An efficient run-based Connected Component Labeling algorithm for processing holes

Florian Lemaitre, Nathan Maurice, Lionel Lacassagne

LIP6, Sorbonne University, CNRS, France

BNBW 2022

What are Connected Component **L**abeling and **A**nalysis ?

Connected Components **L**abeling (CC**L**) consists in assigning a unique number (label) to each connected component of a binary image to cluster pixels

Connected Components **A**nalysis (CC**A**) consists in computing some features associated to each connected component like the bounding box $[x_{min},x_{max}]$ x $[y_{min},y_{max}]$, the sum of pixels S, the sums of x and y coordinates Sx, Sy

gray level image

binary level image (segmentation by (motion detection)

connected component labeling

connected component analysis

- seems easy for a human being who has a global view of the image
- ill-posed problem: the computer has only a local view around a pixel (neighborhood)

Black & White Labeling and Adjacency Tree

Interest in labeling both foreground (FG) and background (BG) pixels (black & white)

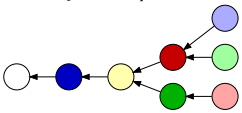
Adjacency tree shows which components are “inside” which component

- indicate the shape of the image
- compute Euler number from adjacency
- fill holes from adjacency

binary image

labeled image

adjacency tree



holes filled

Surrounding relation and Adjacency Tree

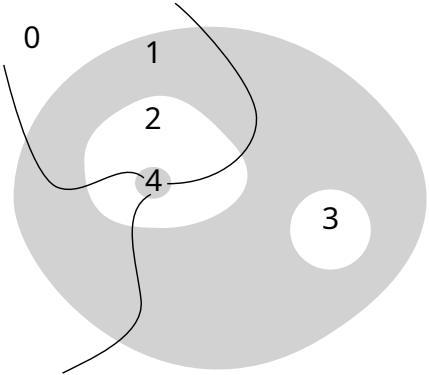
A component C_1 is surrounded by another component C_2 *iif* all paths from C_1 to the exterior ⓪ intersect C_2.

- noted: $C_1 \sqsubset C_2$

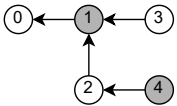
In this example:

- ① ⊏ ⓪
- ② ⊏ ⓪
- ② ⊏ ①
- ③ ⊏ ⓪
- ③ ⊏ ①
- ④ ⊏ ⓪
- ④ ⊏ ①
- ④ ⊏ ②

The adjacency tree encodes which components are "touching": there is an edge between two components *iif* they are adjacent. The edges are oriented using the surrounding relation.



Adjacency tree:

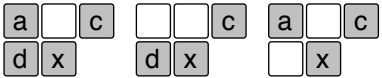


How to detect holes?

Closing pixel •

When a label is *merged* with itself, a hole has been closed

- 8-connex



$a \equiv c$ or $d \equiv c$

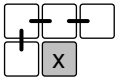
- 4-connex

$d \equiv b$

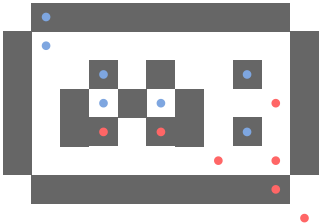
Initial adjacency •

If x is the first pixel of the component, its neighborhood is also its surrouding

- 8-connex



- 4-connex



Algorithm technicalities

Based on segment processing (and FLSL algorithm)

The foreground and background connectivity must be complementary:

- either 4-connex/8-connex, or **8-connex**/**4-connex** for FG/BG

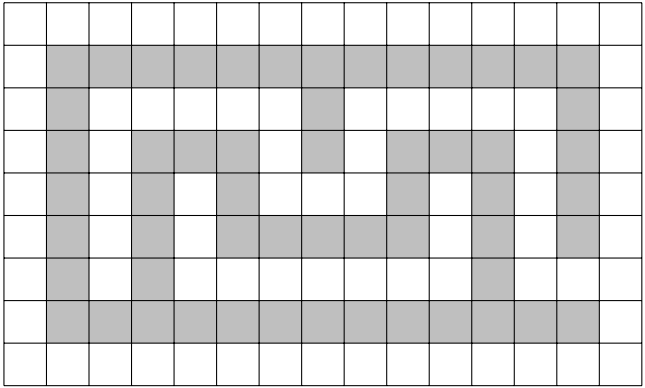
The total number of labels is higher than for FG only

- number of pixels is a practical upper bound

In addition to the equivalence table T, the algorithm also requires a feature table F and the *Initial Adjacency* table I

Hole filling is done by transforming an adjacency edge into an equivalence edge during the transitive closure

Example: Init



label	T	I	F_S
0	0	-1	0
1	N/A	N/A	N/A
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

⟵ Adjacency

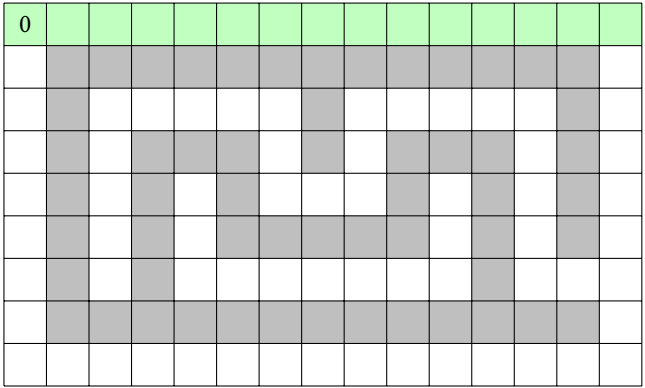
⟵ Adjacency discarded

◁⟵ BG equivalence

◀⟵ FG equivalence

⓪

Example: First scan



label	T	I	F_S
0	0	-1	15
1	N/A	N/A	N/A
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

⓪

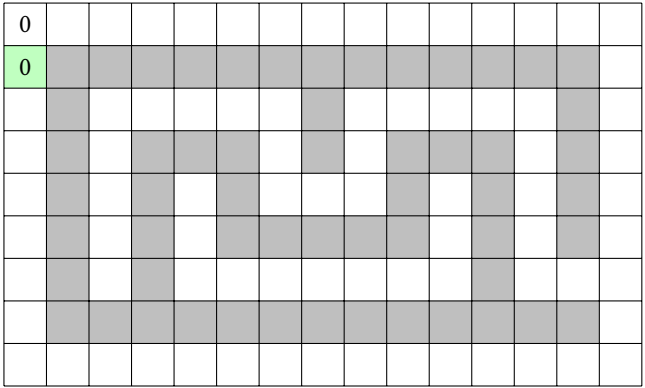
⟵ Adjacency

⟵ Adjacency discarded

◁⟵ BG equivalence

◀⟵ FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	16
1	N/A	N/A	N/A
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

⟵ Adjacency

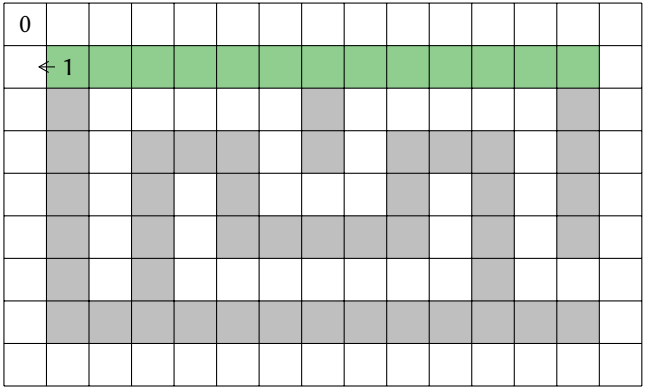
⟵ Adjacency discarded

◁⟵ BG equivalence

⟵● FG equivalence

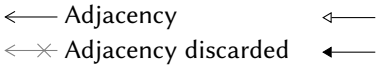
⓪

Example: First scan

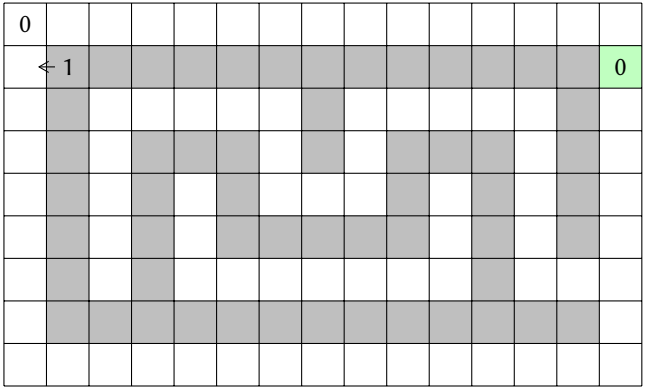


label	T	I	F_S
0	0	-1	16
1	1	0	13
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

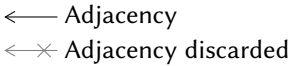


Example: First scan

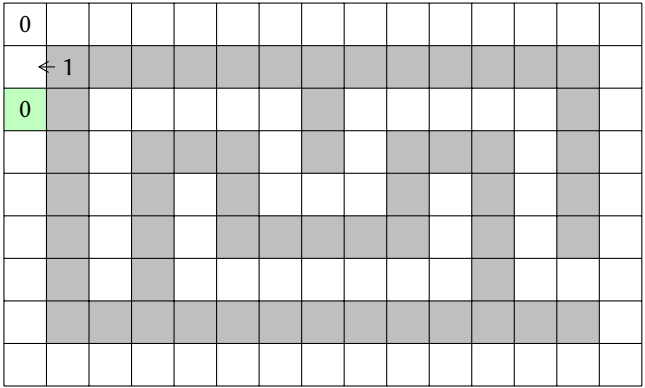


label	T	I	F_S
0	0	-1	17
1	1	0	13
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

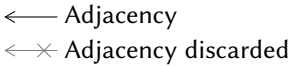


Example: First scan

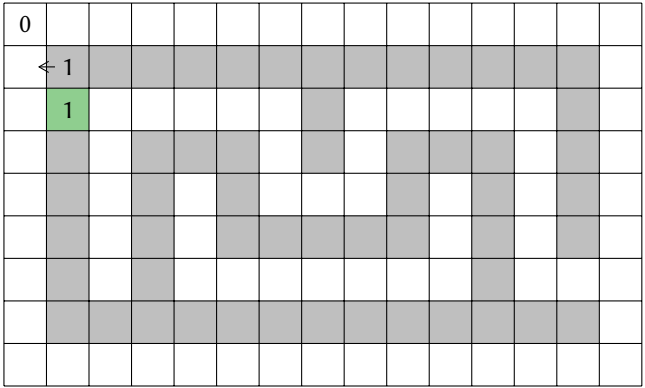


label	T	I	F_S
0	0	-1	18
1	1	0	13
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

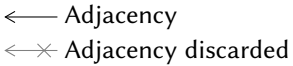


Example: First scan

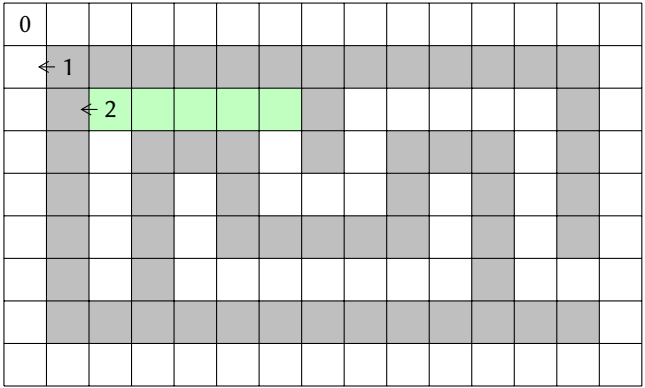


label	T	I	F_S
0	0	-1	18
1	1	0	14
2	N/A	N/A	N/A
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Example: First scan

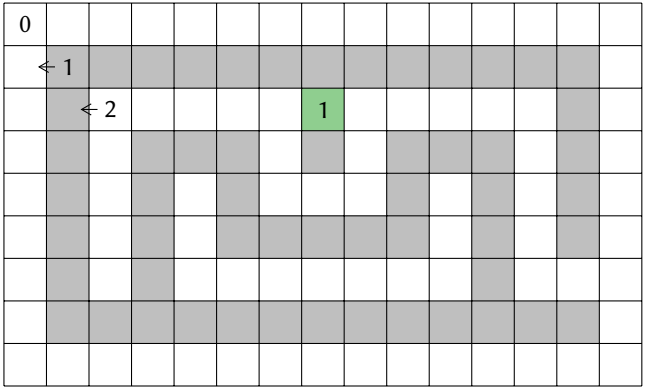


label	T	I	F_S
0	0	-1	18
1	1	0	14
2	2	1	5
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	18
1	1	0	15
2	2	1	5
3	N/A	N/A	N/A
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

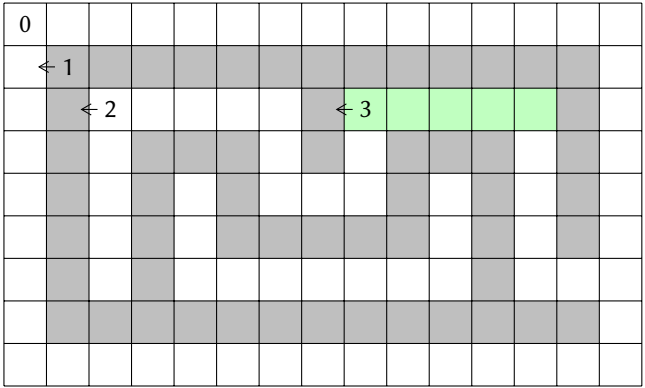
Adjacency

Adjacency discarded

BG equivalence

FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	18
1	1	0	15
2	2	1	5
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

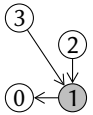
Adjacency

Adjacency discarded

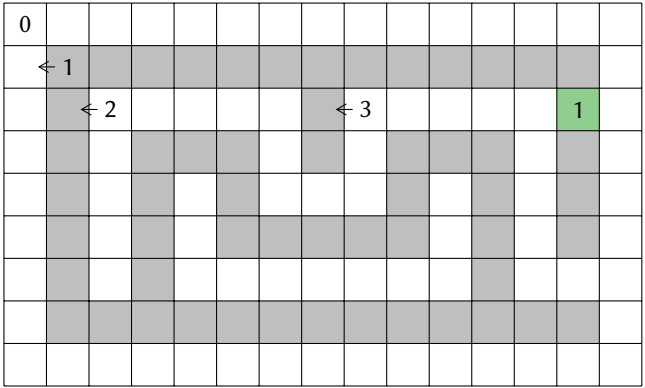
BG equivalence

FG equivalence

Equivalence & Adjacency tree

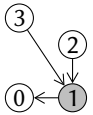


Example: First scan



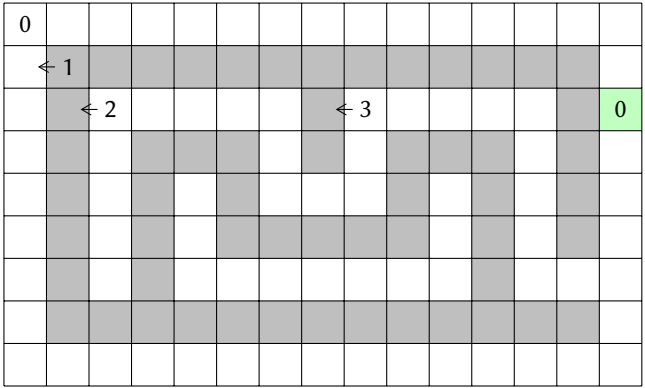
label	T	I	F_S
0	0	-1	18
1	1	0	16
2	2	1	5
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

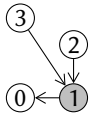
Example: First scan



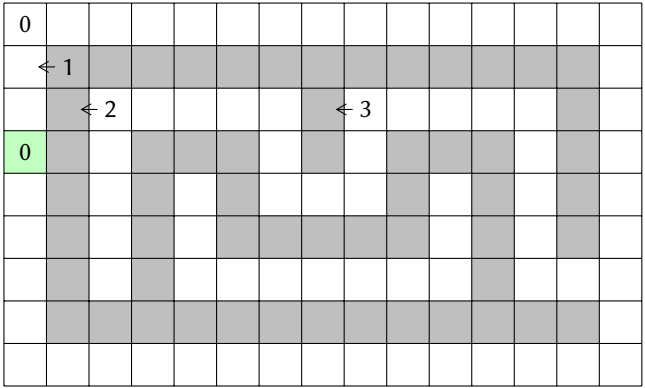
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	19
1	1	0	16
2	2	1	5
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

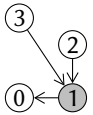


Example: First scan



label	T	I	F_S
0	0	-1	20
1	1	0	16
2	2	1	5
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



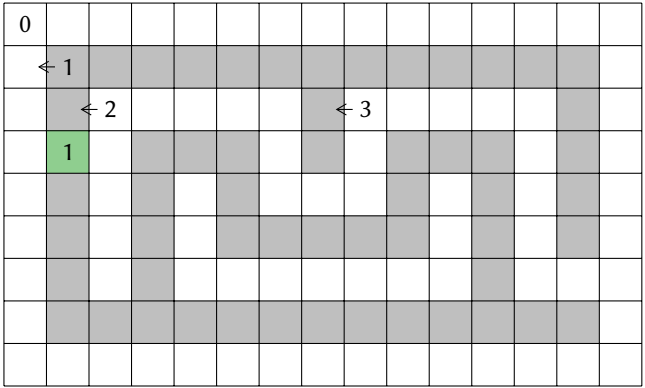
Adjacency

Adjacency discarded

BG equivalence

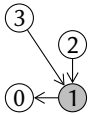
FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	20
1	1	0	17
2	2	1	5
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



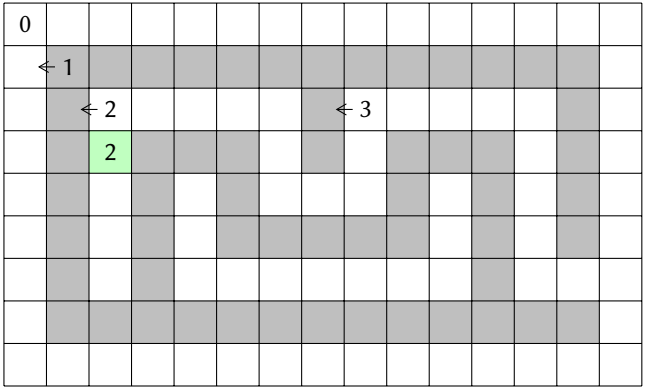
Adjacency

Adjacency discarded

BG equivalence

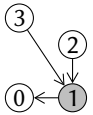
FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	20
1	1	0	17
2	2	1	6
3	3	1	5
4	N/A	N/A	N/A
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



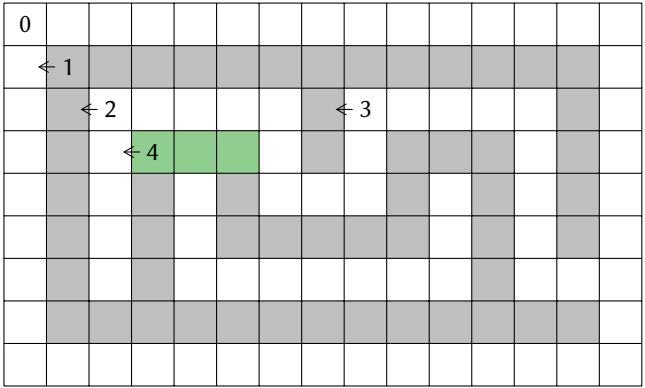
Adjacency

Adjacency discarded

BG equivalence

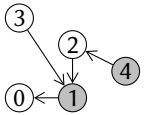
FG equivalence

Example: First scan



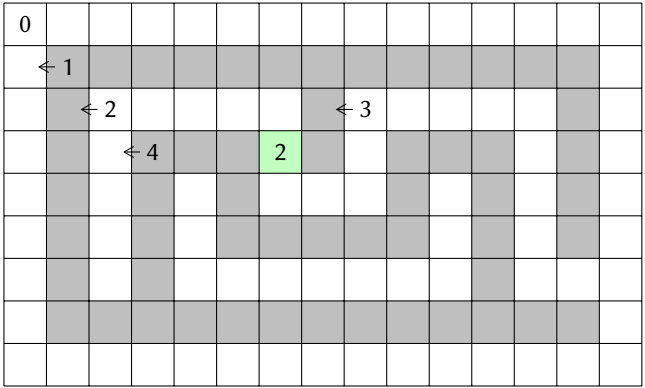
label	T	I	F_S
0	0	-1	20
1	1	0	17
2	2	1	6
3	3	1	5
4	4	2	3
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

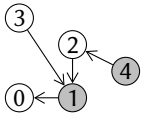
Example: First scan



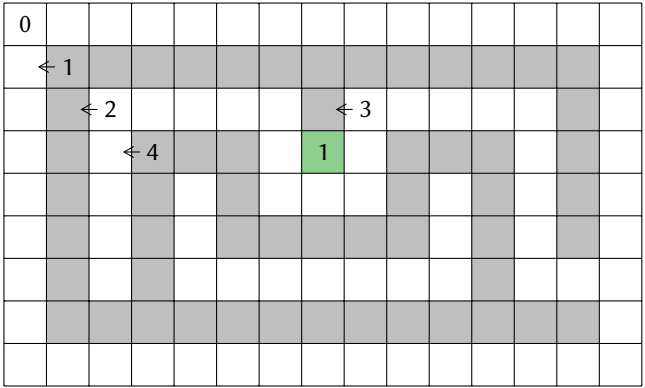
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	20
1	1	0	17
2	2	1	7
3	3	1	5
4	4	2	3
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

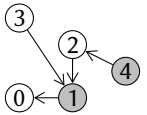


Example: First scan



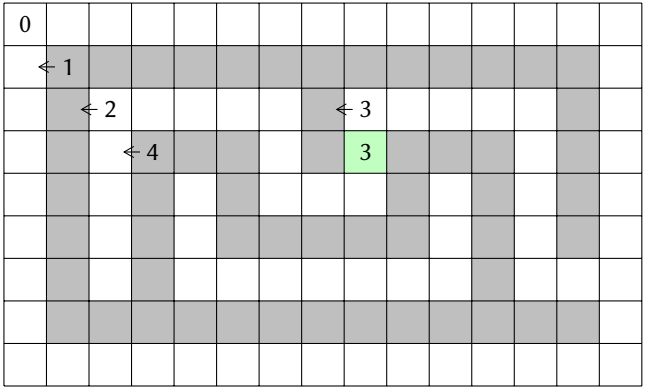
label	T	I	F_S
0	0	-1	20
1	1	0	18
2	2	1	7
3	3	1	5
4	4	2	3
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



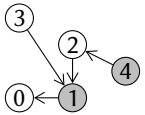
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Example: First scan



label	T	I	F_S
0	0	-1	20
1	1	0	18
2	2	1	7
3	3	1	6
4	4	2	3
5	N/A	N/A	N/A
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



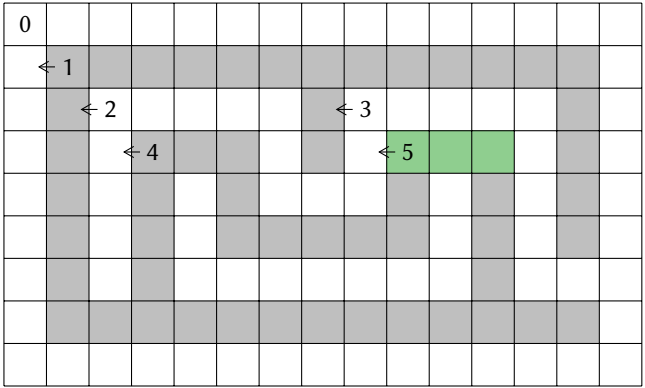
Adjacency

Adjacency discarded

BG equivalence

FG equivalence

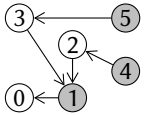
Example: First scan



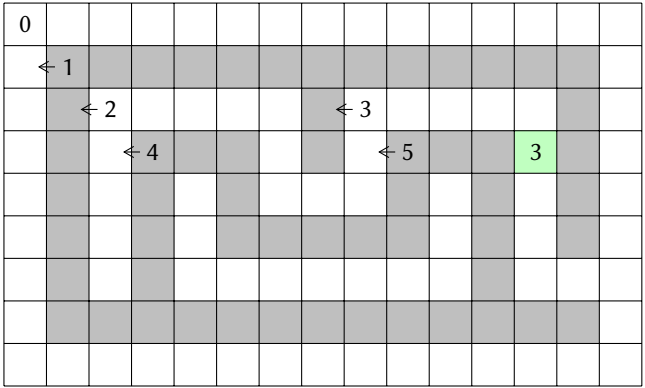
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	20
1	1	0	18
2	2	1	7
3	3	1	6
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree

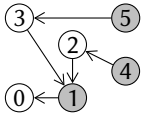


Example: First scan



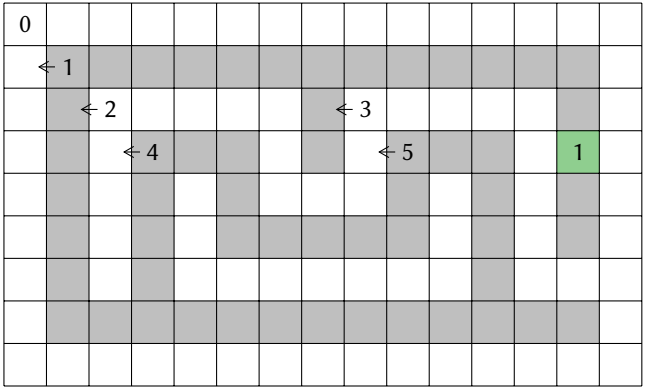
label	T	I	F_S
0	0	-1	20
1	1	0	18
2	2	1	7
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



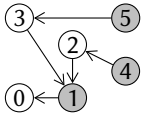
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Example: First scan



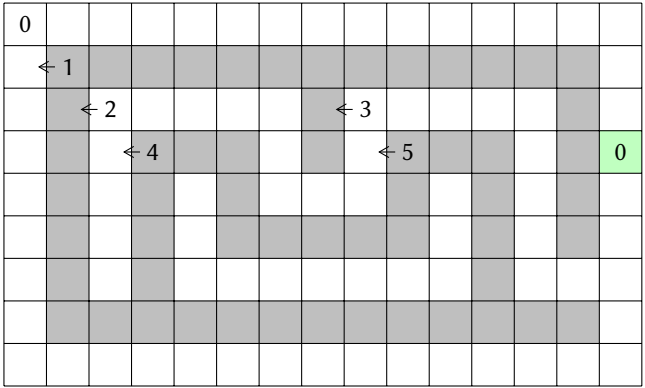
label	T	I	F_S
0	0	-1	20
1	1	0	19
2	2	1	7
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

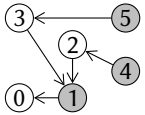
Example: First scan



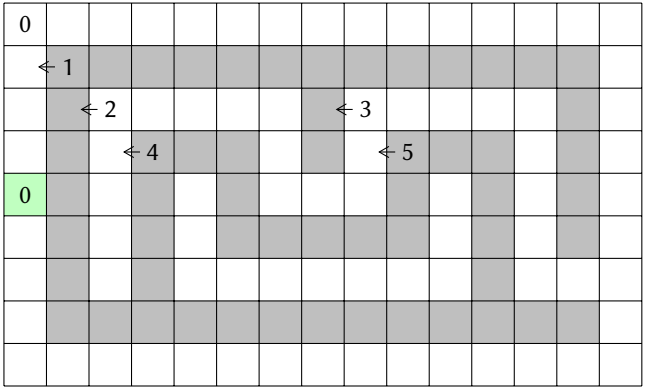
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	21
1	1	0	19
2	2	1	7
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



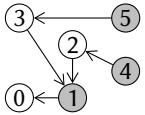
Example: First scan



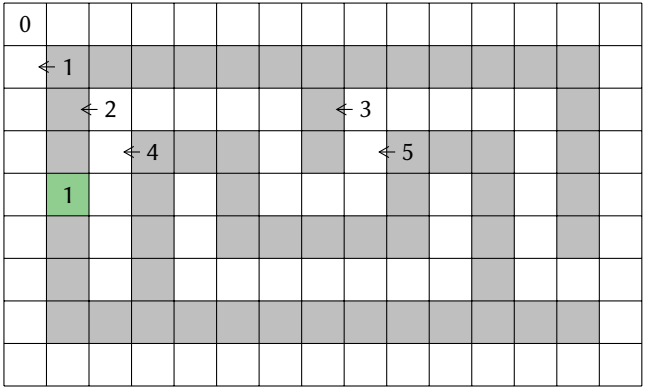
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	19
2	2	1	7
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Example: First scan



label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	7
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

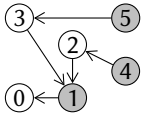
Adjacency

Adjacency discarded

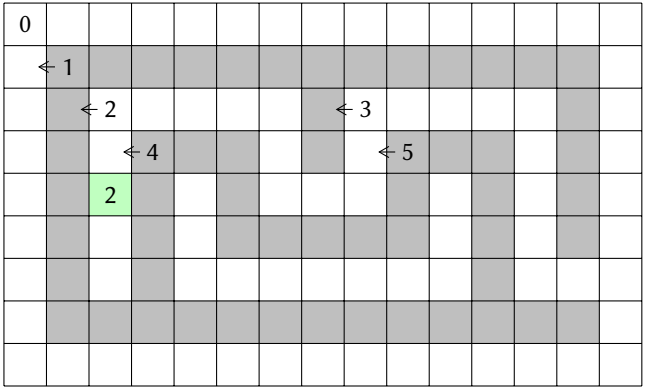
BG equivalence

FG equivalence

Equivalence & Adjacency tree

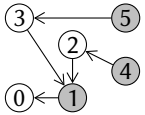


Example: First scan



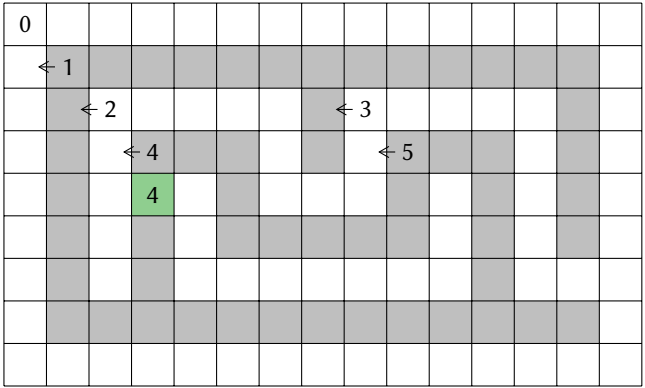
label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	8
3	3	1	7
4	4	2	3
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

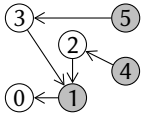
Example: First scan



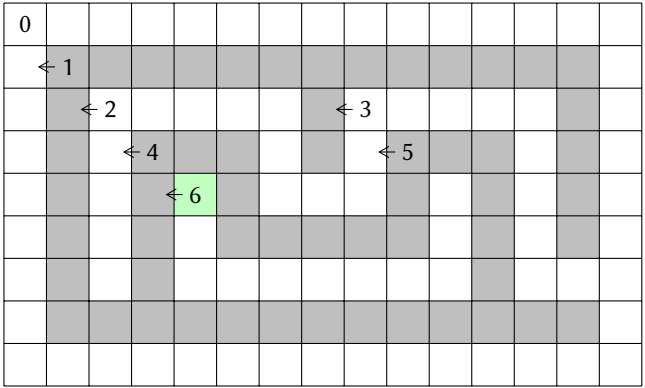
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	8
3	3	1	7
4	4	2	4
5	5	3	3
6	N/A	N/A	N/A
7	N/A	N/A	N/A

Equivalence & Adjacency tree



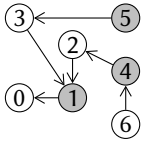
Example: First scan



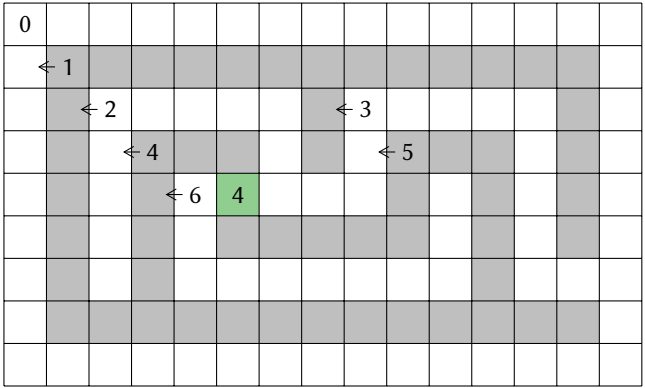
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	8
3	3	1	7
4	4	2	4
5	5	3	3
6	6	4	1
7	N/A	N/A	N/A

Equivalence & Adjacency tree

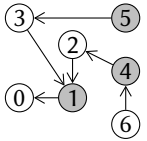


Example: First scan



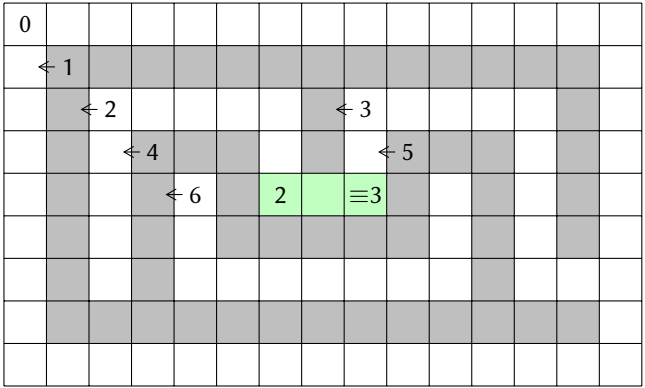
label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	8
3	3	1	7
4	4	2	5
5	5	3	3
6	6	4	1
7	N/A	N/A	N/A

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

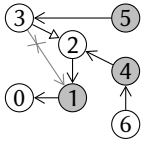
Example: First scan



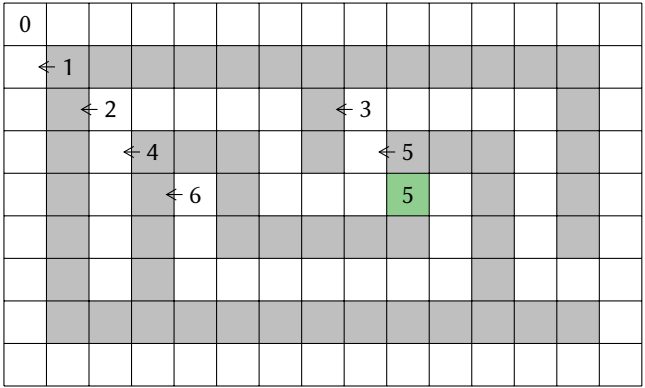
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	11
3	2	~~1~~	7
4	4	2	5
5	5	3	3
6	6	4	1
7	N/A	N/A	N/A

Equivalence & Adjacency tree

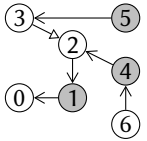


Example: First scan



label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	11
3	2	~~1~~	7
4	4	2	5
5	5	3	4
6	6	4	1
7	N/A	N/A	N/A

Equivalence & Adjacency tree



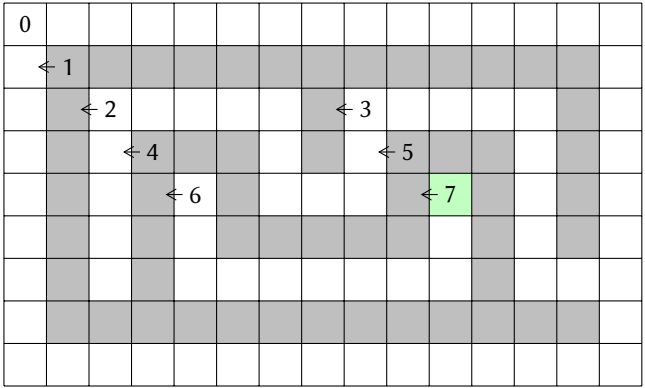
Adjacency

Adjacency discarded

BG equivalence

FG equivalence

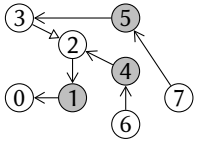
Example: First scan



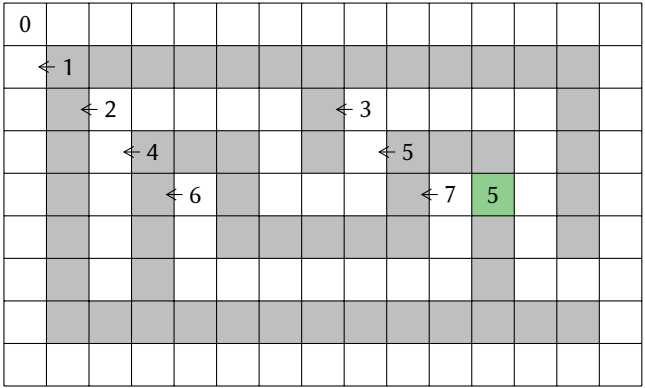
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	11
3	2	✳	7
4	4	2	5
5	5	3	4
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree

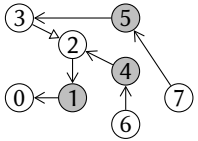


Example: First scan



label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	11
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



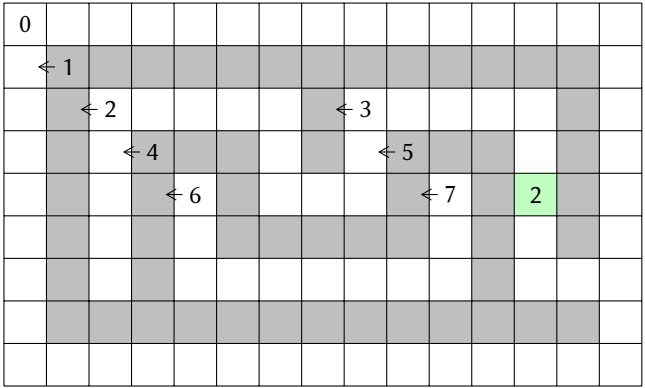
Adjacency

Adjacency discarded

BG equivalence

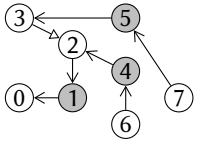
FG equivalence

Example: First scan



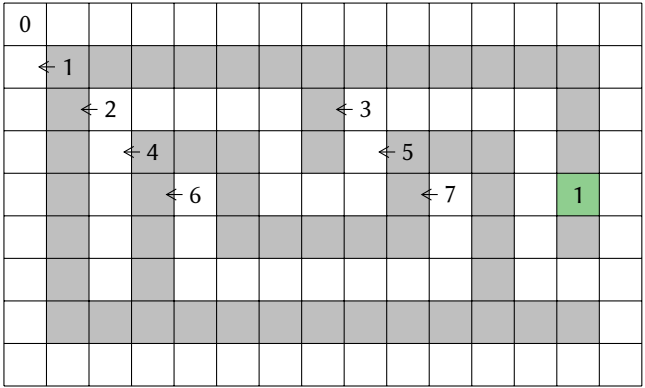
label	T	I	F_S
0	0	-1	22
1	1	0	20
2	2	1	12
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

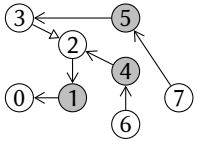
Example: First scan



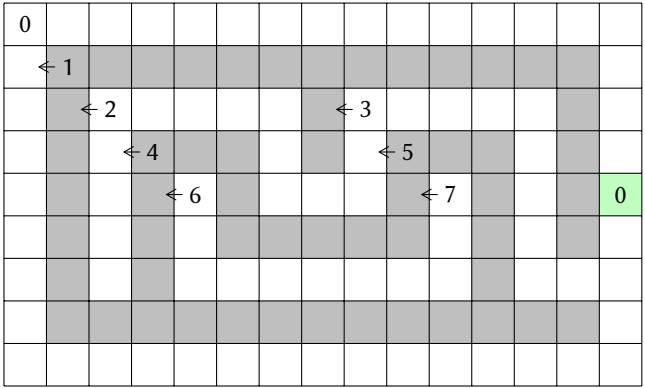
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	22
1	1	0	21
2	2	1	12
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



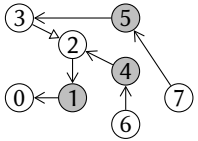
Example: First scan



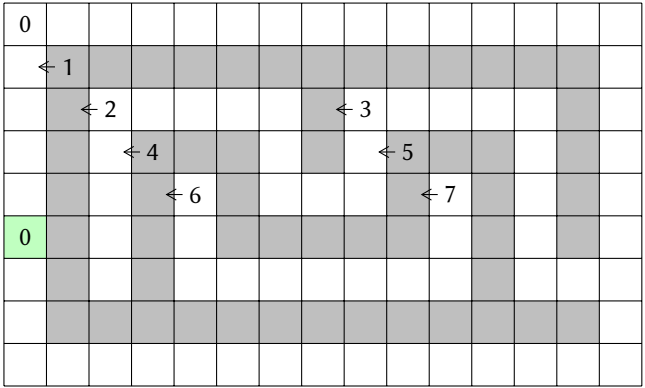
label	T	I	F_S
0	0	-1	23
1	1	0	21
2	2	1	12
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



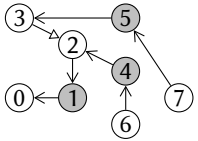
Example: First scan



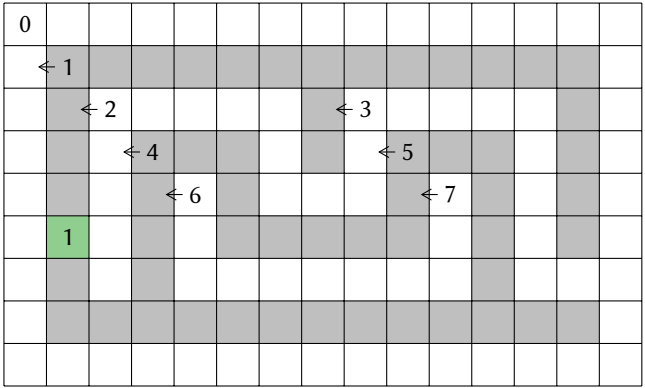
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	21
2	2	1	12
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



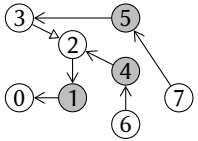
Example: First scan



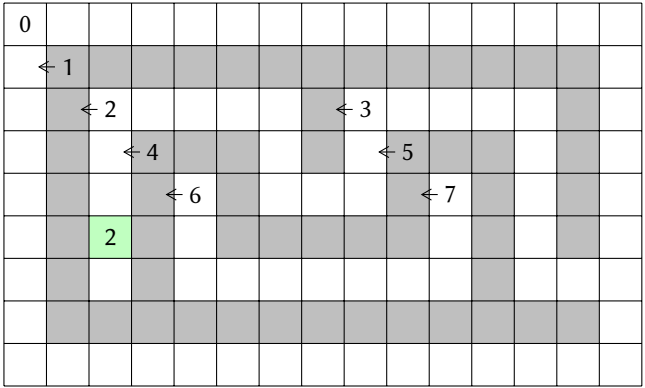
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	12
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



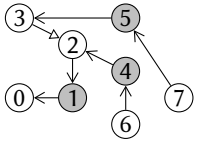
Example: First scan



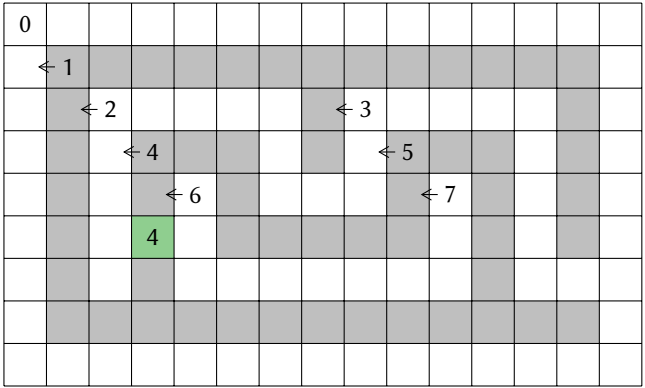
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	✳	7
4	4	2	5
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



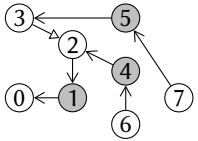
Example: First scan



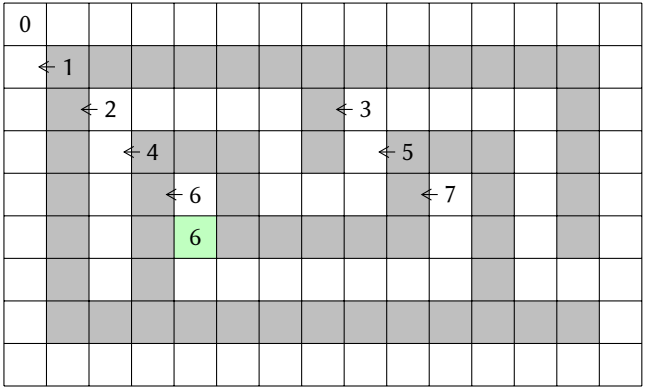
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	✳	7
4	4	2	6
5	5	3	5
6	6	4	1
7	7	5	1

Equivalence & Adjacency tree



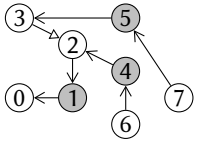
Example: First scan



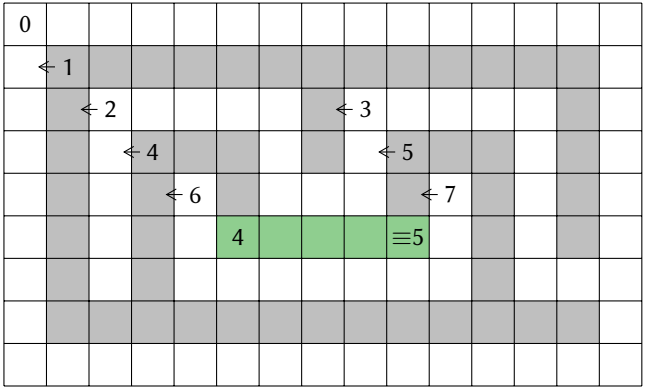
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	✳	7
4	4	2	6
5	5	3	5
6	6	4	2
7	7	5	1

Equivalence & Adjacency tree

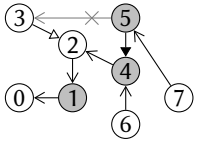


Example: First scan



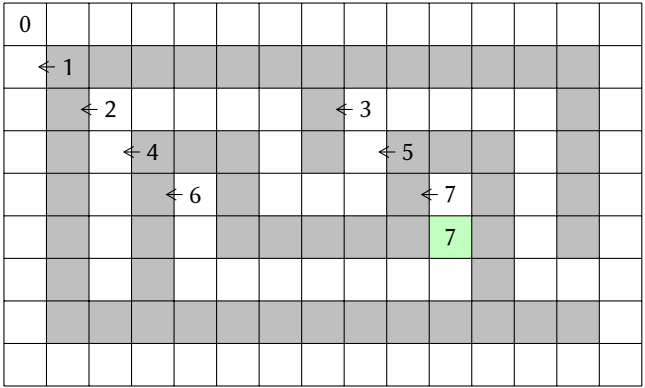
label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	~~1~~	7
4	4	2	11
5	4	~~2~~	5
6	6	4	2
7	7	5	1

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

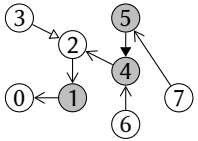
Example: First scan



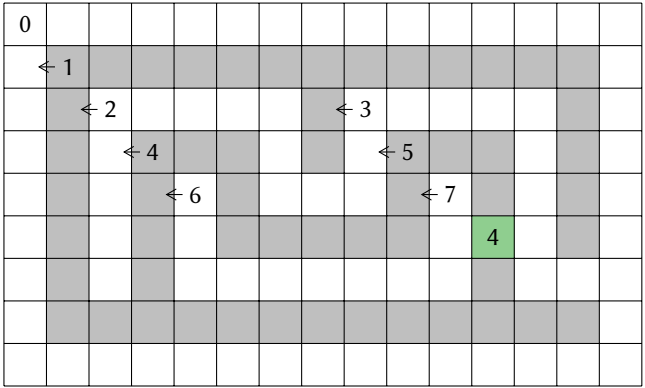
label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	~~1~~	7
4	4	2	11
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



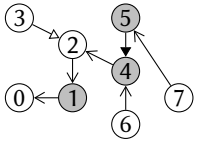
Example: First scan



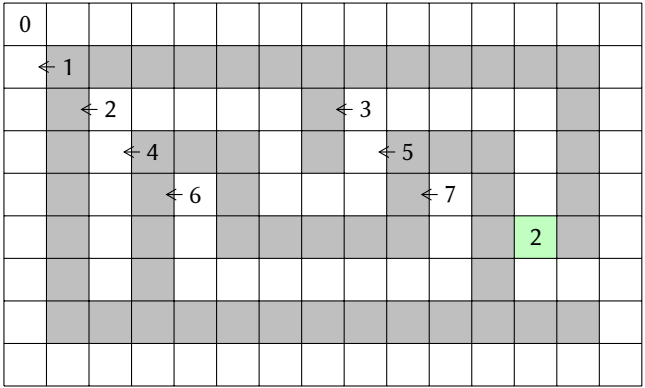
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	13
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Equivalence & Adjacency tree



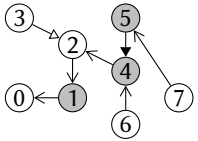
Example: First scan



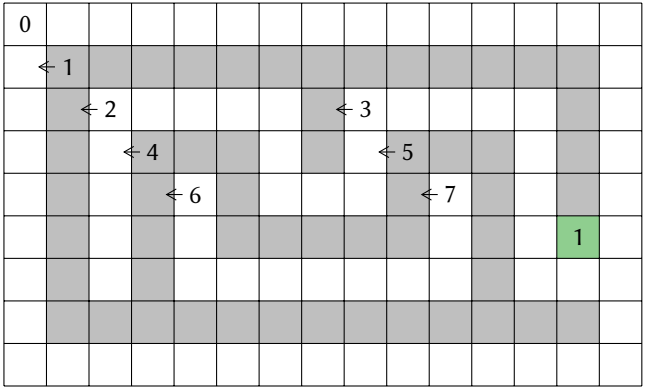
label	T	I	F_S
0	0	-1	24
1	1	0	22
2	2	1	14
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



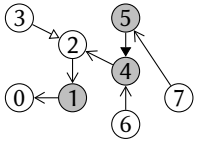
Example: First scan



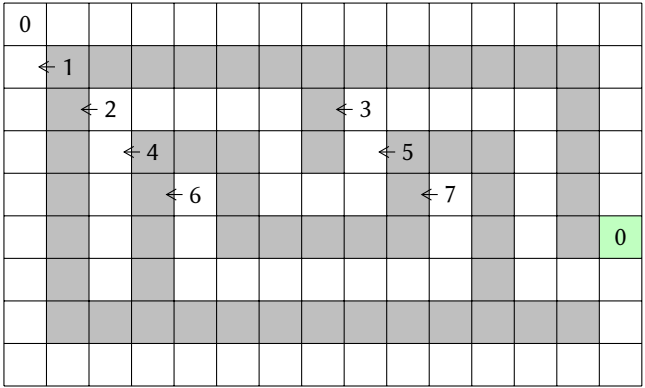
label	T	I	F_S
0	0	-1	24
1	1	0	23
2	2	1	14
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree

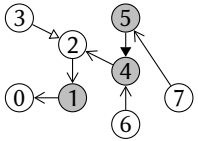


Example: First scan



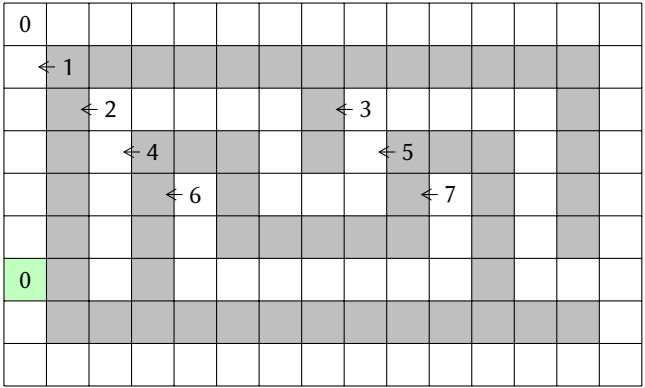
label	T	I	F_S
0	0	-1	25
1	1	0	23
2	2	1	14
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

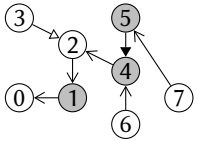
Example: First scan



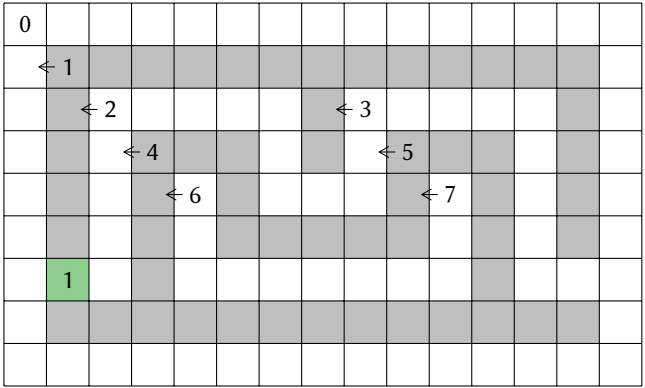
label	T	I	F_S
0	0	-1	26
1	1	0	23
2	2	1	14
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



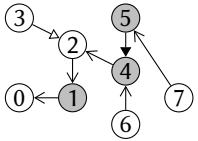
Example: First scan



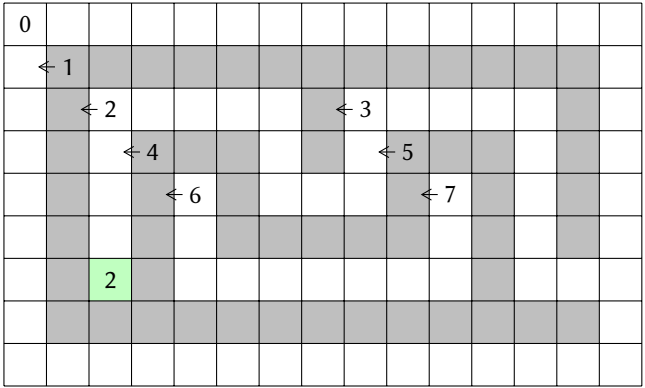
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	26
1	1	0	24
2	2	1	14
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Equivalence & Adjacency tree



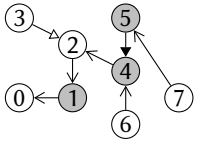
Example: First scan



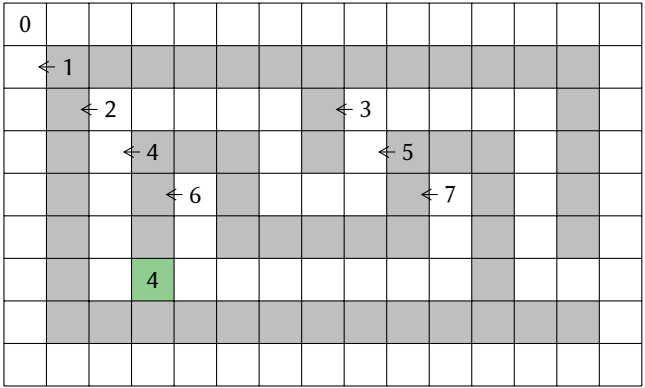
label	T	I	F_S
0	0	-1	26
1	1	0	24
2	2	1	15
3	2	~~1~~	7
4	4	2	12
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



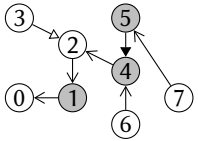
Example: First scan



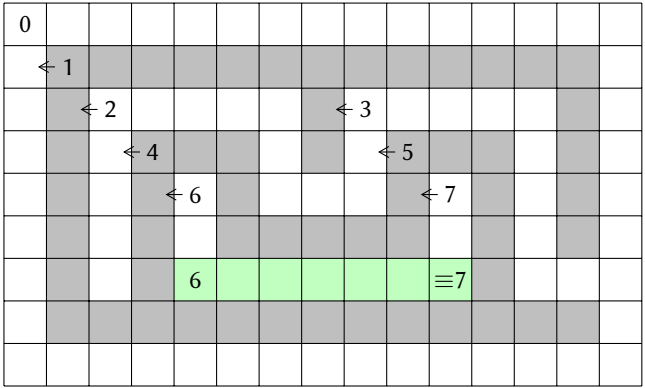
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	26
1	1	0	24
2	2	1	15
3	2	~~1~~	7
4	4	2	13
5	4	~~2~~	5
6	6	4	2
7	7	5	2

Equivalence & Adjacency tree



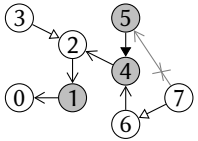
Example: First scan



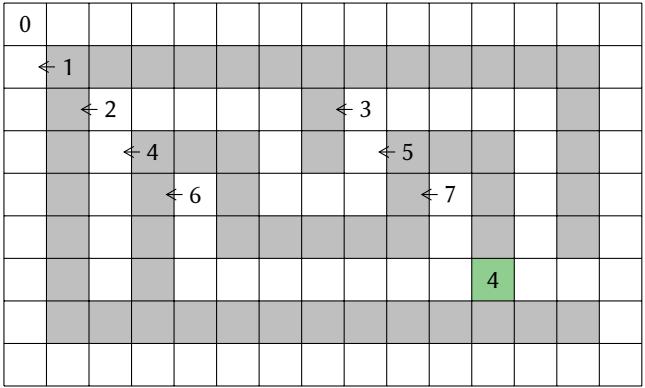
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	26
1	1	0	24
2	2	1	15
3	2	~~1~~	7
4	4	2	13
5	4	~~2~~	5
6	6	4	9
7	6	~~4~~	2

Equivalence & Adjacency tree



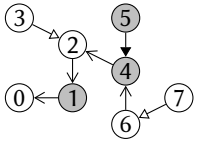
Example: First scan



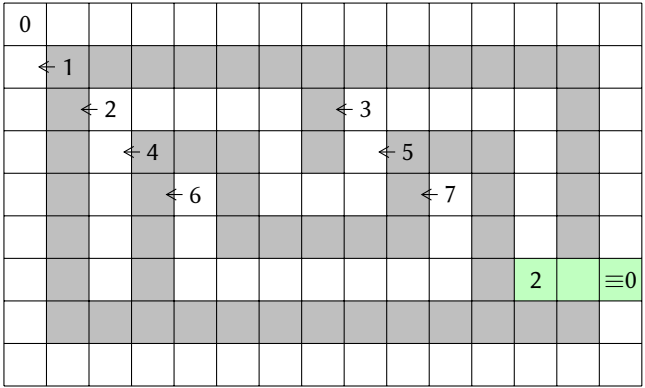
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	26
1	1	0	24
2	2	1	15
3	2	~~1~~	7
4	4	2	14
5	4	~~2~~	5
6	6	4	9
7	6	~~4~~	2

Equivalence & Adjacency tree



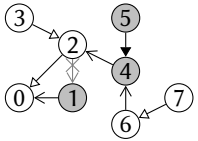
Example: First scan



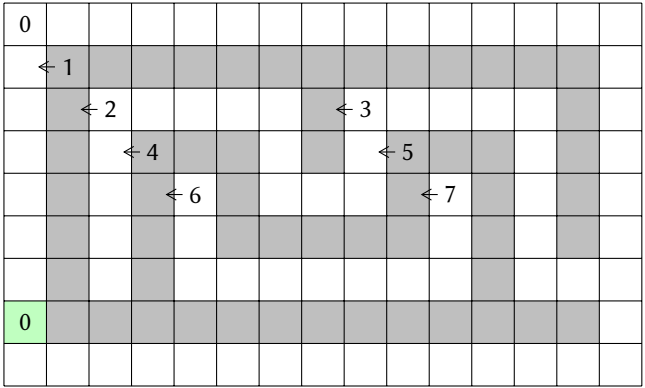
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	29
1	1	0	24
2	0	~~1~~	15
3	2	~~1~~	7
4	4	2	14
5	4	~~2~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree

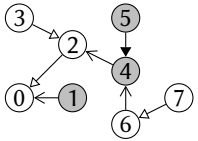


Example: First scan



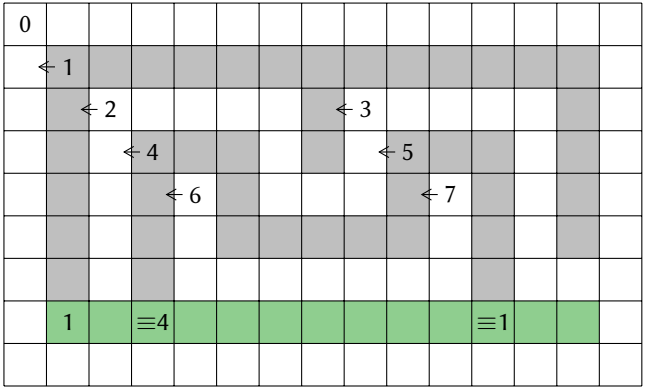
label	T	I	F_S
0	0	-1	30
1	1	0	24
2	0	~~1~~	15
3	2	~~1~~	7
4	4	2	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



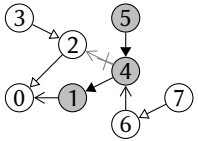
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Example: First scan



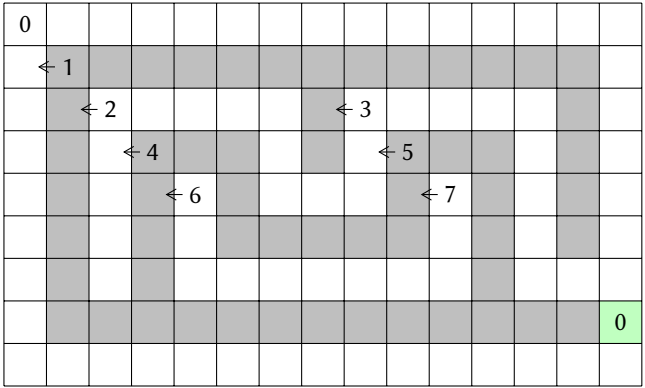
label	T	I	F_S
0	0	-1	30
1	1	0	37
2	0	~~1~~	15
3	2	~~1~~	7
4	1	~~2~~	14
5	4	~~2~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



Adjacency
Adjacency discarded
BG equivalence
FG equivalence

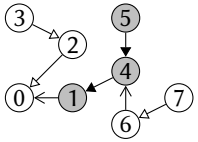
Example: First scan



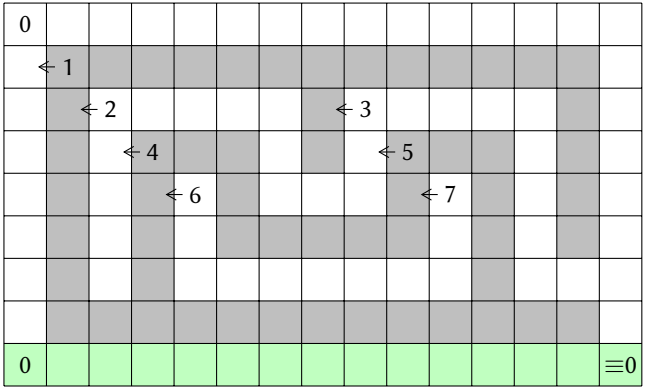
label	T	I	F_S
0	0	-1	31
1	1	0	37
2	0	~~1~~	15
3	2	~~1~~	7
4	1	~~2~~	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

Adjacency
Adjacency discarded
BG equivalence
FG equivalence

Equivalence & Adjacency tree



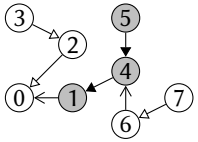
Example: First scan



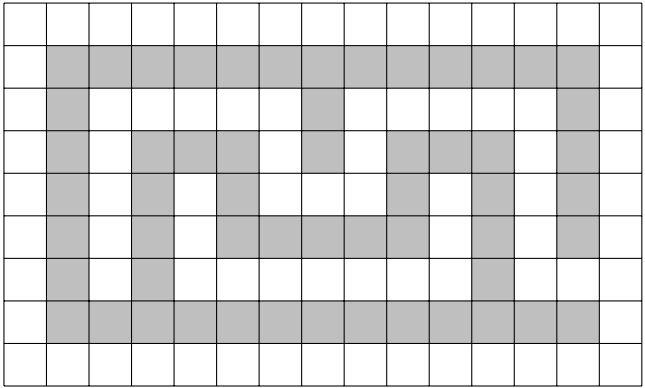
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	46
1	1	0	37
2	0	~~0~~	15
3	2	~~0~~	7
4	1	~~2~~	14
5	4	~~2~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree

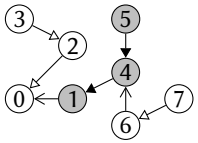


Example: Transitive closure



label	T	I	F_S
0	0	-1	46
1	1	0	37
2	0	~~1~~	15
3	2	~~1~~	7
4	1	~~2~~	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



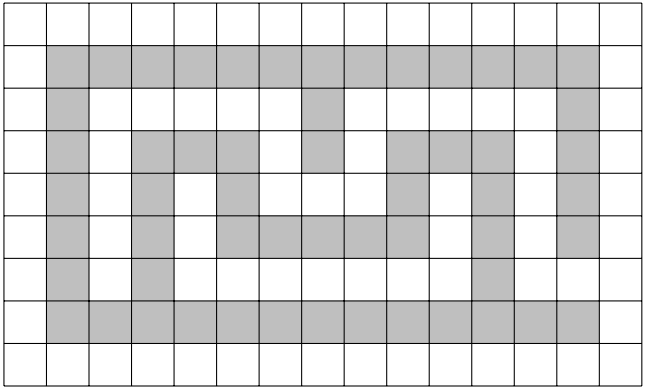
Adjacency

Adjacency discarded

BG equivalence

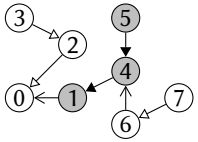
FG equivalence

Example: Transitive closure



label	T	I	F_S
0	0	-1	46
1	1	0	37
2	0	~~1~~	15
3	2	~~1~~	7
4	1	~~2~~	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



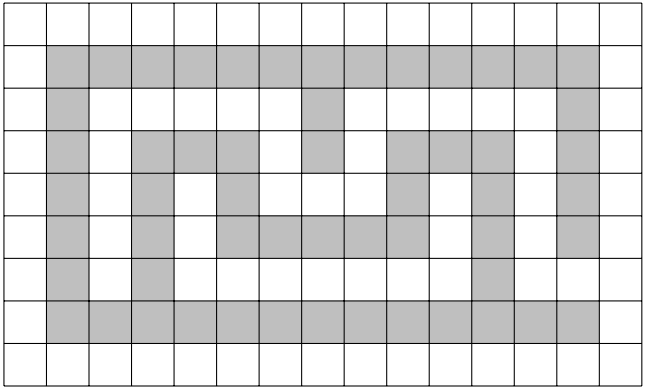
Adjacency

Adjacency discarded

BG equivalence

FG equivalence

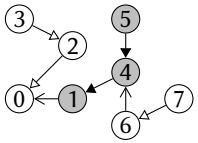
Example: Transitive closure



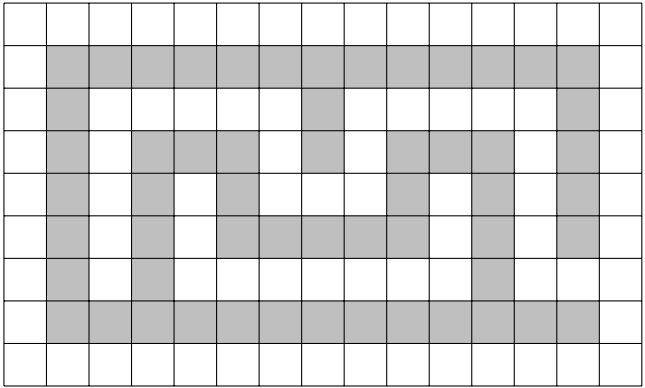
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	61
1	1	0	37
2	0	~~1~~	~~15~~
3	2	~~1~~	7
4	1	~~2~~	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



Example: Transitive closure



label	T	I	F_S
0	0	-1	68
1	1	0	37
2	0	~~1~~	~~15~~
3	0	~~1~~	~~7~~
4	1	~~2~~	14
5	4	~~3~~	5
6	6	4	9
7	6	~~5~~	2

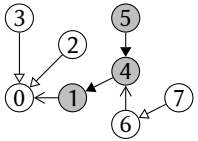
Adjacency

~~Adjacency discarded~~

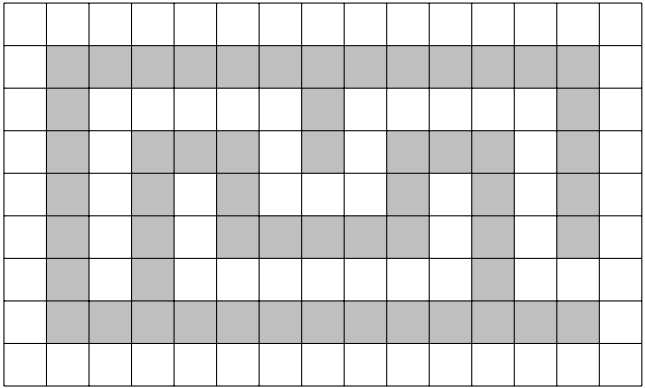
BG equivalence

FG equivalence

Equivalence & Adjacency tree



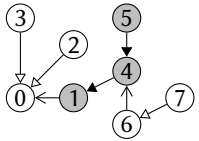
Example: Transitive closure



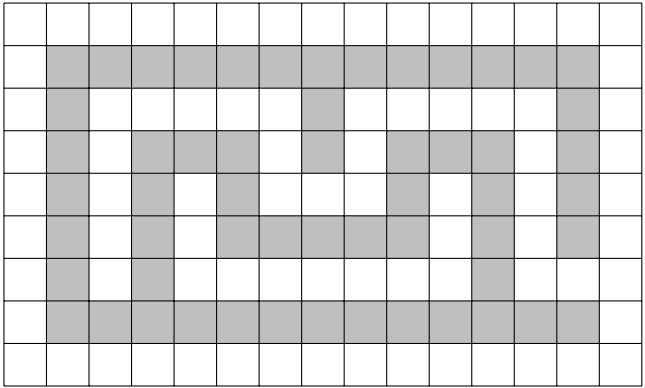
Adjacency
Adjacency discarded
BG equivalence
FG equivalence

label	T	I	F_S
0	0	-1	68
1	1	0	51
2	0	~~1~~	~~15~~
3	0	~~1~~	~~7~~
4	1	~~2~~	~~14~~
5	4	~~2~~	5
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree

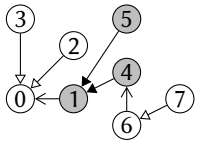


Example: Transitive closure



label	T	I	F_S
0	0	-1	68
1	1	0	56
2	0	~~1~~	~~15~~
3	0	~~1~~	~~7~~
4	1	~~2~~	~~14~~
5	1	~~3~~	~~8~~
6	6	4	9
7	6	~~5~~	2

Equivalence & Adjacency tree



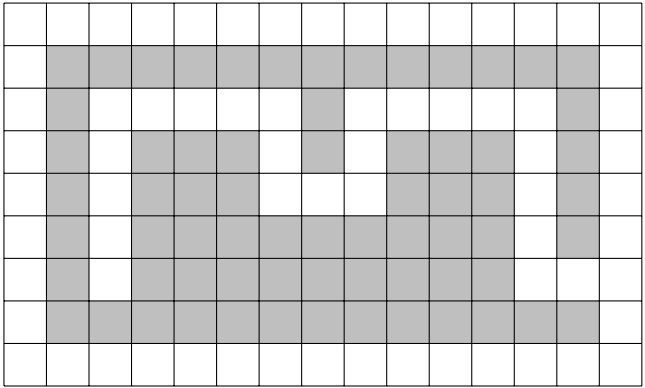
Adjacency

Adjacency discarded

BG equivalence

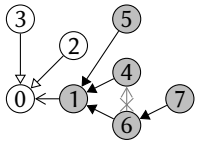
FG equivalence

Example: Transitive closure



label	T	I	F_S
0	0	-1	68
1	1	0	65
2	0	~~1~~	~~15~~
3	0	~~1~~	~~7~~
4	1	~~2~~	~~14~~
5	1	~~3~~	~~5~~
6	1	~~4~~	~~9~~
7	6	~~5~~	2

Equivalence & Adjacency tree



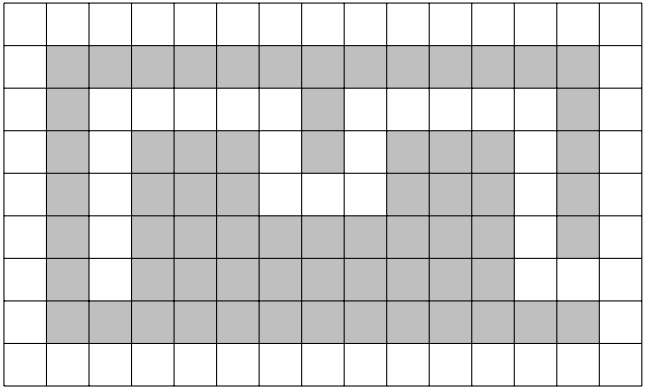
Adjacency

Adjacency discarded

BG equivalence

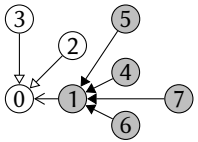
FG equivalence

Example: Transitive closure



label	T	I	F_S
0	0	-1	68
1	1	0	67
2	0	~~1~~	~~15~~
3	0	~~1~~	~~7~~
4	1	~~2~~	~~14~~
5	1	~~3~~	~~5~~
6	1	~~4~~	~~9~~
7	1	~~5~~	~~2~~

Equivalence & Adjacency tree



Adjacency

Adjacency discarded

BG equivalence

FG equivalence

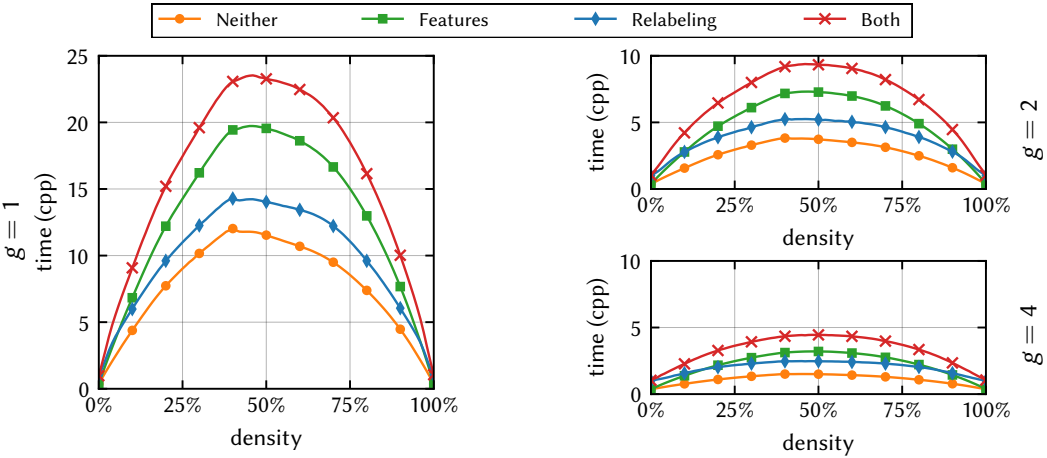
Benchmark protocol

We tested randomly generated 2048×2048 images with varying density and granularity on a Skylake Gold 6126 Xeon @2.60GHz.

We focus our analysis on $g = 1$ as it is the *worst* case for run-based algorithms like FLSL.

Some State-of-the-Art algorithms could not be tested, so we took the results from their paper directly and normalized performance in cycle per point (cpp) to make them comparable.

Computation performance



Computation performance

time (cpp)		min	max
BW + Adjacency	(BWA)	0.36	12.7
+Euler number	(E)	+ 0	+ 0.29
+Hole Filling	(H)	+ 0	+ 0.50
+Feature Computation	(F)	+ 0	+ 8.59
+Relabeling	(R)	+ 0.59	+ 3.66

Impact on the processing time of each additional computations

- E, H, F have zero impact on empty images
- E, H are very fast
- R should be avoided if not required

Performance comparison with State-of-the-Art

algorithm		compute	min	avg	max
[1]He bit-quad		E	2.87	14.0	23.7
[2]He BW	(with R)	BWER	16.5	51.0	79.6
[3]Diaz	(with R)	BWAR	18.4	36.8	59.0
[4]Spaghetti(FG) + Spaghetti(BG)	(with R)	BWR	5.76	32.7	51.2
[5]FLSL(FG+R) + FLSL(BG+R)	(with R)	BWR	2.34	17.0	24.4
[5]FLSL(FG+F) + FLSL(BG+F)	(with F)	BWF	1.68	20.5	30.3
[5]FLSL(BG) + FLSL(FG+F)	(with F)	WF**H**	1.89	19.8	33.7
BW FLSL+ER		BWAER	0.98	10.0	14.6
BW FLSL+F		BWAF	0.38	13.0	20.0
BW FLSL+FH		BWAF**H**	0.38	14.0	20.7

B : Black labeling (BG) A: Adjacency tree F : Feature Computation R: Relabel
W: White labeling (FG) E: Euler number **H**: Hole filling

Conclusion

Introducing our new algorithm BW FLSL:

- fast Black & White processing
- simple Adjacency Tree computation
- low overhead Adjacency Tree computation
- fast features computation (Component Analysis)
- able to fill holes and propagate features
- easily extensible to arbitrary connected filters

Thank you!

References I

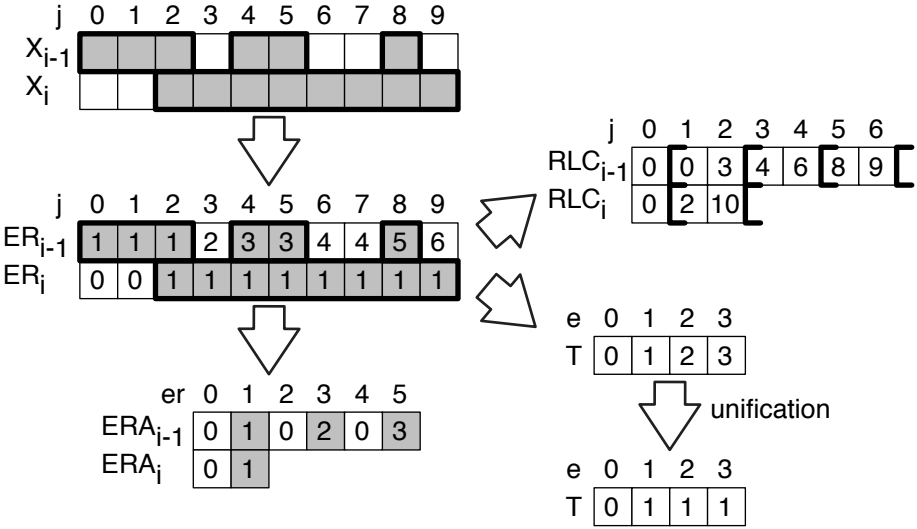
B. Yao, L. He, S. Kang, Y. Chao, and X. Zhao, "Bit-quad-based euler number computing," *Transaction on Information and Systems*, vol. E100-D,9, pp. 2197–2204, 2017.

L. He, Y. Chao, and K. Suzuki, "An algorithm for connected-component labeling, hole labeling and euler number computing," *Journal of Computer Science and Technology*, vol. 28,3, pp. 468–478, 2013.

F. D. del Rio, P. Sanchez-Cuevas, H. Molina-Abril, and P. Real, "Parallel connected-component-labeling based on homotopy trees," *Pattern Recognition Letters*, vol. 131, pp. 71–78, 2020.

F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *Transactions on Image Processing*, vol. PP, pp. 1–14, 2019.

F. Lemaitre, A. Hennequin, and L. Lacassagne, "How to speed connected component labeling up with SIMD RLE algorithms," in *Proceedings of the 2020 Sixth Workshop on Programming Models for SIMD/Vector Processing*, pp. 1–8, 2020.



Tables for the LSL (*ER* is actually not needed anymore).

Algorithm 1: New BW FLSL overview.

This paper contribution is highlighted with gray boxes.

```
1  ne ← 1                          ▷ Reset number of labels
2  I[0] ← −1                       ▷ Exterior component has no surrounding
3  F[0] ← ∅
4  for i = 0 to h − 1 do
5      RLE(i)                      ▷ Step 1a: Detect segments
6      Unify(i)                    ▷ Step 1b: Merge labels from adjacent line segments
7  Close()                         ▷ Step 2: Transitively close the equivalence graph
8  if relabel then
9      for i = 0 to h − 1 do
10         Relabel(i)              ▷ Step 3: Write the label image
```

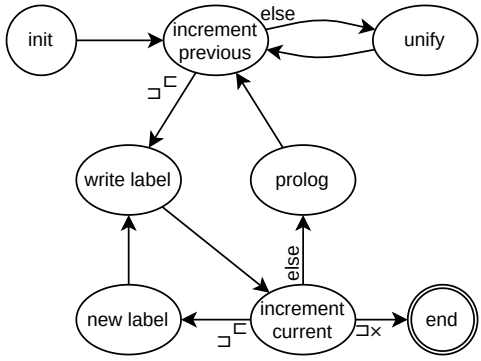
```
int16_t* detect_segment(int w, const uint8_t* X, int16_t* RLC, int16_t* ER) {
  __m128i last = _mm_setzero_si128();
  __m128i incr8 = _mm_set1_epi16(8);
  __m128i J = _mm_set_epi16(7, 6, 5, 4, 3, 2, 1, 0);
  __m128i ver = _mm_setzero_si128();
  for (int i = 0; i < w; i += 8) {
    __m128i in = _mm_loadl_epi64((const __m128i*)(X + i));
    in = _mm_cvtepi8_epi16(in);
    // edge detection
    __m128i prev = _mm_alignr_epi8(in, last, 14);
    __m128i f = _mm_cmpeq_epi8(in, prev);
    f = _mm_xor_si128(f, _mm_set_epi8(-1));
    last = in;
    // compress and store indices: LUT-based emulation
    RLC = compress_storeu_m16_epi16(RLC, J, f);
    J = _mm_add_epi16(J, incr8);
    // ER is a plus scan of the edges (f < 0)
    ver = _mm_sub_epi16(ver, inclusive_plus_scan_epi16(f));
    _mm_store_si128((__m128i*)(ER + j), ver);
    ver = _mm_shuffle_epi8(ver, _mm_set_epi16(0x0f0e));
  }
  *RLC++ = w;
  return RLC;
}
```

Listing: `compress`-based RLE encoder targeting SSE

Algorithm 2: New BW unification (BW FLSL step 1b).

Black and White related processing is highlighted with gray boxes.

```
1  init:
2      er ← 0    ▷ Index of current line segment (relative label)
3      er' ← 0   ▷ Index of previous line segment
4      a ← 0     ▷ Label of current segment, the first is necessarily the exterior
5      j0 ← RLC_0[er]    ▷ Starting position of current segment
6      j1 ← RLC_0[er + 1]    ▷ End position of current segment
7      c8 ← 0    ▷ c8 = 1 if current segment is 8-adjacent, here, BG is 4-adjacent
8      ▷ virtual segments allowing to avoid testing for the end of previous line
9      S0 ← RLC_{i-1}[ner_{i-1}]    S1 ← RLC_{i-1}[ner_{i-1} + 1]    ▷ Save past-the-end
10     if ner_{i-1} is odd then RLC_{i-1}[ner_{i-1}] ← w + 1
11     RLC_{i-1}[ner_{i-1} + 1] ← w + 2
12     goto increment previous
13 new label:
14     T[ne] ← ne    ▷ On-the-fly initialization of the equivalence table
15     F[ne] ← ∅    ▷ On-the-fly initialization of the feature table
16     F[ne] ← a    ▷ Initial adjacency: a is the label of previous segment
17     a ← ne
18     ne ← ne + 1
19 write label:
20     F[a] ← F[a] ∪ computeFeatures(i, j0, j1)
21     ERA_i[er] ← a
22 increment current:
23     er ← er + 1    ▷ Next segment of current line
24     er' ← er' − 1    ▷ Previous segment of previous line intersects current segment
25     c8 ← c8 ⊕ 1    ▷ Adjust adjacency for current component
26     j0 ← j1
27     j1 ← RLC_i[er + 1]
28     if er = ner_i then goto end
29     if RLC_{i-1}[er'] ⩾ j1 + c8 then goto new label
30 prolog:
31     a ← Find(T, ERA_{i-1}[er'])
32 increment previous:
33     er' ← er' + 2
34     if RLC_{i-1}[er'] ⩾ j1 + c8 then goto write label
35 unify:
36     e ← Find(T, ERA_{i-1}[er'])
37     ▷ Union of the two root labels e and a
38     if e ≠ a then
39         if e < a then swap e, a
40         T[e] ← a
41     goto increment previous
42 end:
43     if ner_i is odd and a ≠ 0 then T[a] ← 0
44     RLC_{i-1}[ner_{i-1}] ← S0    RLC_{i-1}[ner_{i-1} + 1] ← S1    ▷ Restore past-the-end
```



Algorithm 3: New BW Transitive closure (BW FLSL step 2).

Black and White related processing is highlighted with gray boxes.

```
1  for e = 0 to ne − 1 do
2      a ← T[e]    ▷ ancestor
3      if Hole filling and e = a then    ▷ If label is root
4          i ← I[e]    ▷ label of the surrounding component
5          if T[i] > 0 then a ← i    ▷ e has a surrounding (is not the exterior)
6      if a < e then
7          r ← T[a]
8          T[e] ← r    ▷ Transitive Closure: r = T[T[e]]
9          F[r] ← F[r] ∪ F[e]    ▷ Feature merge
10     else    ▷ e is a root
11         I[e] ← T[I[e]]    ▷ point adjacency to root
12         if Euler number computation then E[e] ← 0
13 if Euler number computation then
14     for e = ne − 1 to 0 step −1 do
15         if T[e] = e then
16             i ← I[e]
17             E[i] ← E[i] + 1 − E[e]
```

Algorithm 4: New BW Relabeling (BW FLSL step 3).

$j_0 \leftarrow RLC_i[0]$ ▷ j_0 is 0
for $er = 0$ **to** $ner_i - 1$ **step** 1 **do**
    $e \leftarrow ERA_i[er]$ ▷ provisional label
    $r \leftarrow T[e]$ ▷ final label
    $j_1 \leftarrow RLC_i[er + 1]$
    $Y_i[j_0, j_1[\leftarrow r$ ▷ Memset
    $j_0 \leftarrow j_1$ ▷ Register rotation