



ELTE

FACULTY OF
INFORMATICS

编程

第3讲

Zsuzsa Pluhár

pluharzs@inf.elte.hu

算法的模式 (PoA) --本质

其目的。

一个成熟的模板作为基础，我们以后可以在此基础上建立我们的解决方案。(开发会更快、更安全)

其结构。

1. 抽象任务规范[+程序规范]
2. 抽象算法

上一条评论：输入至少是一个序列（后文：intervall）。

算法的模式 (PoA) --本质

我们如何使用它们。

1. 为给定的任务创建一个规范
2. 根据规格猜测PoA
3. 将给定任务的参数映射到相应的抽象任务的参数中。
4. 根据PoA的算法，通过改变第3点的参数，"生成"特定任务的算法。
5. 创建 a 更加 的效率 算法 使用 程序转

算法的模式 (PoA) --本质 换



算法的模式

什么是PoA？它是一个典型编程任务的一般解决方案。

- 序列单值
- 序列 序列
- 序列 序列
- 序列 序列

1.序列计算

任务（例子）。

1. 我们知道一个人的月收入 and 支出。让我们来计算一下，到年底他的资产会有**多大的**变化!
2. 我们知道一个汽车赛车手的单圈时间。让我们来**确定**他的**平均**单圈时间!
3. 让我们来**计算**一下**N的阶乘**!
4. 我们知道**N**个单词。请给出我们通过**连接**它们得到的句子。



1.序列计算

分组。

- 数字的总和。"资产", "圈速"
- 数字的乘积。"阶乘"
- 词的加入。"单词"

什么是常见的？

我们有一连串的 "东西", 我们必须从它们的元素中计算出一个单一的 "东西"。

例如。- 收入/时间；- 阶乘；以及 - 词语

1.序列计算

规格

输入: $X[1 \dots]S^*$

输出: scS 前提

条件: $\bar{}$

后置条件: $sc = F(X[1] \dots X[\text{length}(X)])$ $F:$

$S^* \square$

- N 个元素的总和。
- N 个元素的乘积。

S : 一个任意的集合

。

这些项目的索引从1到

$\text{Length}(X)$

1. 序列计算

& - N个文本的串联...



1. 序列计算

$S: \mathbb{N}, \mathbb{Z} \text{ 或 } \mathbb{R}$

规范（求和） 输入。

Input:

$X[1 \dots]S^*$ 输出：

Output

Precondition

Postcondition

sumS 前提条件：

– $length(X) \geq 0$

后置条件。

$sum = \sum_{i=1}^{length(X)} X[i]$

递归定义（数学）。

$$\sum_{i=1}^N X_i := \begin{cases} 0 & , N = 0 \\ \left(\sum_{i=1}^{N-1} X_i \right) + X_N & , N > 0 \end{cases}$$



1.序列计算

一般问题。

F : 有N个参数的操作, 其中N是变量 variable

解决方案。

分解为2个参数的操作 (例如到+) 和一个 中性值 (在+的情况下是0)

。

$F(X_1 \dots X_N) = f(F(X_1 \dots X_{N-1}), X_N)$, 如果 $N > 0$

$F(-) = F_0$ 否则

1. 序列计算

规格（一般） 输入。

$X[1..]S^*$ 输出： scS

前提条件：-

后置条件： $sc = F(X[1]..X[\text{length}(X)])$

定义。 $F:S^*S$

$$F(X_{1..N}) := \begin{cases} F_0 & , N = 0 \\ f(F(X_{1..N-1}), X_N) & , N > 0 \end{cases}$$

$$f: S \times S \rightarrow S, \quad F_0 \square$$

1. 序列计算

规格(进一步概括)

输入。 $X[1 \dots] S_1^*$ 输

出： $s \in S_2$ 前提条件：

\neg 。

后置条件： $s \in F(X[1] \dots X[\text{length}(X)])$

定义。 $F: S_1^* \rightarrow S_2$

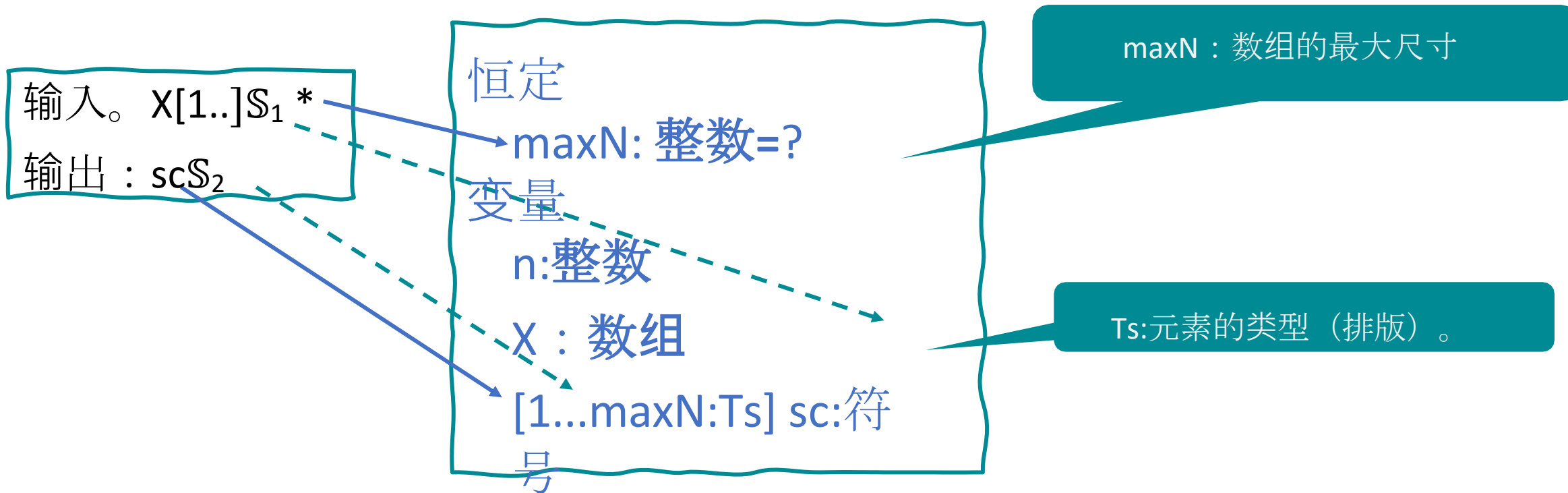
$$F(X_{1..N}) := \begin{cases} F_0 & , N = 0 \\ f(F(X_{1..N-1}), X_N) & , N > 0 \end{cases}$$

$$f: S_2 \times S_1 \rightarrow S_2, \quad F_0 \in S_2$$



1.序列计算

算法--声明程序变量



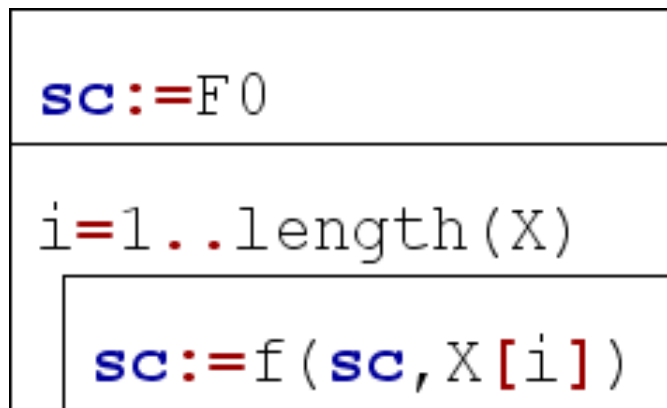
1.序列计算

以**静态**方式声明序列的数组

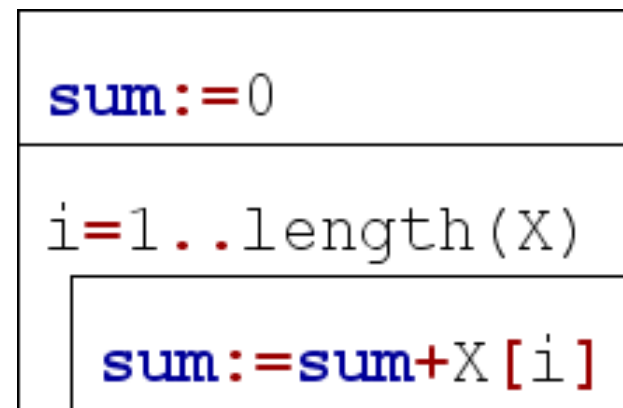


1. 序列计算

算法（一般）



算法（求和，）。



1.顺序计算--实例

我们知道一个汽车赛车手的单圈时间。让我们来确定他的平均单圈时间!

规格

输入: $n \in \mathbb{N}$, $X[1 \dots n] \in \mathbb{N}^n$

输出: $AV \in \mathbb{R}$

前提条件: -

后置条件: -

$$av = \left(\begin{array}{c} \text{length}(X) \\ \square \\ \circ \\ i=1 \end{array} X[i] \right) : \text{length}(n) \circ$$

n : 圈速的数量

$n = \text{length}(X)$

sum - 帮助变量

$\text{sum} := 0$

$i = 1 \dots \text{length}(X)$

$\text{sum} := \text{sum} + X[i]$

$av := \text{sum} / \text{length}(X)$



1.顺序计算--实例

我们知道一个人的月收入 and 支出。让我们来计算一下，到年底他的资产会有多大的变化！

规格

输入: $n \in \mathbb{N}$, $Inc[1..n] \in \mathbb{N}^n$, $Outc[1..n] \in \mathbb{N}^n$

输出: $sum \in \mathbb{Z}$

前提条件: -

后提条件。

$$\left(\begin{array}{l} \text{length}(X) \\ \square \\ i=1 \end{array} \circ (Inc[i] - Outc[i]) \right)$$

1.顺序计算--实例

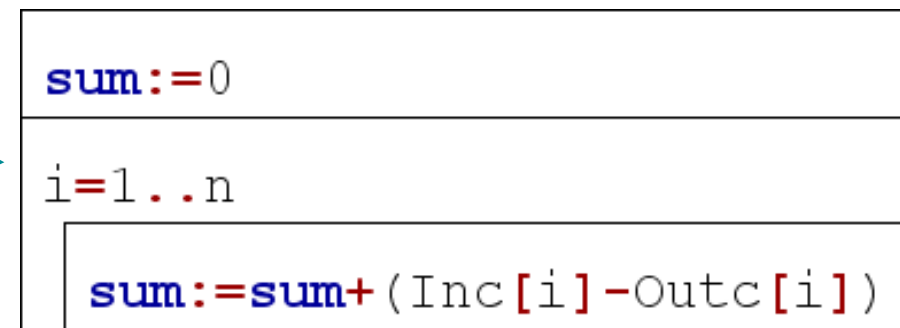
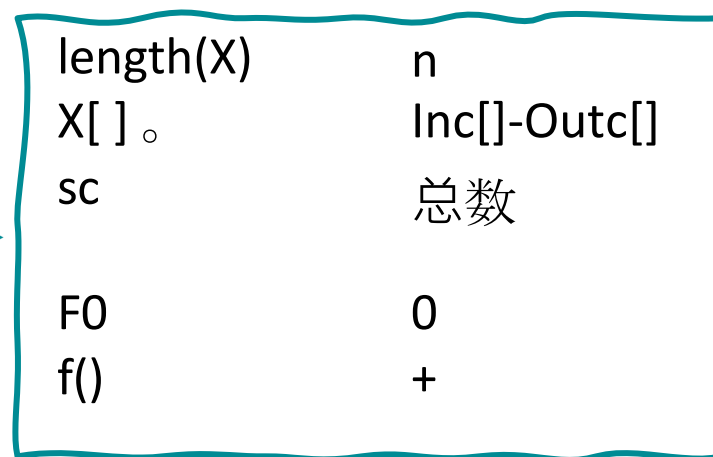
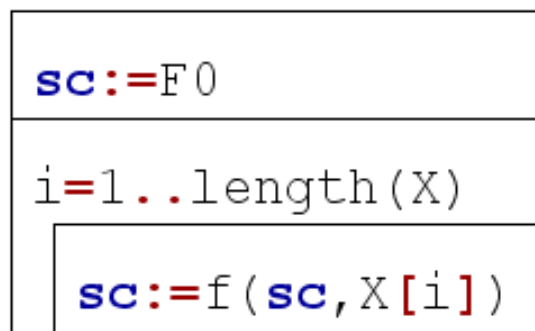
sum=



1.顺序计算--实例

我们知道一个人的月收入 and 支出。让我们来计算一下，到年底他的资产会有多大的变化!

算法



需要学习的教训

1. 一个特定任务的**前提条件**可能比算法的使用模式**更严格**。
2. 某项任务的**后置条件**可能比所使用的算法模式的**后置条件更弱**（我们将满足这一点）。
3. 索引：从**1**到**N**，从**B**到**E**，使用**length()**。
4. 该函数可以要求第*i*个元素的值（从更多的数组中，从更多的数组元素中，或用一个独立于数组的函数），而不是与数组中的元素一起工作。

2.计数

任务（例子）。

1. 我们知道一个人的每月收入和支出。让我们计算一下他的资产增长的月份数吧!
2. 让我们来计算一个整数的除数!
3. 让我们确定在一个人的名字中可以找到多少个字母 "a"!
4. 根据每年的日常统计，让我们来算算被冻住的天数吧!
5. 我们有N个人的出生日期（月）数据。让我们计算一下，有多少

2.计数

人的生日是在冬季!



2.计数

什么是常见的？

我们有一个 "某物 "的序列，我们必须计算它们中有多少个项目有一个给定的属性。



2.计数

S : 一个任意的集

规格

输入: $x[1 \dots]S^*$, $a:S \rightarrow \mathbb{N}$

A 是一个任意的属性函数

输出: $cnt \in \mathbb{N}$

前提条件: -
后置条件。

$length(X)$

$A(X[i])$

$cnt = \sum_{i=1}^{length(X)} 1$



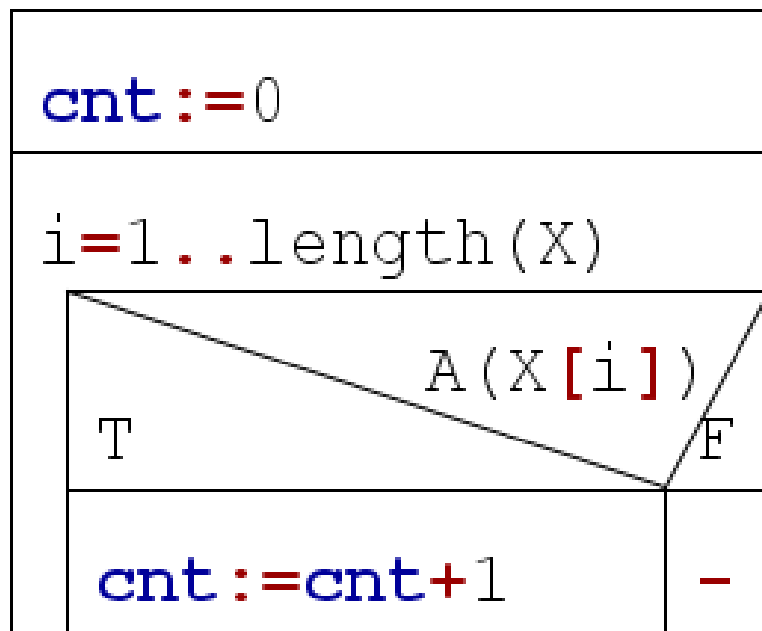
2. 计数

注: A 是一个可以被赋予布尔函数的属性。人们可以知道 x (和 S) 的每个元素是否具有这个属性。



2.计数

算法



2.计数 - 例子

我们有N个人的出生日期（月）数据。让我们计算一下，有多少人的生日是在冬季！
规格

输入: $n \in \mathbb{N}$, $\text{Mon}[1..n] \subseteq \mathbb{N}$.

冬季: $\mathbb{N} \rightarrow \mathbb{B}$; $\text{冬季}(x) = (x < 3 \vee x = 12)$

输出: $\text{cnt} \in \mathbb{N}$

前提条件: $\forall i (1 \leq i \leq n) : 1 \leq \text{Mon}[i] \leq 12$

后置条件:

$$\text{cnt} = \sum_{i=1}^n \text{Winter?}(\text{Mon}[i])$$

注: 前提条件可以更严格
比所使用的模式要好。

2.计数 - 例子

我们有N个人的出生日期（月）数据。让我们计算一下，有多少人的生日是在冬季！
算法

<code>cnt:=0</code>	
<code>i=1..length(X)</code>	
<code>T</code>	<code>A(X[i])</code>
<code>cnt:=cnt+1</code>	<code>-</code>

`length(X)`

`X[]`

`cnt`

`A(X[i])`

`n`

`Mon[]`

`o`

`cnt`

`Mon[i]<3或Mon[i]=12`

<code>cnt:=0</code>	
<code>i=1..n</code>	
<code>T</code>	<code>Mon[i]<3 or Mon[i]=12</code>
<code>cnt:=cnt+1</code>	<code>-</code>

如果不满足前提条件，会发生什么？



ELTE

FACULTY OF
INFORMATICS

程序设计 - Zsuzsa Pluhár

3.最大限度的选择

任务（例子）。

1. 我们知道一个人的每月收入和支出。让我们确定他的资产增长**最多的**月份!
2. 让我们来挑选按字母顺序排在**最后**的人的名字吧!
3. 让我们找出喜欢**最多**种类食物的人吧!
4. 根据每年的日常统计，让我们来定义一年中**最温暖**的一天！"。
5. 我们有**N**个人的出生日期，让我们找出谁的生日是在这一年的第

3.最大限度的选择

一时间!



3.最大限度的选择

什么是常见的？

我们必须从一连串的 "东西 "中挑选/找到最大（或最小）的东西。

重要的是。

- 这些东西有一个属性，我们可以根据这个属性对它们进行排序（排序关系）。
- 如果我们至少有1个元素，那么我们就知道存在一个最大（或最



3.最大限度的选择

小)。



3.最大限度的选择

规格

输入。 $X[1 \dots]S^*$

输出： $\text{maxIndN}, \text{maxValS}$

前提条件： $\text{length}(X) > 0$

后置条件。 $1 \leq \text{maxInd} \leq \text{length}(X)$ 且

$\forall i (1 \leq i \leq \text{length}(X)) : X[\text{maxInd}] \geq X[i]$ 和 $\text{maxVal} = X[\text{maxInd}]$

很短 $(\text{maxInd}, \text{MaxVal}) = \underset{i=1}{\overset{\text{length}(X)}{\text{MAX}}}(X[i])$ 。



3.最大限度的选择

算法。

注意：如果有更多的元素等于最大，那么这个算法将找到第一个元素。问题。

- 我们怎样才能找到最后一个最伟大的人呢
- 我们怎样才能找到（第一个）最小的元素

修改



该算法

<code>maxInd:=1</code>	
<code>maxVal:=X[1]</code>	
<code>i=2..length(X)</code>	
<div><div><code>X[i]>maxVal</code></div><div>T</div><div>F</div></div>	
<code>maxInd:=i</code>	-
<code>maxVal:=X[i]</code>	



ELTE

FACULTY OF
INFORMATICS

程序设计 - Zsuzsa Pluhár

3.最大限度的选择

评论。

- 我们假设存在以下排序操作符。

`SometingSomething` → `Boolean`。

- 一个元素的序列号是更一般的信息，因此我们提供这个信息而不是元素本身。



3.最大选择 (指数)

规格

输入。 $X[1 \dots]S^*$

输出： $\text{maxInd}N$

前提条件： $\text{length}(X) > 0$

后置条件。 $1 \leq \text{maxInd} \leq \text{length}(X)$ 且
 $\forall i (1 \leq i \leq \text{length}(X)) : X[\text{maxInd}] \geq X[i]$

很短
$$(\text{maxInd}) = \underset{i=1}{\text{length}(X)} \text{MAXIND}(i)。$$

3.最大选择（指数）

算法

<code>maxInd:=1</code>	
<code>i=2..length(X)</code>	
T	$X[i] > X[\text{maxInd}]$
	F
<code>maxInd:=i</code>	-

3.最大选择 (值)

规格

输入。 $X[1 \dots]S^*$

输出： $\text{maxVal}S$ 前提条件：

$\text{length}(X) > 0$ 后提条件：

$\text{maxVal}X$ 和

$i (1 \leq i \leq \text{length}(X)) : \text{maxVal} = X[i]$ 和

$i (1 \leq i \leq N) : \text{maxVal}X[i]$ 。

很短。 $(\text{maxVal}) = \mathbf{MAXVL} (X[i])$
 $i = 1$ $\text{length}(X)$ 。

3.最大选择（值）

算法

<code>maxVal := X[1]</code>	
<code>i = 2 .. length(X)</code>	
T	<code>X[i] > maxVal</code>
	F
<code>maxVal := X[i]</code>	-

3.最大限度的选择--例子

根据每年的日常统计，让我们来定义一年中最温暖的一天！"。

规格

输入。温度 $[1..365]N$ 。³⁶⁵

输出： $w[i]N$

前提条件。 $365 > 0$

后置条件。 $1 \leq w[i] \leq 365$ 和 i ($1 \leq i \leq 365$) : $Temp[w[i]] \geq Temp[i]$

$$(w[i] = \underset{i = 1}{\overset{365}{\text{MAXIND}}} (Temp[i])) .$$

3.最大限度的选择--例子

根据每年的日常统计，让我们来定义一年中最温暖的一天！"

算法

<code>maxInd:=1</code>	
<code>i=2..length(X)</code>	
<code>T</code>	<code>X[i]>X[maxInd]</code>
<code>maxInd:=i</code>	<code>F</code>
	-

<code>wdi:=1</code>	
<code>i=2..365</code>	
<code>T</code>	<code>Temp[i]>Temp[wdi]</code>
<code>wdi:=i</code>	<code>F</code>
	-

长度(X)

X[]

最大指数

A(X[i])

n

温度[]。

wdi

温度[i]>温度[wdi]

4.搜索

任务（例子）。

1. 我们知道一个人的每月收入和支出。他的资产在年底的时候会增长。让我们给出一个月，当他的资产没有增长时!
2. 让我们定义一个非1和非N的整数N的除数!
3. 让我们在一个人的名字中搜索字母 "a"!
4. 让我们搜索一下该学生没有通过的课程吧!
5. 让我们在一个序列中找到一个比前一个元素大的元素!



4.搜索

什么是常见的？

我们有一个 "某某 "序列，我们必须搜索一个具有给定属性的元素，而我们不知道这样的元素在序列中是否存在。

4.搜索

规格

输入。 $x[1 \dots]S^*$, $a:SL$

输出：存在 L 。 $indN$, $valS$

"是否有" => 决定

前提条件：-

后置条件: $exists = (i (1 \leq i \leq length(X)) : A(X[i]))$ and $exists$
 $1 \leq ind \leq length(X)$ and $A(X[ind])$ and $val = X[ind]$

选择

很短。

$exists, ind, val$

$length ($
 $X) 。$
 $SEARCH (X [i$
 $])$
 $i = 1$

4.搜索



()

[]



ELTE

FACULTY OF
INFORMATICS

4.搜索

算法₁

规格

输入。 $x[1\dots]S^*$, $a:SL$

输出: **exists** L , **ind** N , **val** S

前提条件: -

后置条件:

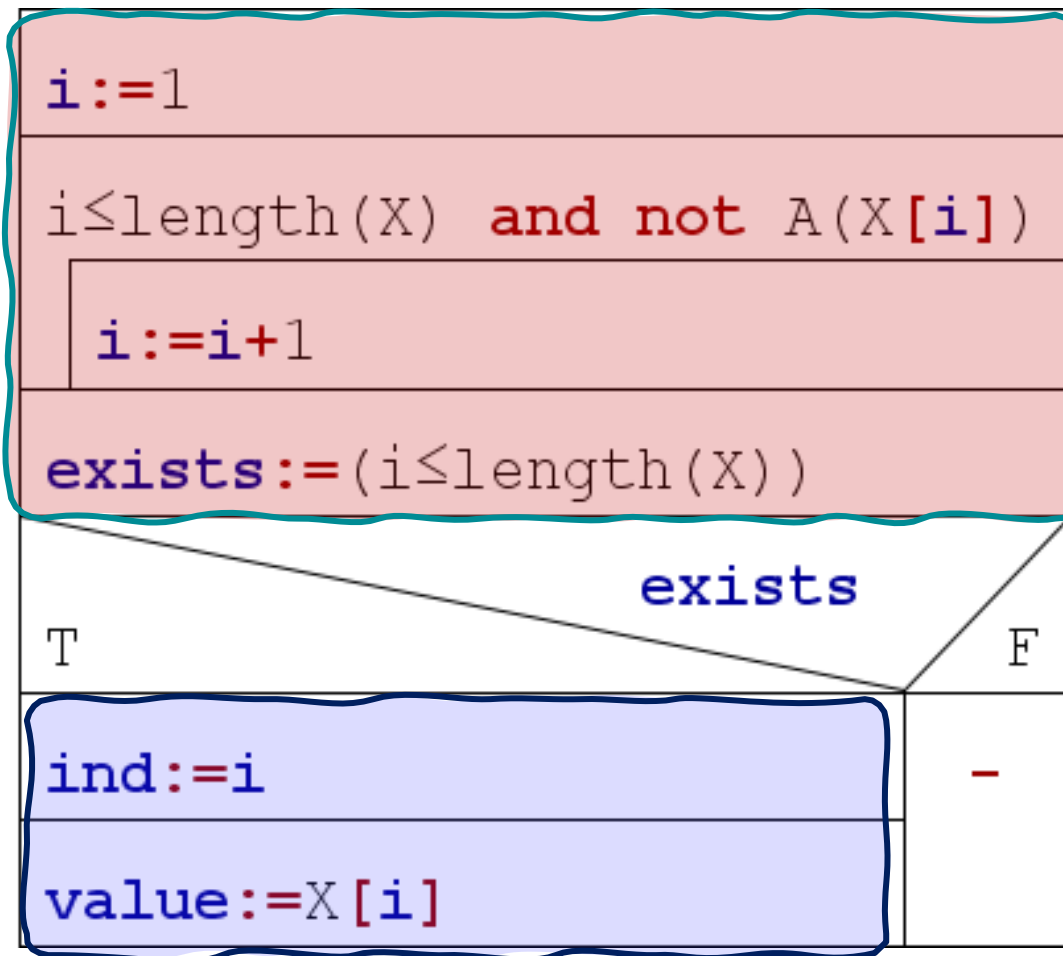
$exists = (i (1 \leq i \leq length(X)) : A(X[i]))$ and
 $exists \ 1 \leq ind \leq length(X)$ and $A(X[ind])$ and
 $val = X[ind]$

修改



该算法

找到最后一项



4. 搜索

算法₂

规格

输入。 $x[1 \dots]S^*$, $a:SL$

输出 : **exists** L , **ind** N , **val** S

前提条件 : -

后置条件:

$exists = (i (1 \leq i \leq \text{length}(X)) : A(X[i])) \text{ and}$
 $exists \ 1 \leq \text{ind} \leq \text{length}(X) \text{ and } A(X[\text{ind}]) \text{ and}$
 $\text{val} = X[\text{ind}]$

i := 0

exists := false

i < length(X) and not exists

i := **i** + 1

exists := A(X[**i**])

exists

T

F

ind := **i**

-

为什么 : 起始索引之间的差异 (0/1) 。



ELTE

FACULTY OF
INFORMATICS

程序设计 - Zsuzsa Pluhár

4.搜索 - 例子

让我们搜索一下该学生没有通过的课程吧!

规格

输入： $n \in \mathbb{N}$, $\text{Mark}[1..n] \subseteq \mathbb{N}^n$, $A: \mathbb{N} \rightarrow \mathbb{L}$

输出：失败 \mathbb{L} , $ci \in \mathbb{N}$

前提条件： i ($1 \leq i \leq n$)。马克[i][1...5]

后置条件： $\text{fails} = (i \text{ ($1 \leq i \leq n$) } : \text{Mark}[i] = 1)$ ，并且
未能 $1 \leq ci \leq n$, 且 $\text{Mark}[ci] = 1$

属性功能

很短。 $(\text{fails}, ci) = \text{SEARCH}^n(X[i])$
 $i = 1$
 $\text{Mark}[i] = 1$

4.搜索 - 例子

让我们搜索一下该学生没有通过的课程吧!
算法

$i := 1$	
$i \leq \text{length}(X) \text{ and not } A(X[i])$	
$i := i + 1$	
$\text{exists} := (i \leq \text{length}(X))$	
T	F
$\text{ind} := i$	
$\text{value} := X[i]$	
$-$	



长度(X)	n
存在	未能成功
ind	ci
$A(X[i])$	马克[i]=1



$i := 1$	
$i \leq n \text{ and Mark}[i] \neq 1$	
$i := i + 1$	
$\text{fails} := (i \leq n)$	
T	F
$\text{ci} := i$	
$-$	

5.决定

任务（例子）。

1. 让我们来**决定**一个整数是否是质数！这是很重要的。
2. 让我们来**判断**一个给定的词是否是一个月的名字!
3. 根据一个学生的最终分数，**让我们来**确定他/她是否在这一学期中不及格!
4. 让我们来找出一个给定的单词是否包含元音!
5. 让我们来**决定**一个序列是否是单调递增的!

5. 决定

6. 根据最后的分数，**我们来确定该学生是否优秀**（所有分数都是最好的）。



5. 决定

什么是常见的？

让我们来确定在一连串的 "某物 "中是否有一个具有给定属性的项目

一个 "缩小" (输出) 的搜索版本



5. 决定

规格

输入。 $x[1 \dots]S^*$, $a:SL$

输出：存在 L

前提条件：-

后置条件: $\text{exists} = (\exists i (1 \leq i \leq \text{length}(X)) : A(X[i]))$

$$\text{很短。} \quad (\text{exists}) = \bigvee_{i=1}^{\text{length}(X)} (A(X[i]))$$

5. 决定

算法1

<code>i:=1</code>
<code>i ≤ length(X) and not A(X[i])</code>
<code>i:=i+1</code>
<code>exists := (i ≤ length(X))</code>

算法2

<code>i:=0</code>
<code>exists:=false</code>
<code>i ≤ length(X) and not exists</code>
<code>i:=i+1</code>
<code>exists:=A(X[i])</code>



为什么：起始索引之间的差异（0/1）。

5. 决定 - 所有

任务变体。

.....所有的项目都是属性为A的.....。

规格（只有差异）。

输出。所有L

岗位条件。全部= $(i \ (1 \leq i \leq \text{length}(X)) : A(X[i]))$

短。 $(all) = \bigwedge_{i=1}^{\text{length}(X)} (A(X[i]))$

$All = \text{NOT}((i \ (1 \leq i \leq \text{length}(X)) : \text{NOT } A(X[i])))$

5. 决定 - 所有

算法

<code>i:=1</code>
<code>i ≤ length(X) and not A(X[i])</code>
<code>i:=i+1</code>
<code>exists := (i ≤ length(X))</code>

<code>i:=1</code>
<code>i ≤ length(X) and A(X[i])</code>
<code>i:=i+1</code>
<code>all := (i > length(X))</code>

$\text{All} = \text{NOT} ((i (1 \leq i \leq \text{length}(X)) : \text{NOT } A(X[i])))$

5.决定 - 例子

根据一个学生的最终分数，让我们来确定他/她是否在这一学期中不及格!

规格

输入： $n \in \mathbb{N}$, $\text{Mark}[1..n] \in \mathbb{N}^n$, $A: \mathbb{N} \rightarrow \mathbb{B}$

输出：失败 \mathbb{B}

前提条件： i ($1 \leq i \leq n$)。马克[i][1...5]

后置条件: $\text{fails} = (\exists i (1 \leq i \leq n) : \text{Mark}[i] = 1)$

属性功能

$i := 1$
$i \leq \text{length}(X)$ and not $A(X[i])$
$i := i + 1$
$\text{exists} := (i \leq \text{length}(X))$



长度(X)n存在 失败
 $A(X[i])$ 马克[i]=1



$i := 1$
$i \leq n$ and $\text{Mark}[i] \neq 1$
$i := i + 1$
$\text{fails} := (i \leq n)$

6.挑选

任务（例子）。

1. 我们知道一个人的月收入 and 支出。他的资产在年底前会增长。让我们**给出**一个月份，**当他的资产增长的时候**!
2. 让我们来**定义**非1整数的最小除数!
3. 让我们在一个英语单词中**找到**一个元音!
4. 让我们来**定义**一个月份的序列号，由它的名字来决定!

6.挑选

什么是常见的？

我们有一个 "某某 "的序列，我们必须选择一个具有给定属性的元素，而且我们知道序列中至少有一个这样的元素存在。

这是*搜索*的版本，我们不必为找不到元素的情况做准备。



6.挑选

规格

输入。 $x[1 \dots]S^*$, $a:SL$

输出 : **ind** N , **val** S

前提条件 : $length(X) > 0$ 且 $i (1 \leq i \leq N) : A(X[i])$

后置条件。 $1 \leq ind \leq length(X)$ and $A(X[ind])$ and $val = X[ind]$

很短。 $(ind, val) = \underset{i=1}{length(X)} . SELEC(X[i])$

6.挑选

[]

(
 X

i

)



6.挑选

算法

```
i := 1
```

```
not A (X[i])
```

```
  i := i + 1
```

```
ind := i
```

```
val := X[i]
```

评论。

我们知道的更多：这个解决方案给出了第一个属性为A的元素--所以程序做的比预期的更多。

我们怎样才能找到最后一个呢？

修改



该算法

6.挑选--举例

让我们在一个英语单词中找到一个元音!

规格输入。输出:

vwN 。

前提条件: $\text{length}(\text{Word}) > 0$ 且

$(1 \leq i \leq \text{length}(\text{Word})) : \text{isVowel}(\text{Word}[i])$

后置条件。 $1 \leq vw \leq \text{length}(\text{Word})$ 且 $\text{isVowel}(\text{Word}[vw])$

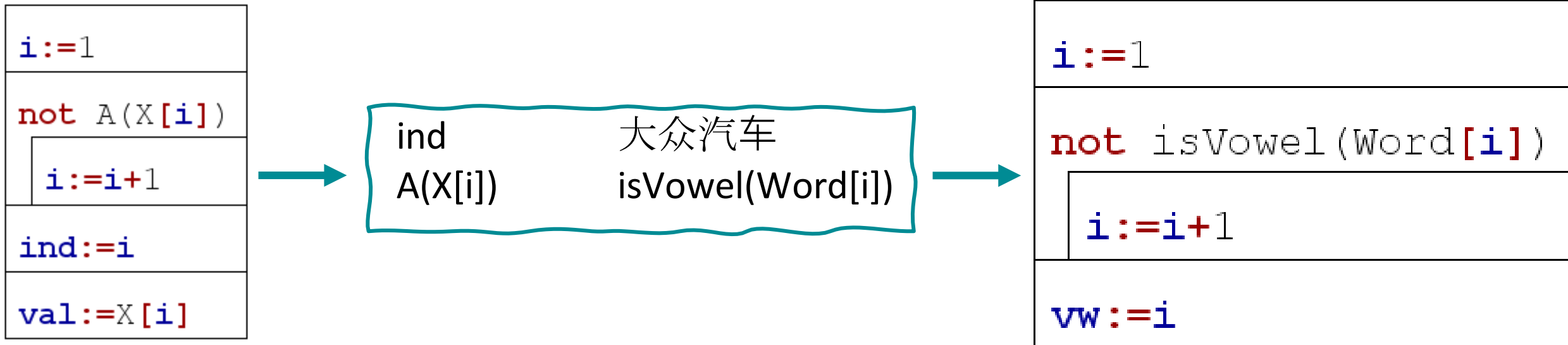
T = 文本集 (字符串)。

属性功能

定义: $\text{isVowel} : \mathbb{C}h \mathbb{L} \rightarrow \mathbb{B}$ (字符布尔) $\text{isVowel}(ch) =$
 $\text{capital}(ch) \in \{'A', \dots, 'U'\}$ 。

6.挑选--举例

让我们在一个英语单词中找到一个元音!
算法



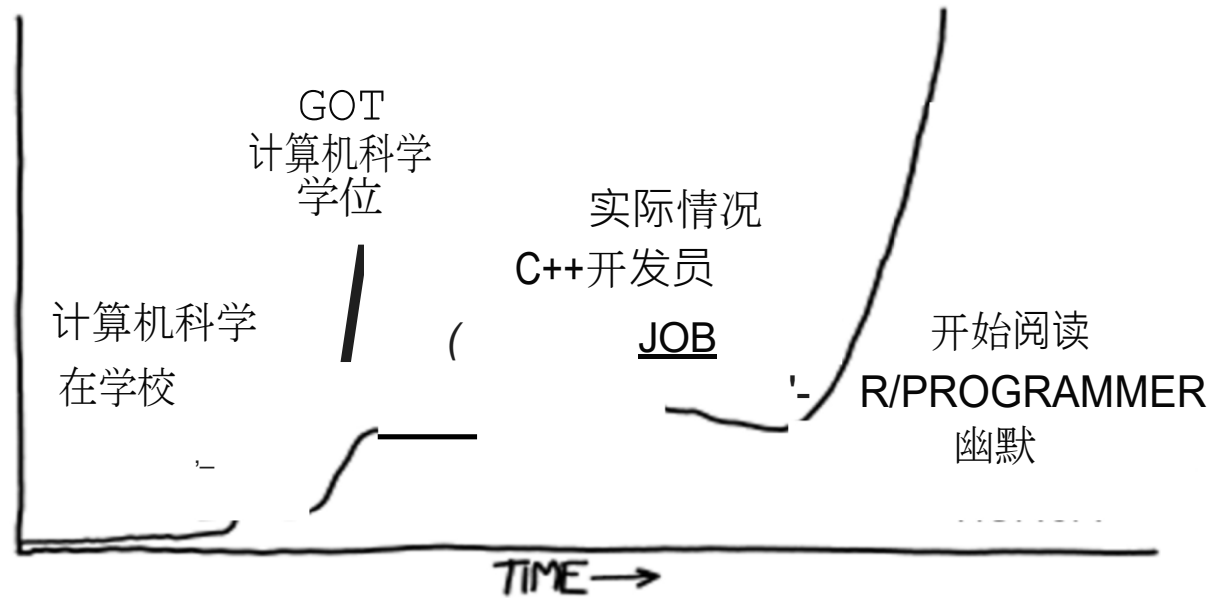
isVowel函数中使用的是哪种模式？



ELTE

FACULTY OF
INFORMATICS

HOW I LEARN / UNDERSTAND PROGRAMMING:



感谢您的关注!