


Eötvös Loránd University
Faculty of Informatics

PROGRAMMING

2019/20 1st semester

Lecture 4

ZS. PLUHÁR, H. TORMA

Content

- More Patterns of Algorithms (PoA)
  - Copy – calculation with function
  - Multiple item selection
  - Partitioning
  - Intersection
  - Union
- PoAs – summary


Pattern of Algoritm

What is PoA? It is the general solution of a typical programming task.

- sequence → single value
- sequence → sequence
- sequence → sequences
- sequences → sequence

7. Copy – function evaluation

Example tasks:

› Let's define the absolute values of **all elements in an array**!

› Let's convert **all letters** of a text to lowercase!

› Let's **calculate** the sum of two vectors!

› Let's calculate sin(x) values for a given x **series**!

› We know the ordinal number of N months, let's give their names!

7. Copy – function evaluation

What is common?

We have N „something”, and we have to assign N other things to these. The type of assigned values can differ from the type of original values, but the count (N) remains the same, as well as the order.

7. Copy – function evaluation

Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}_1^N$

$f: \mathbb{S}_1 \to \mathbb{S}_2$

Output: $Y_{1..N} \in \mathbb{S}_2^N$

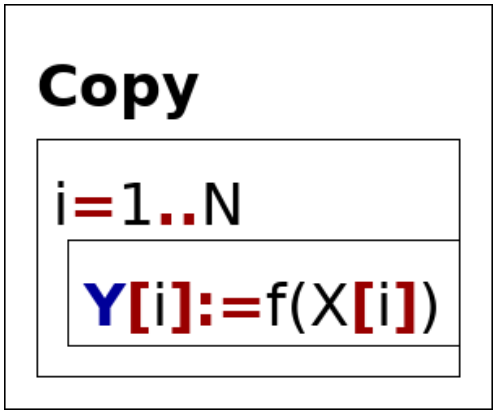
Precondition: –

Postcondition: $\forall i (1 \le i \le N) : Y_i = f(X_i)$

another way: $Y_{1..N} = f(X_{1..N})$

7. Copy – function calculation

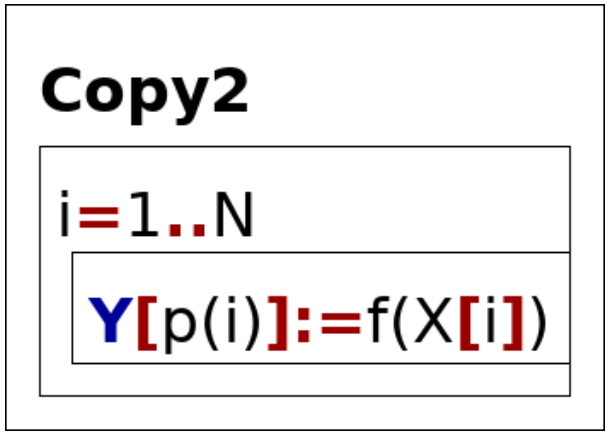
Algorithm:



Note: It's not a must to use the same i index for both arrays. Eg.

Postcondition: $\forall i\,(1 \le i \le N) : Y_{p(i)} = f(X_i)$

p(i) could be `2*i` or `N-i+1` (defined with the needed array size and index-interval



7. Copy – function evaluation

Specification:
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}_1^N$
$f: \mathbb{S}_1 \to \mathbb{S}_2$
Output: $Y_{1..N} \in \mathbb{S}_2^N$
Precondition: –
Postcondition: $\forall i (1 \le i \le N) : Y_i = f(X_i)$

Specification (a frequent special case)$_1$:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$

$g: \mathbb{S} \to \mathbb{S}$, $A: \mathbb{S} \to \mathbb{L}$

f is often a function with a conditional statement

Output: $Y_{1..N} \in \mathbb{S}^N$

Precondition: –

Postcondition: $\forall i (1 \le i \le N) : Y_i = f(X_i)$

definition: $f(x) = \begin{cases} g(x), & \text{if } A(x) \\ x, & \text{otherwise} \end{cases}$

7. Copy – function evaluation

Specification:
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}_1^N$
$f: \mathbb{S}_1 \to \mathbb{S}_2$
Output: $Y_{1..N} \in \mathbb{S}_2^N$
Precondition: –
Postcondition: $\forall i\,(1 \le i \le N): Y_i = f(X_i)$

Specification (a frequent special case)$_1$:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$

$g: \mathbb{S} \to \mathbb{S}$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Y_{1..N} \in \mathbb{S}^N$

Precondition: –

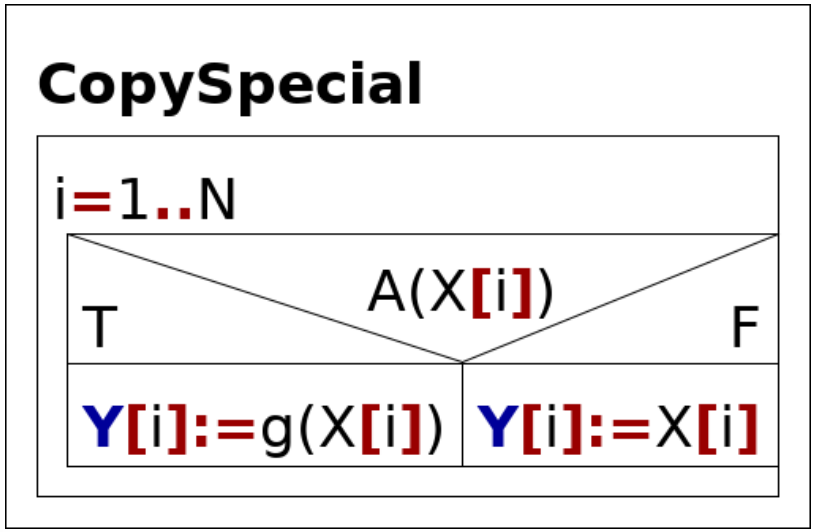
Postcondition:

$\forall i\,(1 \le i \le N): \mathbf{(A(X_i) \to Y_i = g(X_i)\ and\ not\ A(X_i) \to Y_i = X_i)}$

the algorithm can be generated automathicaly

7. Copy – function calculation

Algorithm$_1$:



Specification (a frequent special case)$_1$:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$

$g: \mathbb{S} \to \mathbb{S}$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Y_{1..N} \in \mathbb{S}^N$

Precondition: –

Postcondition:

$\forall i\,(1 \le i \le N)$: $\mathbf{(A(X_i) \to Y_i = g(X_i)\ and\ not\ A(X_i) \to Y_i = X_i)}$

7. Copy – function evaluation

Specification:
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}_1^N$
$f: \mathbb{S}_1 \to \mathbb{S}_2$
Output: $Y_{1..N} \in \mathbb{S}_2^N$
Precondition: –
Postcondition: $\forall i (1 \le i \le N) : Y_i = f(X_i)$

Specification (another special case)$_2$:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$

Output: $Y_{1..N} \in \mathbb{S}^N$

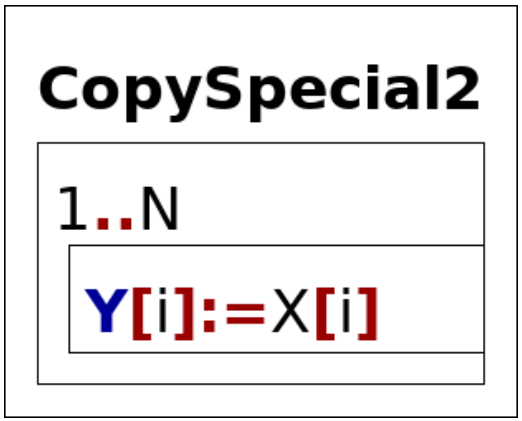
Precondition: –

Postcondition: $\forall i (1 \le i \le N) : \mathbf{Y_i = X_i}$

Note: there is no `f` function, or more precisely it is identical `(f(x):=x)`

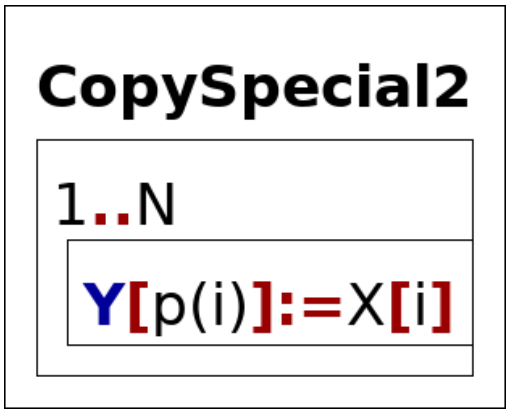
7. Copy – function calculation

Algorithm$_2$:



Note: It can be replaced by the `Y:=X` value assignment, if the two arrays have the same size.

Except when the indexes are different.



Let's **calculate** the sum of two vectors!

7. Copy – function evaluation

Specification:

Input: $N \in \mathbb{N}$, $P_{1..N}$, $Q_{1..N} \in \mathbb{R}^N$

$(P,Q) \in (\mathbb{R} \times \mathbb{R})^N$

$f: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $f((p_i, q_i)) := p_i + q_i$

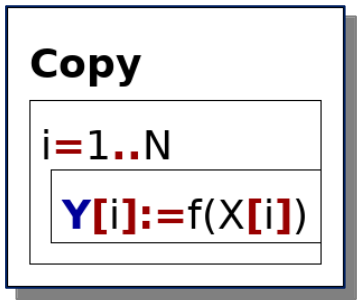
Output: $R_{1..N} \in \mathbb{R}^N$

Precondition: –

Postcondition:

$\forall i\,(1 \le i \le N): R_i = P_i + Q_i$

Algorithm:



8. Multiple item selection

Example tasks:

- Let's **define** all excellent student**s** in a class!
- Let's **calculate** the divisor**s** of an integer!
- Let's **list** all the vowel**s** from an English word!
- Let's collect **all** the people who are taller than 180cm, from a set of people!
- Let's **list** those day**s** of the year when it didn't freeze at noon!

8. Multiple item selection

What is common?

We have to list **all** elements from N "something", which have a common attribute A.

8. Multiple item selection

Specification:

Input: $N \in \mathbb{N},\ X_{1..N} \in \mathbb{S}^N,\ A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N},\ Y_{1..N} \in \mathbb{N}^N$

Using only the first Cnt elements

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

See PoA „Counting”

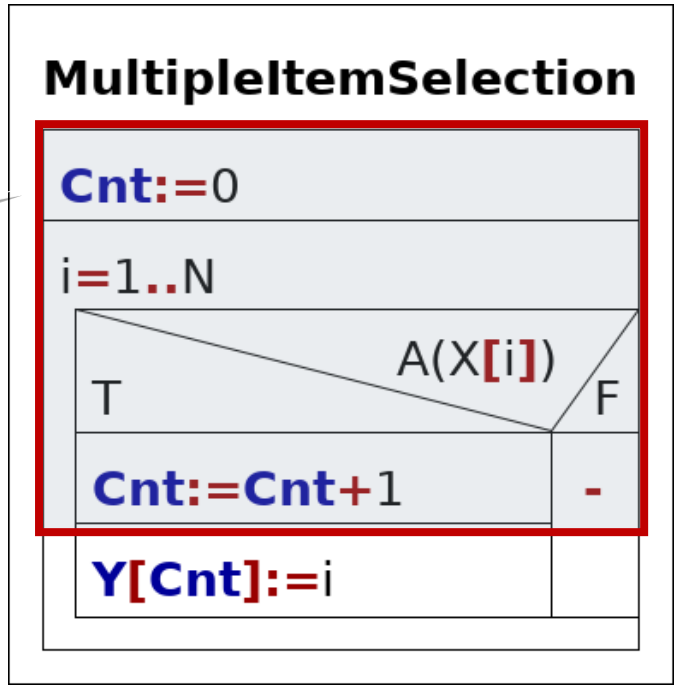
$$\text{and } \forall i (1 \le i \le Cnt): A(X_{Y_i}) \text{ and } Y \subseteq (1, 2, \ldots, N)$$

another way: $(Cnt, Y) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{\text{mutiselect}}} i$

8. Multiple item selection

Algorithm:

See PoA „Counting”



Comment: Collecting indexes is more general than collecting elements. If we needed elements then we would write: `Y[Cnt]:=X[i]` (and accordingly modified specification, as well – see later)

8. Multiple item selection

Specification$_2$ (selecting values):

Input: $N \in \mathbb{N},\ X_{1..N} \in \mathbb{S}^N,\ A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N},\ Y_{1..N} \in \mathbb{S}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt): A(Y_i)$ and $Y \subseteq X$

another way: $(Cnt, Y) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{mutiselect}} X_i$

8. Multiple item selection

Let's **list** those days of the year when it didn't freeze at noon!

Specification:

Input: $N \in \mathbb{N}$, $T_{1..N} \in \mathbb{R}^N$

$Pos : \mathbb{R} \to \mathbb{L}$, $Pos(x) = (x > 0)$

Output: $Cnt \in \mathbb{N}$, $F_{1..N} \in \mathbb{R}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ T_i > 0}}^{N} 1$

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $Y_{1..N} \in \mathbb{N}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : A(X_{Y_i})$ and $Y \subseteq (1, 2, \ldots, N)$

and $\forall i (1 \le i \le Cnt) : T_{NF_i} > 0$ and $NF \subseteq (1, 2, \ldots, N)$

8. Multiple item selection

Let's **list** those days of the year when it didn't freeze at noon!

Algorithm:

MultipleItemSelection

Cnt:=0
i=1..N
T: **Cnt:=Cnt+**1 / **Y[Cnt]:=**i; F: -

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$
Output: $Cnt \in \mathbb{N}$, $Y_{1..N} \in \mathbb{N}^N$
Precondition: –
Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : A(X_{Y_i})$ and $Y \subseteq (1,2,\ldots,N)$

NotFreezingDays

Cnt:=0
i=1..N
T: **Cnt:=Cnt+**1 / **NF[Cnt]:=**i; F: -

Specification:
Input: $N \in \mathbb{N}$, $T_{1..N} \in \mathbb{R}^N$
$Pos: \mathbb{R} \to \mathbb{L}$, $Pos(x) = (x > 0)$
Output: $Cnt \in \mathbb{N}$, $F_{1..N} \in \mathbb{R}^N$
Precondition: –
Postcondition: $Cnt = \sum_{\substack{i=1 \\ T_i > 0}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : T_{NF_i} > 0$ and $NF \subseteq (1,2,\ldots,N)$

8. Local selection

Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $X'_{1..N} \in \mathbb{S}^N$

Precondition: –

Postcondition: $Cnt = \sum_{i=1}^{N} [A(X_i)]$

and $\forall i (1 \le i \le Cnt): A(X'_i)$ and $X'_{1..Cnt} \subseteq X$

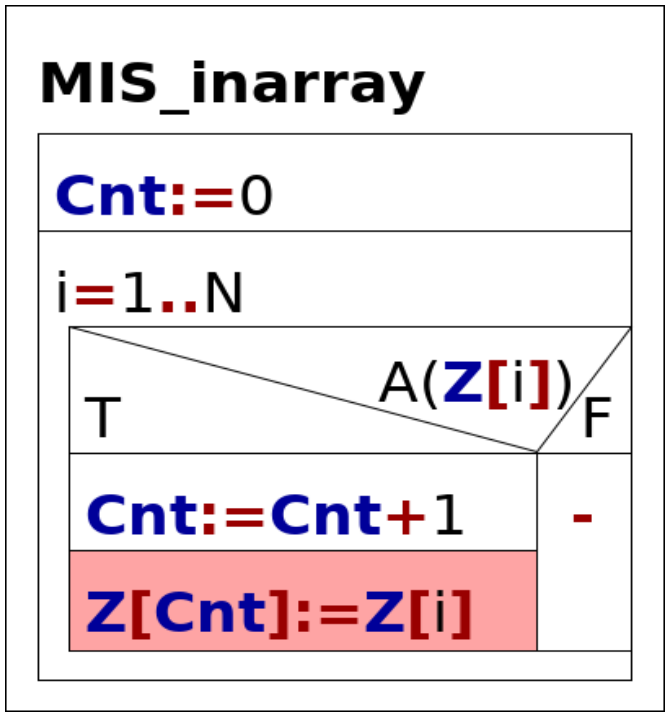
We can use the same variable for X and X' in our programm code.

$$X^{output}_{1..Cnt} \subseteq X^{input}_{1..N} \text{ and } \forall i (1 \le i \le Cnt): A(X_i)$$

8. Local selection

Main idea: we put the selected item to a place which we don't use later.

Algorithm:



9. Partitioning

Example tasks:

- Let's **list** all **even** and **all** **odd** numbers from a series!
- Let's **list** those days of the year when it was **freezing** **and** when it **wasn't** at noon!
- Let's **list** all the **vowels** **and** **consonants** of an English word!
- Let's **collect all** the people who are **shorter** than 140cm, who are **between** 140-180 cm, **and** those who are **taller** than 180cm, from a set of people!
- From a group of people, **list** the people who were born in **summer**, in **autumn**, in **winter** and in **spring**.

9. Partitioning

What is common?

We have to list all elements from N "something", which have a common attribute A, and then also list those ones not having attribute A. So we “assign” all the elements of the input to one of the output sequences.

9. Partitioning

$Y_{1..N} \in \mathbb{N}^{Cnt}$,
$Z_{1..N} \in \mathbb{N}^{N-Cnt}$

Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $Y_{1..N} \in \mathbb{N}^N$, $Z_{1..N} \in \mathbb{N}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt): A(X_{Y_i})$
and $\forall i (1 \le i \le N-Cnt)$: not $A(X_{Z_i})$
and $Y \subseteq (1,2,\ldots,N)$ and $Z \subseteq (1,2,\ldots,N)$

9. Partitioning

Specification$_2$:

...

Postcondition$_2$: $(Cnt, Y, Z) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{partition}} i$

... for values: $(Cnt, Y, Z) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{partition}} X_i$

9. Partitioning

Algorithm:

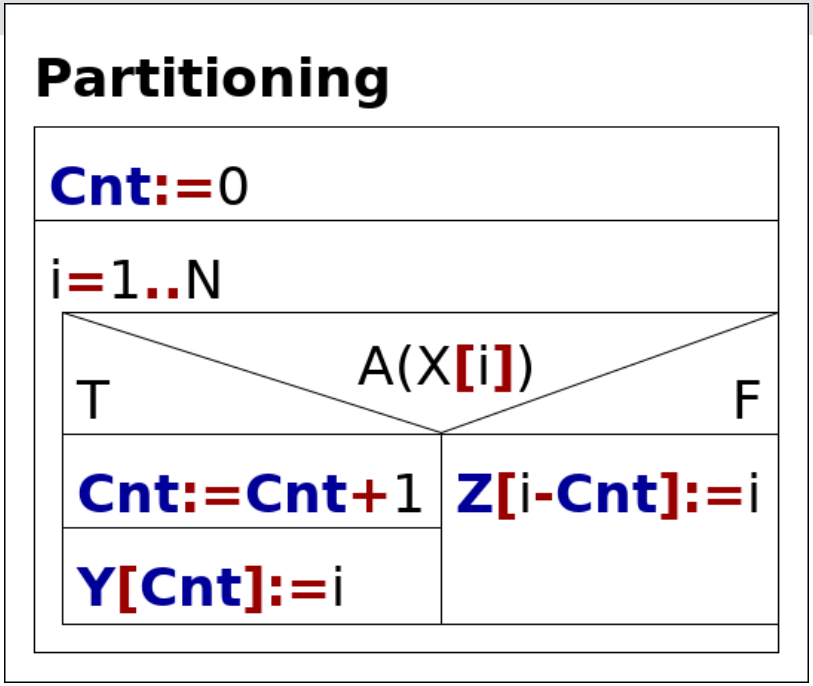
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $Y_{1..N} \in \mathbb{N}^N$, $Z_{1..N} \in \mathbb{N}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : A(X_{Y_i})$
and $\forall i (1 \le i \le N-Cnt) :$ not $A(X_{Z_i})$
and $Y \subseteq (1,2,\ldots,N)$ and $Z \subseteq (1,2,\ldots,N)$



Comment:

If we needed the **values**, than we would write **`:=X[i]`** instead of **`:=i`** (and accordingly specification should be modified).

9. Partitioning

Modification: There are exactly N number of elements in Y and Z together, so it's sufficient to have only one output array. We will collect the A attributed elements at the beginning **of the output array Y.**

Specification:

Input: $N \in \mathbb{N},\ X_{1..N} \in \mathbb{S}^N,\ A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N},\ Y_{1..N} \in \mathbb{N}^N$

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : A(X_{Y_i})$

and $\forall i (Cnt+1 \le i \le N) : \text{not } A(X_{Y_i})$

and $Y \in \textbf{Permutation}(1, 2, \ldots, N)$

Permutation(1,2,...,N):=set of all permutations of 1..N (set of sets)

9. Partitioning

Specification$_2$:

...

Postcondition$_2$: $(\mathrm{Cnt}, \mathrm{Y}) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{\mathrm{partition}_2}}\ i$

... for values: $(\mathrm{Cnt}, \mathrm{Y}) = \underset{\substack{i=1 \\ A(X_i)}}{\overset{N}{\mathrm{partition}_2}}\ X_i$

9. Partitioning

Algorithm:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $Y_{1..N} \in \mathbb{N}^N$

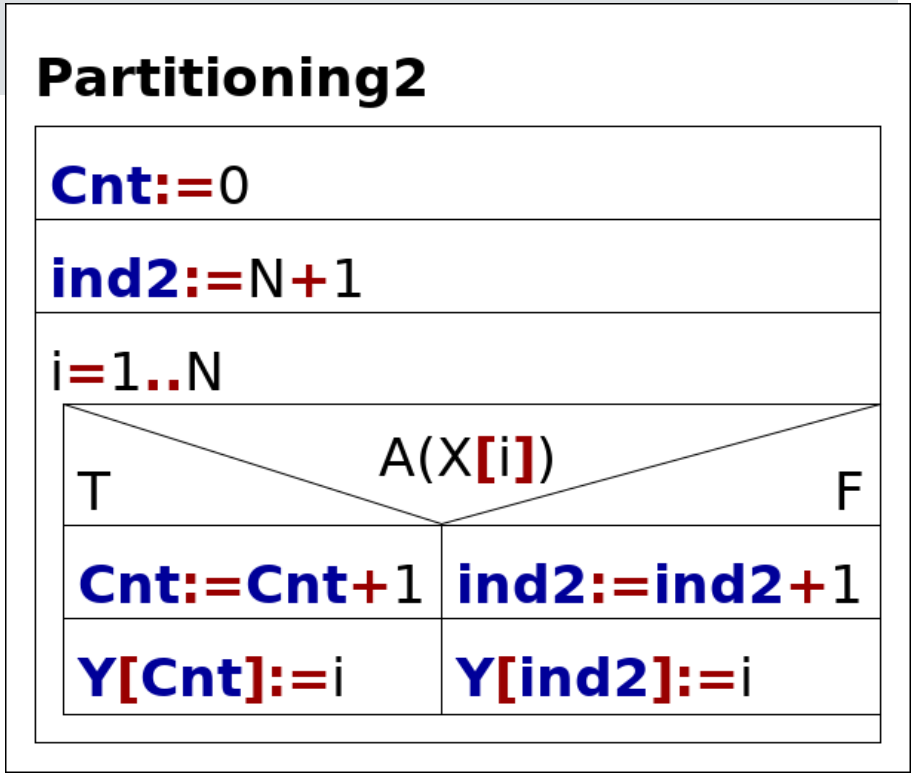
Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$

and $\forall i (1 \le i \le Cnt) : A(X_{Y_i})$

and $\forall i (Cnt+1 \le i \le N) : \text{not } A(X_{Y_i})$

and $Y \in \text{Permutation}(1, 2, \ldots, N)$



Comment:

We can use an extra variable (ind2) to know where we are from back in Y.

9. Local partitioning

Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $A: \mathbb{S} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $X'_{1..N} \in \mathbb{S}^N$

Precondition: –

Postcondition: $Cnt = \sum_{i=1}^{N} [A(X_i)]$

and $\forall i (1 \le i \le Cnt) : A(X'_i)$

and $\forall i (Cnt+1 \le i \le N) : \text{not } A(X'_i)$

and $X' \in Permutation(X)$

We can use the same variable for X and X' in our programm code.

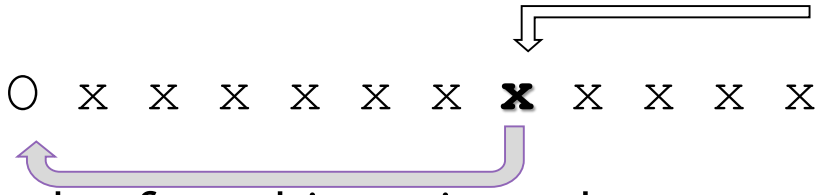
9. Local partitioning

Main idea for the algorithm:

1. Pick out the **first** item of the sequence

O x x x x x x x x x x x

2. Let's search an item from **back** which needs to be in the **front-sequence** (attribute A is true for it)



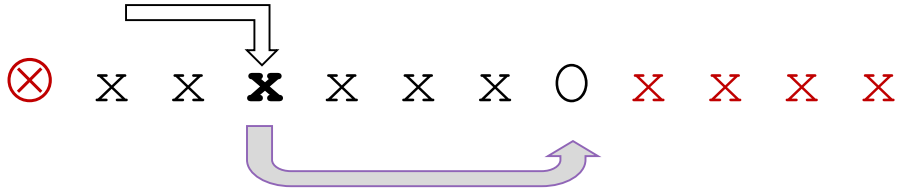
3. Move the found item into the empty (first) place.

⊗ x x x x x x O x x x x

The first item and the items after the new empty place are at the right place.

9. Local partitioning

4. Now we have an empty place at the end-sequence. From the **beginning** (starting after the first item) let's search an item which needs to be in the end-sequence (attribute A is not true for it)



5. Move the founded item into the empty place. And we can search from back away.

The items before the new empty place and after the new place of the moved item are at right place.

9. Local partitioning

6. … and so on …

7. We can finish searching if we reached the empty place from one direction.

x x x x x O x x x x x x

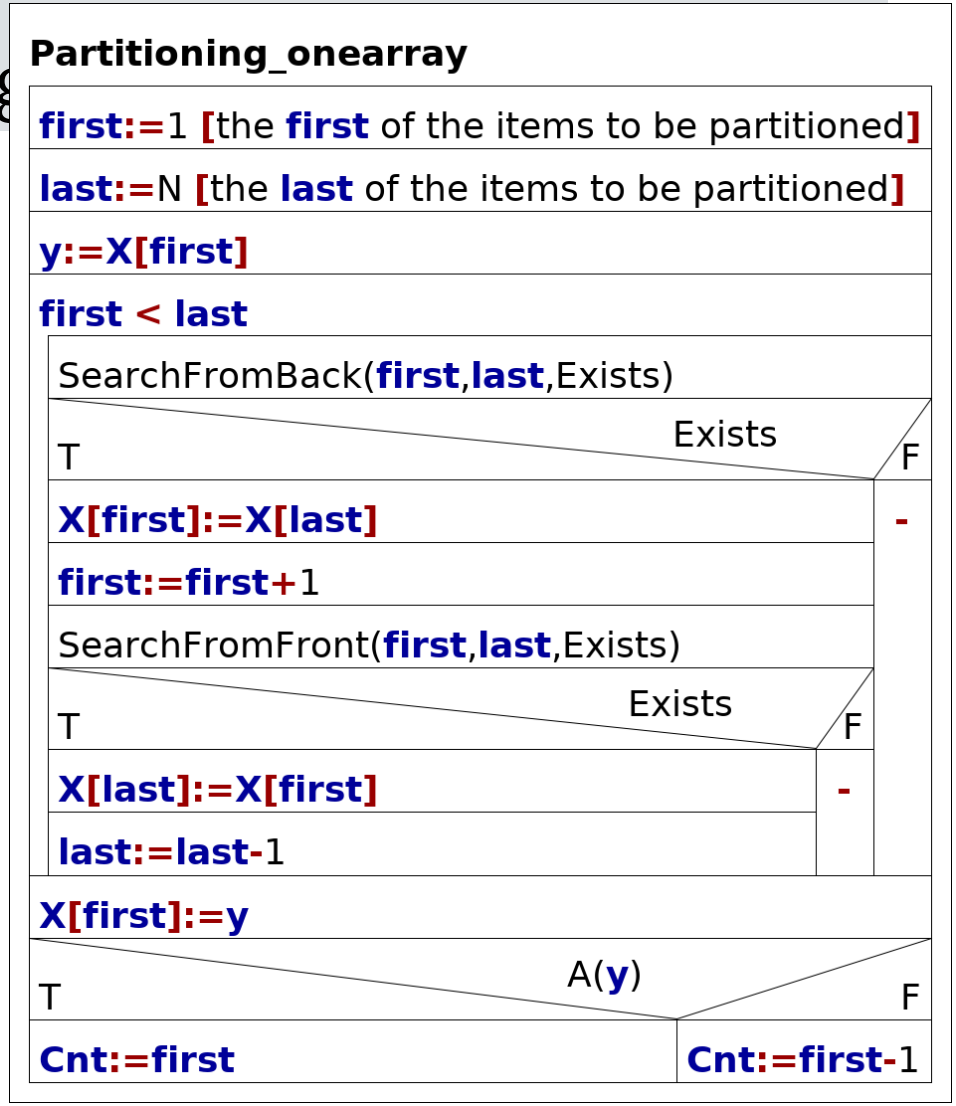
8. We put our first item back into this empty place.

9. Local partitioning

Algorithm:

What we know about X?

- At the beginning: it is the input sequence
- At the end: it is the permutation of the original (input) sequence
- While „running”: to „first” are items with attribute A, from „last” items which are not of attribute A.



9. Local partitioning

```
SearchFromFront(first,last:Integer,Exists:Boolean)
    first<last and A(X[first])
        first:=first+1
    Exists:=first<last
```

```
SearchFromBack(first,last:Integer,Exists:Boolean)
    first<last and not A(X[last])
        last:=last-1
    Exists:=first<last
```

10. Intersection

Example tasks:

- Let's list all **common** divisors of **two** integers!
- We investigate birds in winter **and** summer. Let's **collect** the birds which are not migratory!
- Based on the free hours from the calendars of **two people**, let's **determine** when they can meet.
- Let's lists the animals which could be seen **both** in the zoos of Budapest and Veszprém.

10. Intersection

What is common?

We have two sets as input (with elements of the same type), and we have to list all elements that are **part of both sets**.


10. Intersection

Definition of IsSet function:

$\mathtt{IsSet}(x_{1..n}) := \text{not } \exists i\,(1 \le i \le n) : x_i \in x_{1..i-1}$

$\mathtt{IsSet}(x_{1..n}) := i \neq j \rightarrow x_i \neq x_j$

Specification:

Input: $N, M \in \mathbb{N},\ X_{1..N} \in \mathbb{S}^N,\ Y_{1..M} \in \mathbb{S}^M$

Output: $Cnt \in \mathbb{N},\ Z_{1..MIN(N,M)} \in \mathbb{S}^{MIN(N,M)}$

Precondition: $\mathtt{IsSet}(X)$ and $\mathtt{IsSet}(Y)$

Postcondition: $Cnt = \sum_{\substack{i=1 \\ x_i \in Y}}^{N} 1$

and $\forall i\,(1 \le i \le Cnt) : Z_i \in X$ and $Z_i \in Y$
and $\mathtt{IsSet}(Z)$

10. Intersection

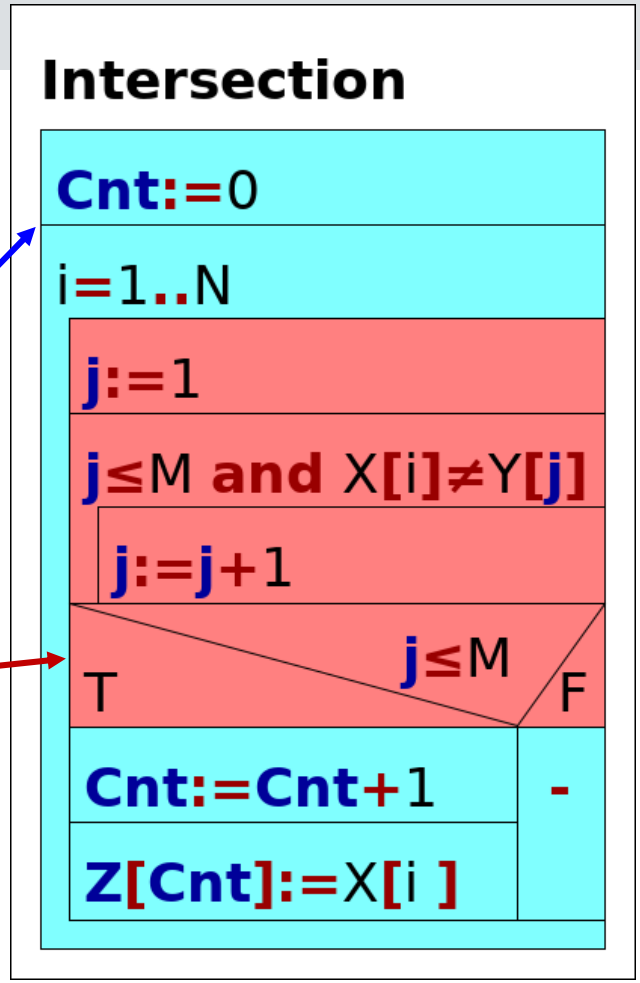
Specification$_2$:

Postcondition$_2$: `(Cnt,Z)=Intersection(N,X,M,Y)`

another way: $(Cnt,Z) = \underset{\substack{i=1 \\ X_i \in Y}}{\overset{N}{\text{multiselect}}} X_i$

10. Intersection

Algorithm:



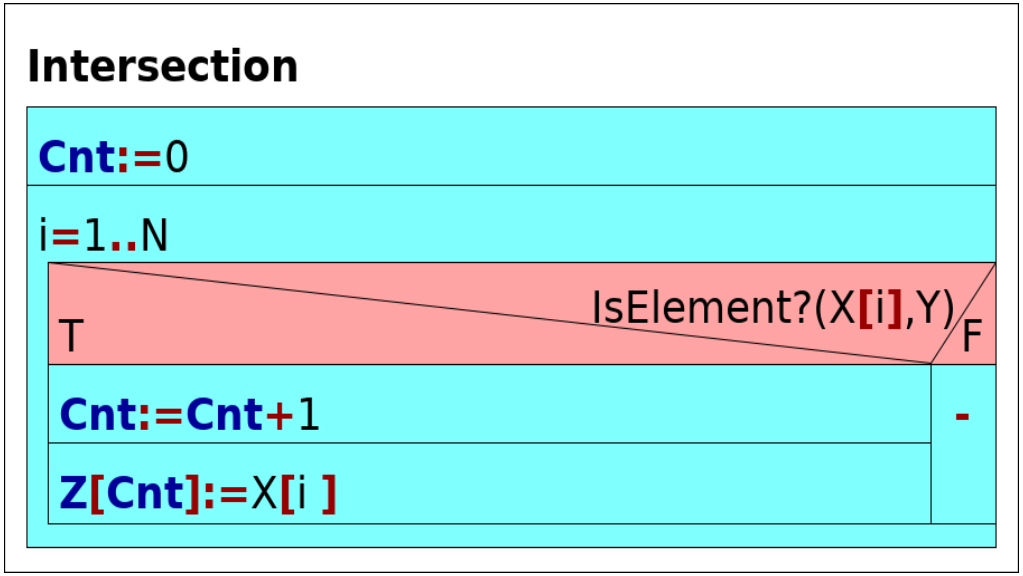
Comment:

The solution is a multiple item selection and a decision.

10. Intersection

As the decision PoA gives back a logical value (boolean), it could appear as a logical expression in the conditional statement

Algorithm:



We need to write a function (IsElement)!

10. Intersection

Variations for intersection:

- We have two sets, we have to determine the **count of common elements**.
- We have two sets, we have to determine if they have **at least one common element**.
- We have two sets, we have to give **one common element**.

11. Union

Example tasks:

- We have two courses, let's **list** the students visiting **both** courses!
- Let's **collect** all the birds we can investigate in winter **as well as** in summer!
- Based on the free hours from the calendars of two persons, let's **determine** when we can reach **at least one of them**!
- Let's **lists** the animals which could be seen **either** in the zoos of Budapest **or** Veszprém.


11. Union

What is common?

We have two sets as input (with elements of the same type), and we have to list all elements that are included at least in one of the sets.


11. Union

Specification:

Input: $N,M \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $Y_{1..M} \in \mathbb{S}^M$

Output: $Cnt \in \mathbb{N}$, $Z_{1..N+M} \in \mathbb{S}^{N+M}$

Precondition: `IsSet(X) and IsSet(Y)`

Postcondition: $Cnt = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$

and $\forall i (1 \le i \le Cnt) : Z_i \in X$ **or** $Z_i \in Y$
and `IsSet(Z)`

11. Union

Specification$_2$:

Postcondition$_2$: `(Cnt, Z)=Union(N,X,M,Y)`

another way: $(Cnt,Z) = X + \underset{\substack{i=1 \\ Y_j \notin X}}{\overset{N}{\texttt{multiselect}}}\ Y_j$

11. Union

Algorithm

Specification:

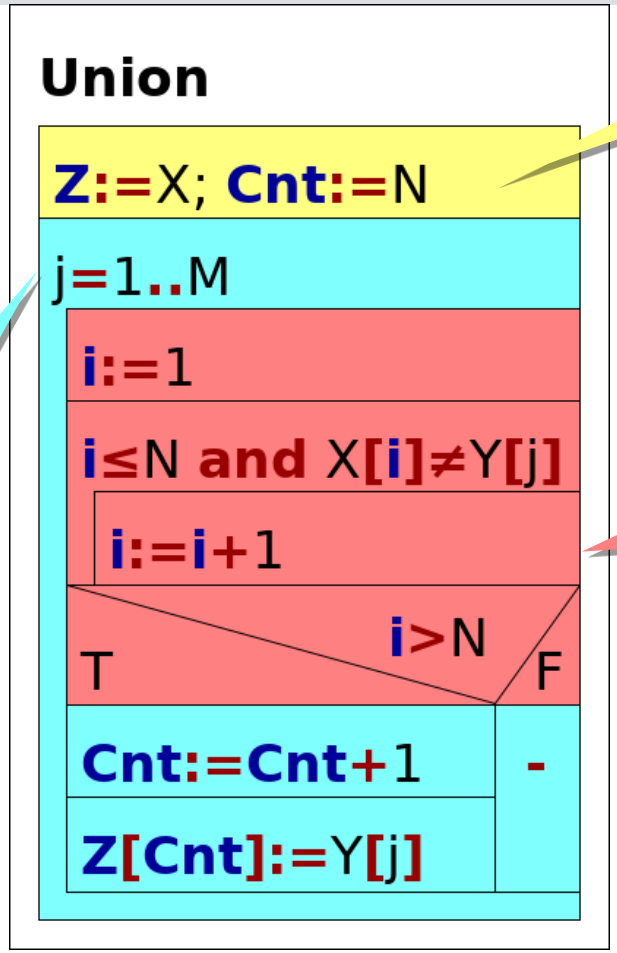
Input: $N, M \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$, $Y_{1..M} \in \mathbb{S}^M$

Output: $Cnt \in \mathbb{N}$, $Z_{1..N+M} \in \mathbb{S}^{N+M}$

Precondition: IsSet(X) and IsSet(Y)

Postcondition: $Cnt = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$

and $\forall i (1 \le i \le Cnt) : Z_i \in X$ **or** $Z_i \in Y$
and IsSet(Z)



copy

decision

multiple item selection

11. Union

Variations for union:

- We have two sets, we have to determine the count of all the elements altogether.
- We have two sets, we have to determine their difference (`X\Y`).
- We have two sets, we have to give those elements which are only in one of the sets (`X\Y ∪ Y\X`).

Patterns of Algorithms II.

Sequence → Sequence

- Copy – calculation with function
- Multiple item selection
- Sorting (later)

Sequence → Sequences

- Partitioning

Sequences → Sequence

- Intersection
- Union