From Visualization to Coding: Practicing Graphical Loop Invariants in CAFÉ 2.0

Géraldine Brieven, Benoit Donnet

Introduction

Context : CS1 Course

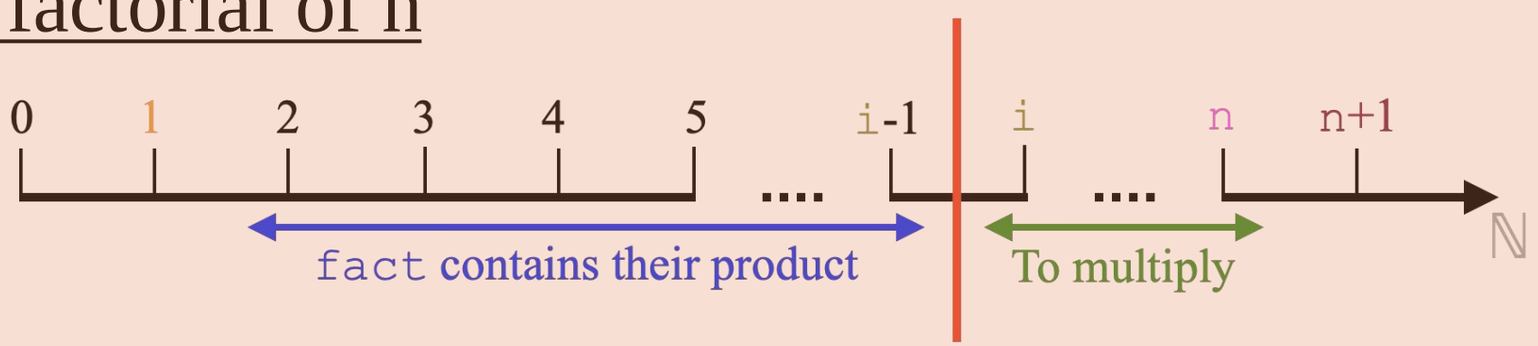
Skills : Problem Solving and Abstract thinking skills

How? : The Graphical Loop Invariant (GLI)

Each time you implement a loop, all the variables that are handled are ***characterized (individually and/or with respect to each other)****.*
That variables' state must be ***true at each evaluation of the guard loop.***

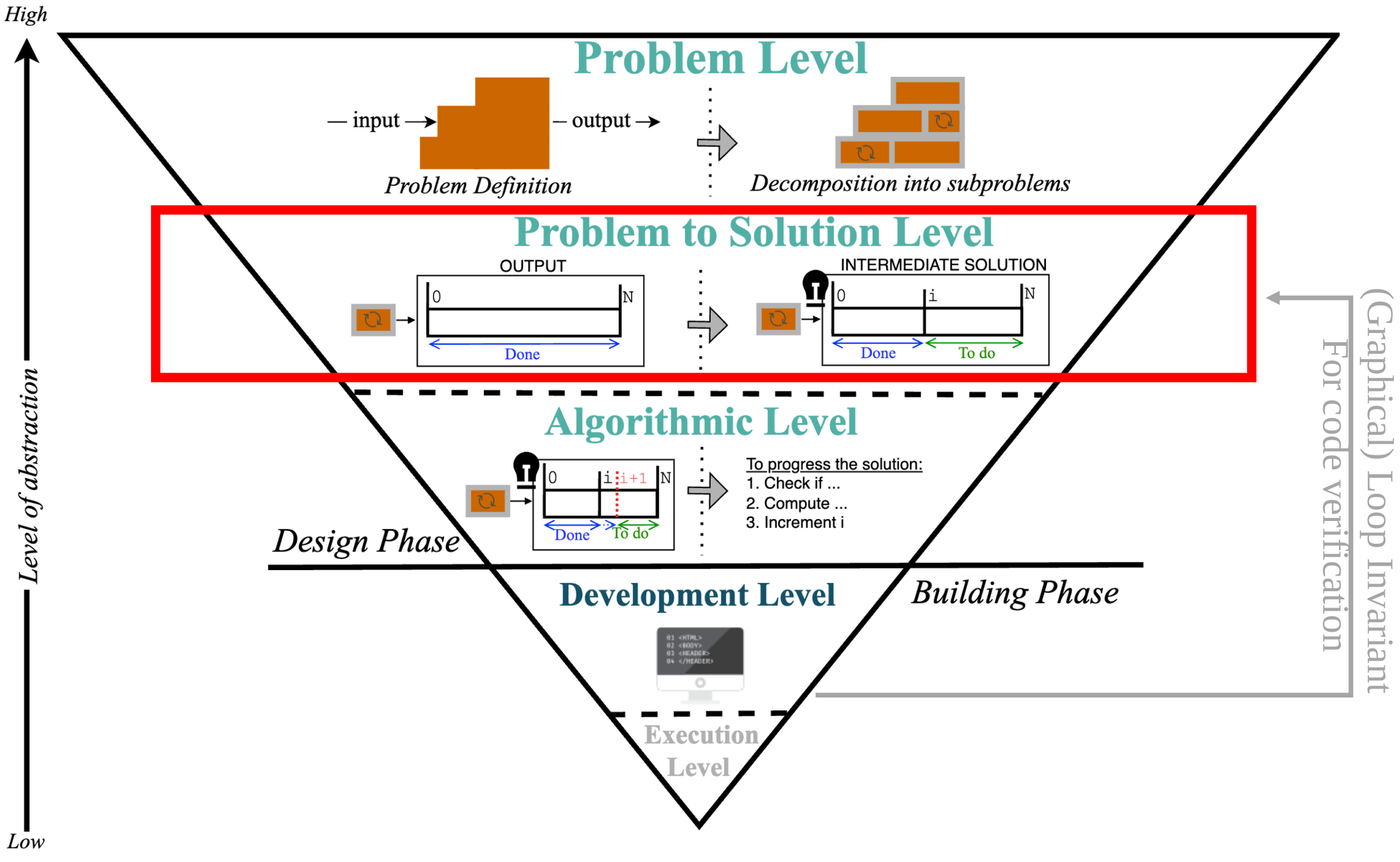
Eg: Compute the factorial of n

Graphical LI:



Loop Invariant (LI): $\equiv\ 2 \leq i \leq n\ \wedge\ fact = (i-1)!$

Level of abstraction where the GLI stands



Agenda

Agenda:

Section 1	Section 2	Section 3	Section 4	Section 5

- Section 1: Graphical Loop Invariant (GLI)
- Section 2: From the GLI to the code
- Section 3: Learning Tools
- Section 4: Preliminary Results
- Section 5: Further work and Conclusion

Agenda

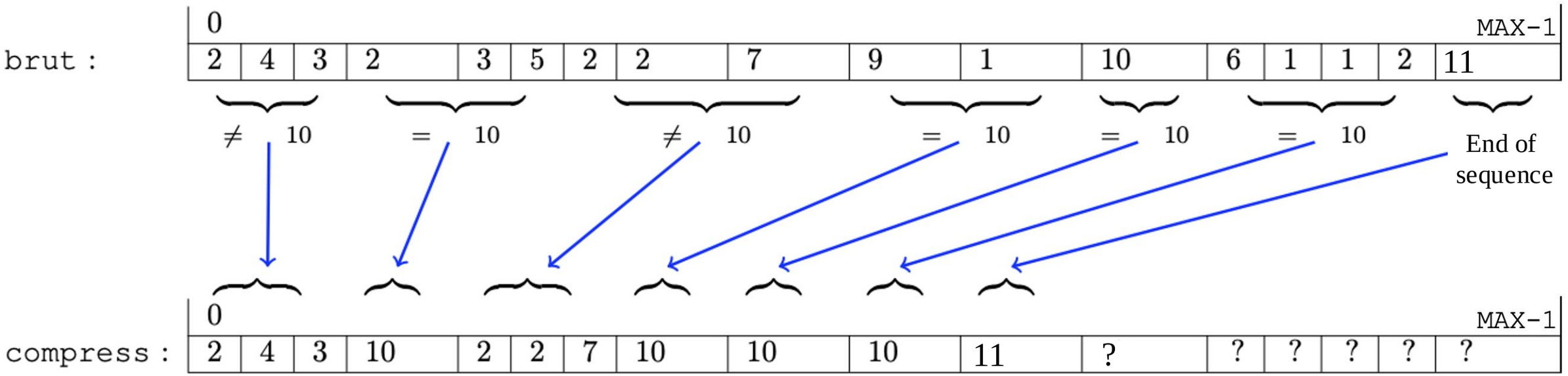
Agenda:

- Section 1: Graphical Loop Invariant (GLI)
- Section 2: From the GLI to the code
- Section 3: Learning Tools
- Section 4: Preliminary Results
- Section 5: Further work and Conclusion

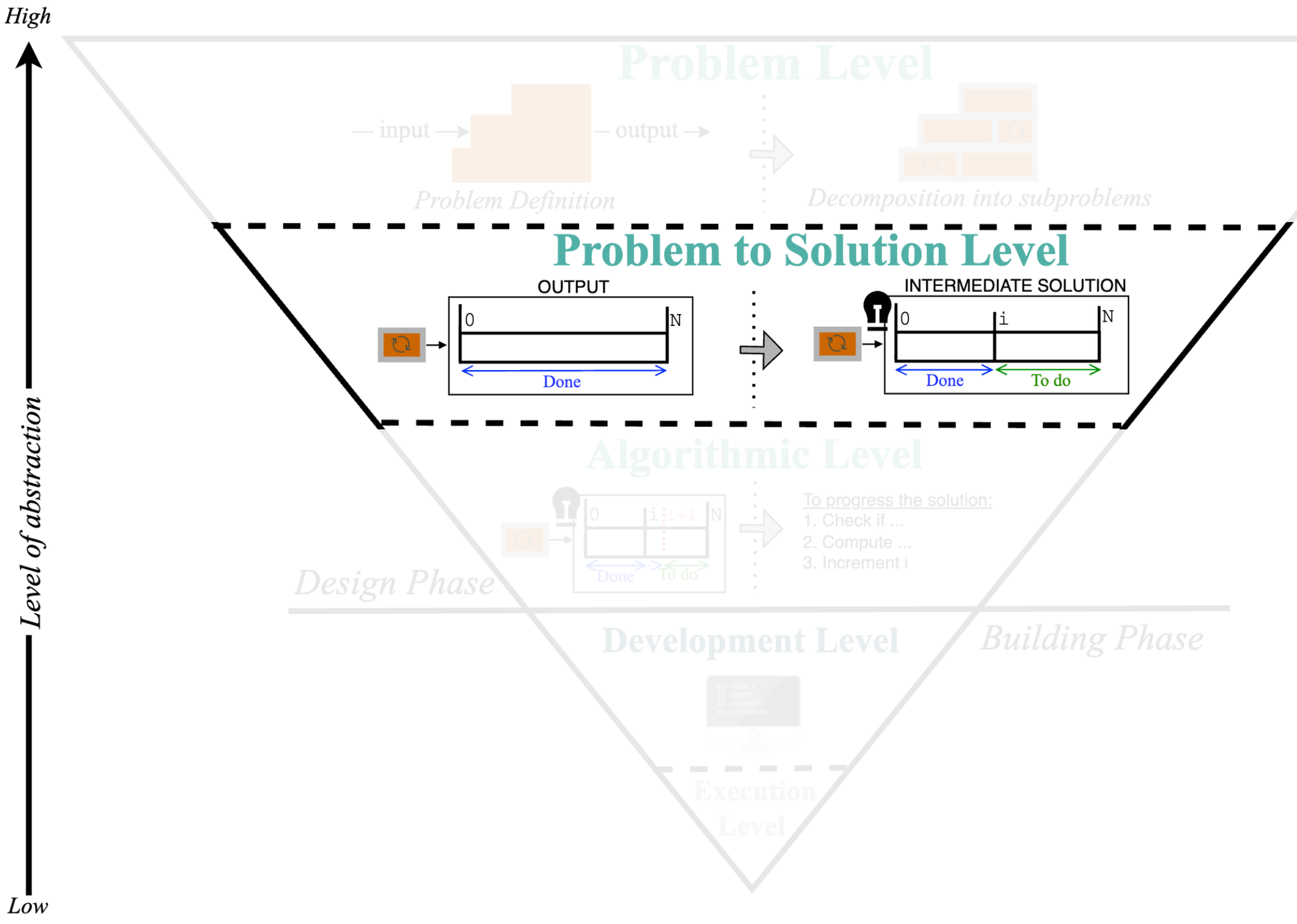
Graphical Loop Invariant

through an example

Problem : Compressing an array `brut` of size `MAX` based on sums of `10`. The last element of `brut` is always `11`.

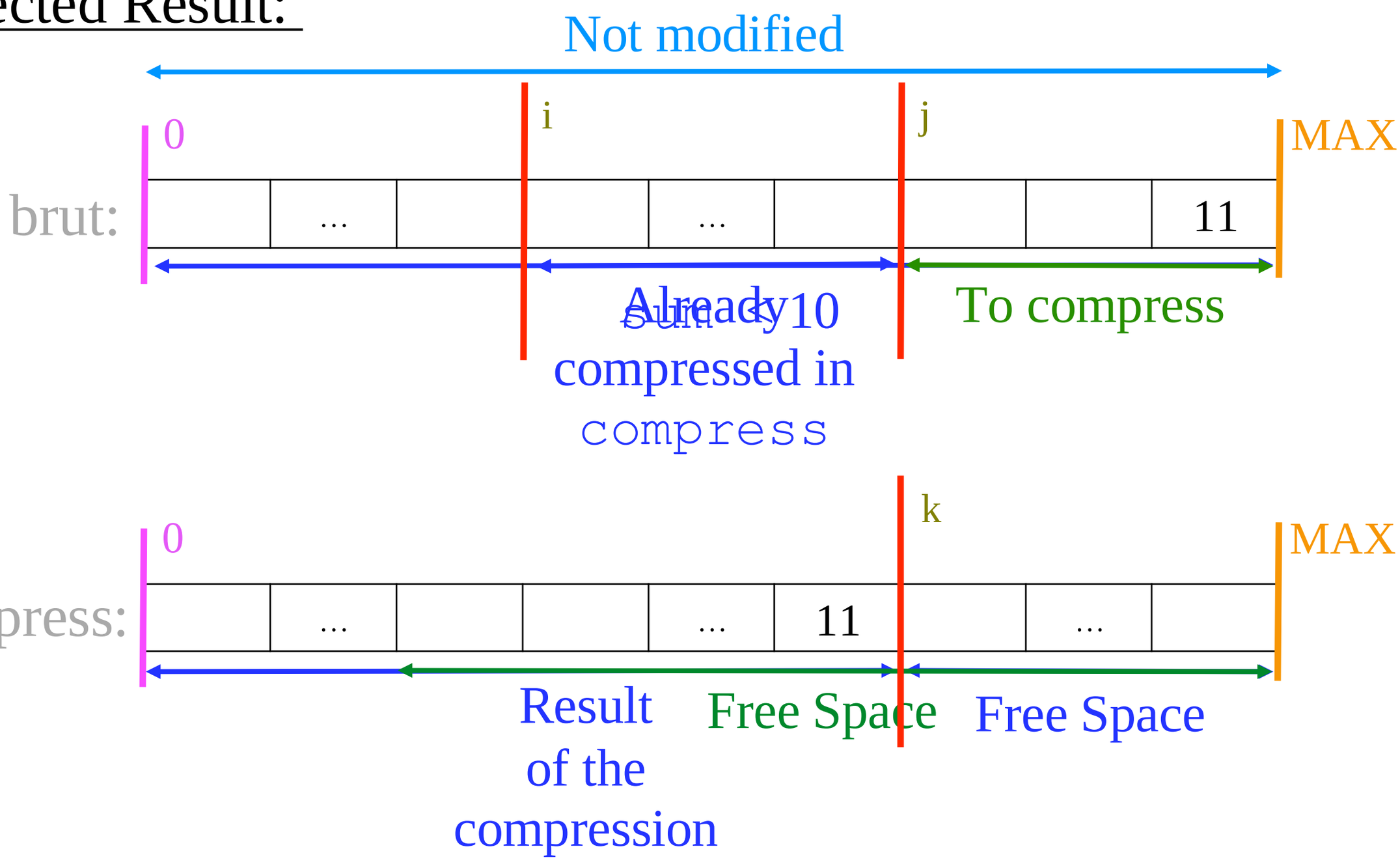


From the output to the intermediate solution (GLI)



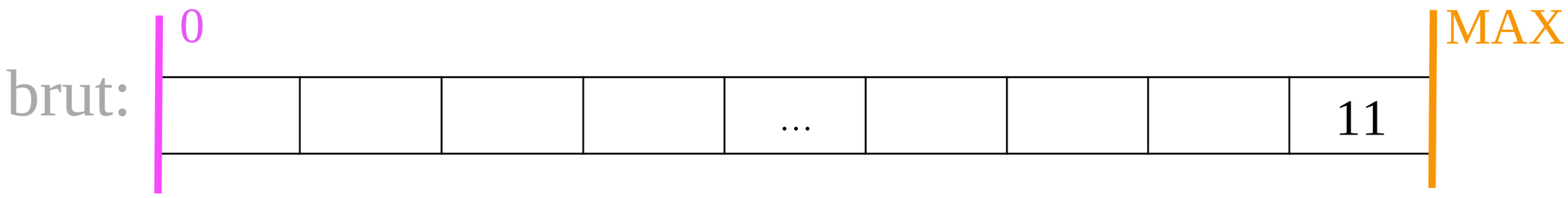
Finding a GLI: constant relaxation

Expected Result:



Rules supporting the GLI

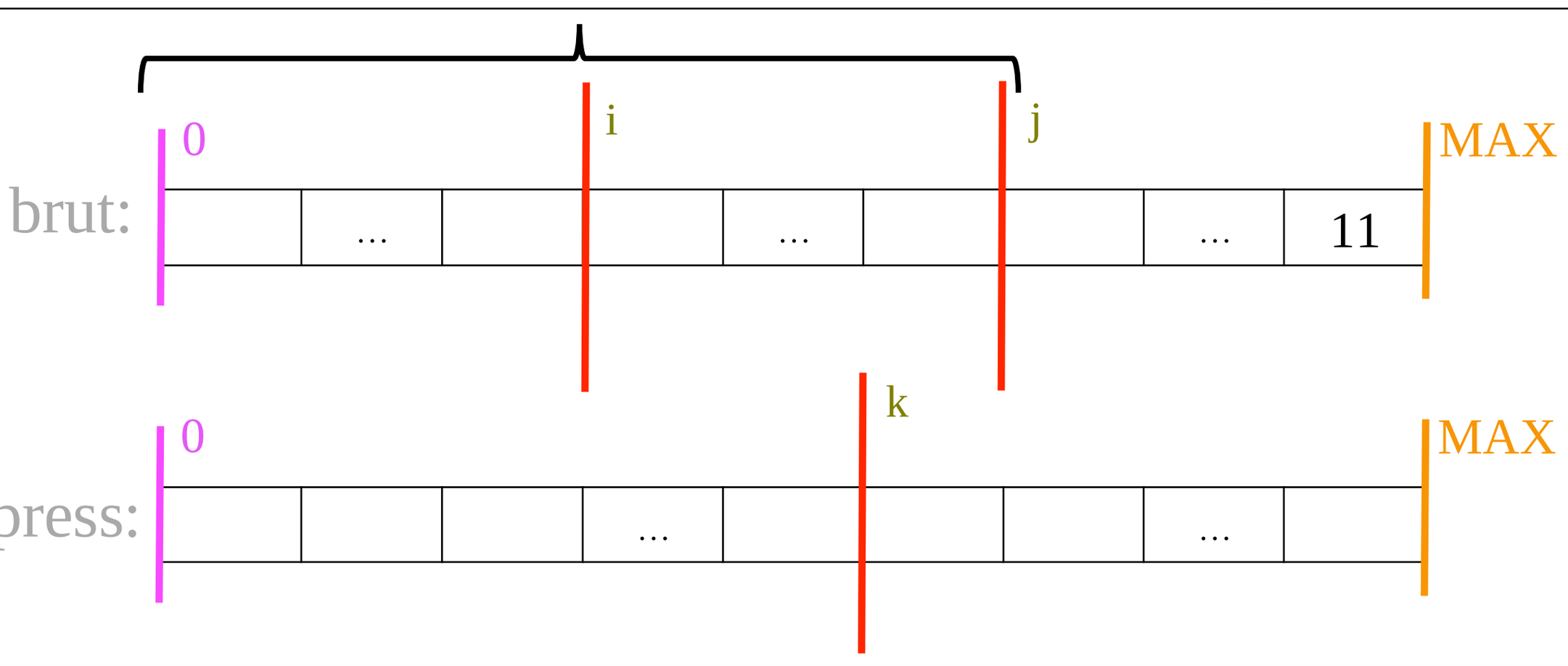
Rules 1 & 2 :



Rules supporting the GLI

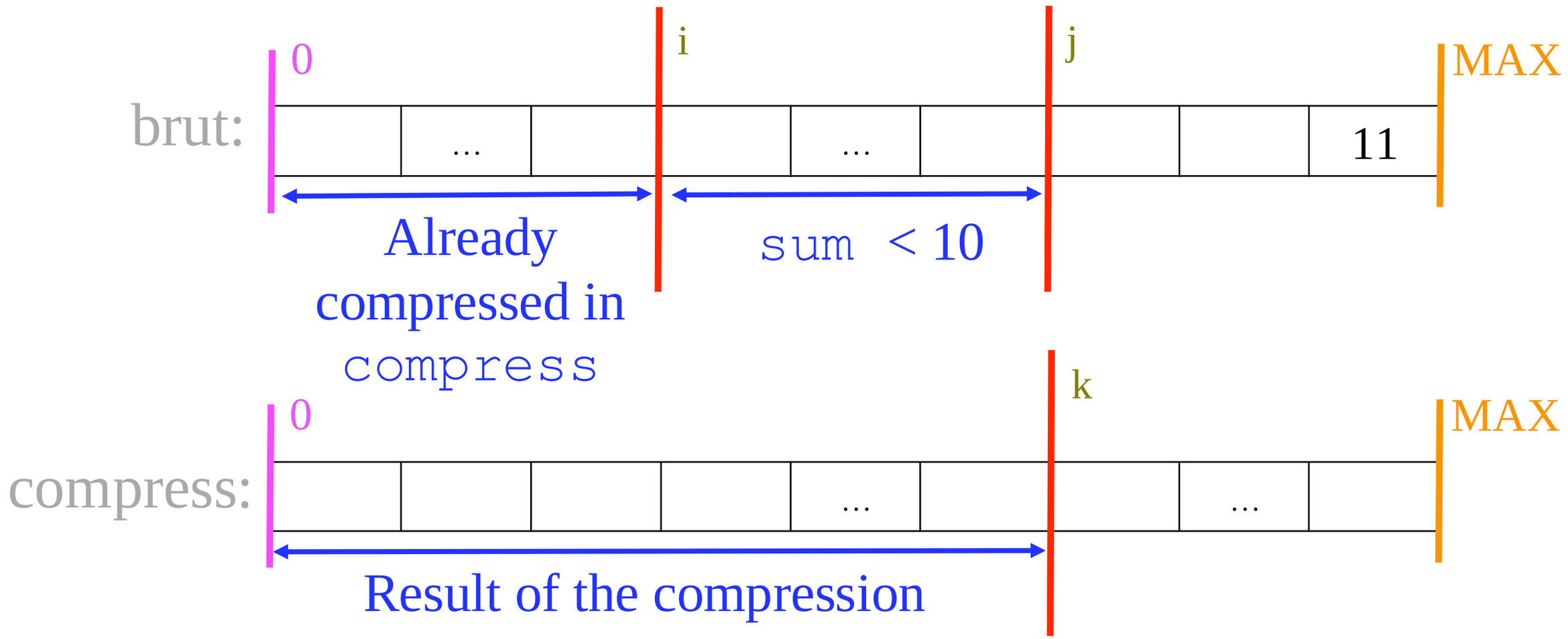
Rules 3 & 4 :

2 subzones treated in `brut` : - What has been compressed
- What is candidate to be compressed



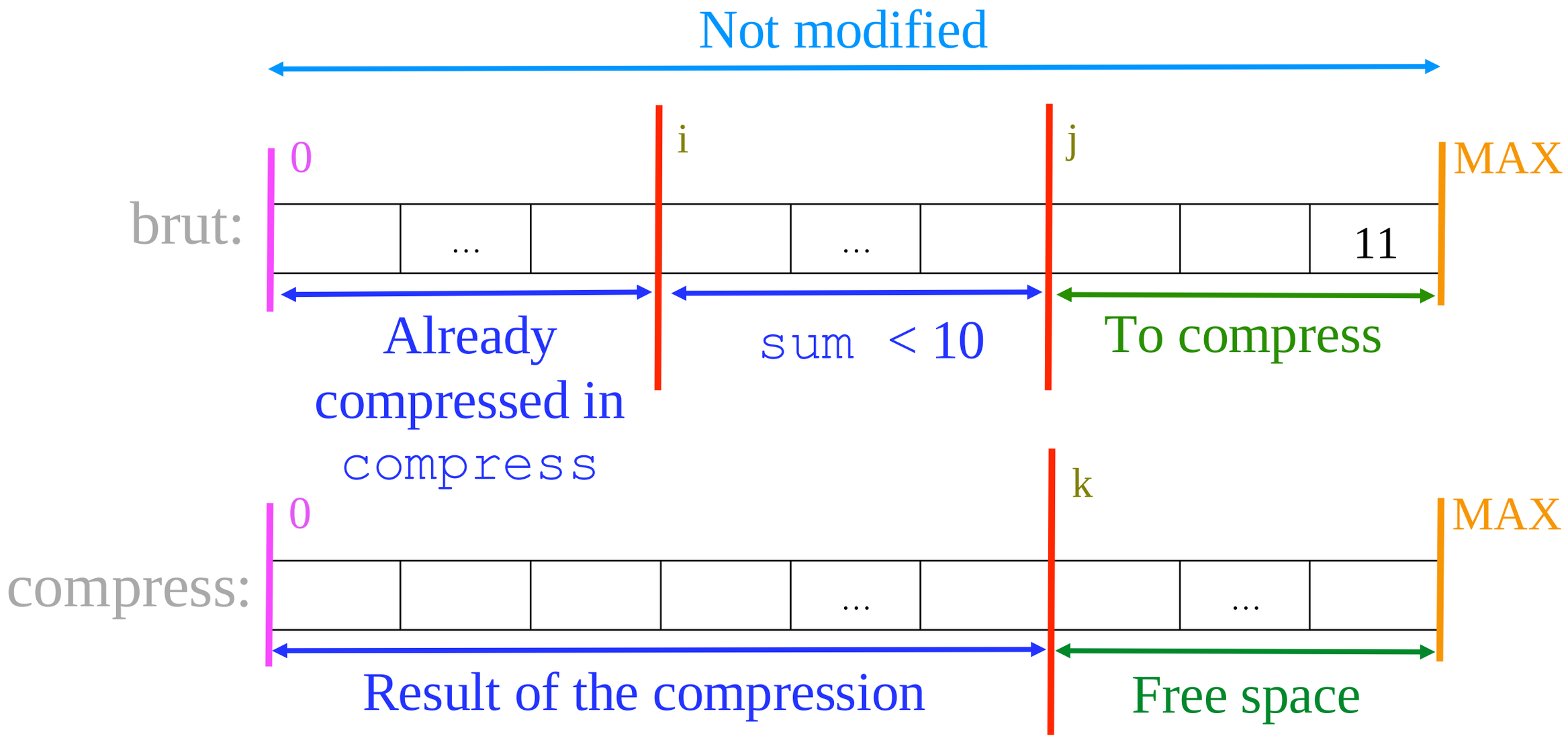
Rules supporting the GLI

Rule 5 :



Rules supporting the GLI

Rule 6 :



Different possible patterns

Object	Pattern
Number	var name: d_{k-1} d_{k-2} ... d_j d_{j-1} ... d_1 d_0
Array	var name: 0 … N-1 N
Integers	$\mathbb{Z}$
Linked List	L : e_0 e_1 ⋯ e_{n-1}

Agenda

Not modified

From the GLI to the code

Preliminary Results

s_curr

Section 1

Section 6

Agenda:

Section 1	Section 2	Section 3	Section 4	Section 5

Graphical Loop Invariant

Learning Tools

Further work and Conclusion

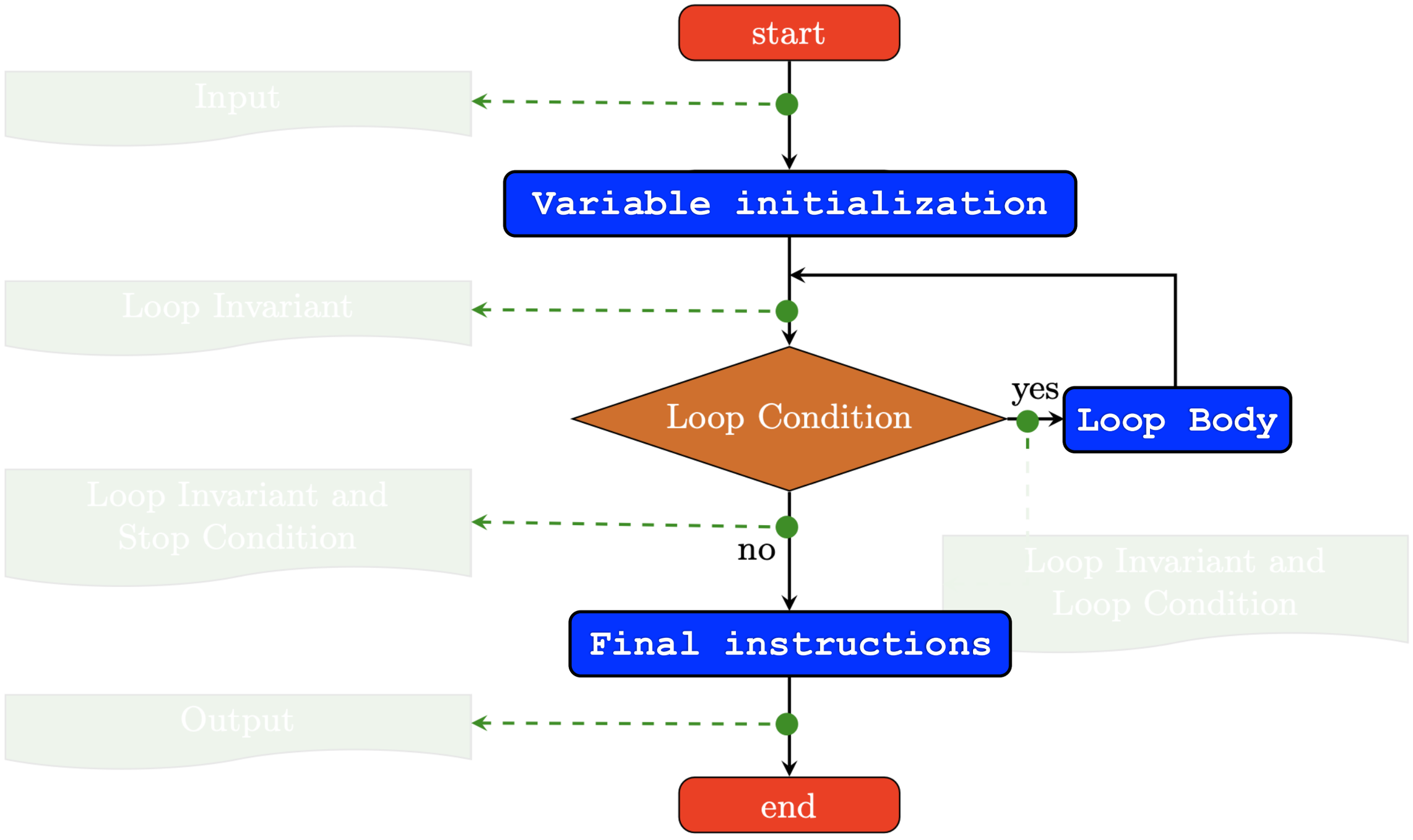
Already discussed

To be discussed

Legend :

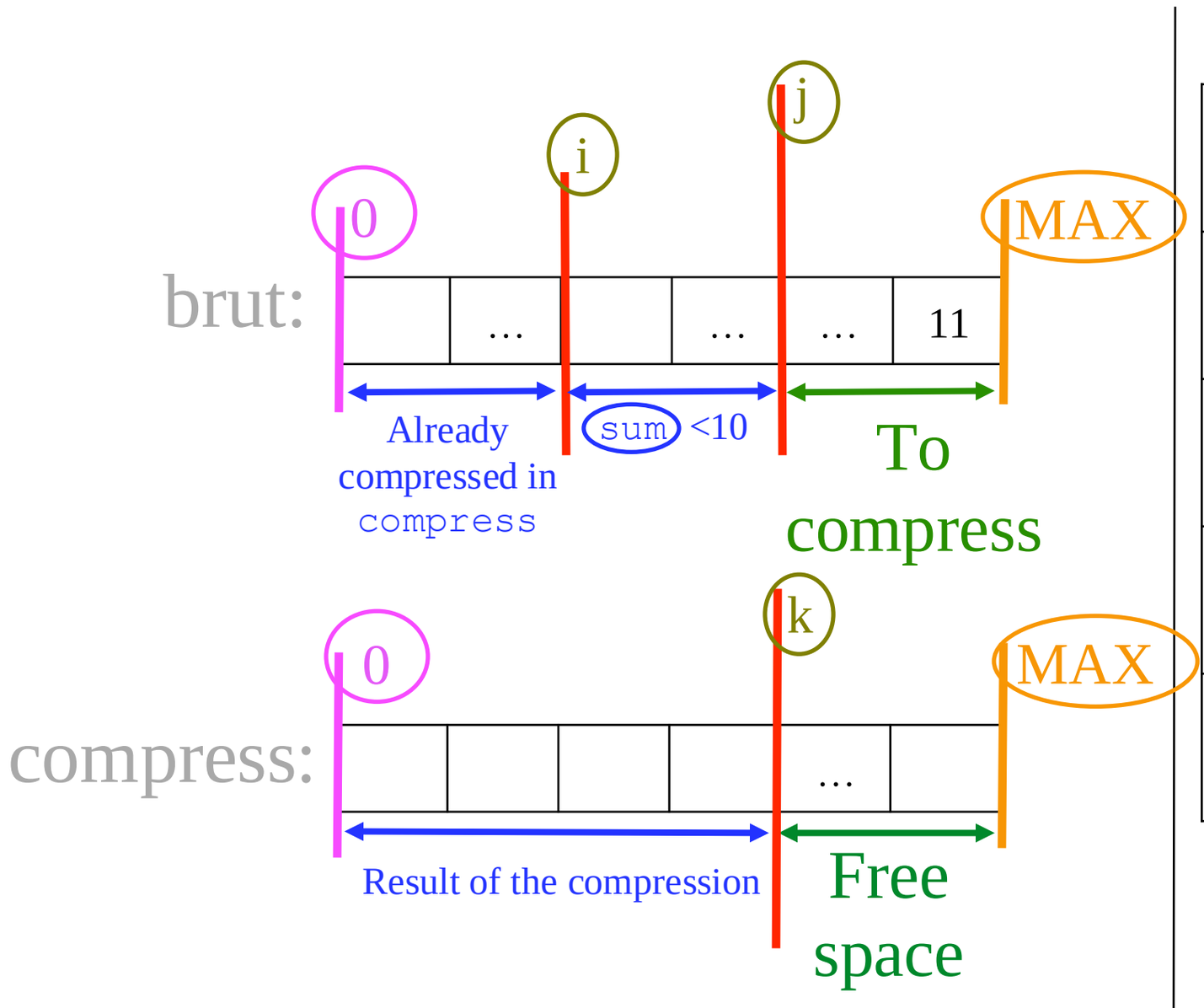
- Rule 1
- Rule 2
- Rule 3
- Rule 4
- Rule 5
- Rule 6

Programming Methodology



Programming Methodology

Initial State:



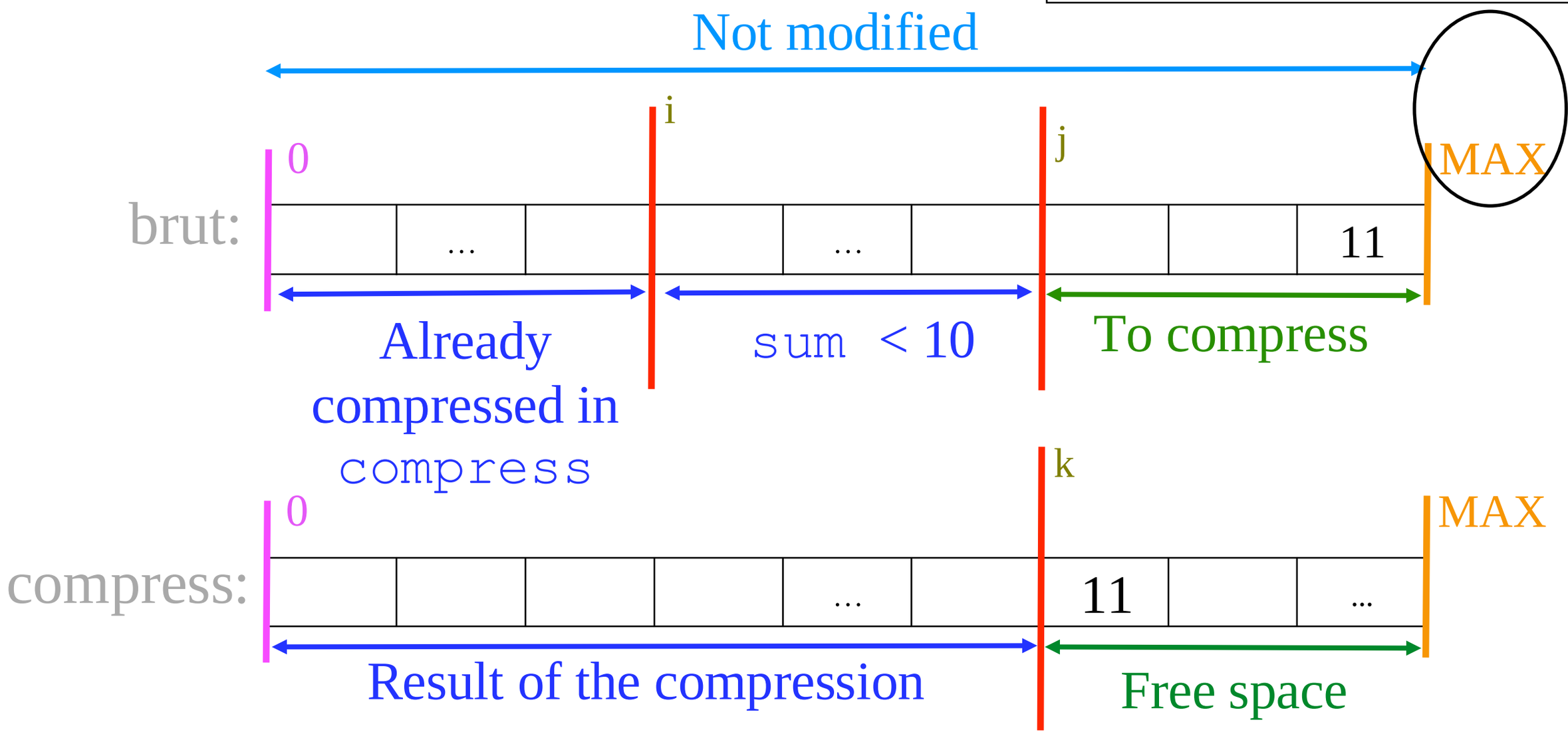
Variable	Type	Initial value
i	unsigned int	0
j		0
sum		0
k		0

Programming Methodology

Final State:

Stopping Condition :

```
i == MAX && j == MAX
```



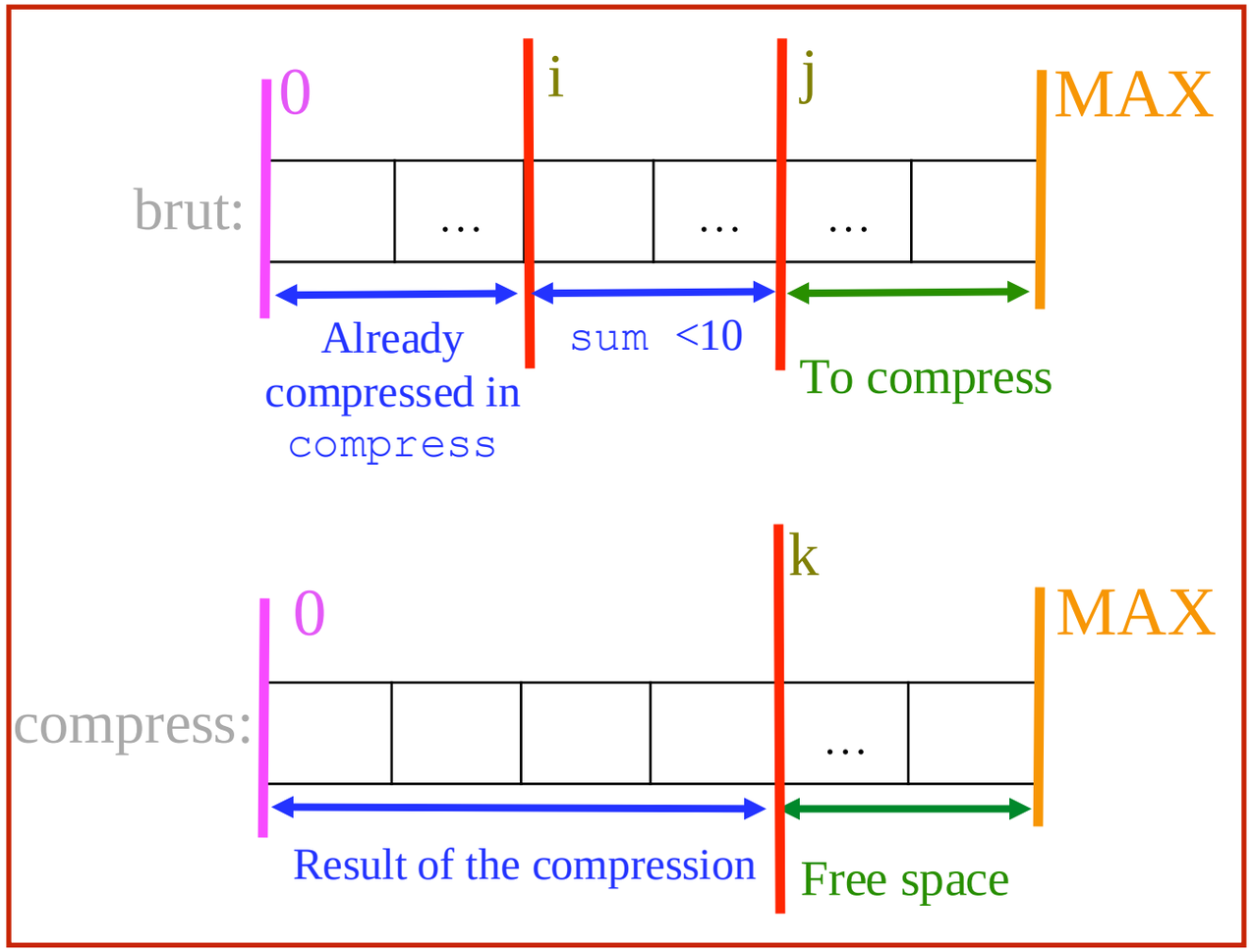
Programming Methodology

Resulting Code :

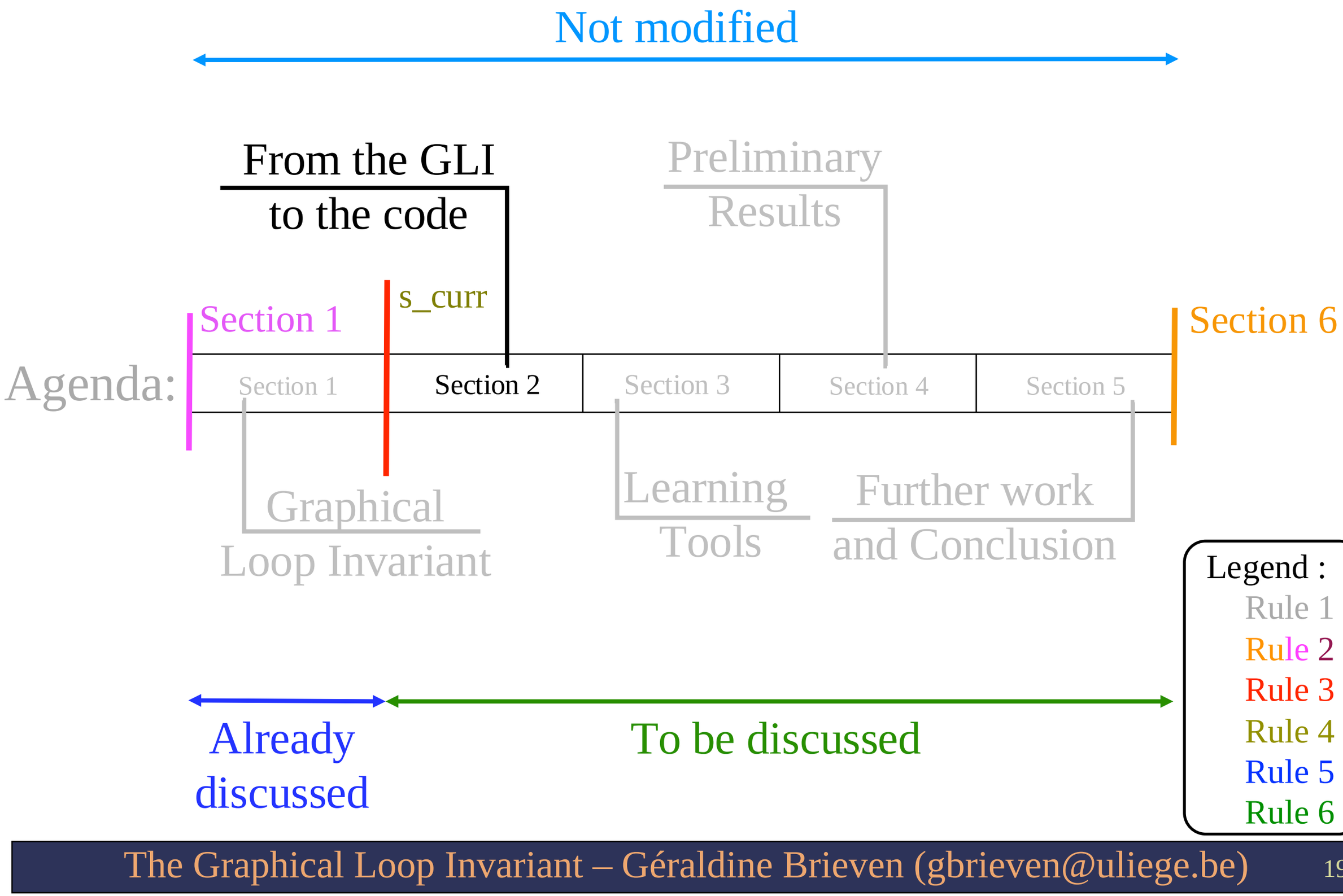
```
unsigned int i = 0, j = 0, k=0, sum = 0;
```
from Initial State

```
while( !(i==MAX && j==MAX) )
{
     sum += brut[j];
     j++;
     if(sum<=10){
          if(sum == 10){ //Compression
               compress[k] = 10;
               i = j;
          }else{ // Copy of element
               compress[k] = brut[i];
               i++;
               j = i;
          }
          k++;
          sum = 0;
     }
}
```
from Final State

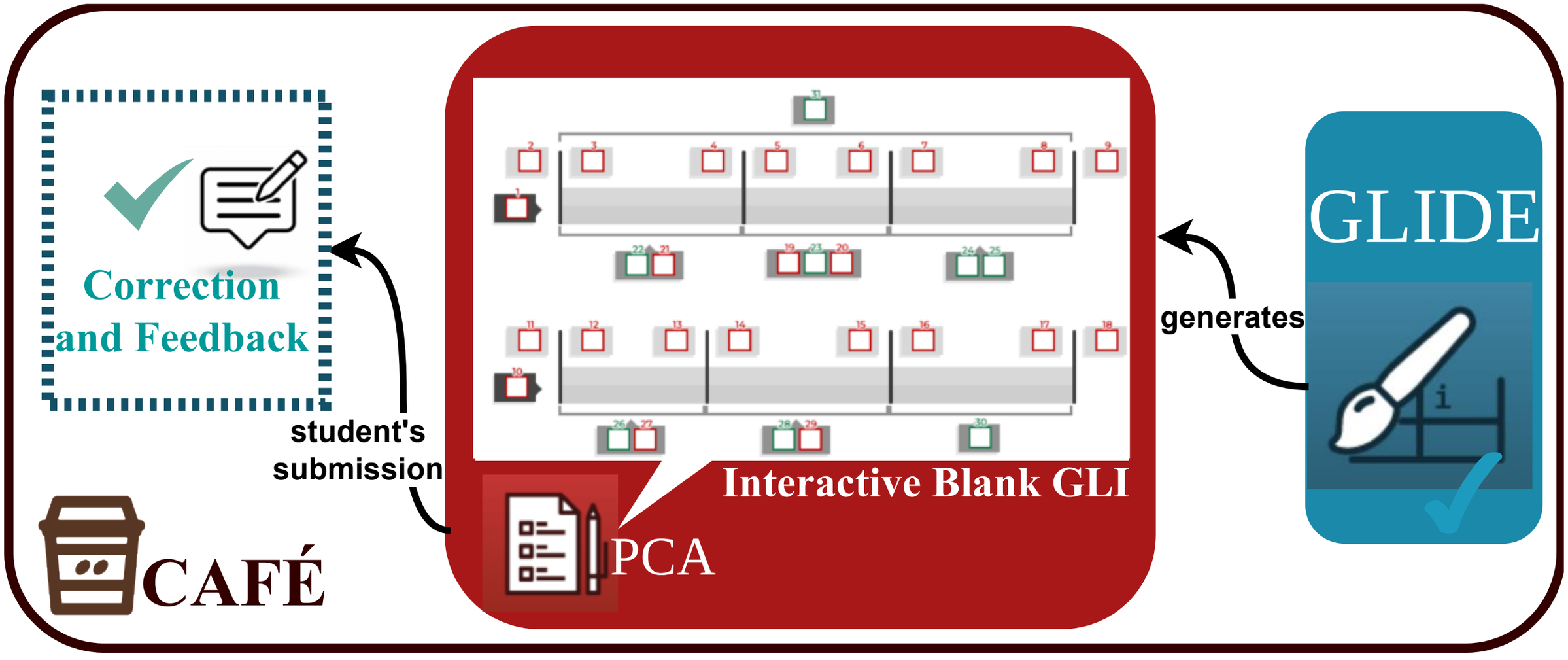
From in-loop state



Agenda

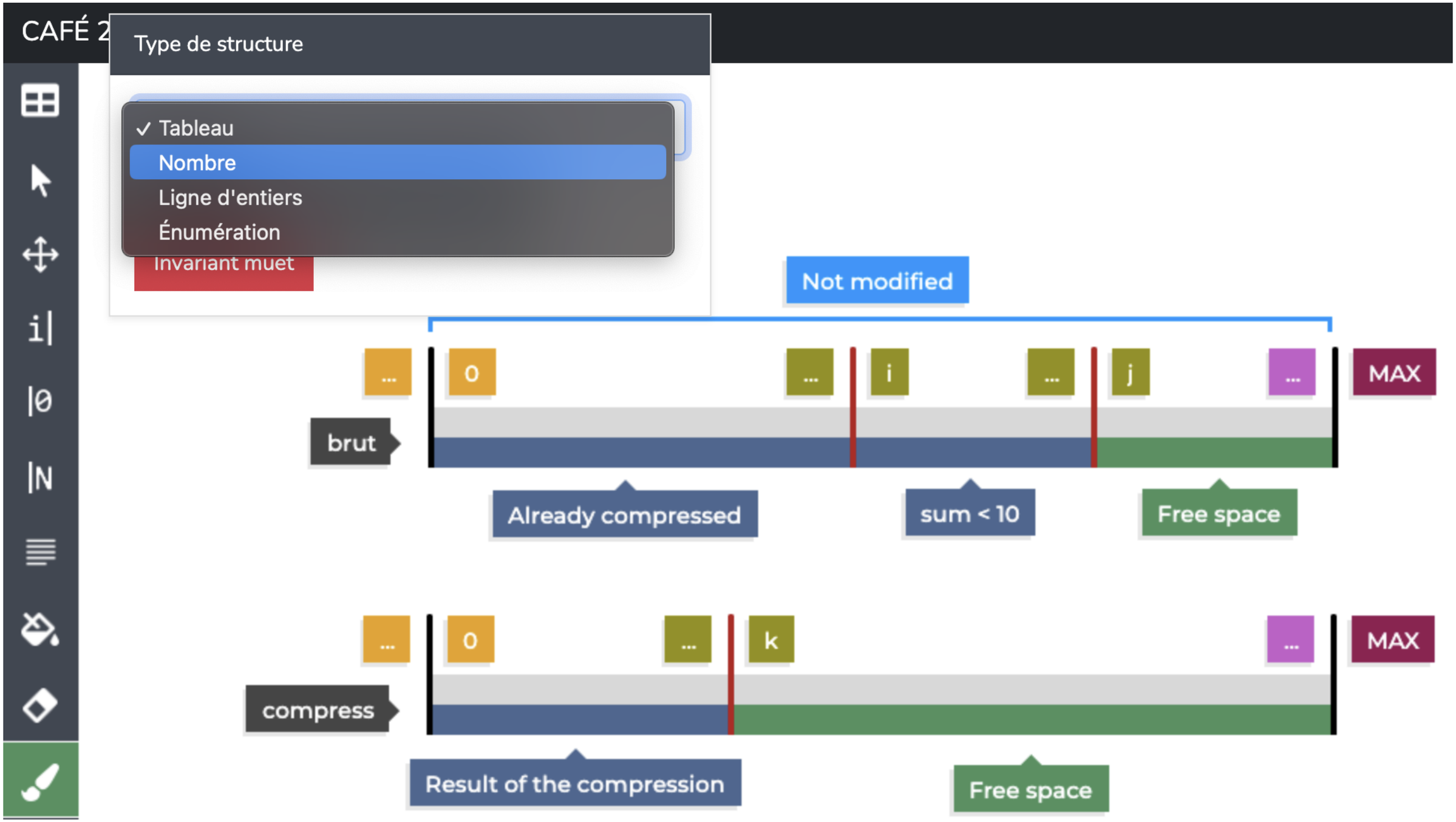

Not modified
From the GLI
to the code
Preliminary
Results
s_curr
Section 1
Section 6
Agenda:
Section 1
Section 2
Section 3
Section 4
Section 5
Graphical
Loop Invariant
Learning
Tools
Further work
and Conclusion
Legend :
Rule 1
Rule 2
Rule 3
Rule 4
Rule 5
Rule 6
Already
discussed
To be discussed

Learning Tools



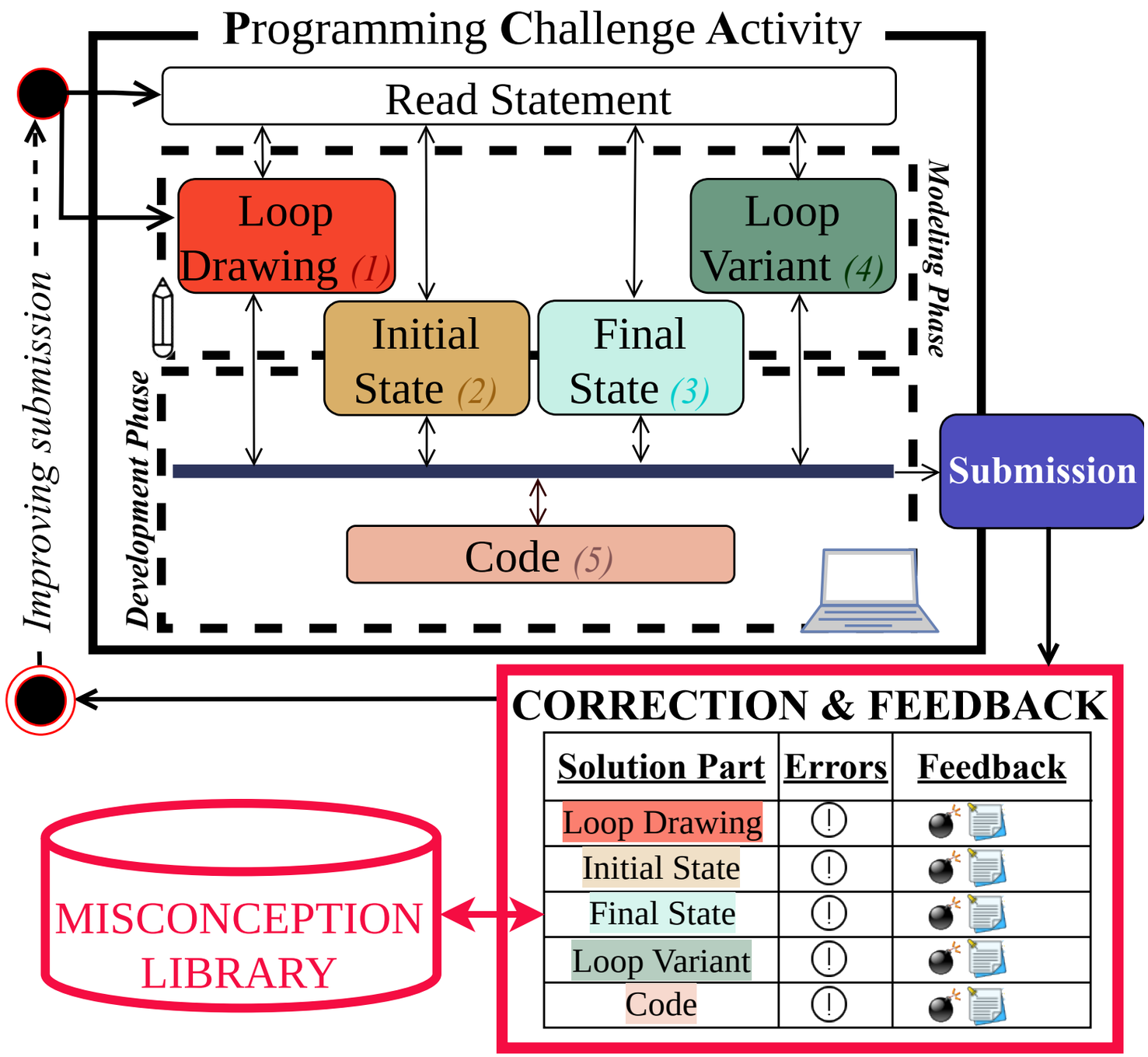
PCA = Programming Challenge Activity
GLIDE = Graphical Loop Invariant Drawing Editor
CAFÉ = Correction et Feedback Automatique pour les Étudiants

GLIDE (graphical editor)



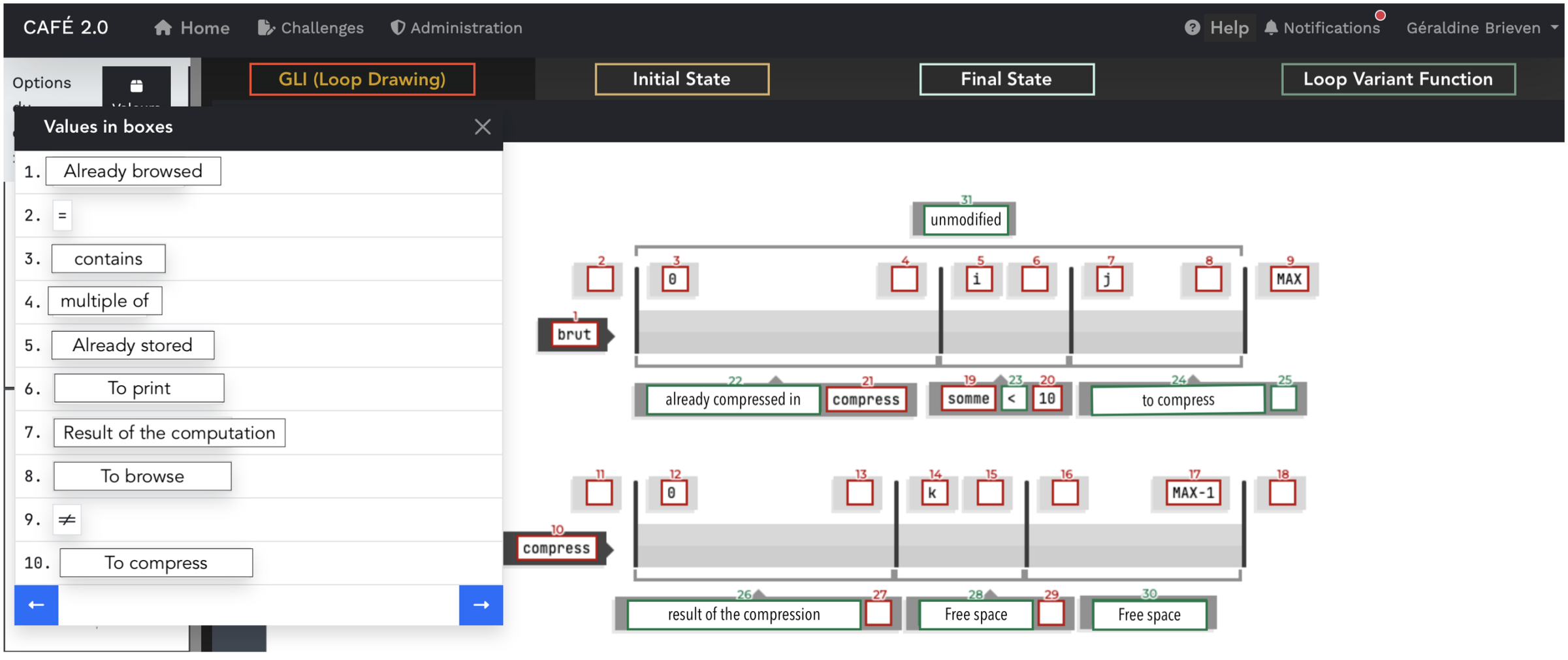
Programming Challenges Activity

Activity Diagram:



Programming Challenges Activity

Screenshot:



Agenda

Not modified

From the GLI to the code

Preliminary Results

s_curr

Section 6

Agenda:

Section 1 | Section 2 | Section 3 | Section 4 | Section 5

Graphical Loop Invariant

Learning Tools

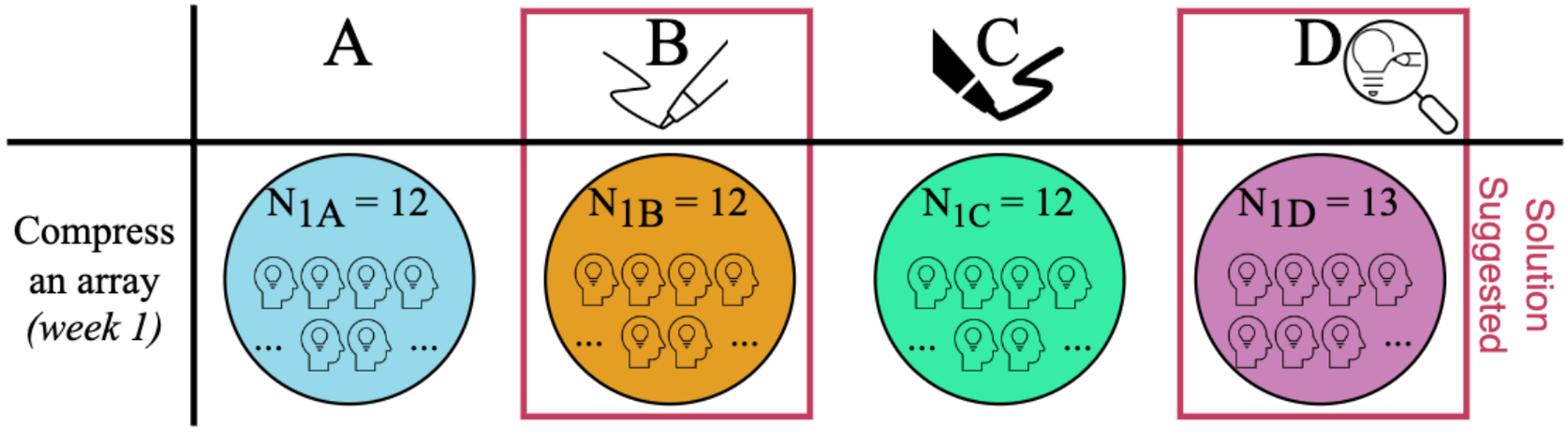
Further work and Conclusion

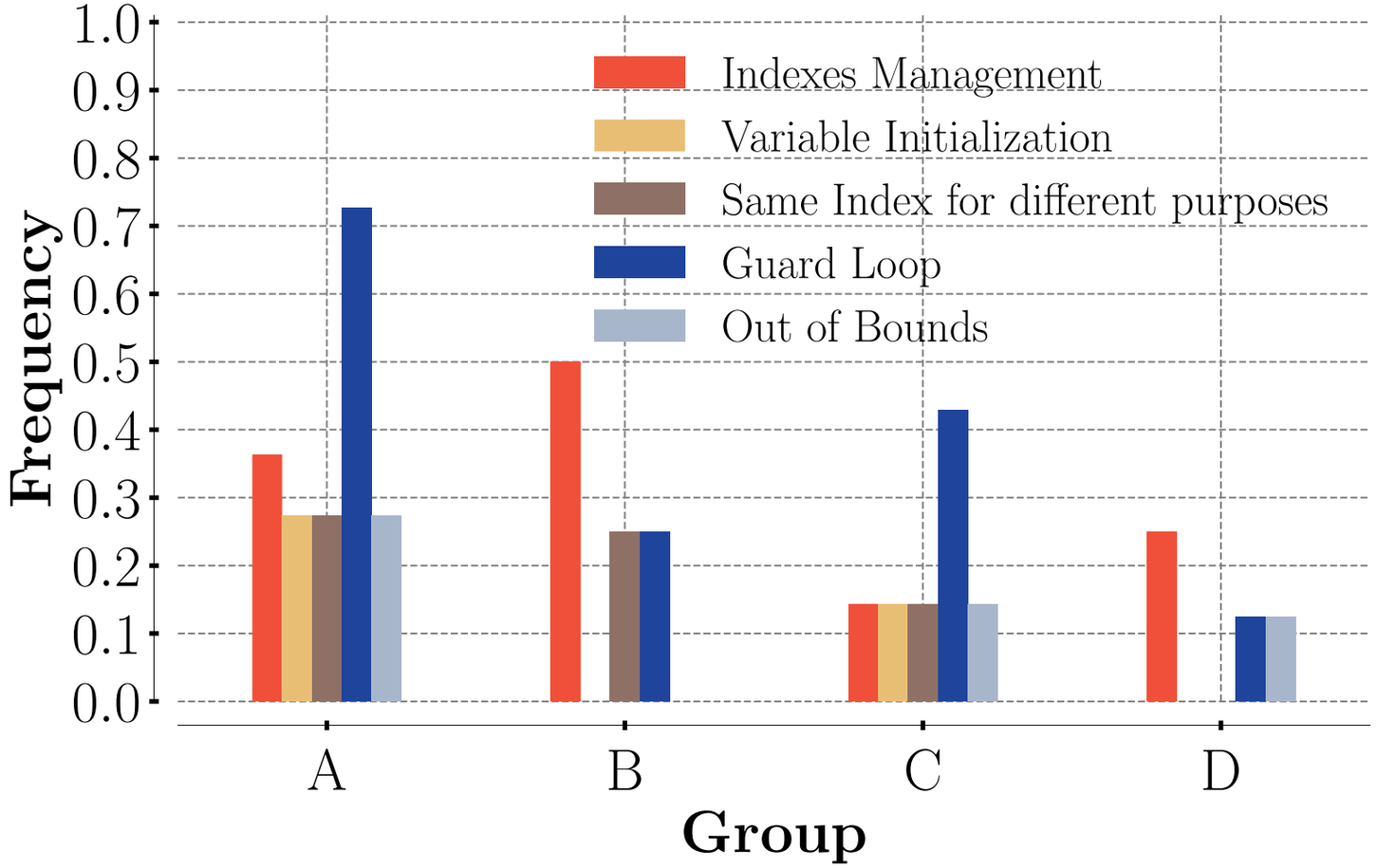
Already discussed

To be discussed

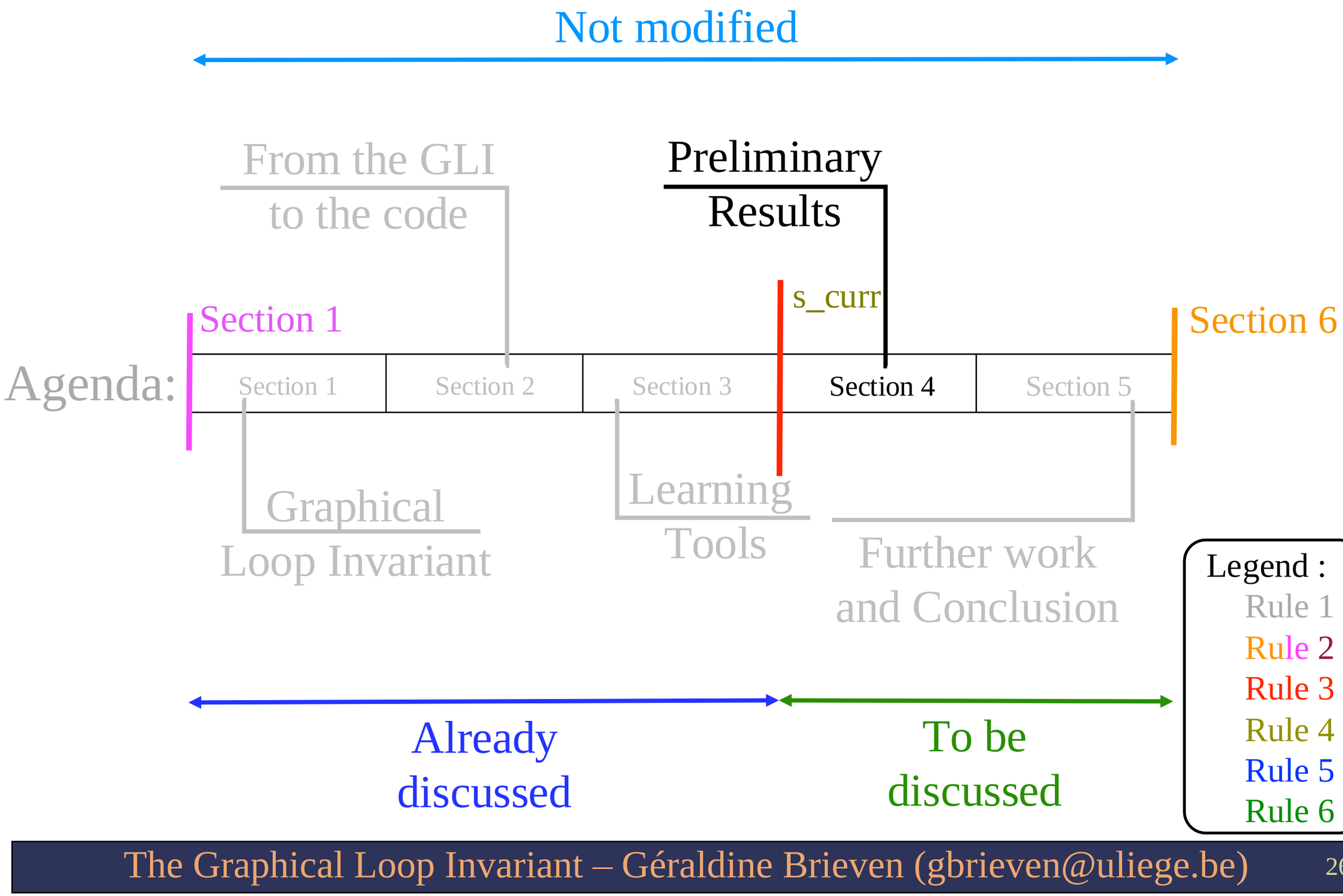
Legend :
- Rule 1
- Rule 2
- Rule 3
- Rule 4
- Rule 5
- Rule 6

Preliminary Results: Errors



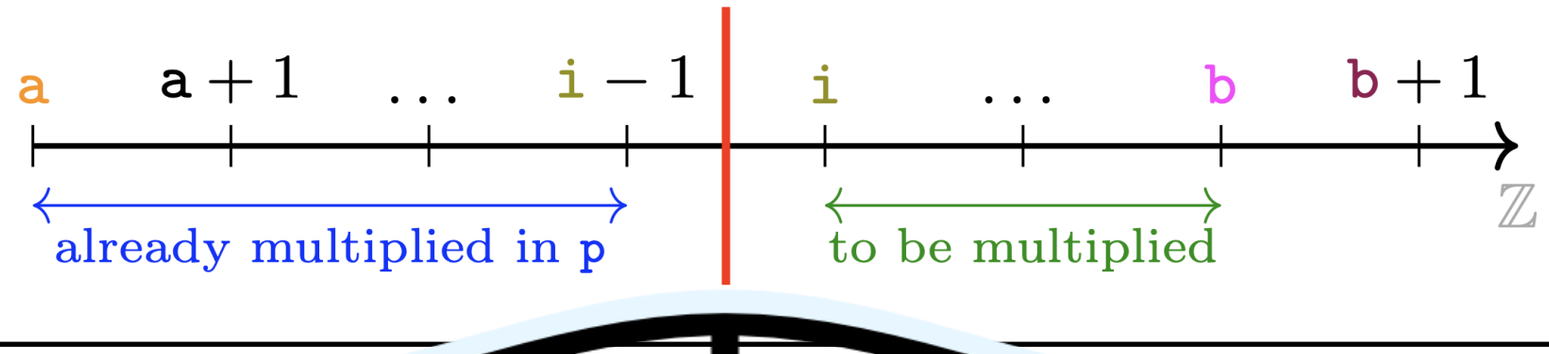


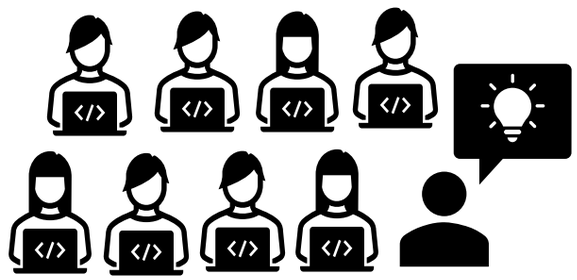
Agenda


Not modified
From the GLI to the code
Preliminary Results
s_curr
Section 1
Section 6
Agenda:
Section 1
Section 2
Section 3
Section 4
Section 5
Graphical Loop Invariant
Learning Tools
Further work and Conclusion
Legend :
Rule 1
Rule 2
Rule 3
Rule 4
Rule 5
Rule 6
Already discussed
To be discussed

Further Work & Conclusion

Graphical Loop Invariant (Informal)





First year students

$$a \leq i \leq b+1 \wedge p = \prod_{j=a}^{i-1} j$$

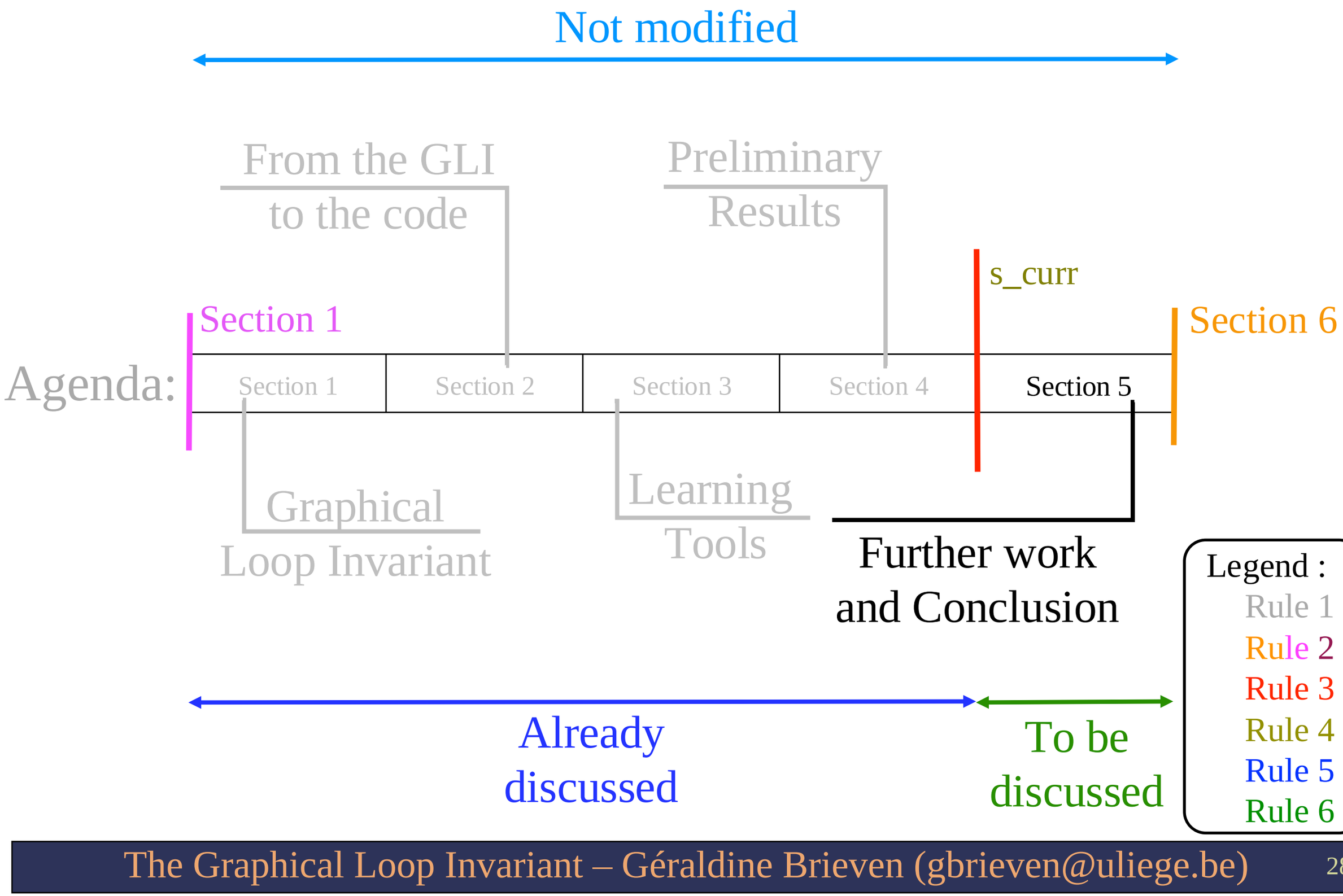
Loop Invariant (Formal)

? <u>On-going Research</u> ?

How students actually manage the GLIBP ?

How to smoothly bridge the GLIBP to the Formal Invariant ?

Agenda

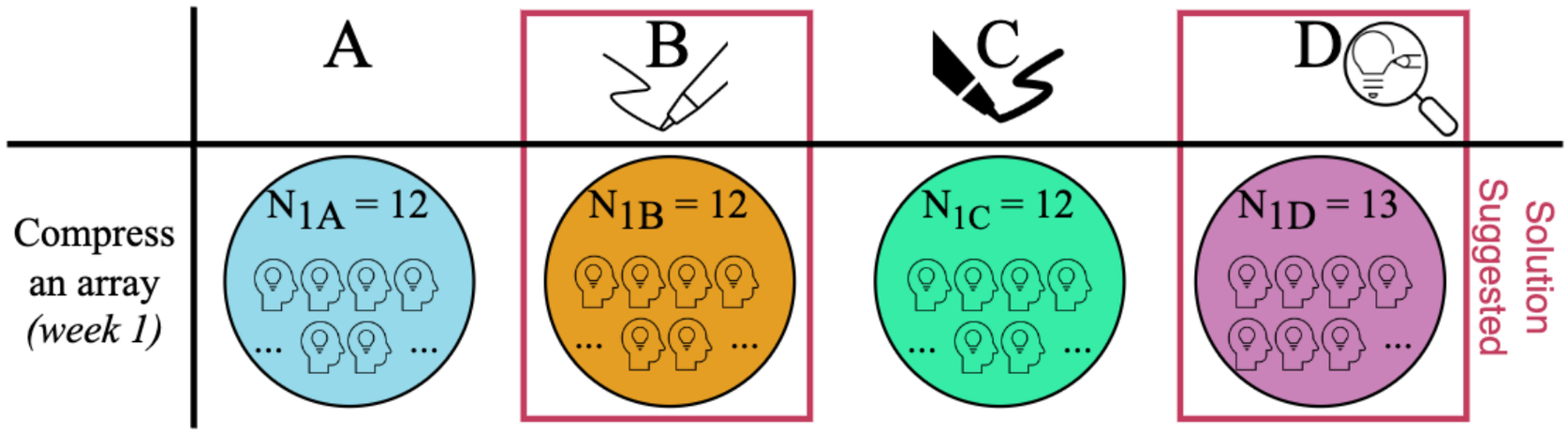

Not modified
From the GLI to the code
Preliminary Results
s_curr
Section 1
Section 6
Agenda:
Section 1
Section 2
Section 3
Section 4
Section 5
Graphical Loop Invariant
Learning Tools
Further work and Conclusion
Legend :
Rule 1
Rule 2
Rule 3
Rule 4
Rule 5
Rule 6
Already discussed
To be discussed

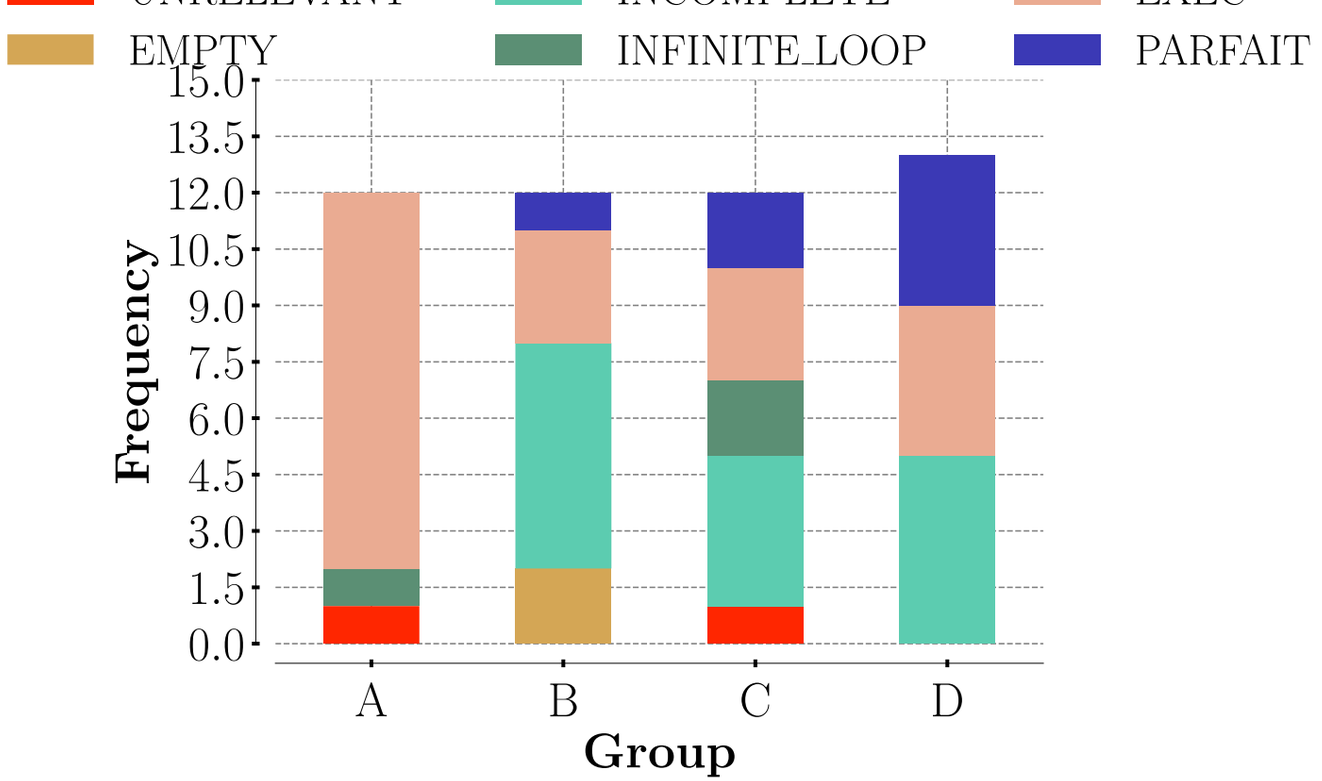
Our Research

Visit our research

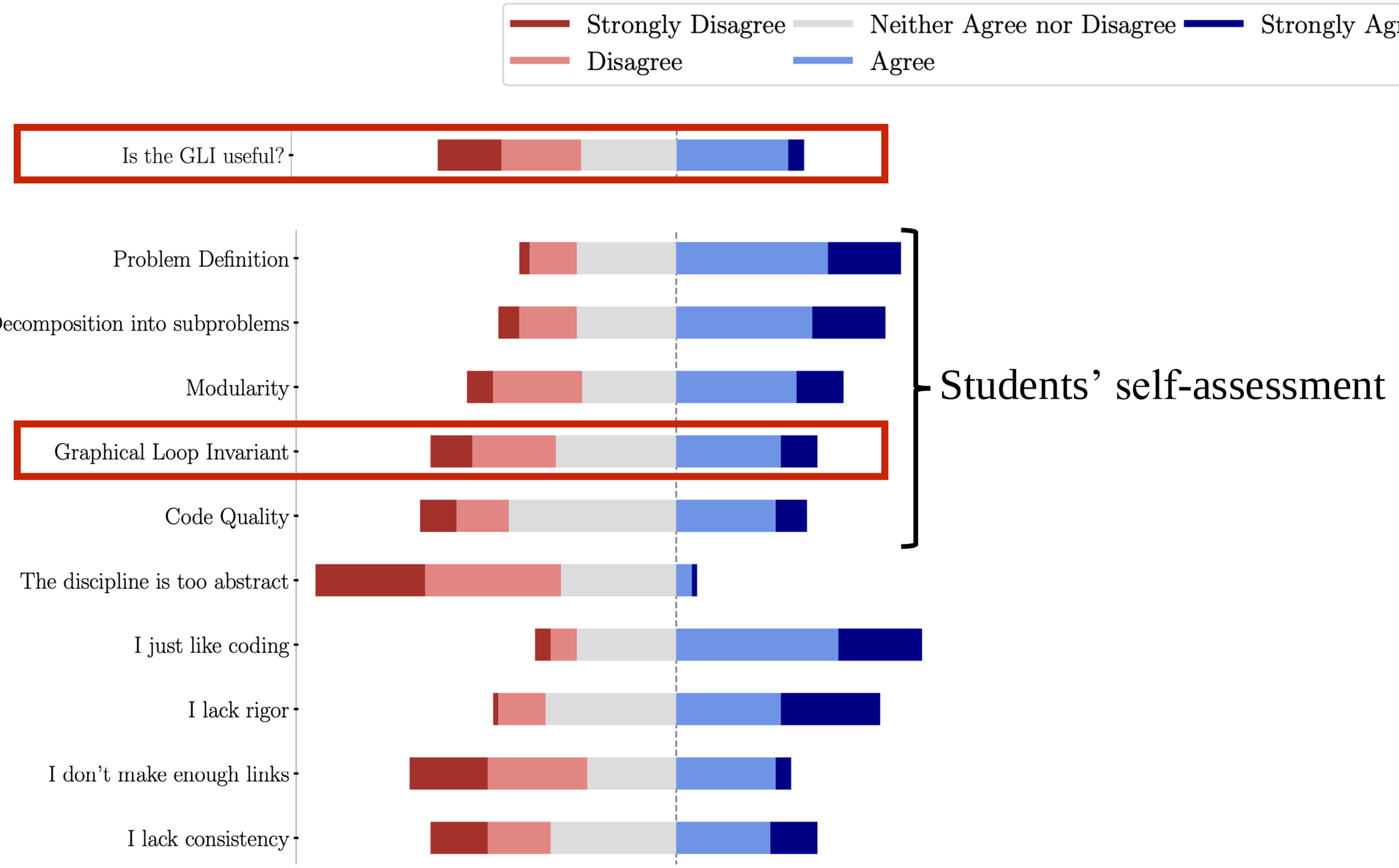
Contribute to our research
(survey for CS1 teachers)
(It takes 10min to fill in ☺)

Preliminary Results: General





Preliminary Results : Students Perception



Preliminary Results : Students performance