExp CoreExp) // binding
          | (listop CoreExp) // first, rest, empty?
          | (cons CoreExp CoreExp) // data structure of list
          | (arithop CoreExp CoreExp) // +, -, *, /, >, <, =
          | x // variable
          | c // boolean, number and list
```

Pure Functional Calculus

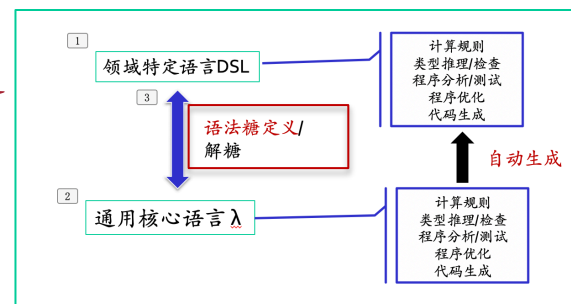
程序设计语言的系统化定制的课题

- 基于语法糖定义的语言定制
 - 能够描述各种各样的“定制”
 - 卫生宏，高阶宏，递归宏，副作用


$$\begin{aligned} S &\rightarrow_d (\text{lambdaN } (x1 \ x2 \ x3) (x1 \ x2 \ (x1 \ x3))) \\ K &\rightarrow_d (\text{lambdaN } (x1 \ x2) x1) \\ I &\rightarrow_d (\text{lambdaN } (x) x) \end{aligned}$$
$$\begin{aligned} (\text{Odd } e) &\rightarrow_d (\text{if } (> \ e \ 0) \ (\text{Even } (- \ e \ 1)) \ \#f) \\ (\text{Even } e) &\rightarrow_d (\text{if } (> \ e \ 0) \ (\text{Odd } (- \ e \ 1)) \ \#t) \end{aligned}$$
$$\begin{aligned} (\text{Map } e_1 \ e_2) &\rightarrow_d \\ &(\text{let } f \ e_1 \ (\text{let } x \ e_2 \ (\text{if } (\text{empty? } x) \ (\text{list}) \ (\text{cons } (f \ (\text{first } x)) \ (\text{Map } f \ (\text{rest } x)))))) \end{aligned}$$

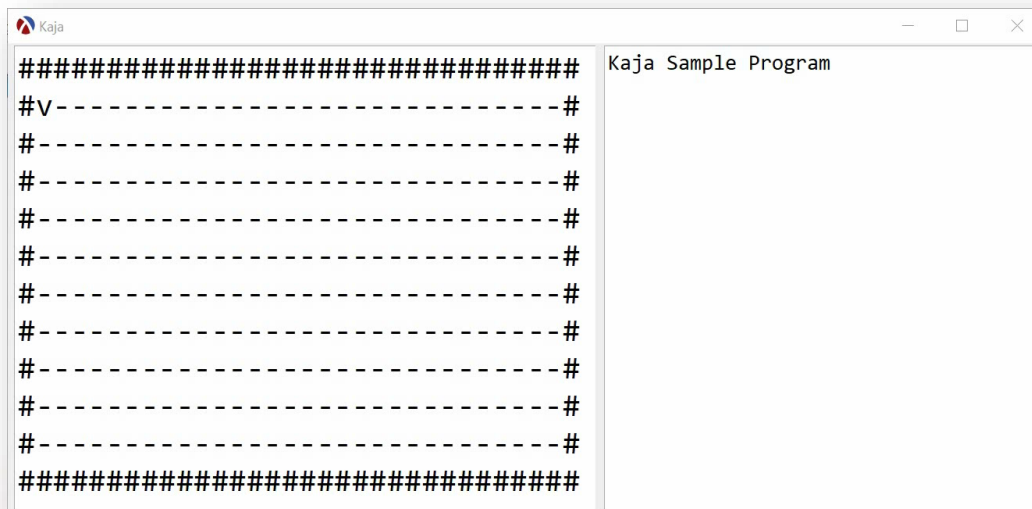
程序设计语言的系统化定制的课题

- 基于语法糖定义的语言定制
 - 能够描述各种各样的“定制”
 - 卫生宏，高阶宏，递归宏，副作用



```
[(while (imperative-not (isMark))
  [(while (isWall)
    [(turnLeft)])
  (while (imperative-not (isWall))
    [(drop)
     (step)]))]])]
```

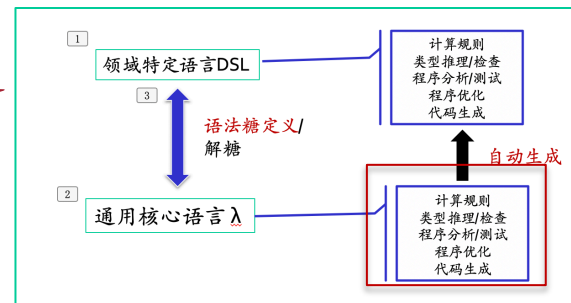
desugar



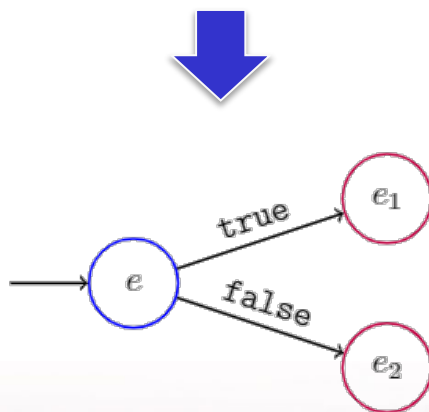
```
((λ (s)
  (if (first
      ((imperative-not
        (isMark))
      s))
    (run
      (do (while
            (isWall)
            ((turnLeft)))
          (while
            (imperative-not
              (isWall))
            ((drop) (step)))
          (while
            (imperative-not
              (isMark))
            ((while
              (isWall)
              ((turnLeft)))
              (while
                (imperative-not
                  (isWall))
                ((drop)
                 (step))))))
      s) (list (list) s))) 0)
```

程序设计语言的系统化定制的课题

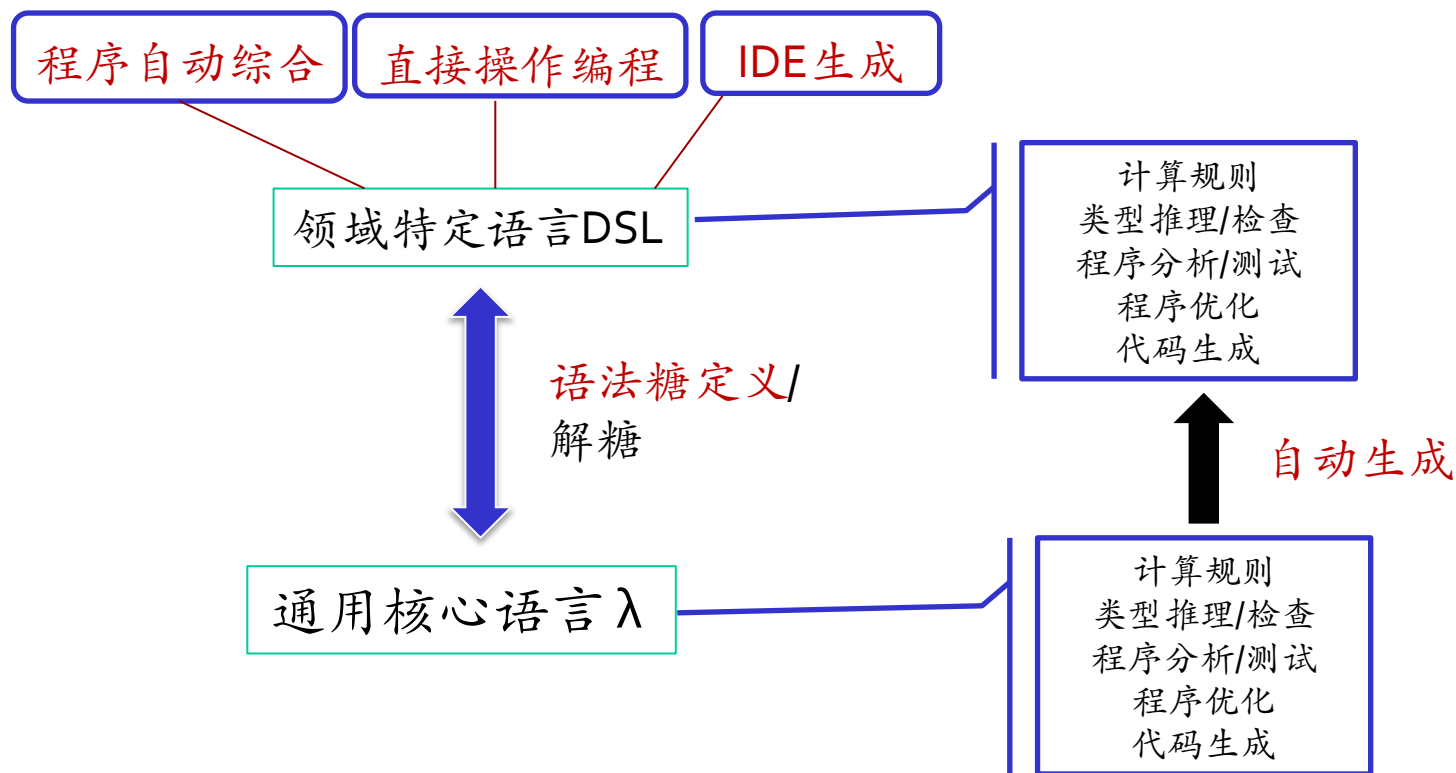
- 计算，分析，优化等规则
 - 必须结构化，以便于自动特殊化
 - 具有统一而有力的描述语言：使用图自动机



$$\frac{e \rightarrow e'}{\text{if } e \text{ then } e_1 \text{ else } e_2 \rightarrow \text{if } e' \text{ then } e_1 \text{ else } e_2}$$
$$\text{if true then } e_1 \text{ else } e_2 \rightarrow e_1$$
$$\text{if false then } e_1 \text{ else } e_2 \rightarrow e_2$$



未来愿景：DSL Programming Environment Builder



欢迎优秀的年轻教师，博士后，博士生加入北大程序设计语言实验室！
欢迎参加CCNC的程序设计语言研究论坛（16:00-18:00）！

谢谢聆听！

