

CprE 281: Digital Logic

Instructor: Alexander Stoytchev

<http://www.ece.iastate.edu/~alexs/classes/>

State Minimization

*CprE 281: Digital Logic
Iowa State University, Ames, IA
Copyright © Alexander Stoytchev*

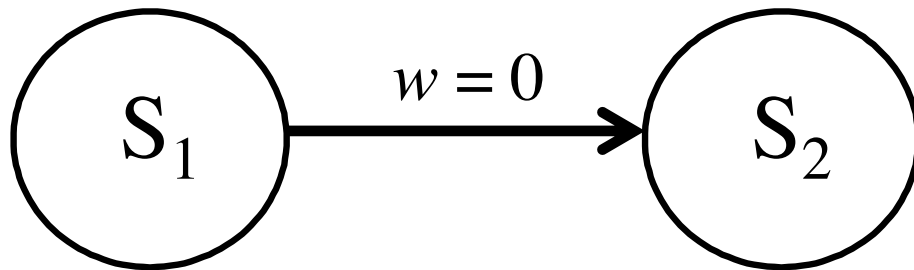
Equivalence of states

“Two states S_i and S_j are said to be equivalent if and only if for every possible input sequence, the same output sequence will be produced regardless of whether S_i or S_j is the initial state.”

Partition Minimization Procedure

0-successor

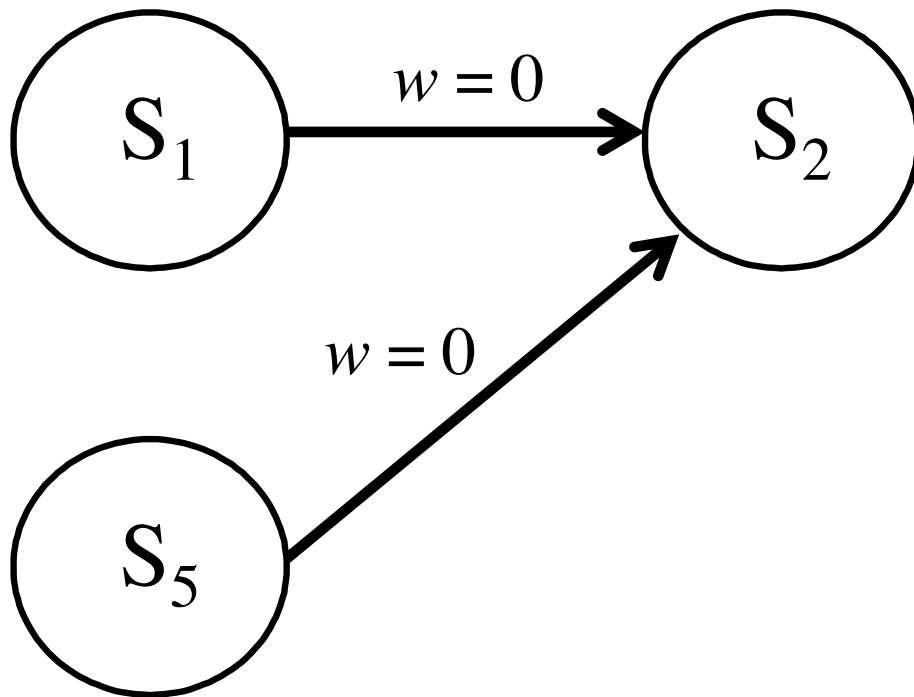
Assuming that we have only one input signal w



S_2 is a 0-successor of S_1

0-successor

Assuming that we have only one input signal w

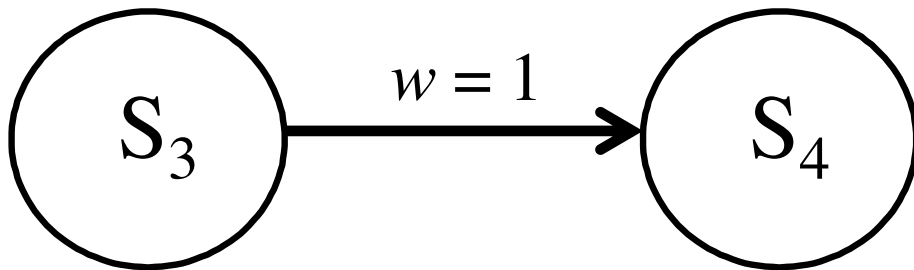


S_2 is a 0-successor of S_1

S_2 is a 0-successor of S_5

1-successor

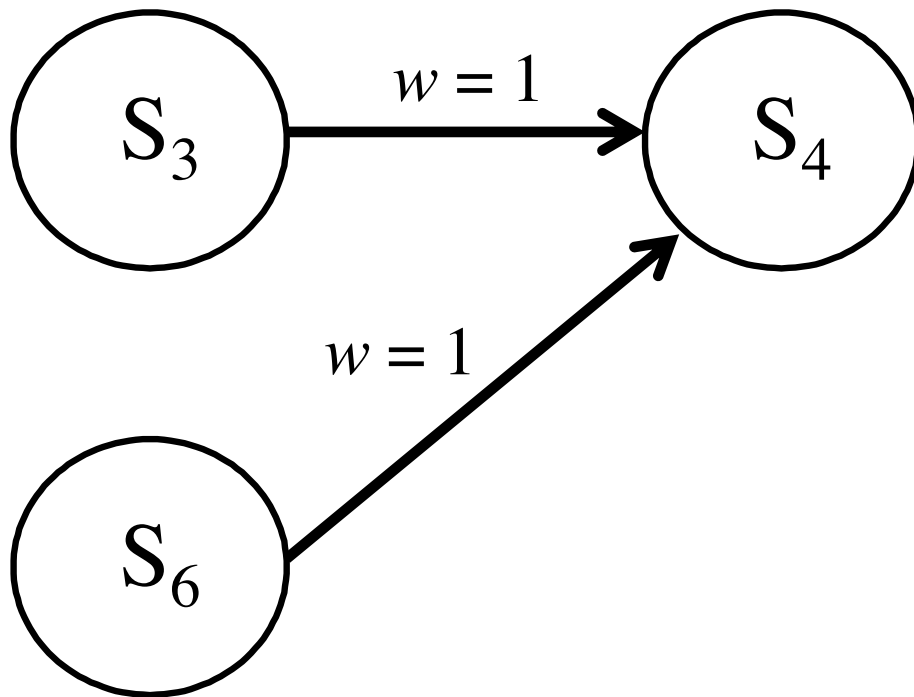
Assuming that we have only one input signal w



S_4 is a 1-successor of S_3

1-successor

Assuming that we have only one input signal w

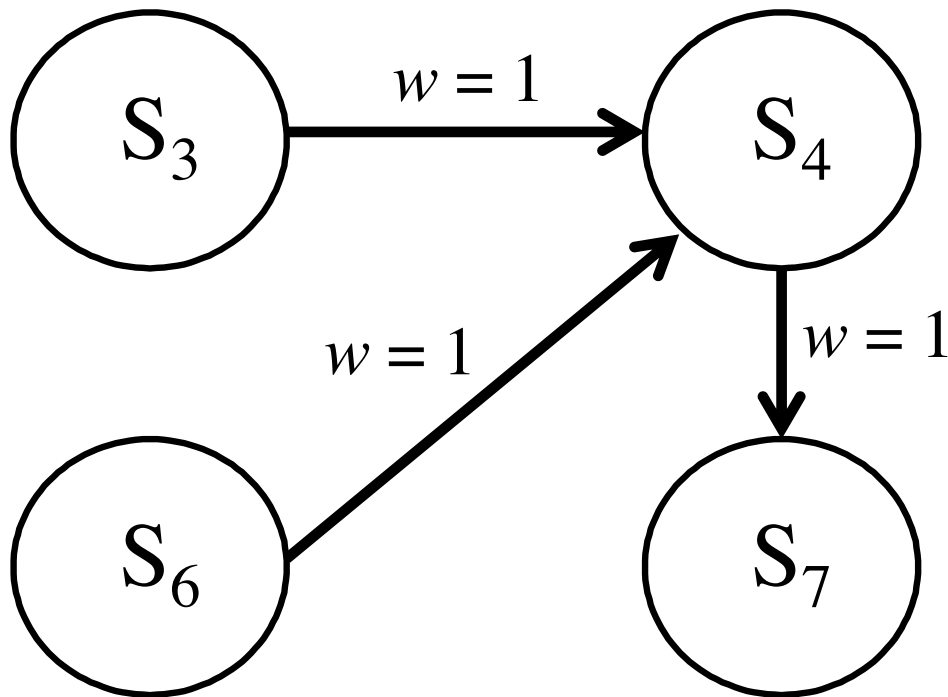


S_4 is a 1-successor of S_3

S_4 is a 1-successor of S_6

1-successor

Assuming that we have only one input signal w



S_4 is a 1-successor of S_3

S_4 is a 1-successor of S_6

S_7 is a 1-successor of S_4

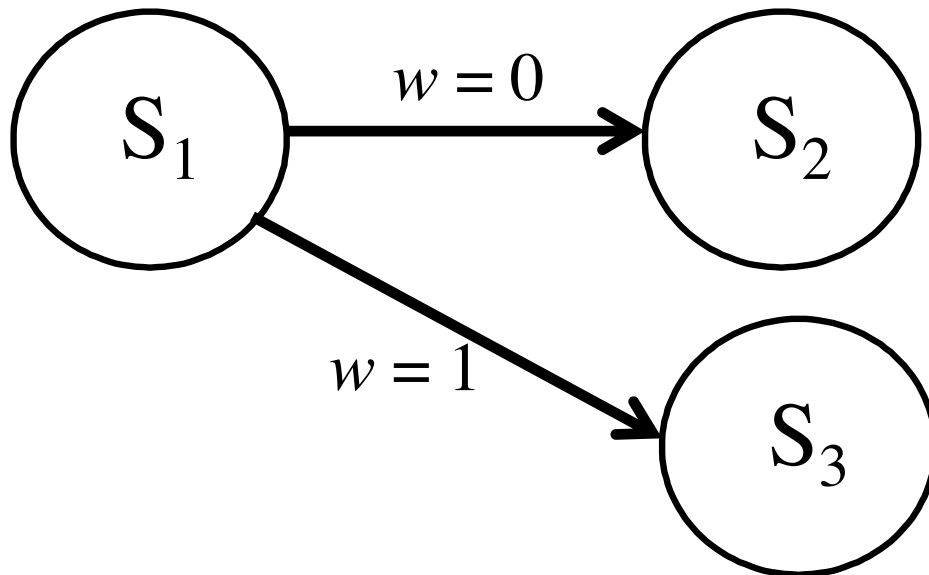
k-successors of a State

Assuming that we have only one input signal w , then k can only be equal to 0 or 1.

k-successors of a State

Assuming that we have only one input signal w , then k can only be equal to 0 or 1.

In other words, this is the familiar 0-successor or 1-successor case.



S_2 is a 0-successor of S_1

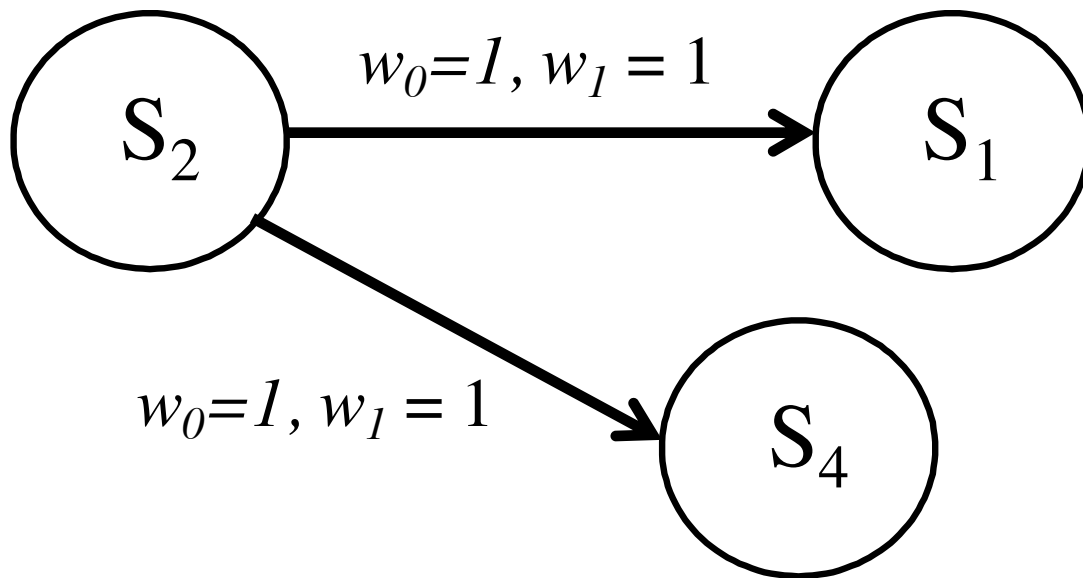
S_3 is a 1-successor of S_1

k-successors of a State

If we have two input signals, e.g., w_0 and w_1 , then k can only be equal to 0, 1, 2, or 3.

k-successors of a State

If we have two input signals, e.g., w_0 and w_1 , then k can only be equal to 0, 1, 2, or 3.



S_1 is a 3-successor of S_2

S_4 is a 3-successor of S_2

Equivalence of states

“If states S_i and S_j are equivalent, then their corresponding k -successors (for all k) are also equivalent.”

Partition

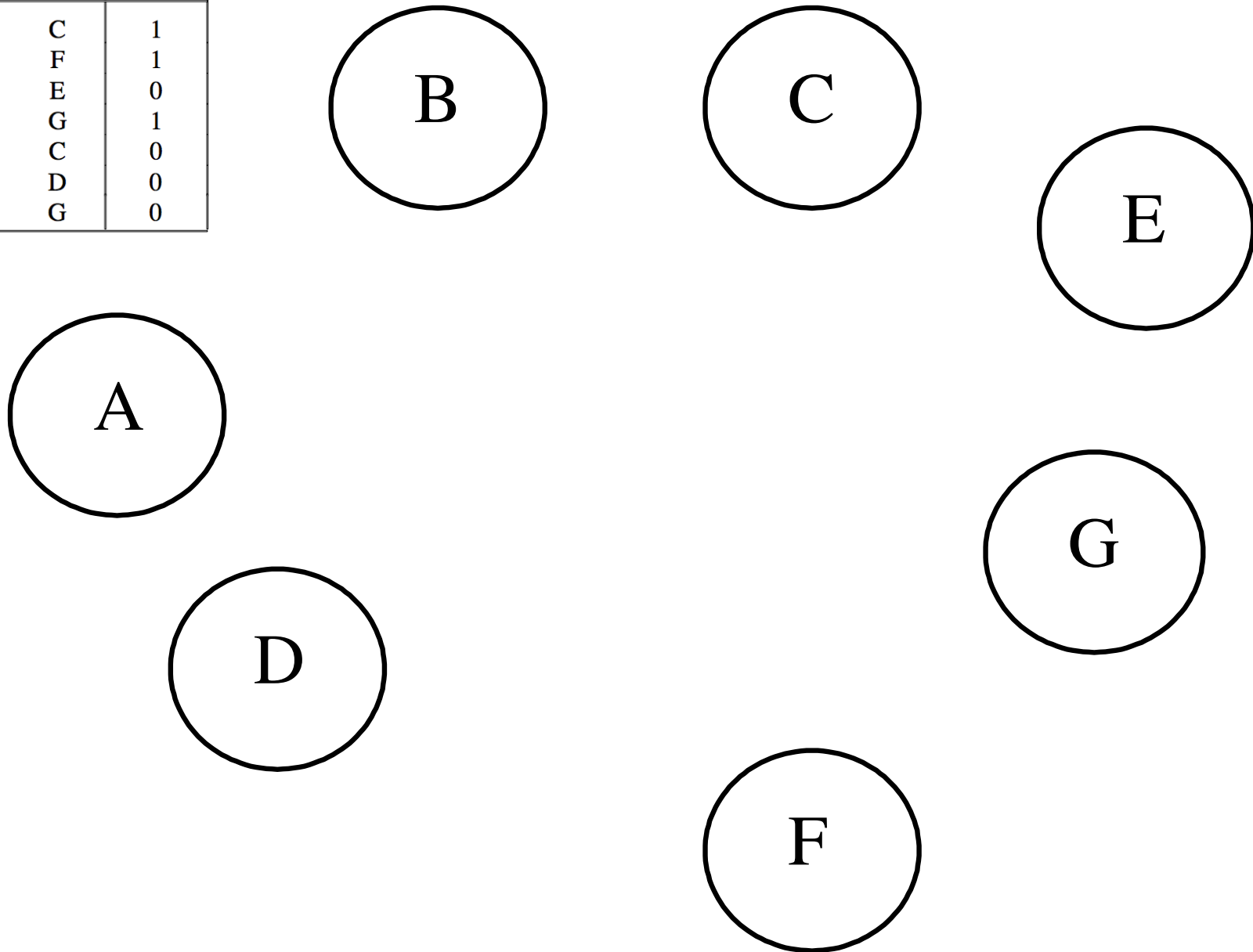
“A partition consists of one or more blocks, where each block comprises a subset of states that may be equivalent, but the states in a given block are definitely not equivalent to the states in other blocks.”

State Table for This Example

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

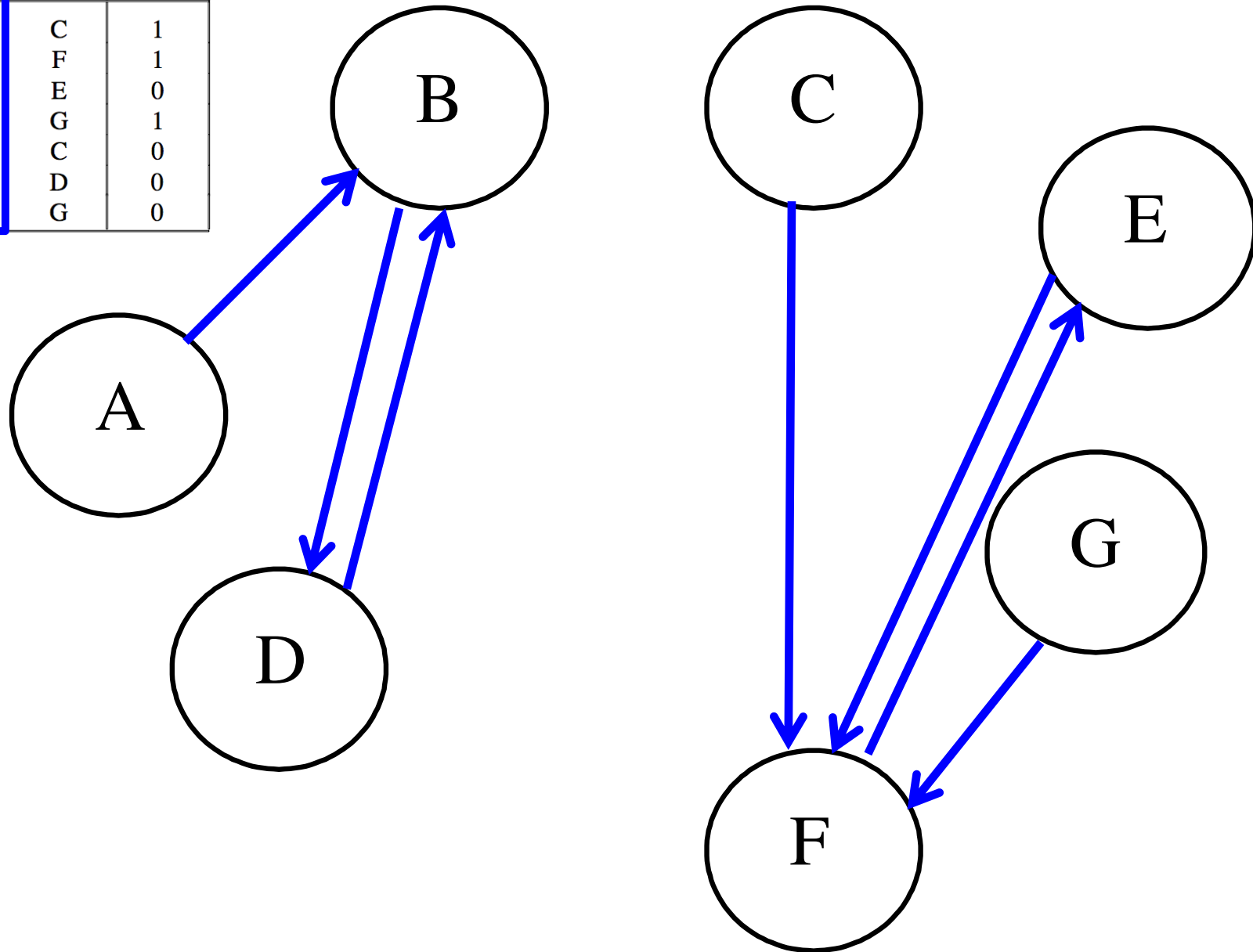
State Diagram (just the states)

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0



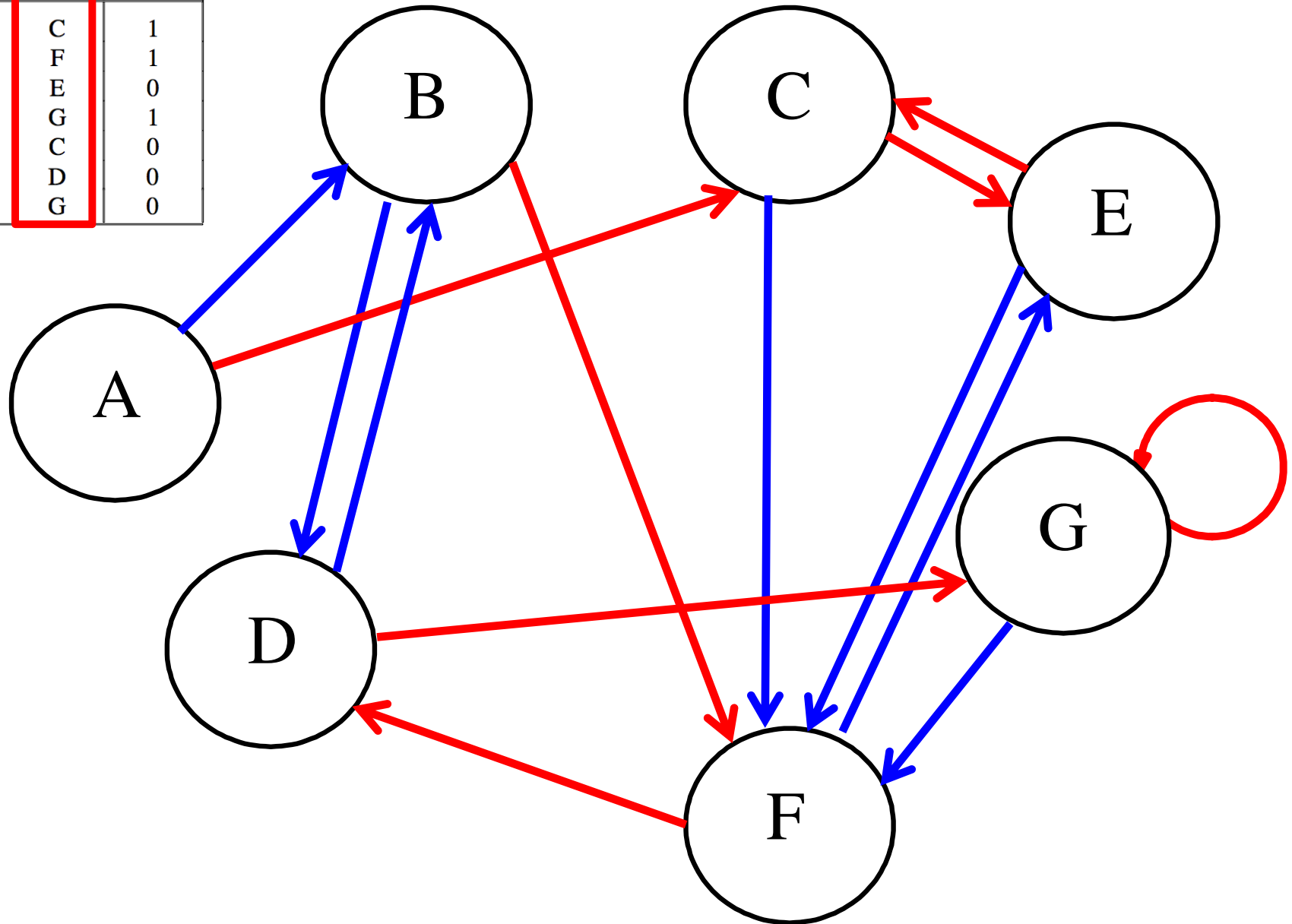
State Diagram (transitions when $w=0$)

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0



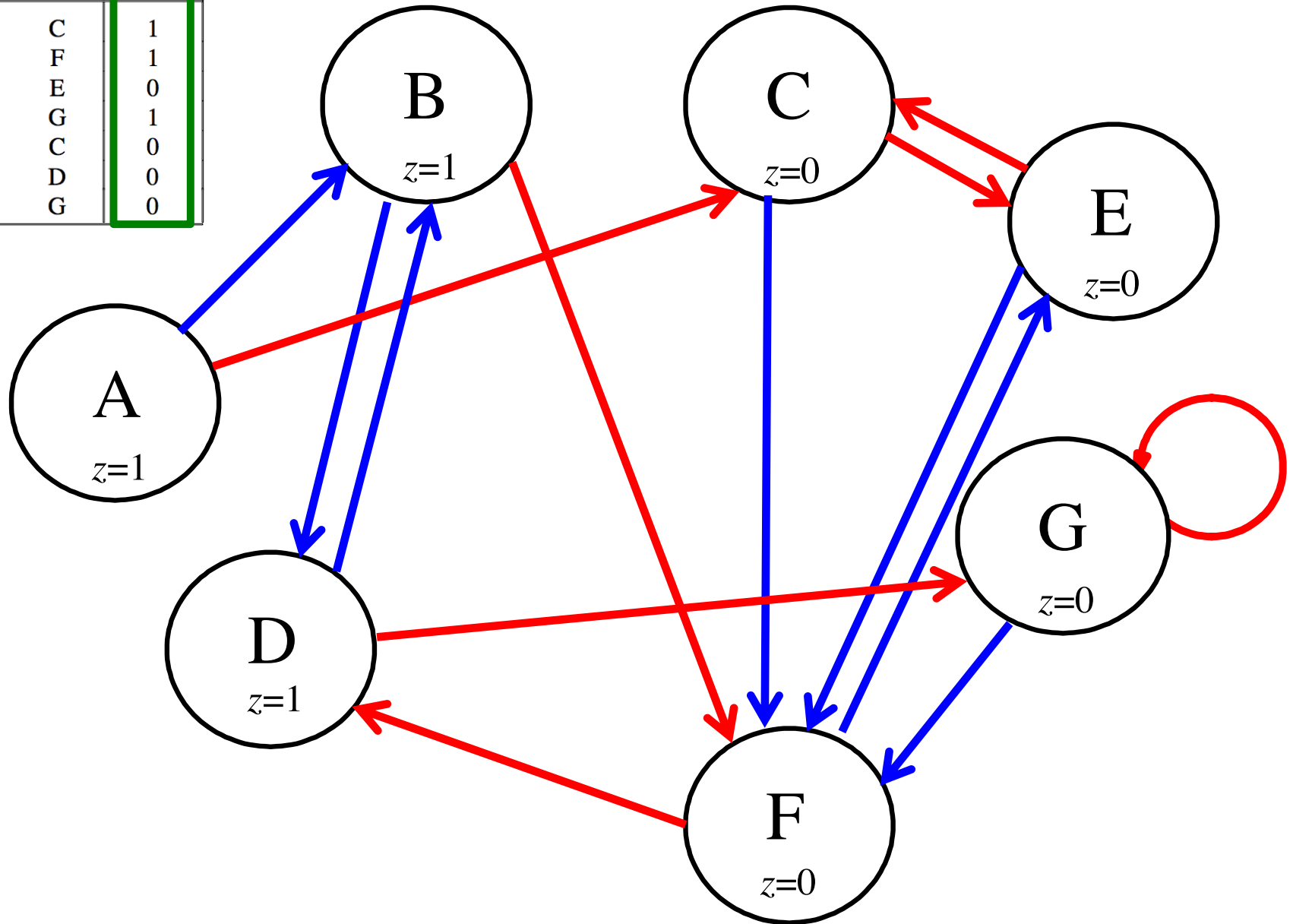
State Diagram (transitions when $w=1$)

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0



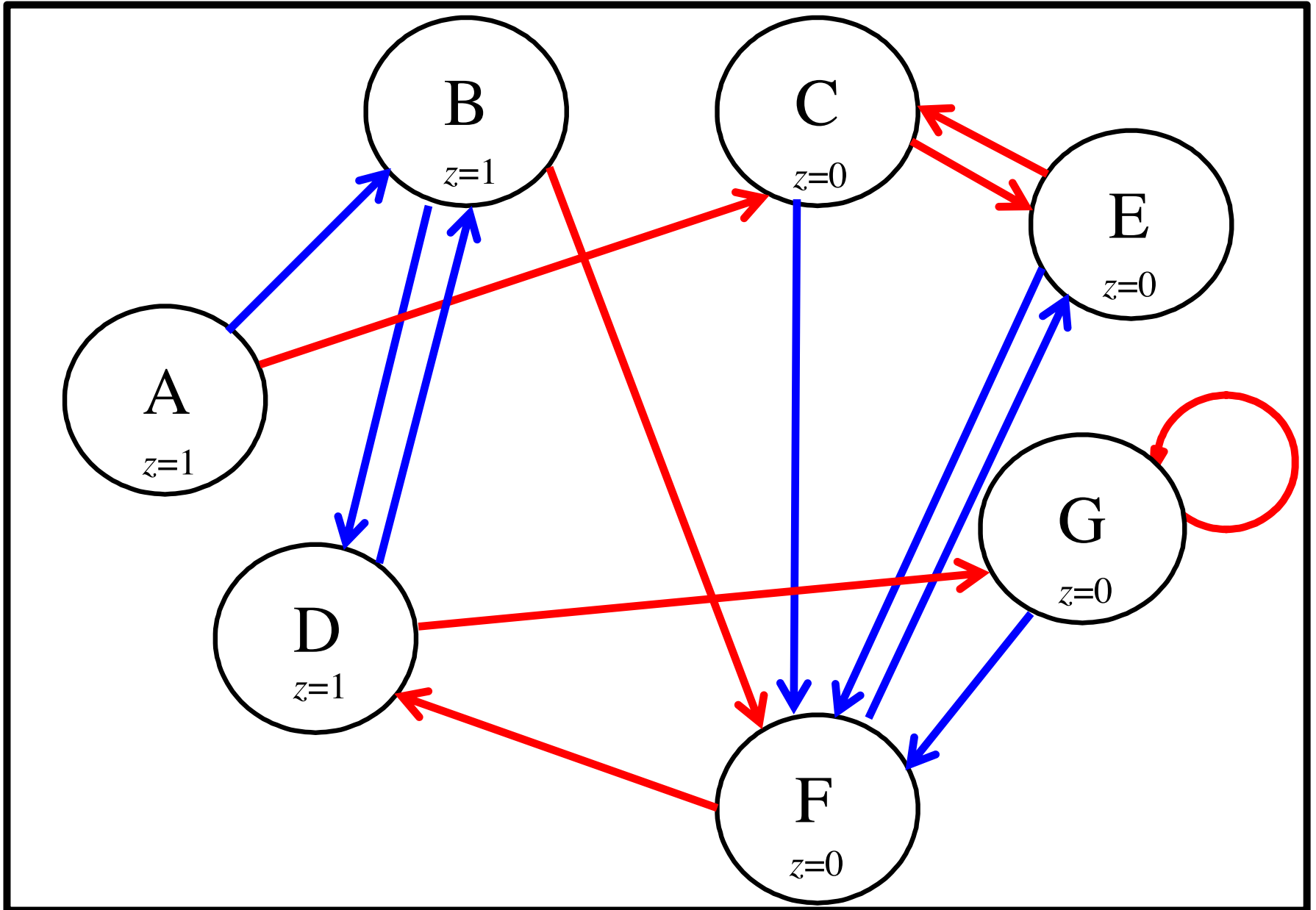
Outputs

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0



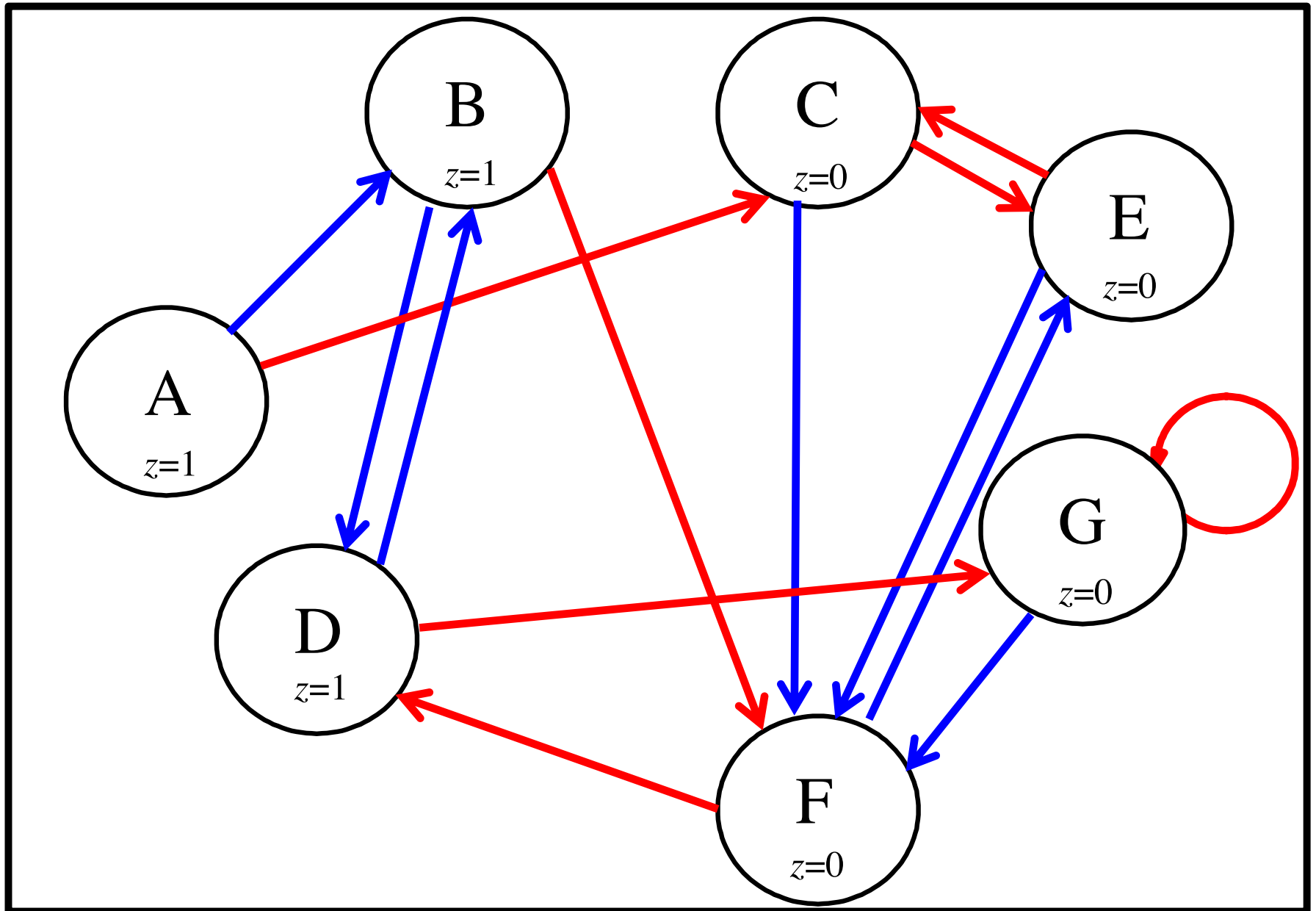
Partition #1

(All states in the same partition)



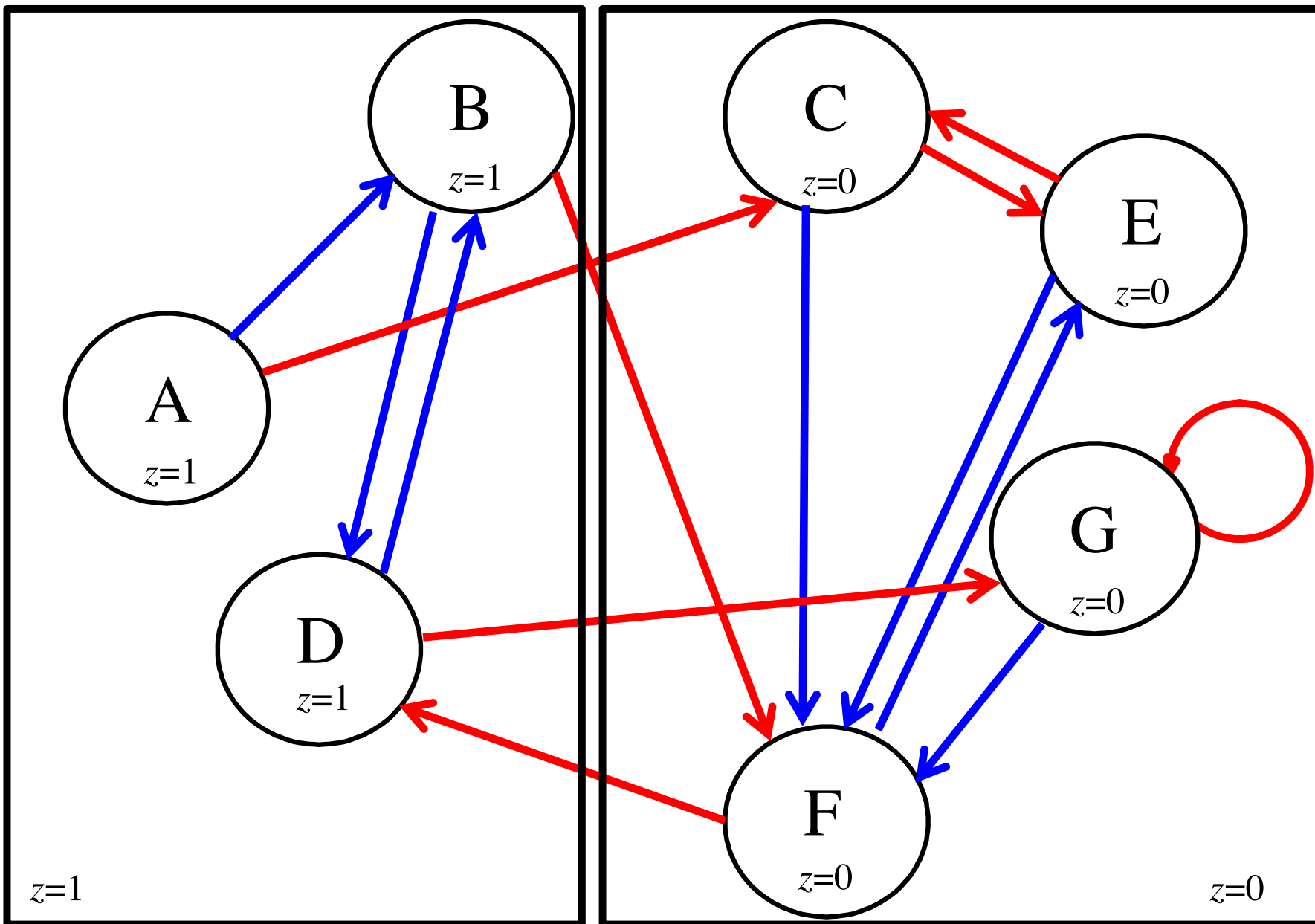
Partition #1

(ABCDEFGG)



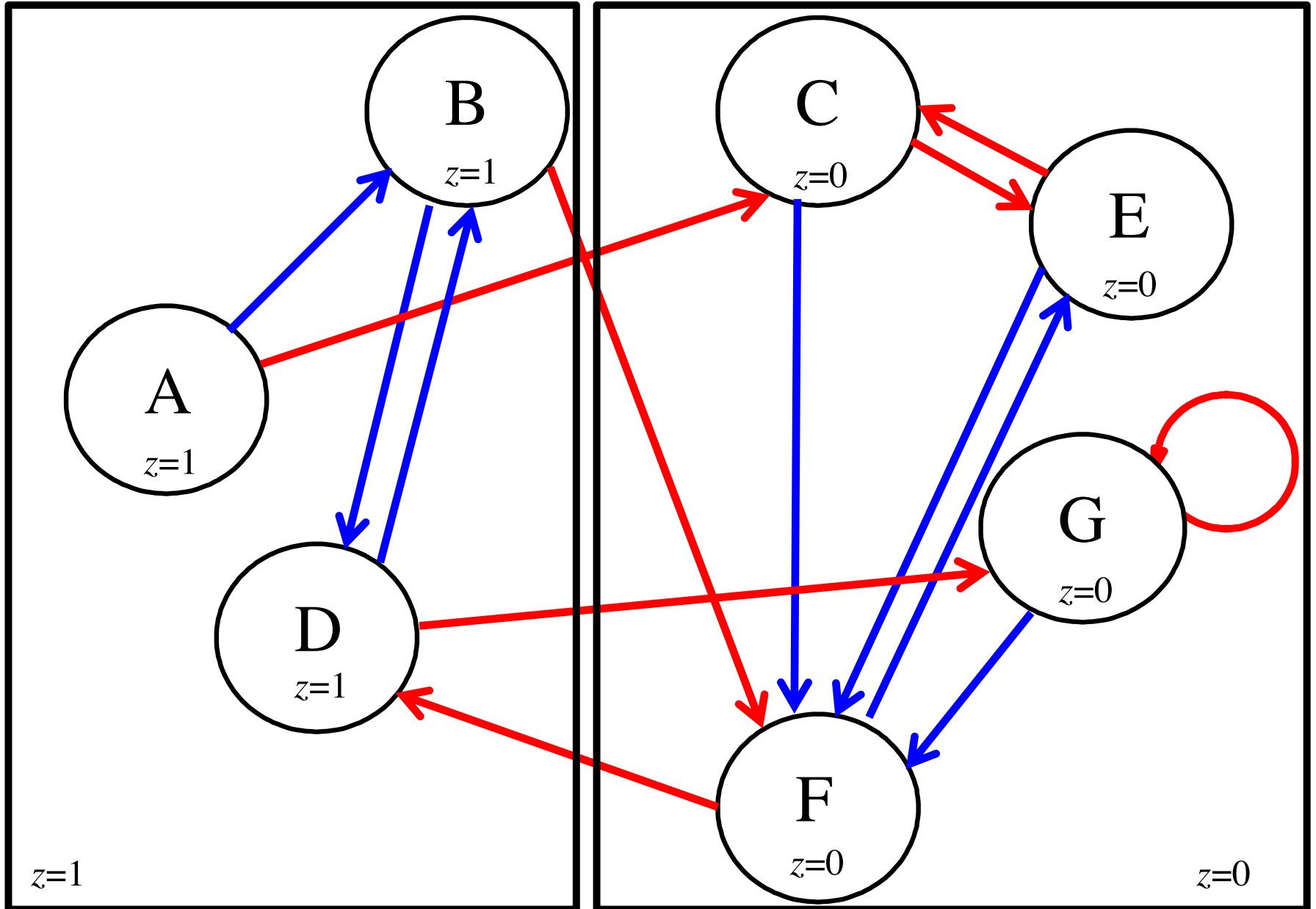
Partition #2

(based on outputs)



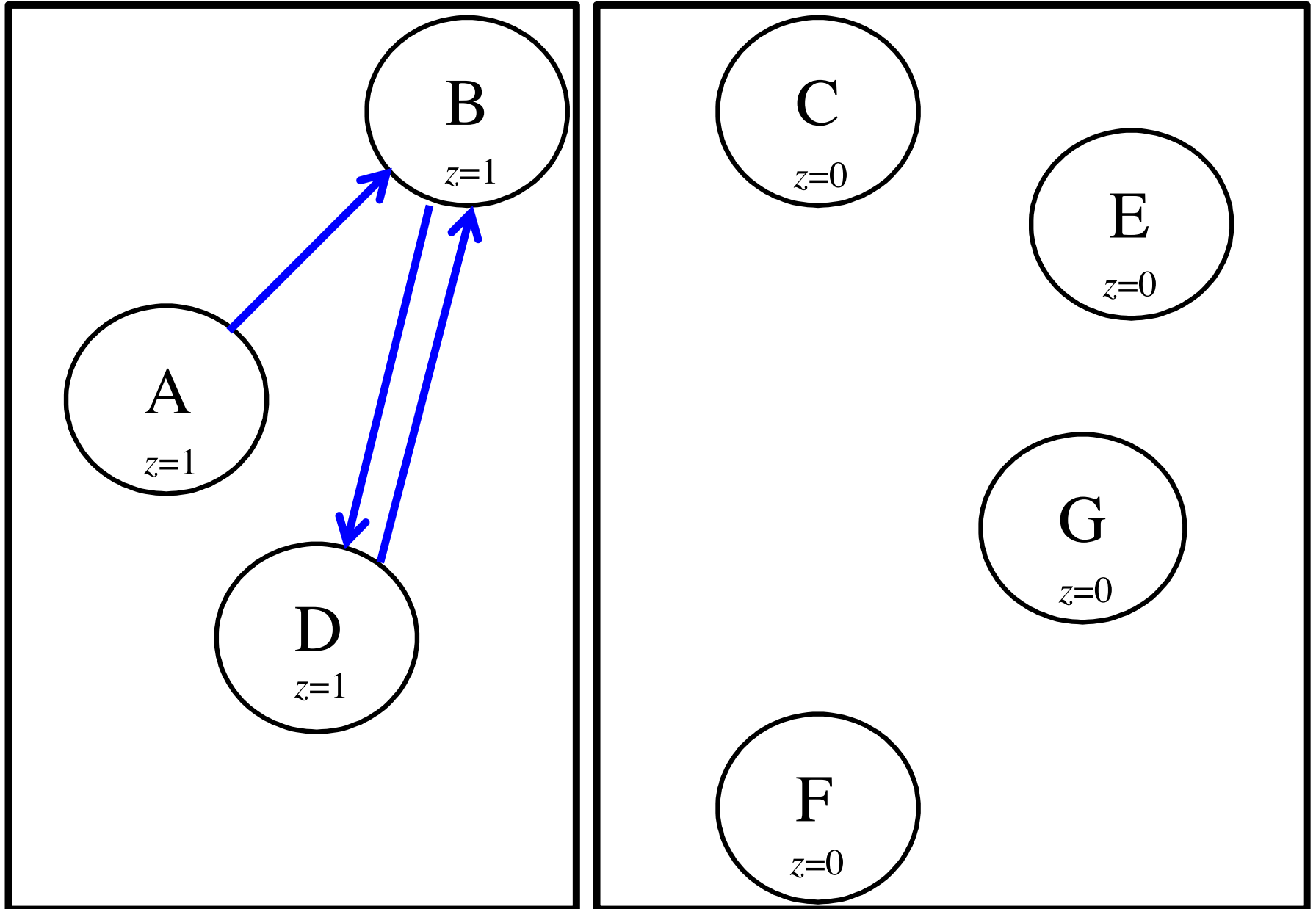
Partition #2

(ABD)(CEFG)



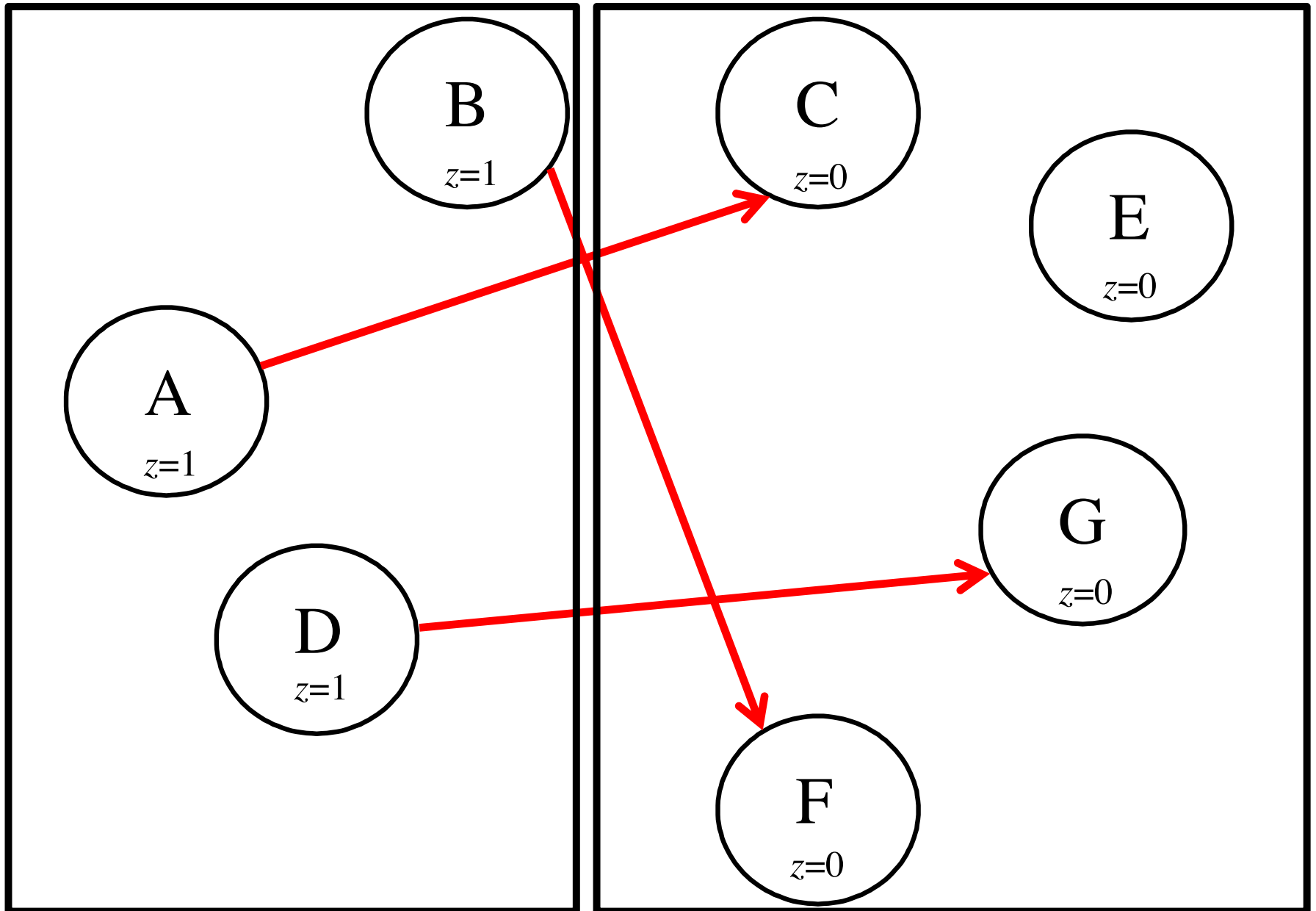
Partition #3.1

(Examine the 0-successors of ABD)



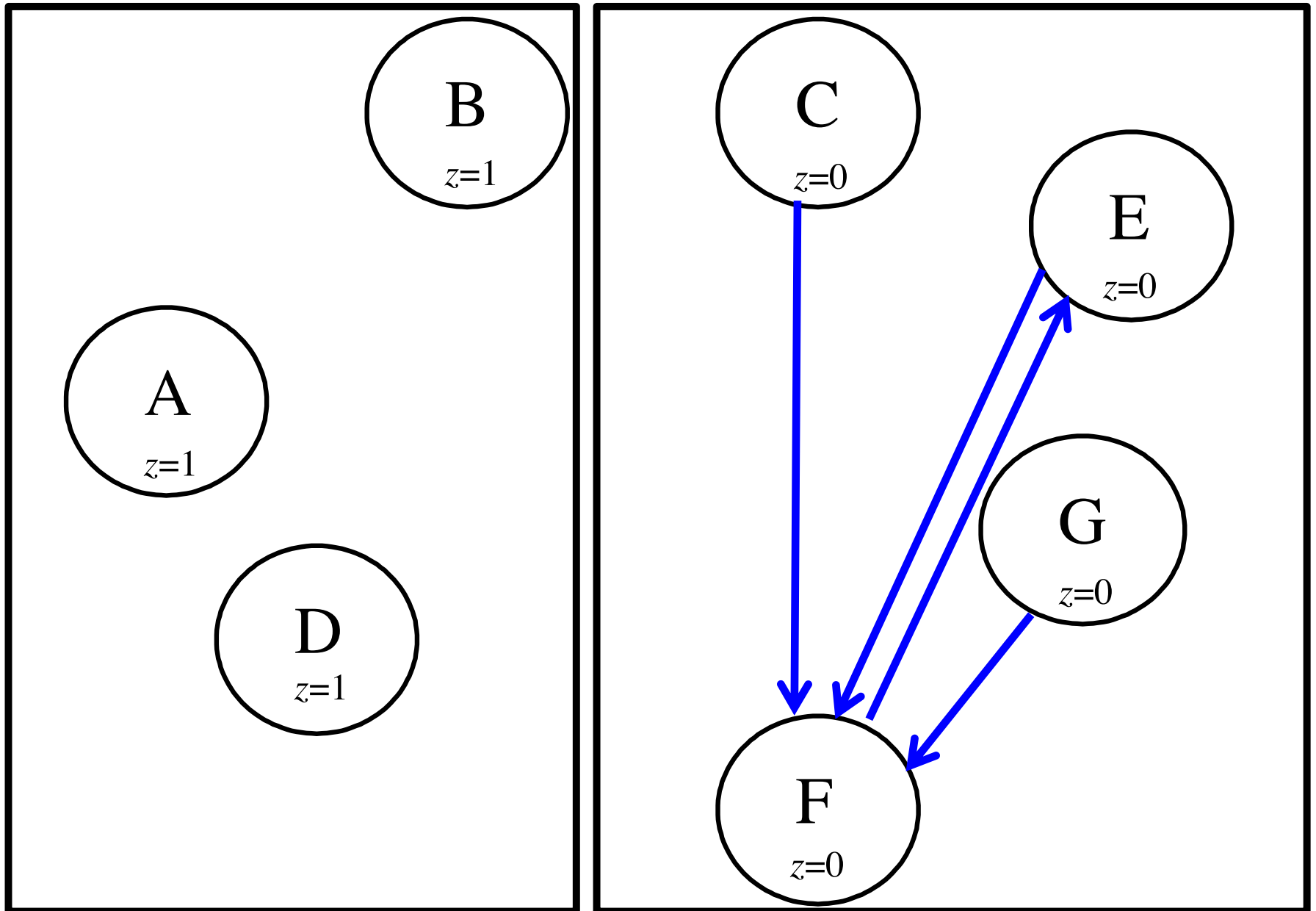
Partition #3.1

(Examine the 1-successors of ABD)



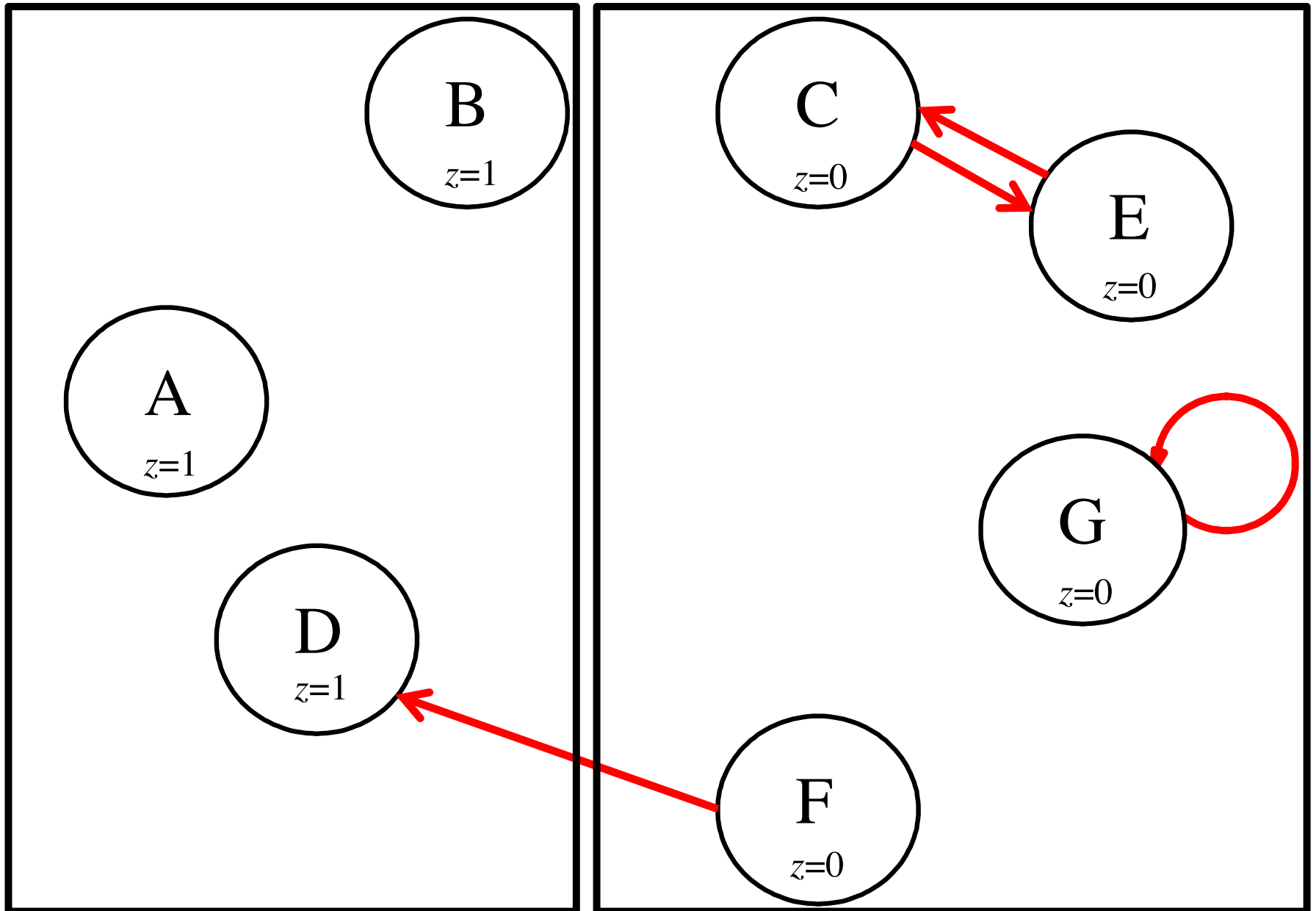
Partition #3.2

(Examine the 0-successors of CEFG)



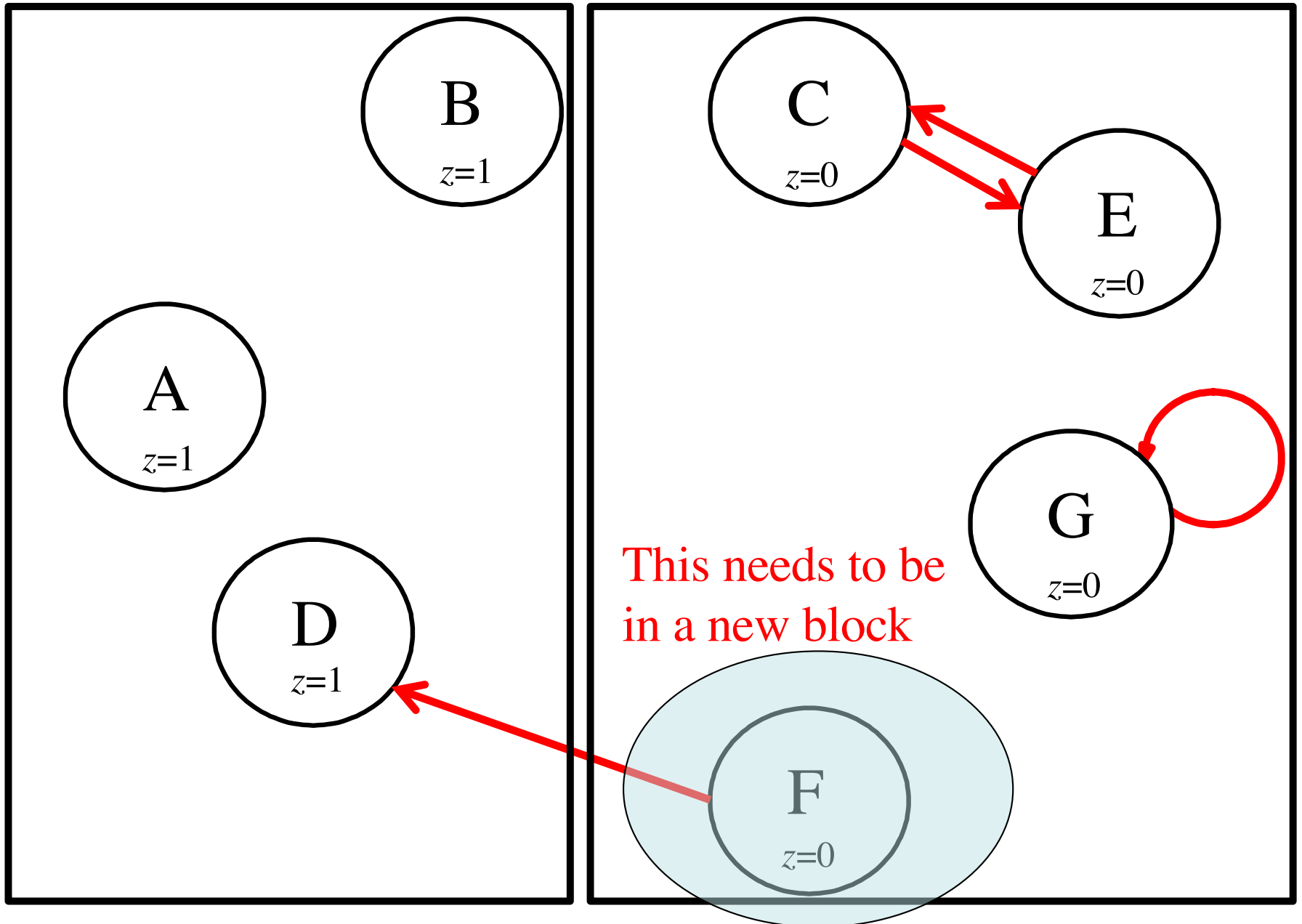
Partition #3.2

(Examine the 1-successors of CEFG)



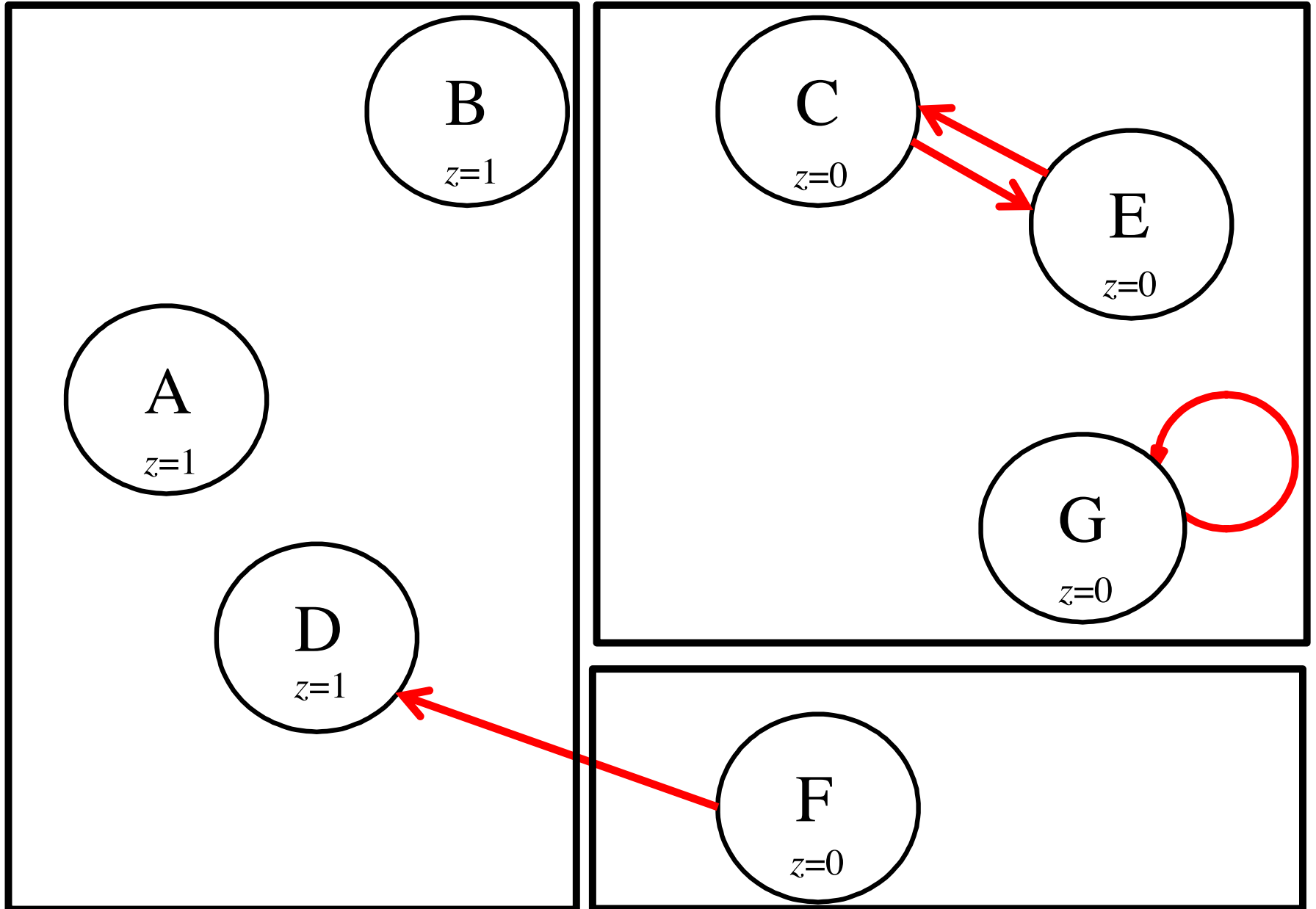
Partition #3.2

(Examine the 1-successors of CEFG)



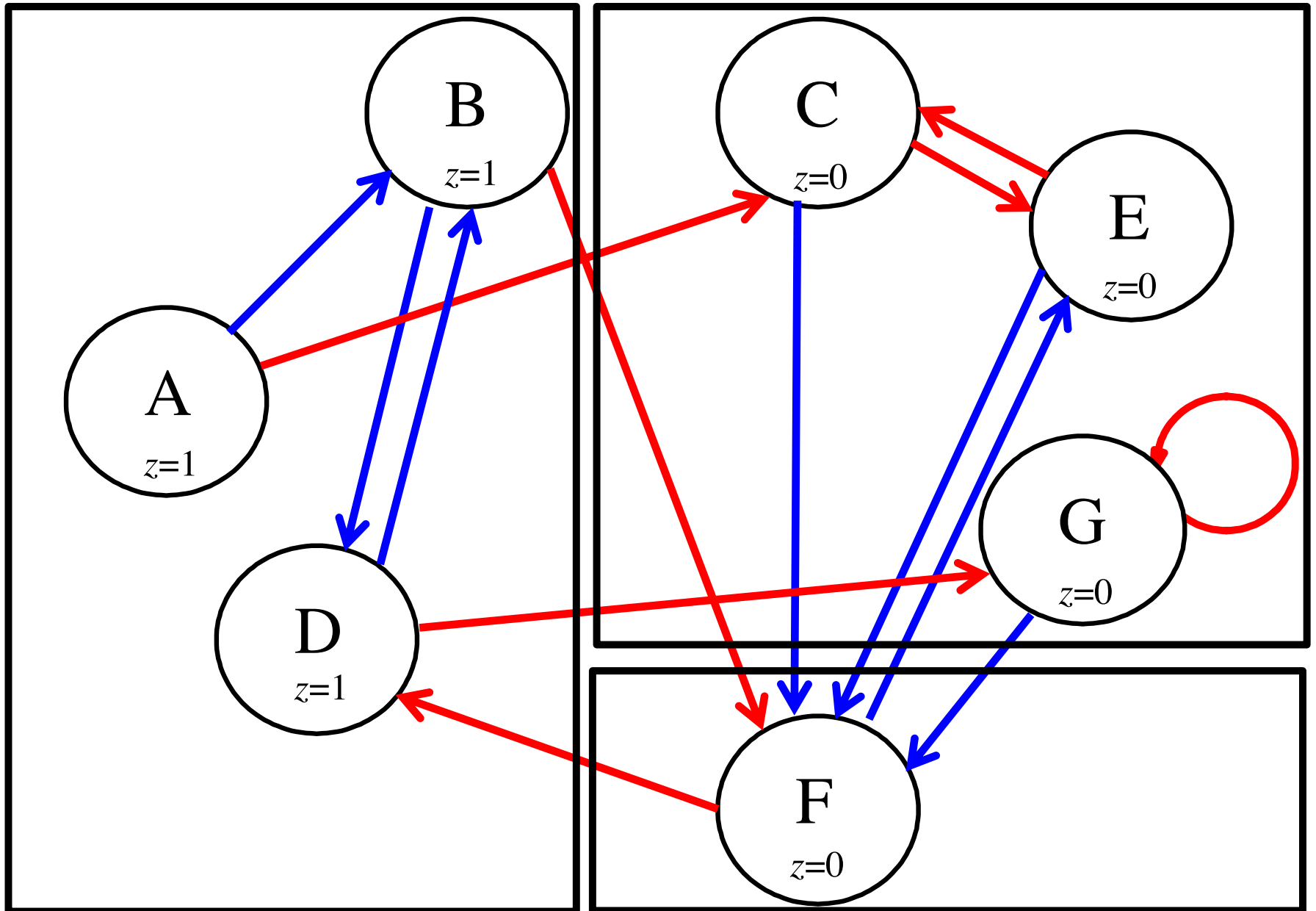
Partition #3

(ABD)(CEG)(F)



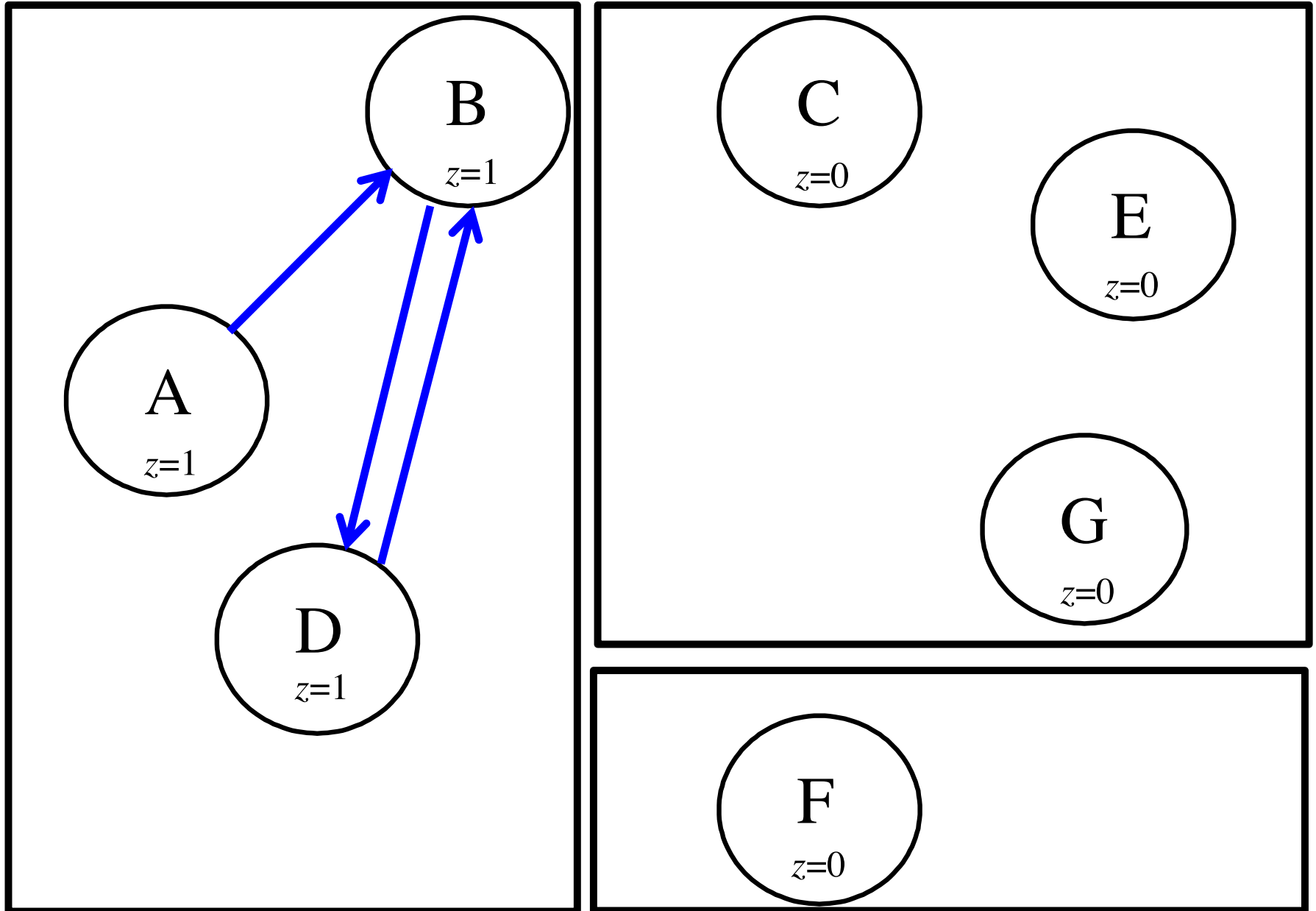
Partition #3

(ABD)(CEG)(F)



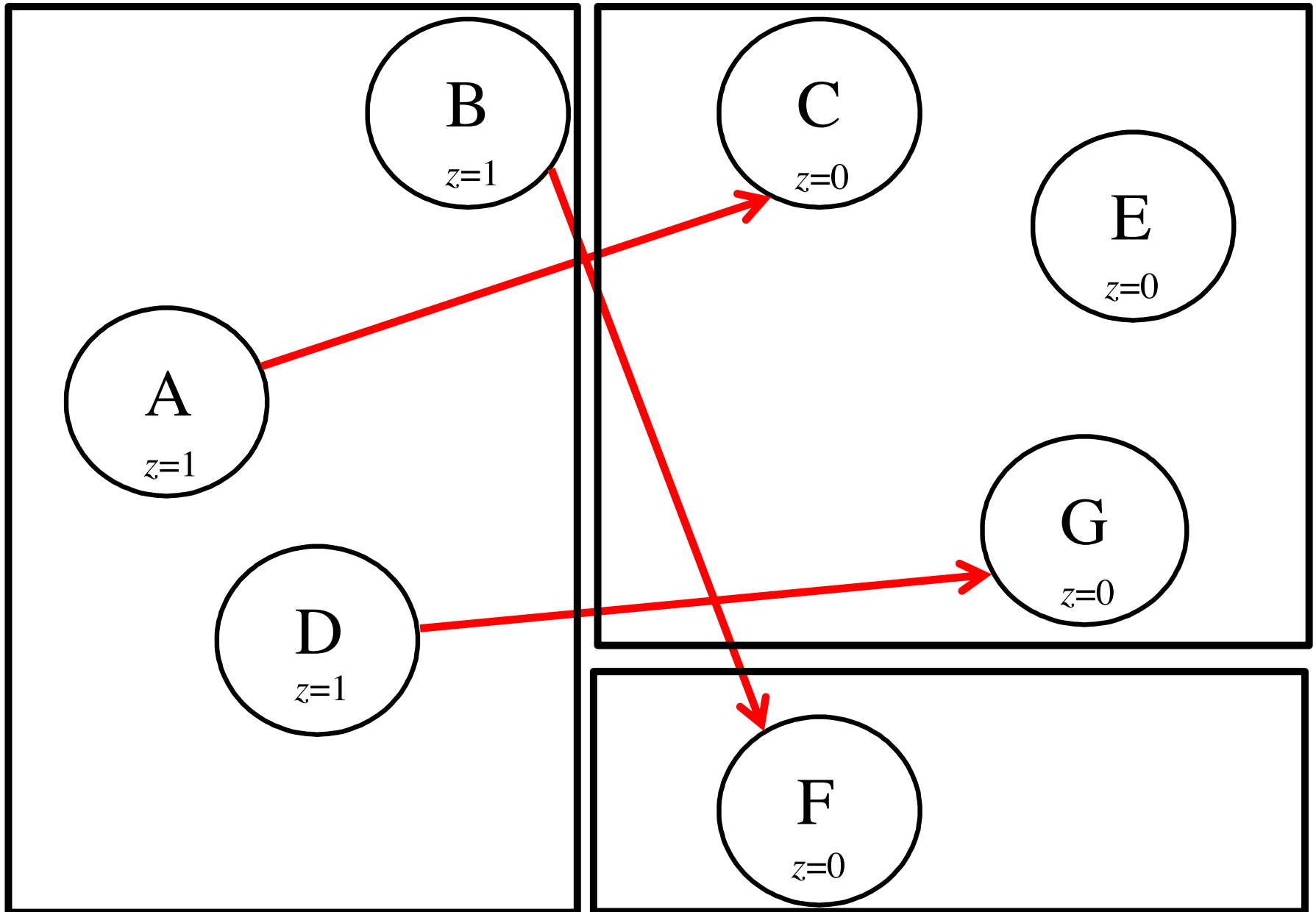
Partition #4.1

(Examine the 0-successors of ABD)



Partition #4.1

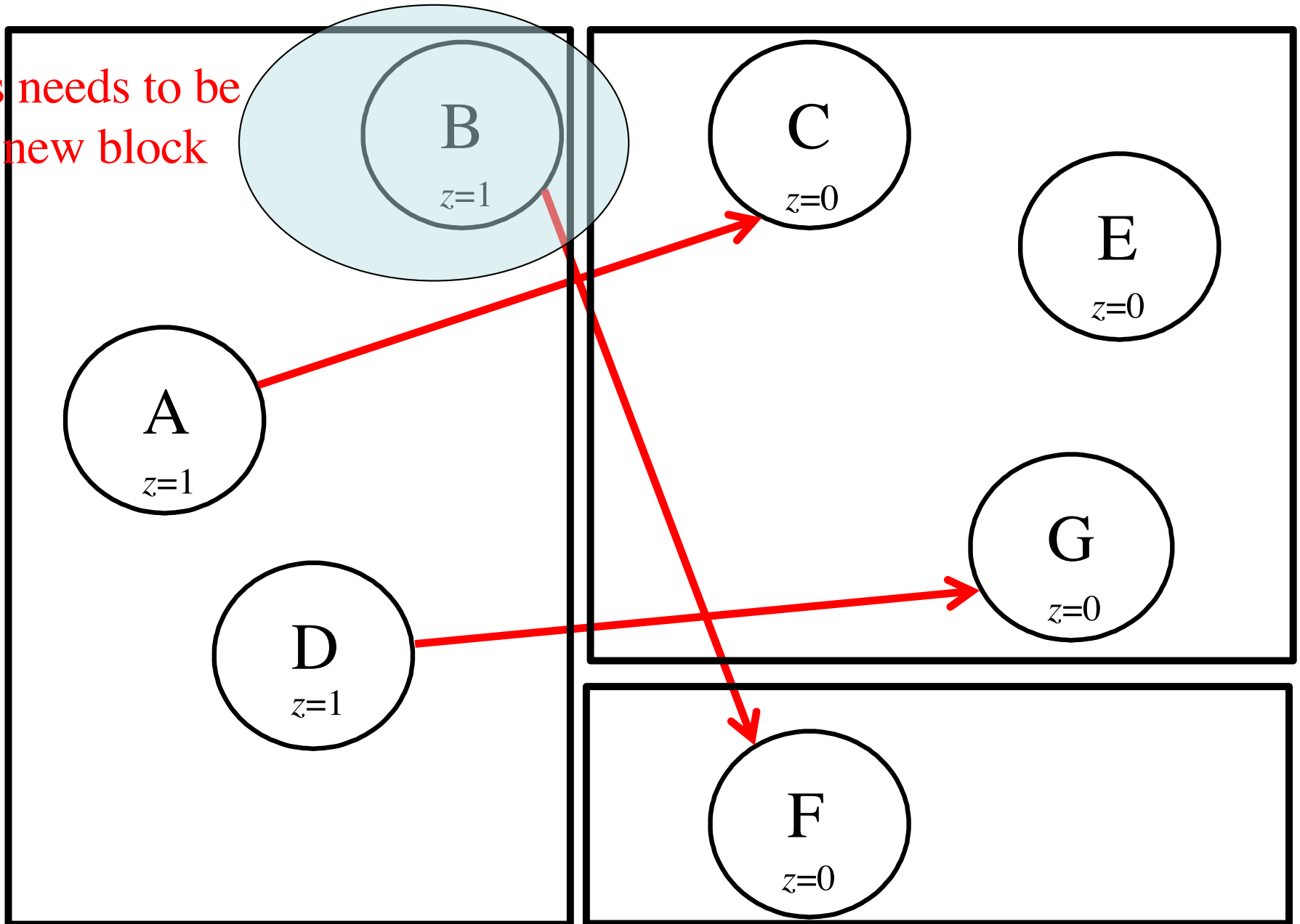
(Examine the 1-successors of ABD)



Partition #4.1

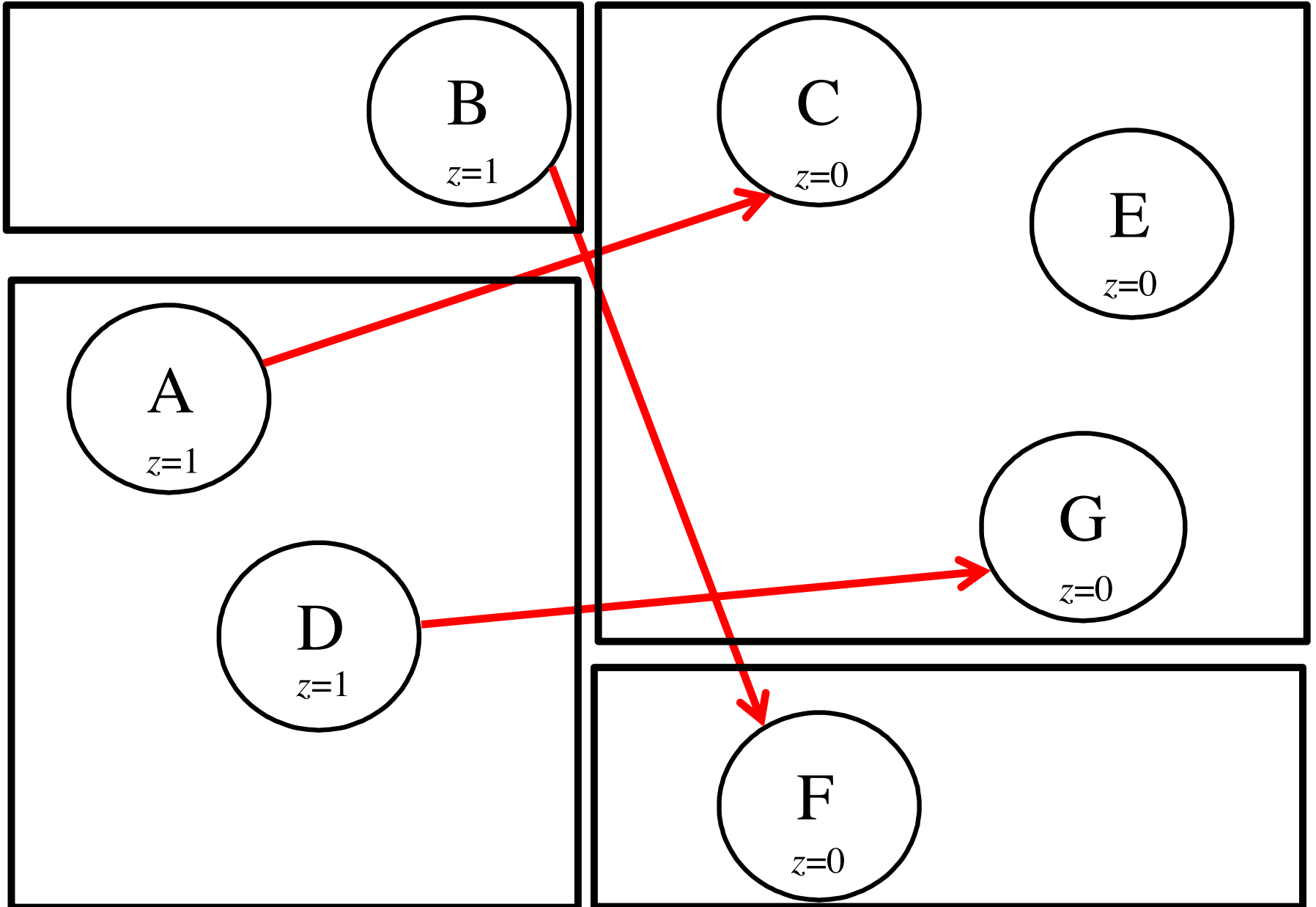
(Examine the 1-successors of ABD)

This needs to be
in a new block



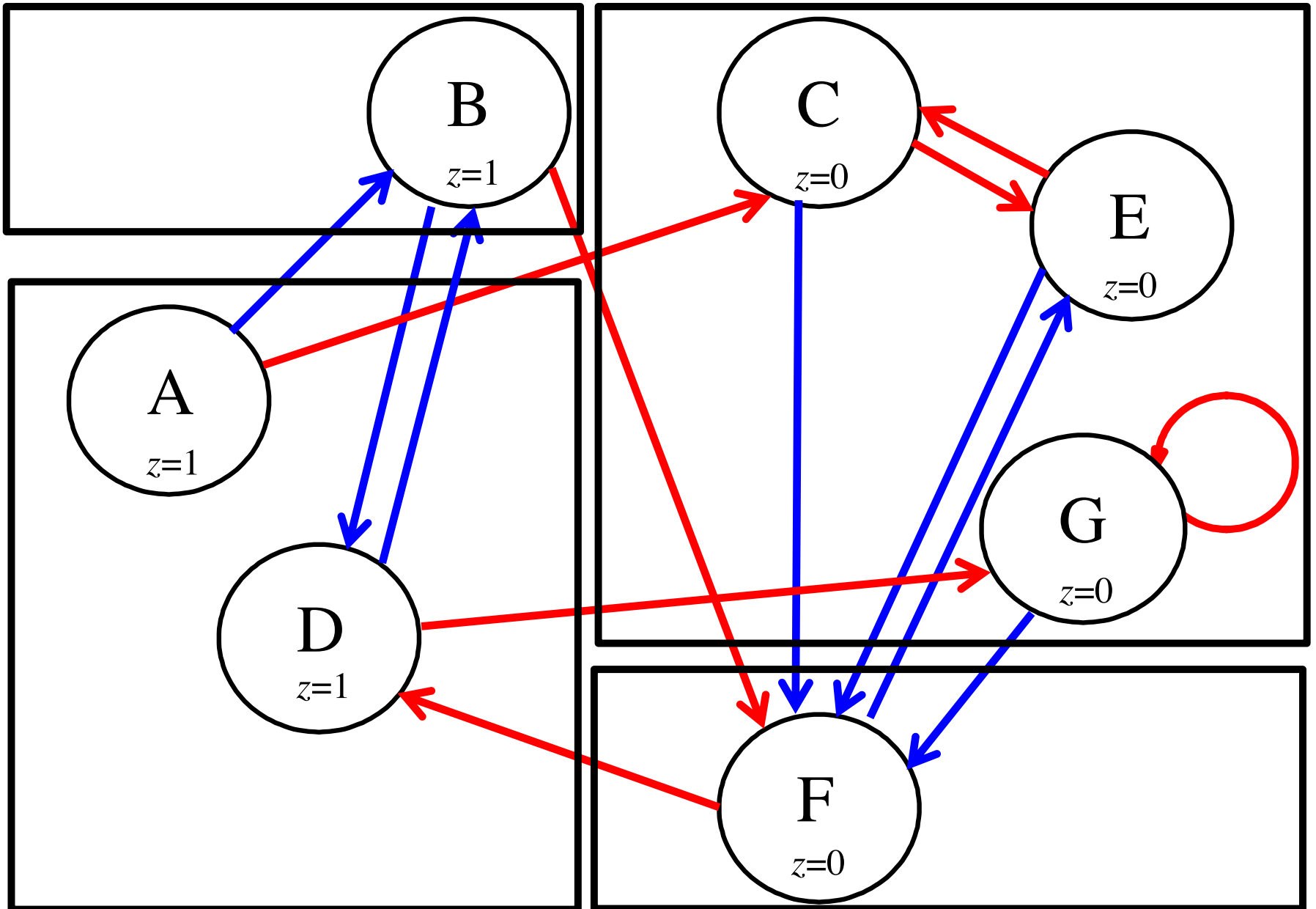
Partition #4

(AD)(B)(CEG)(F)



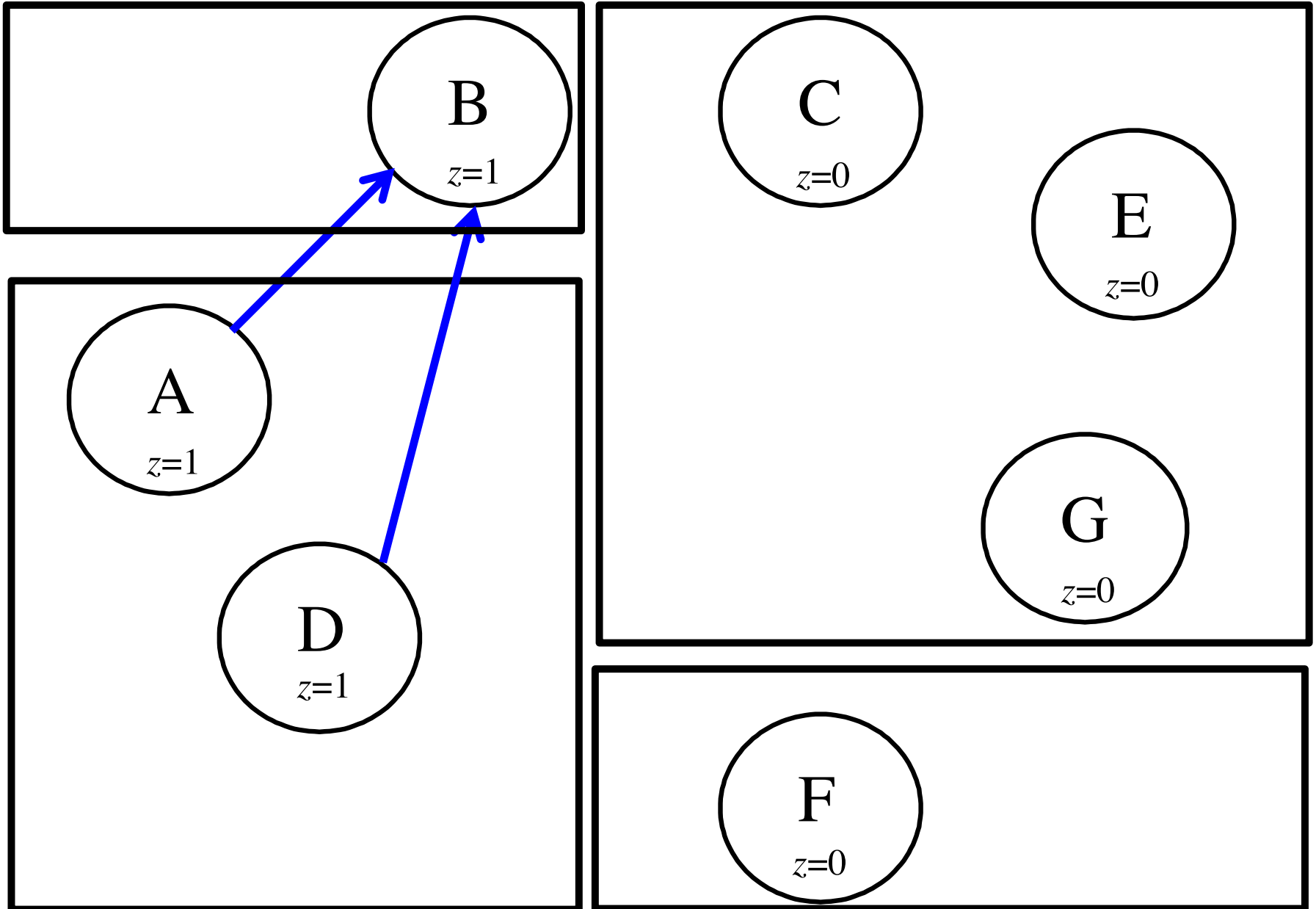
Partition #4

(AD)(B)(CEG)(F)



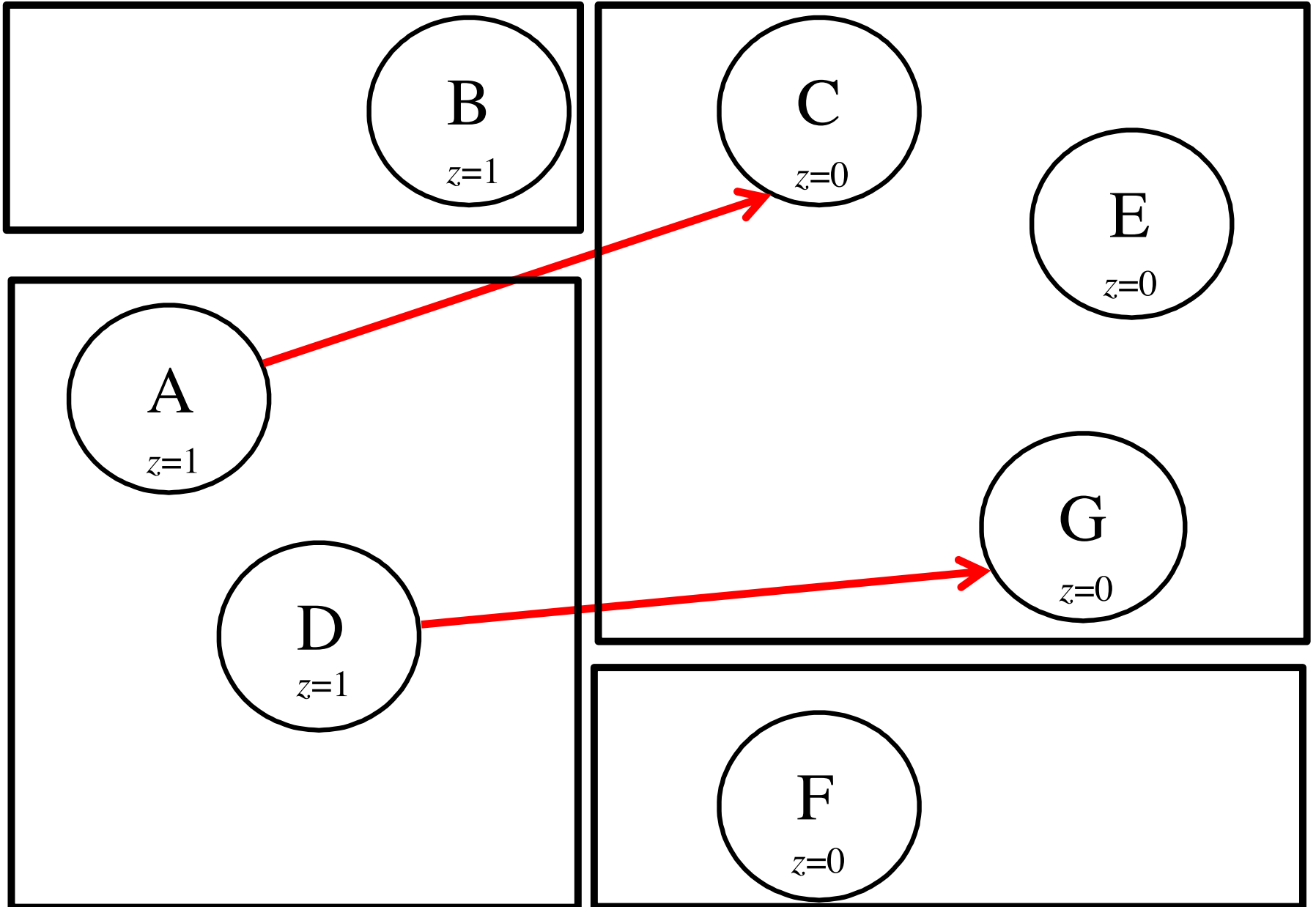
Partition #5.1

(Examine the 0-successors of AD)



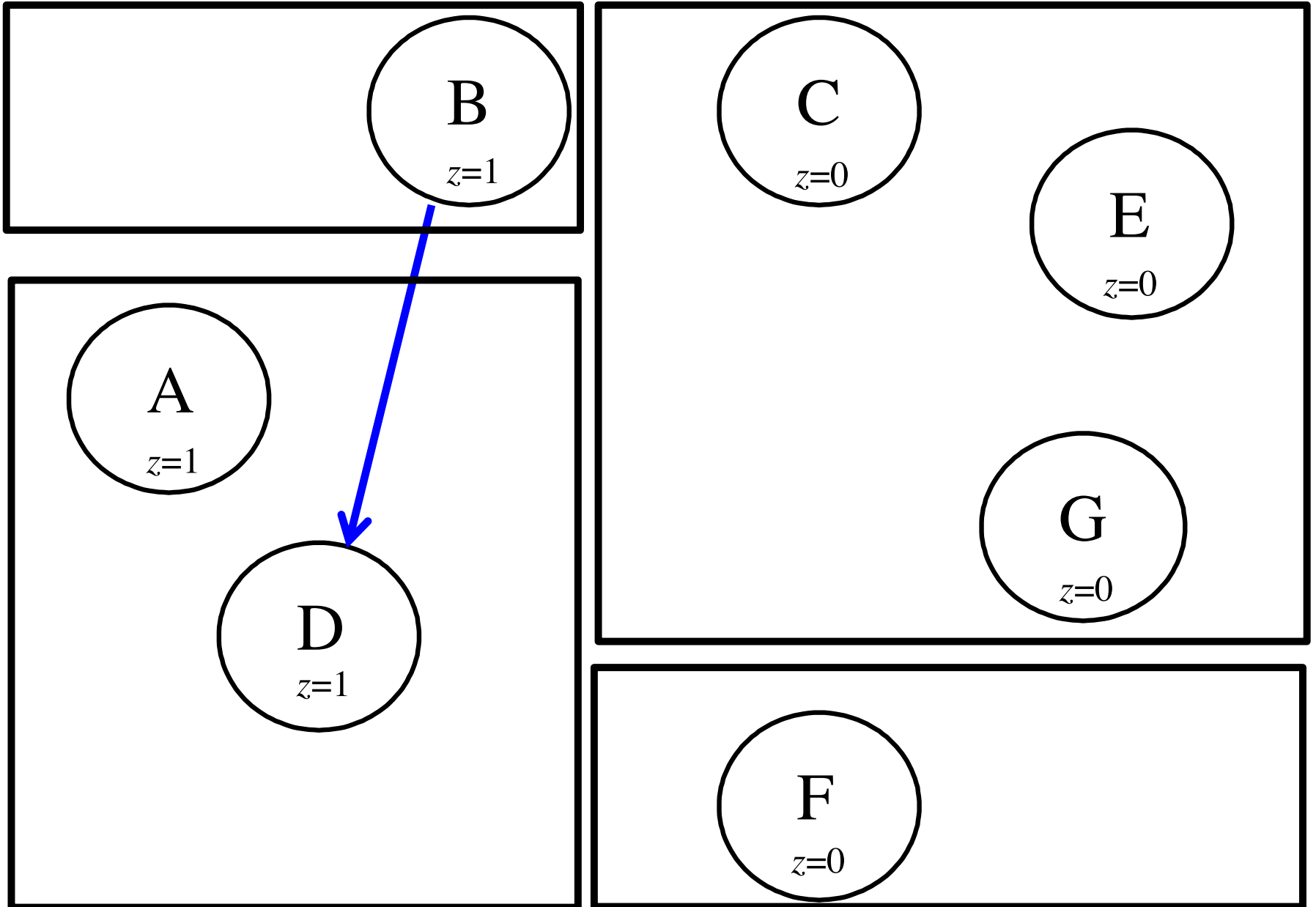
Partition #5.1

(Examine the 1-successors of AD)



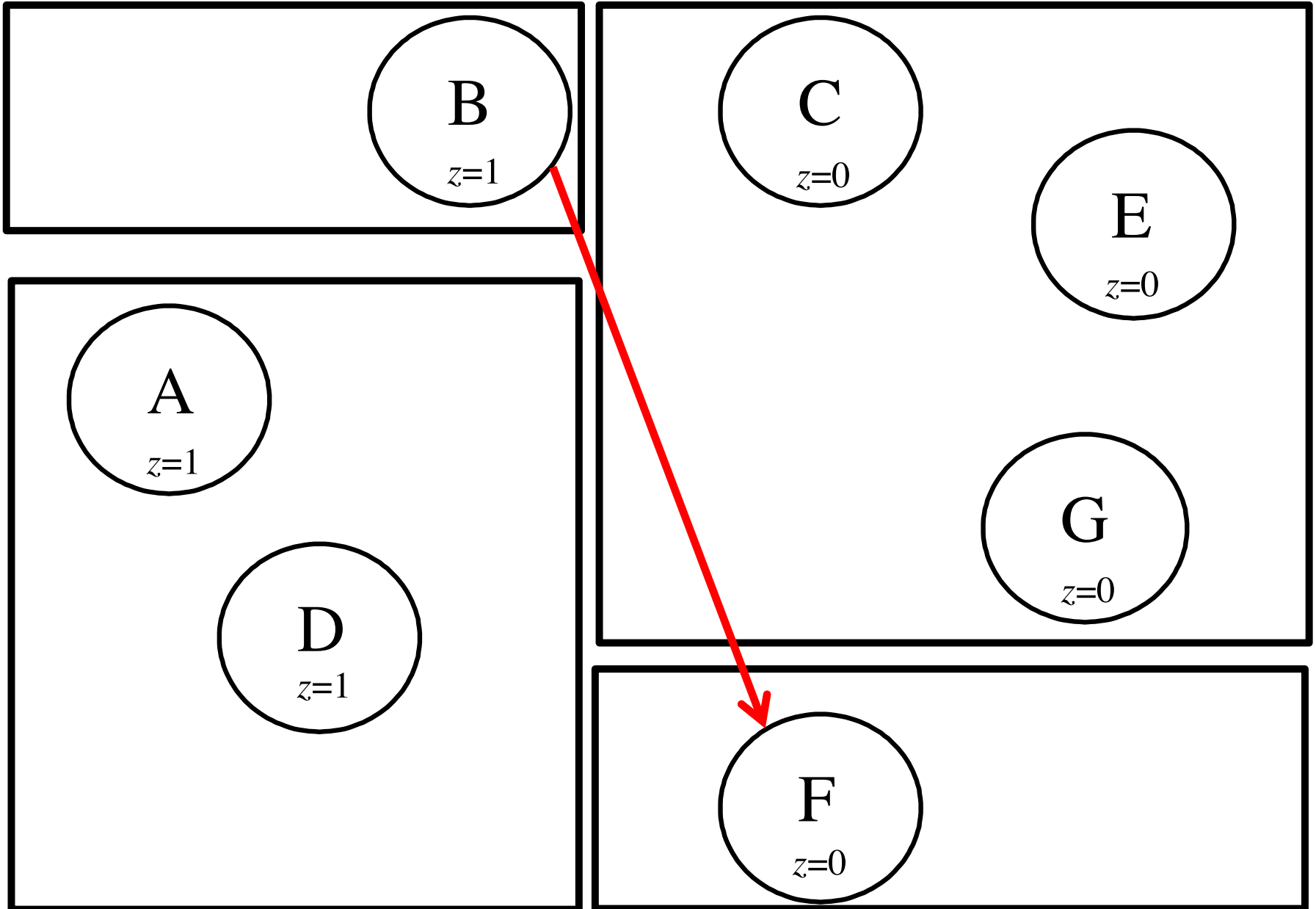
Partition #5.2

(Examine the 0-successors of B)



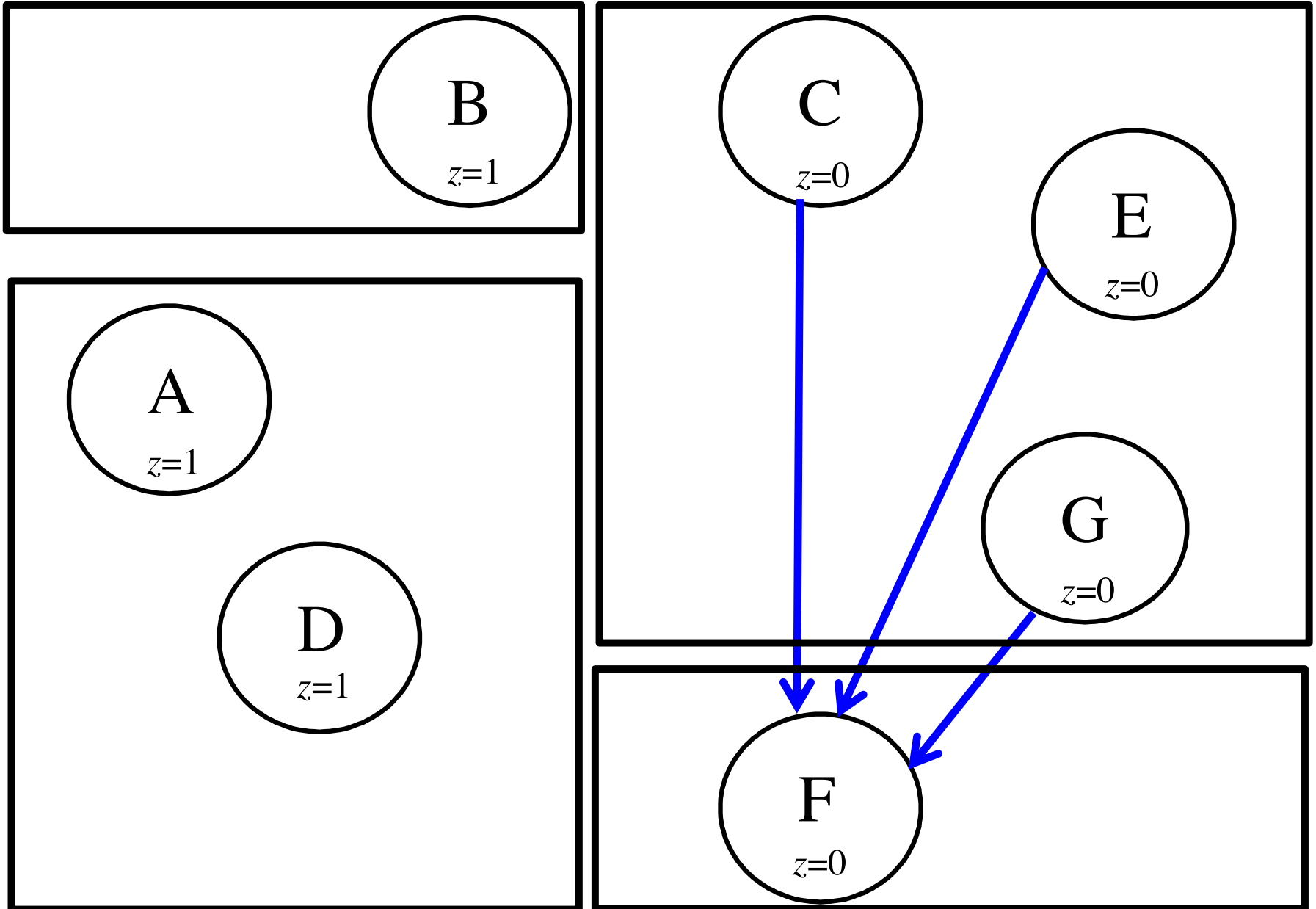
Partition #5.2

(Examine the 1-successors of B)



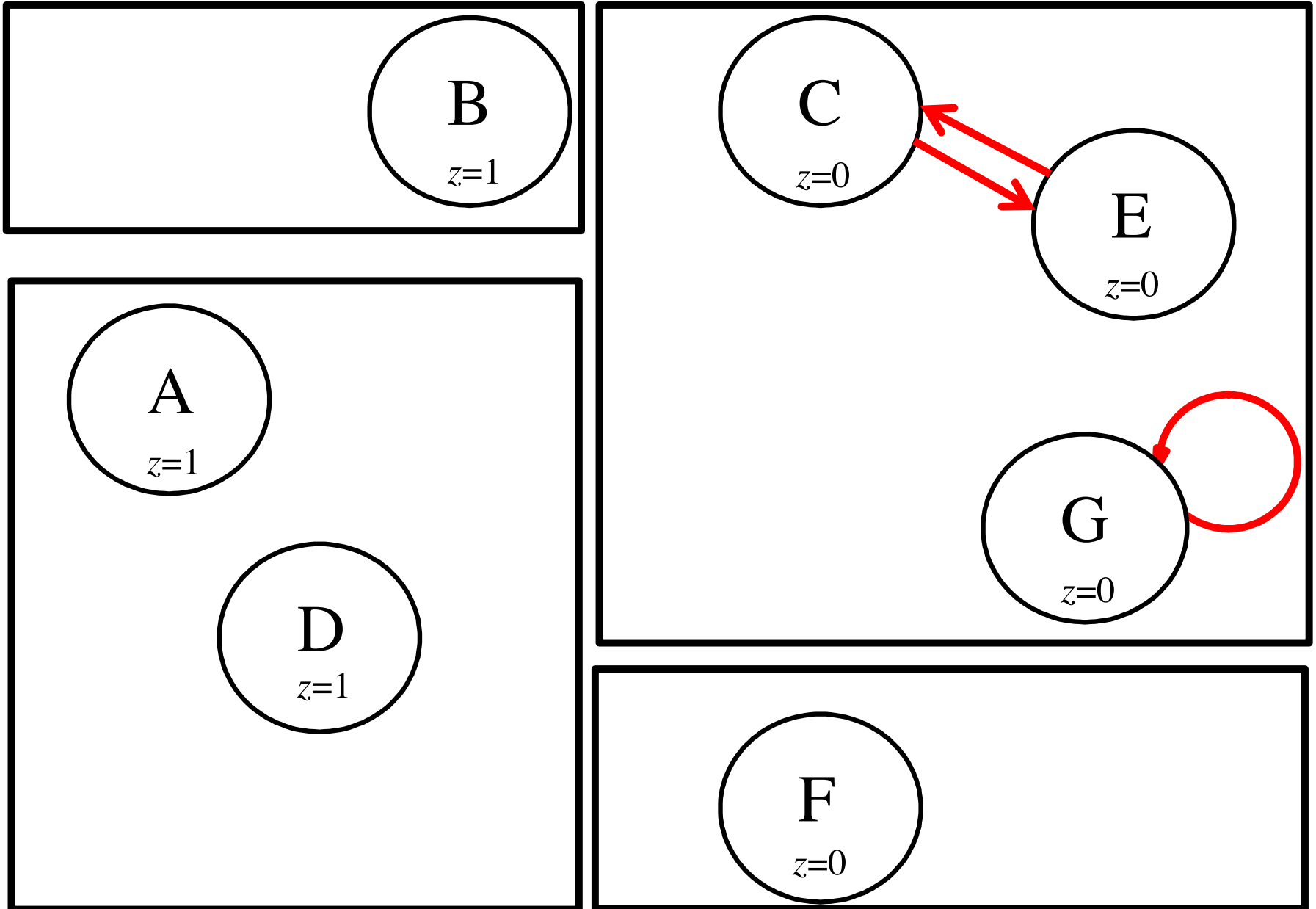
Partition #5.3

(Examine the 0-successors of CEG)



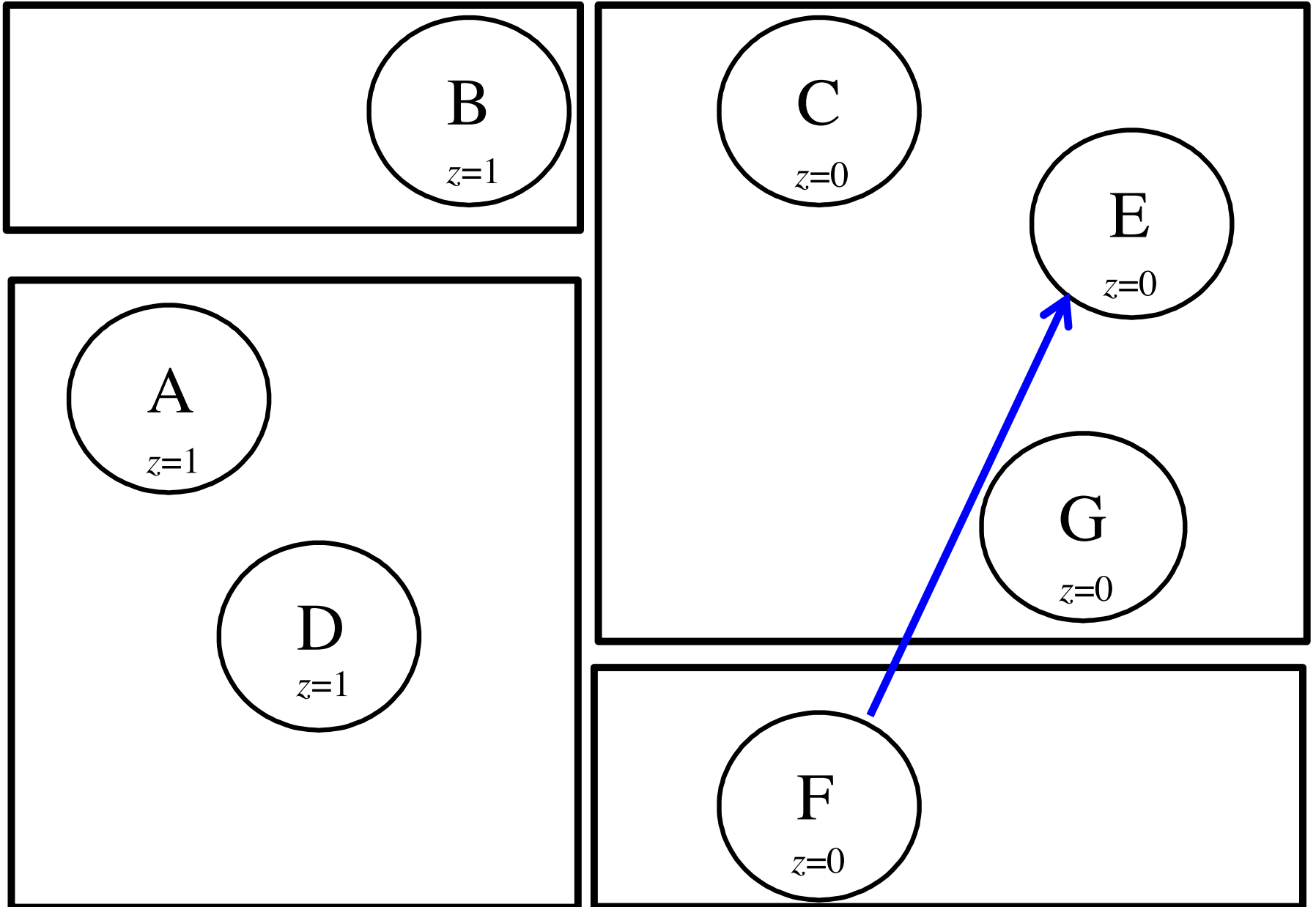
Partition #5.3

(Examine the 1-successors of CEG)



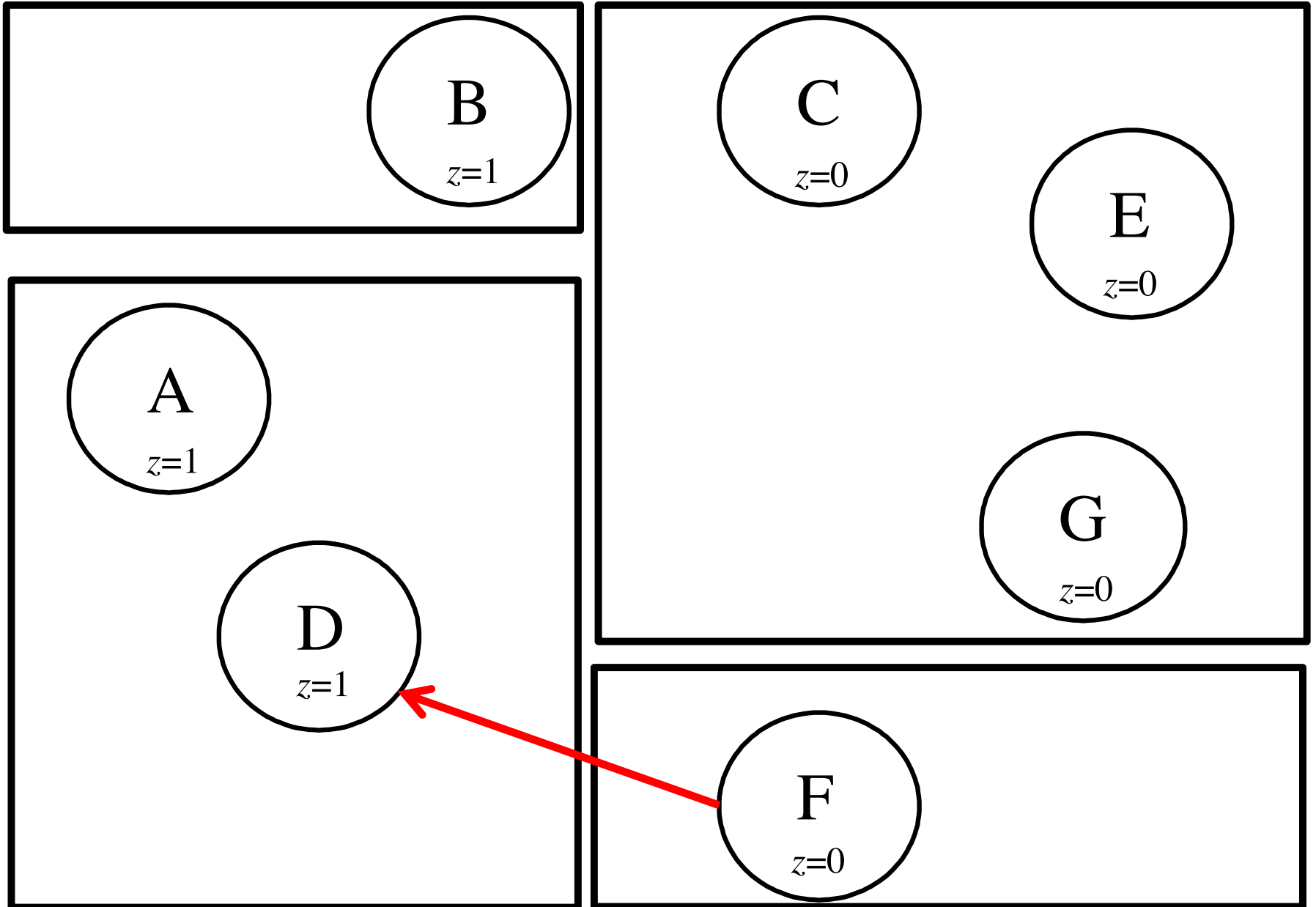
Partition #5.4

(Examine the 0-successors of F)



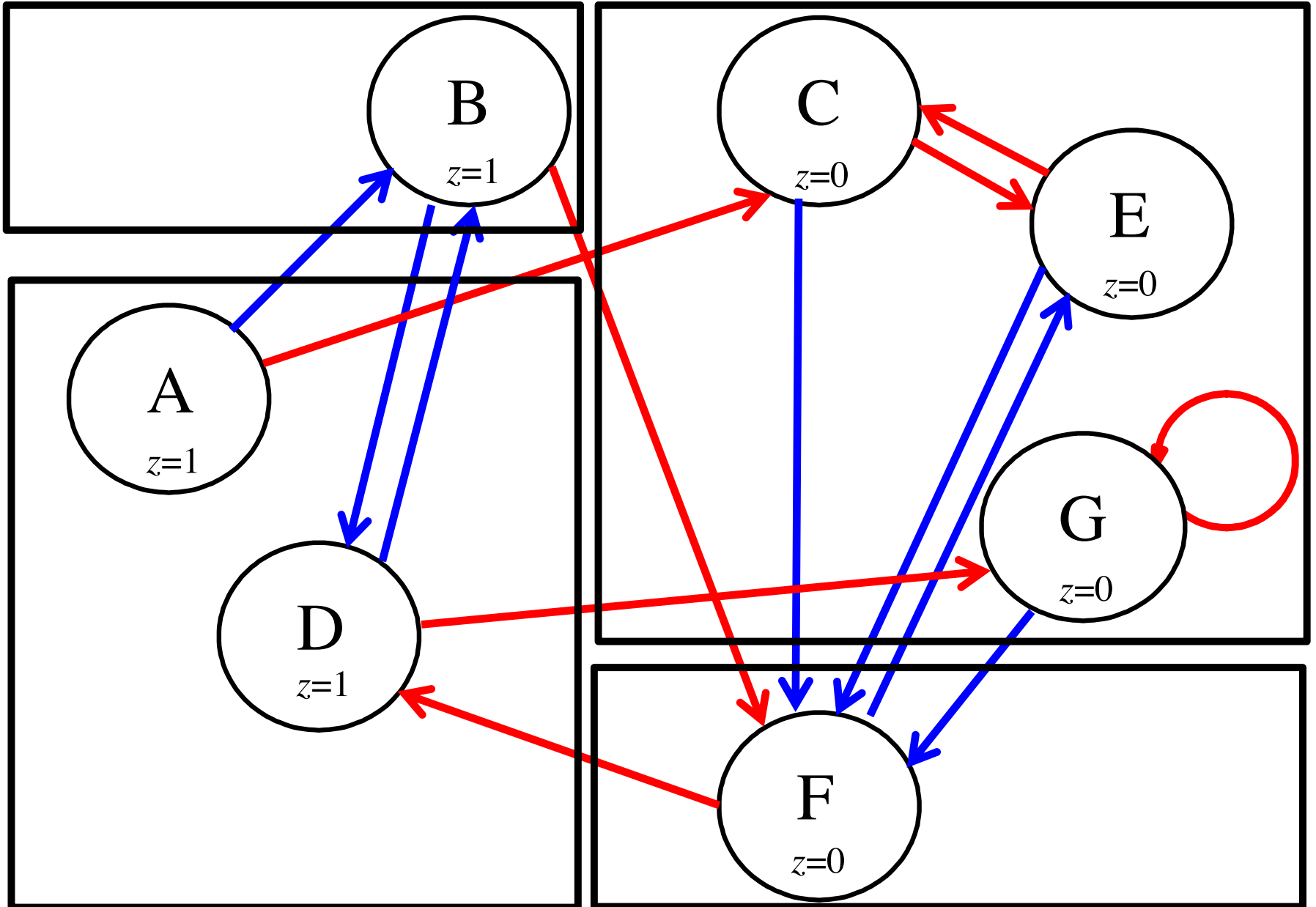
Partition #5.4

(Examine the 1-successors of F)



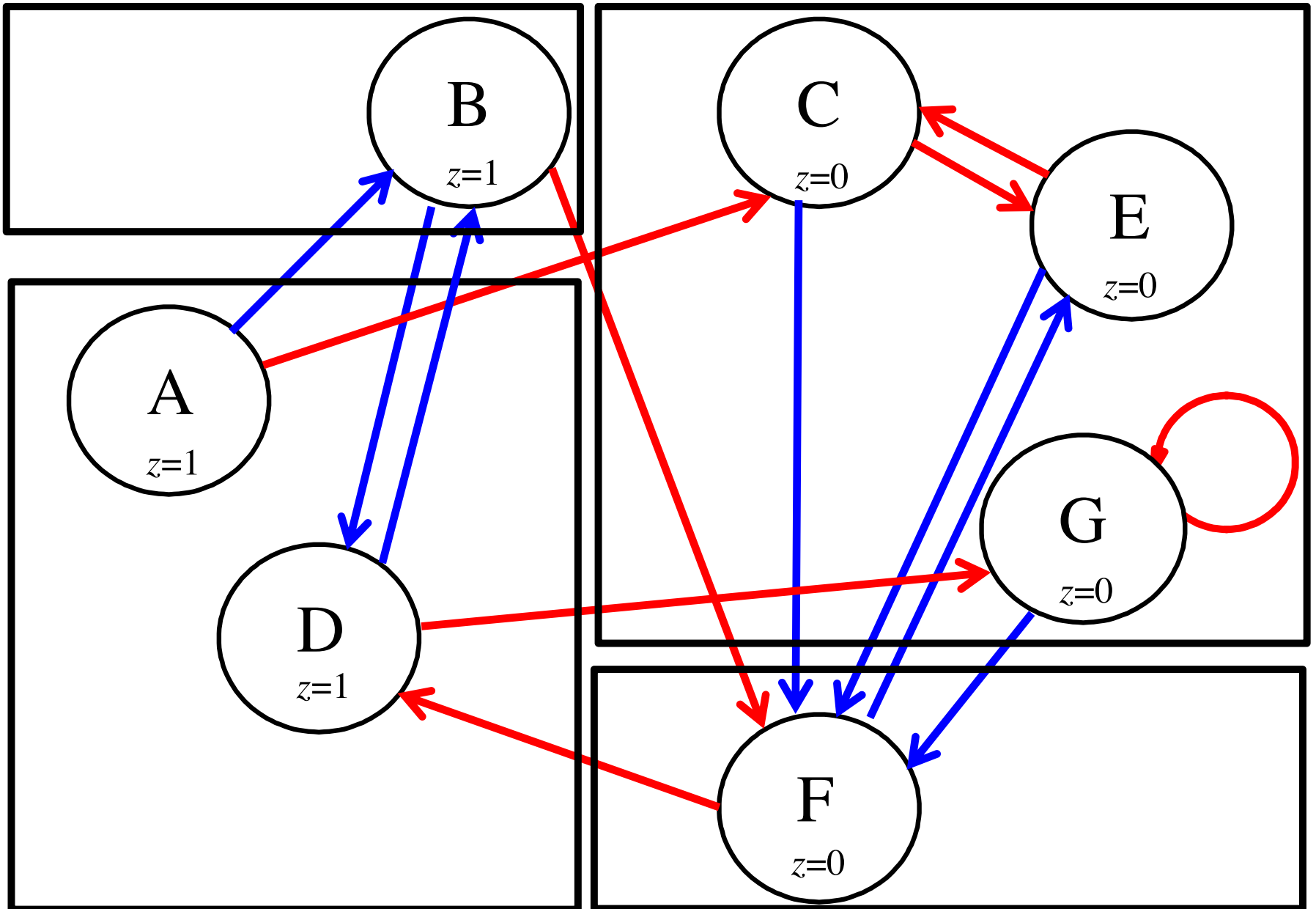
Partition #5

(AD)(B)(CEG)(F)



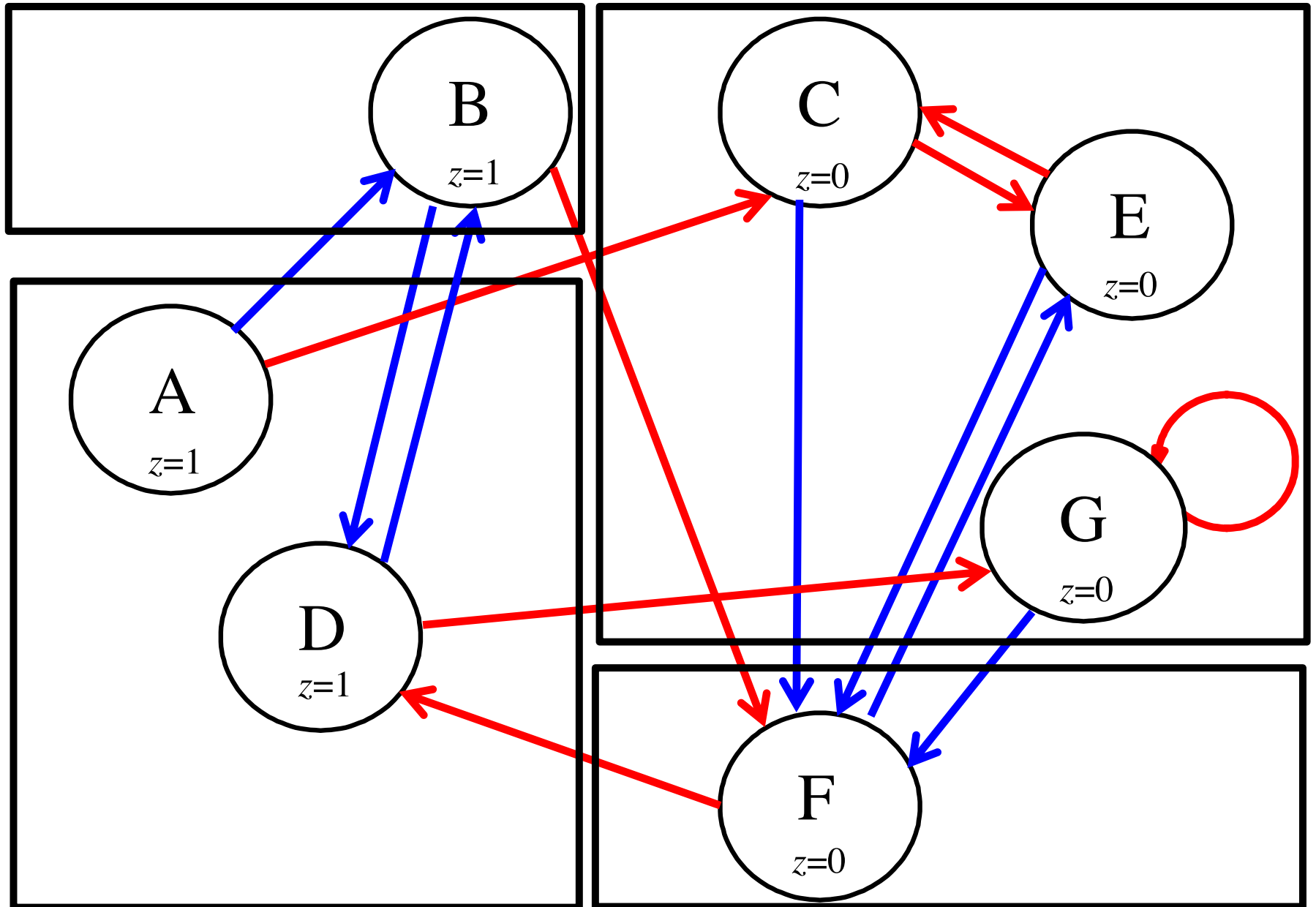
Partition #4

(AD)(B)(CEG)(F)

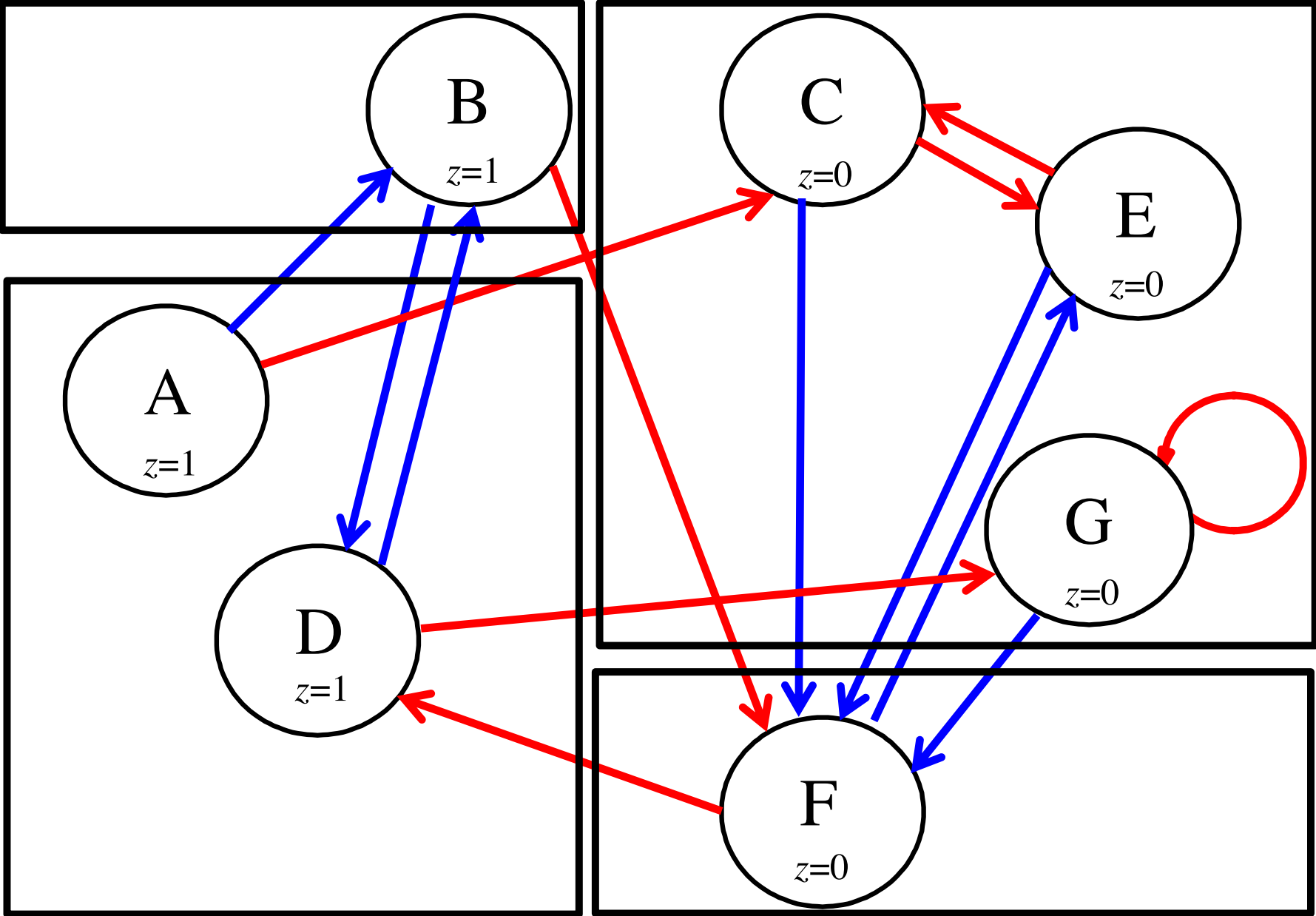


Partition #5

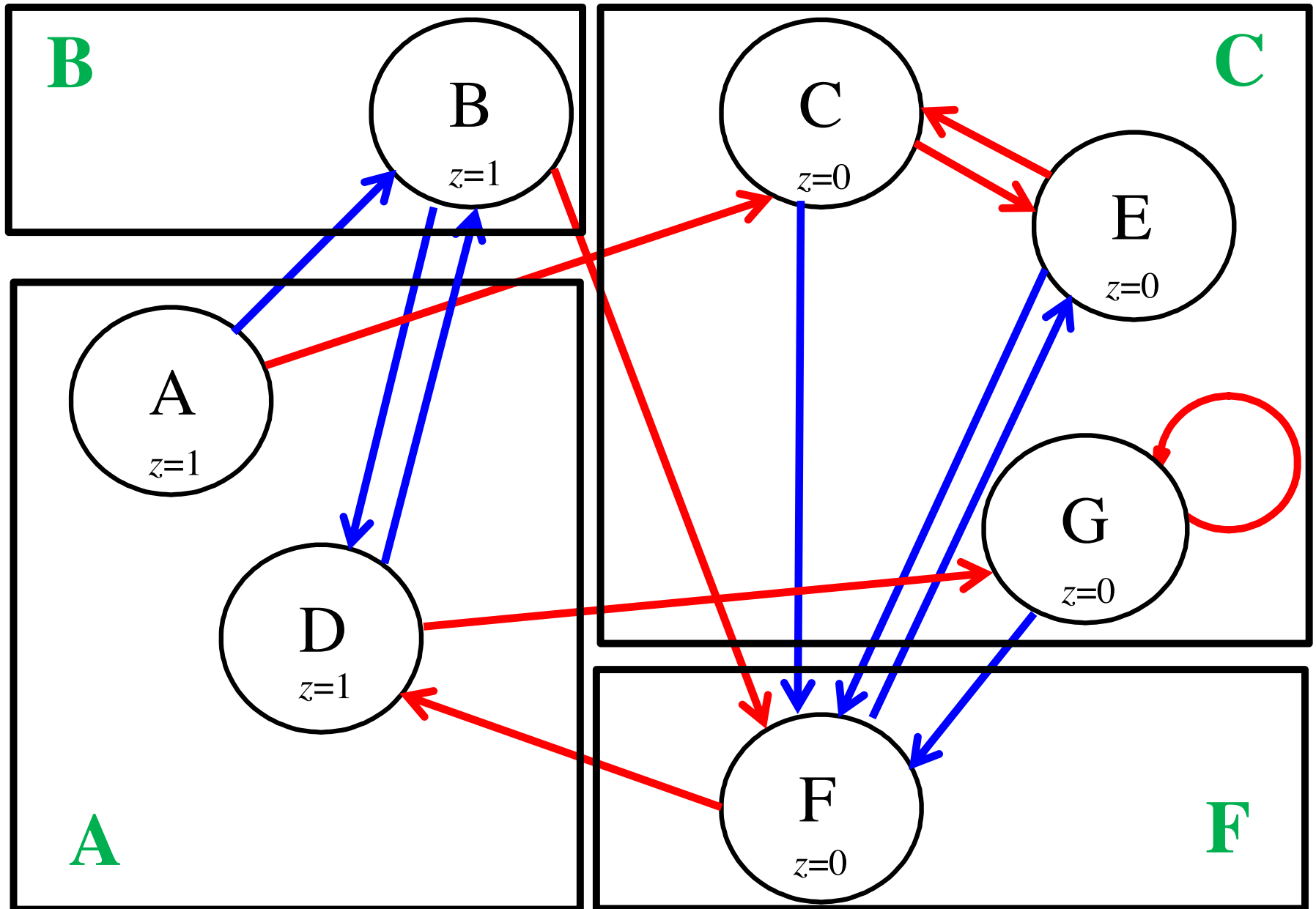
(This is the same as #4 so we can stop here)



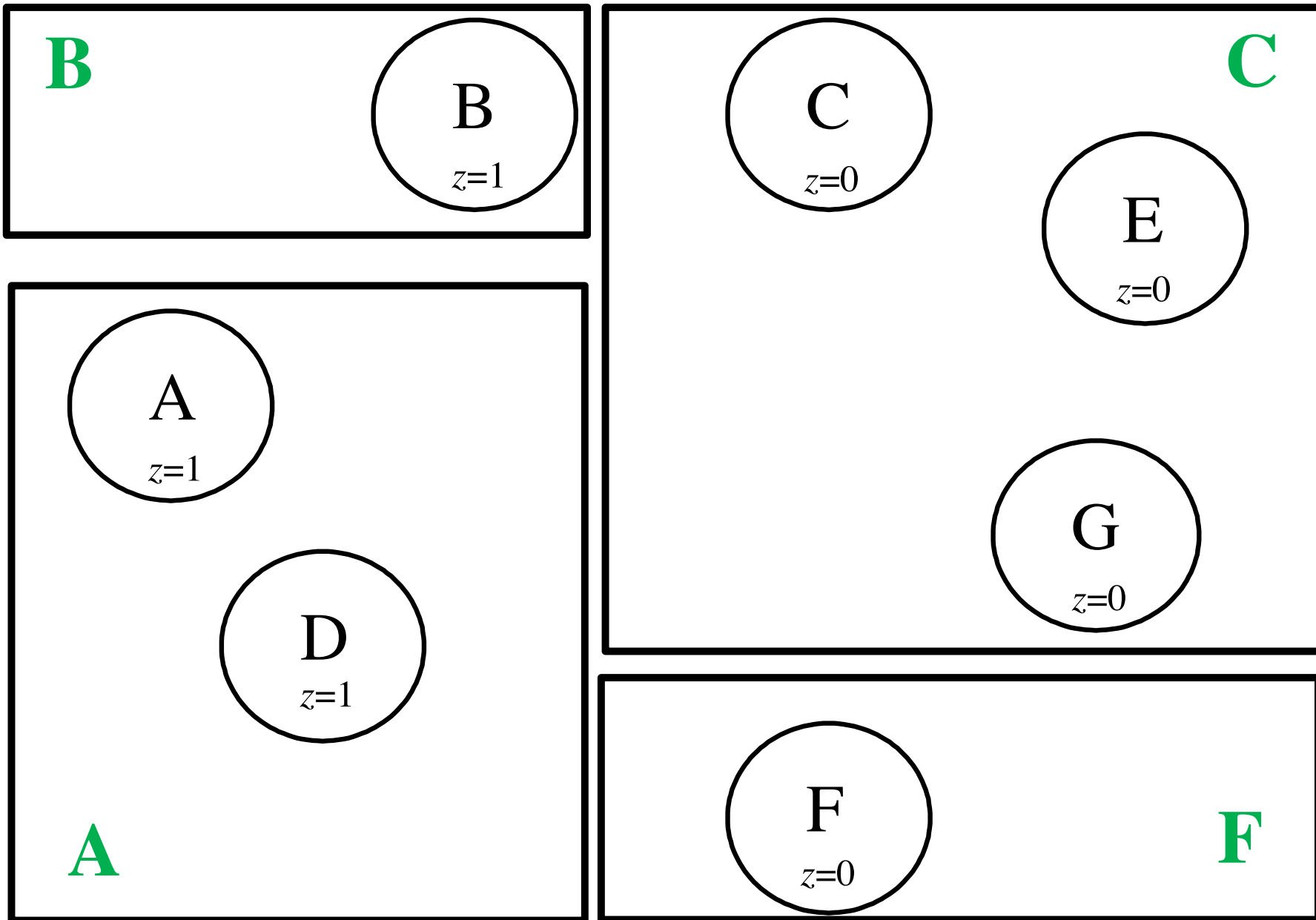
Stop Here ...



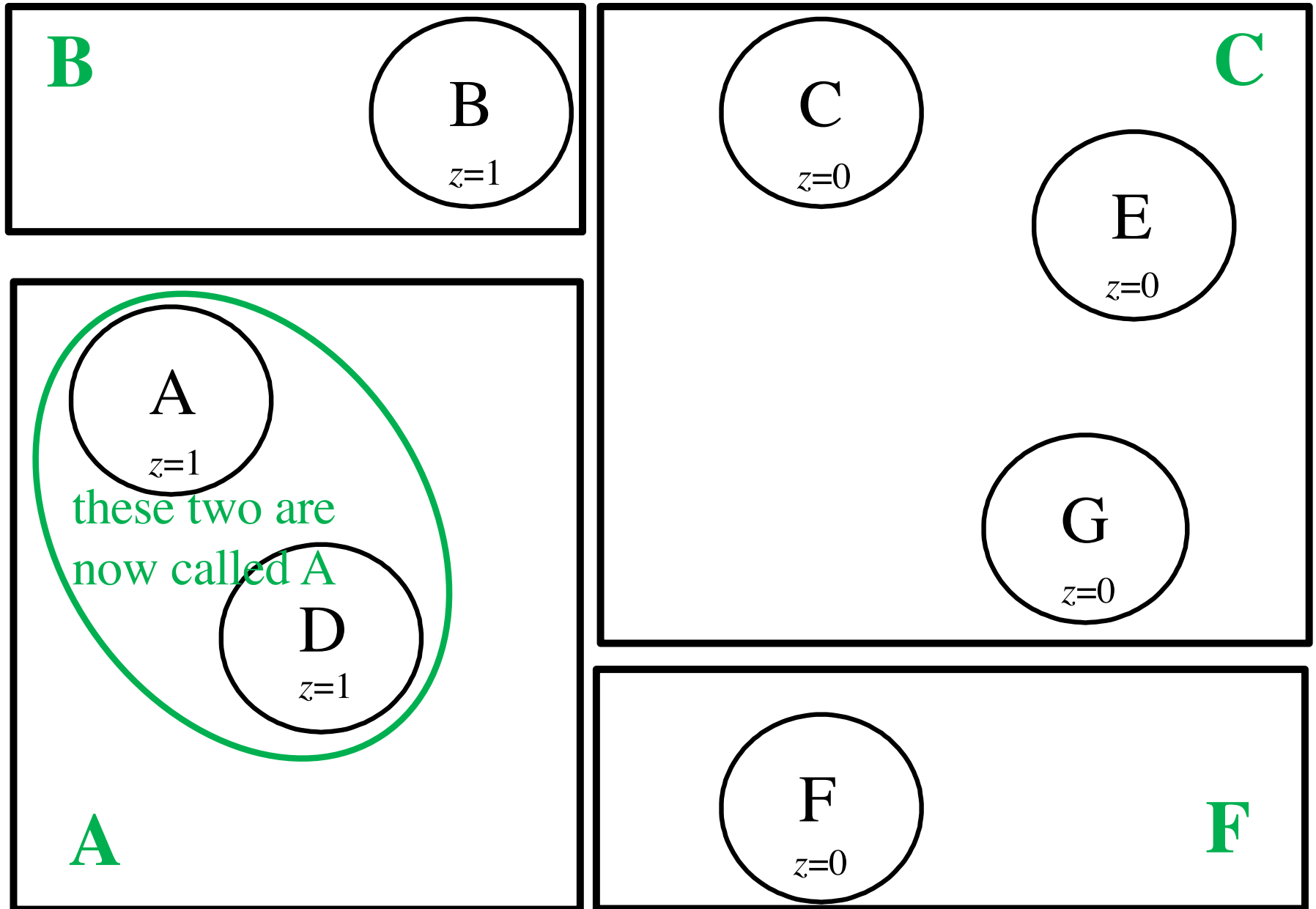
... and Relabel All Partitions



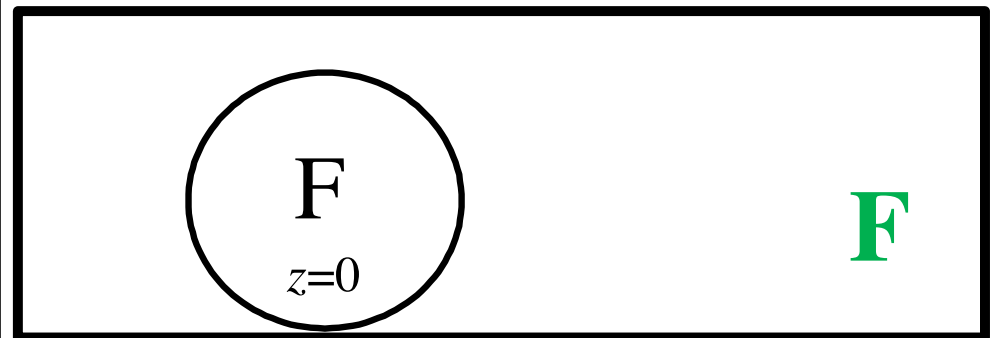
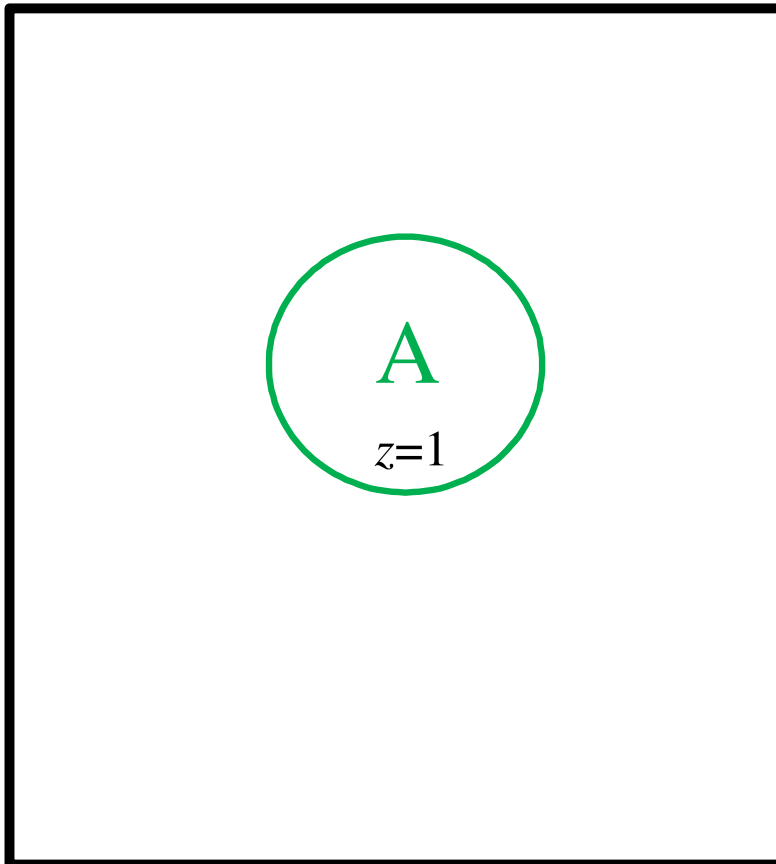
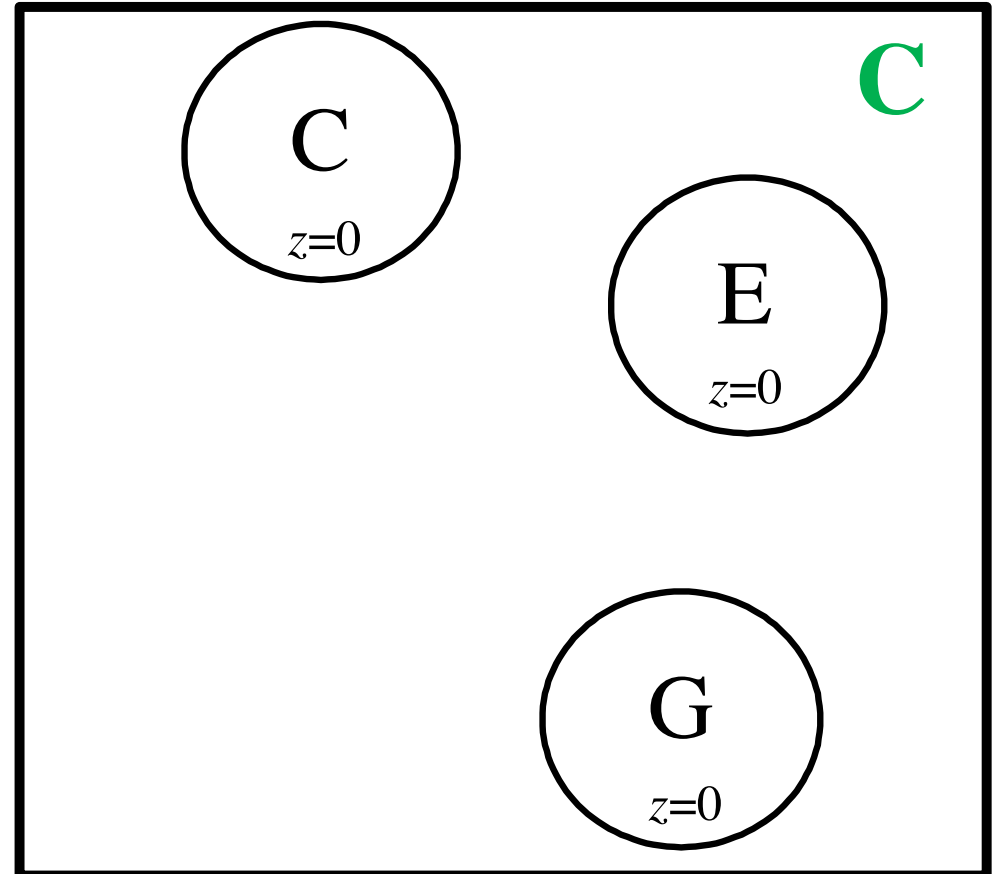
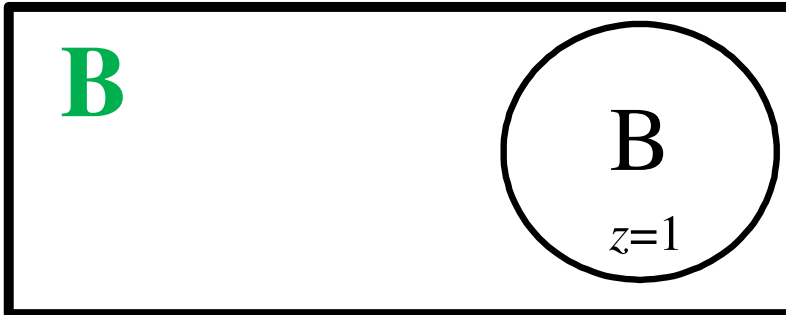
... and Relabel All Partitions



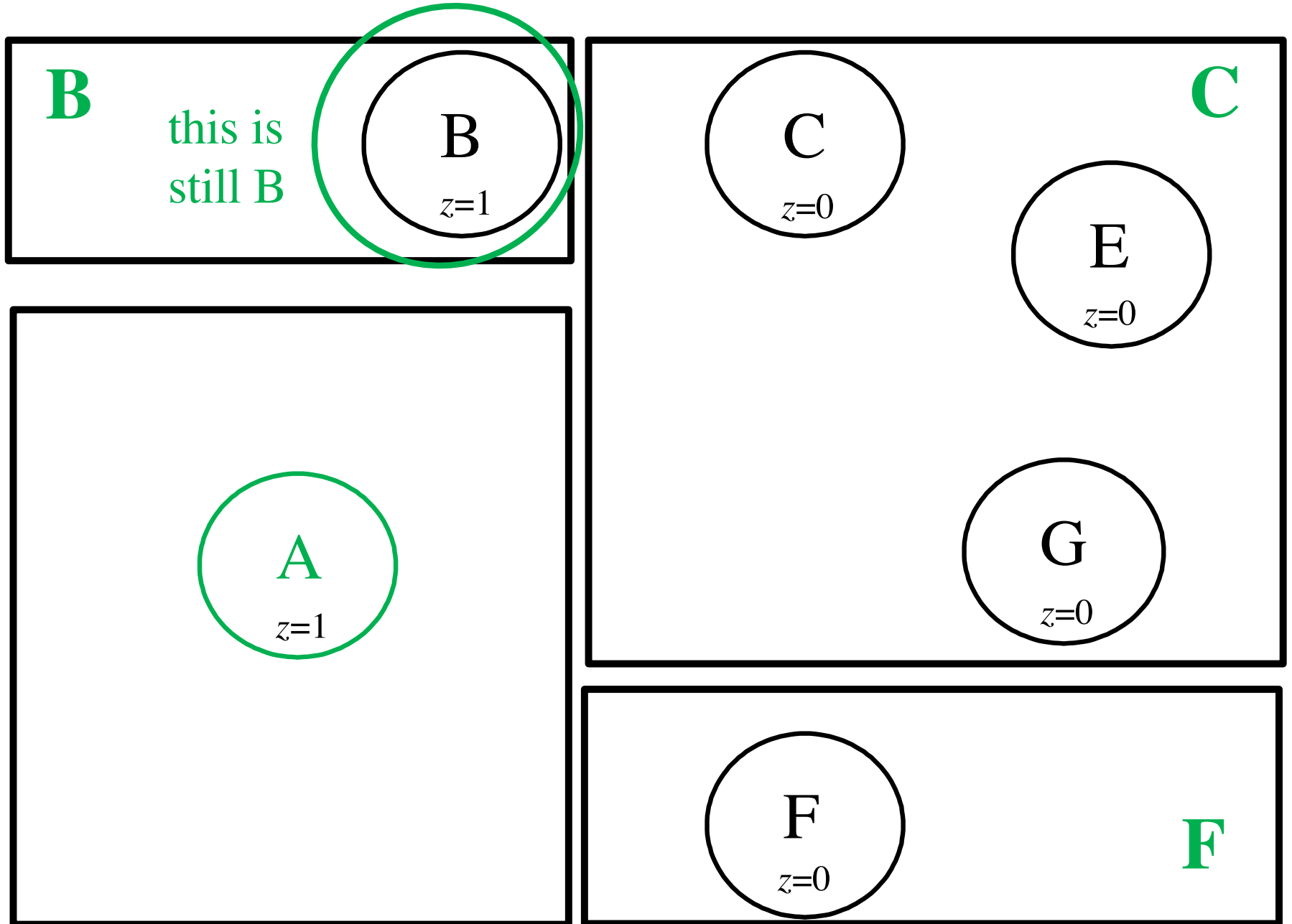
Merge the states in the same partition



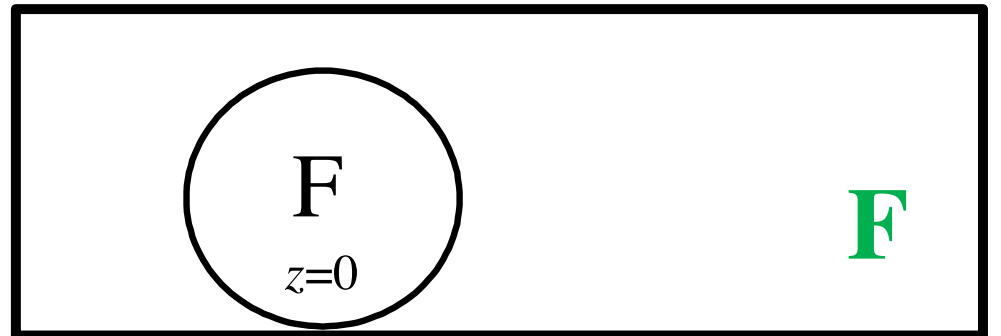
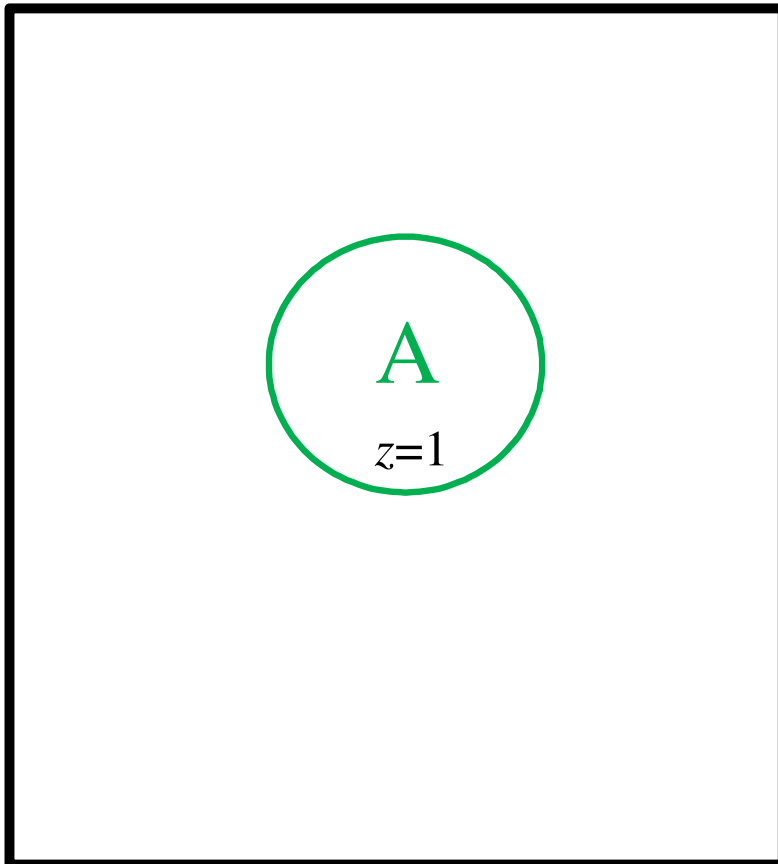
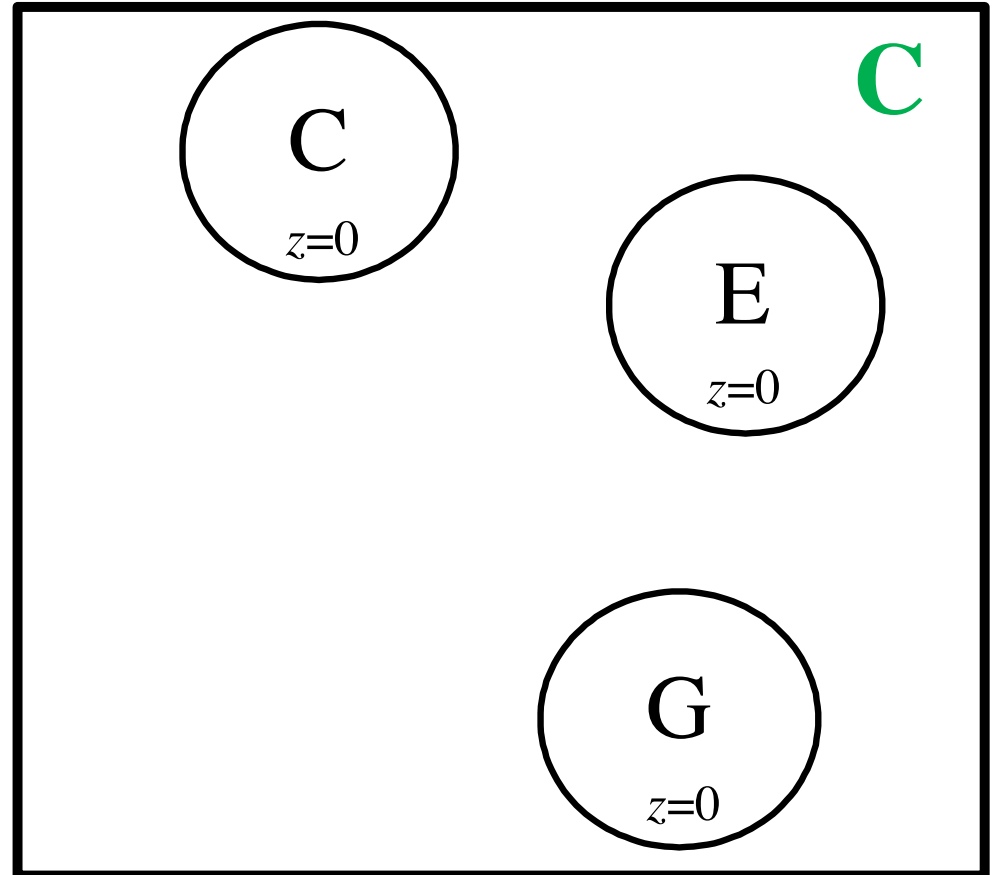
Merge the states in the same partition



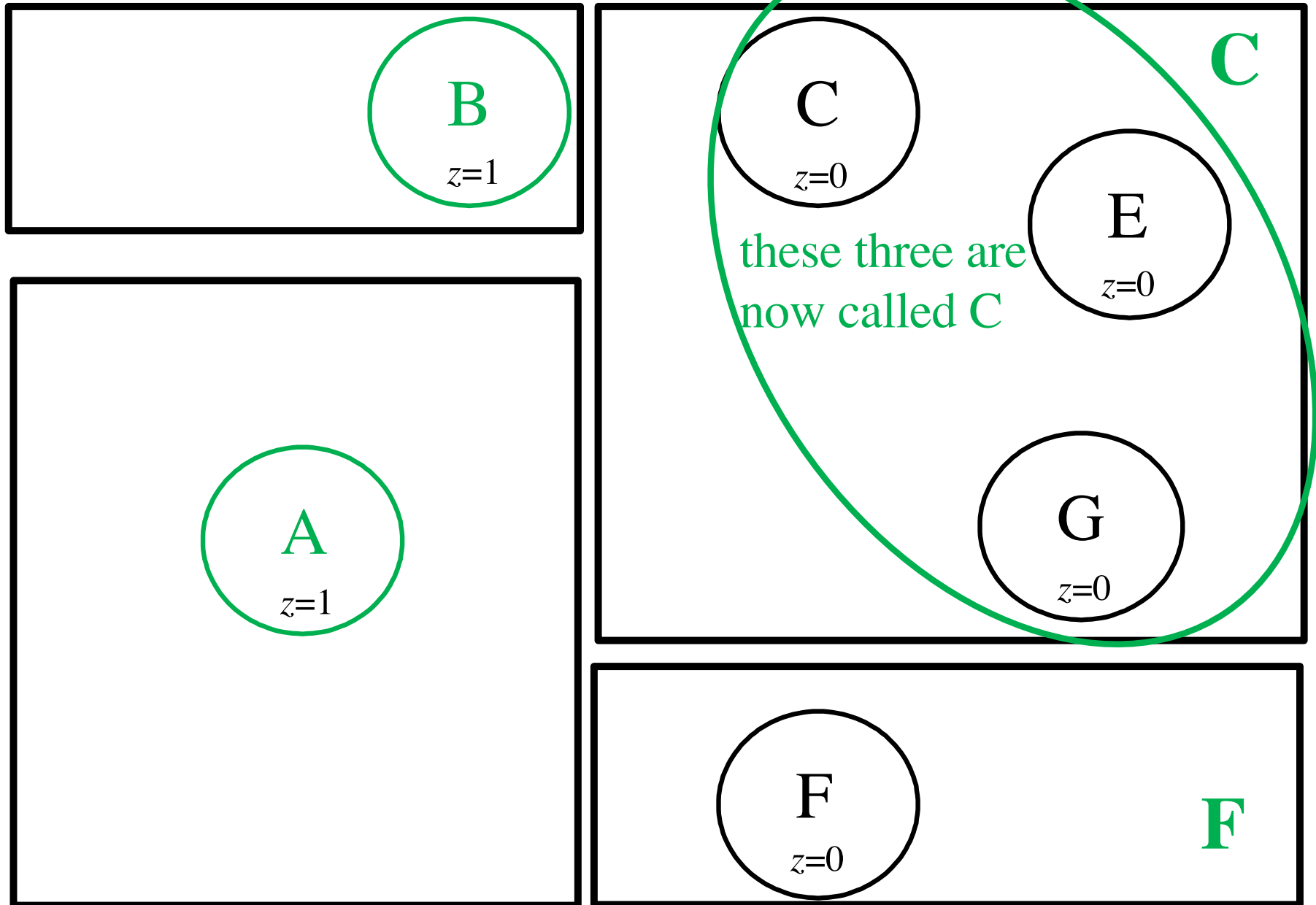
Merge the states in the same partition



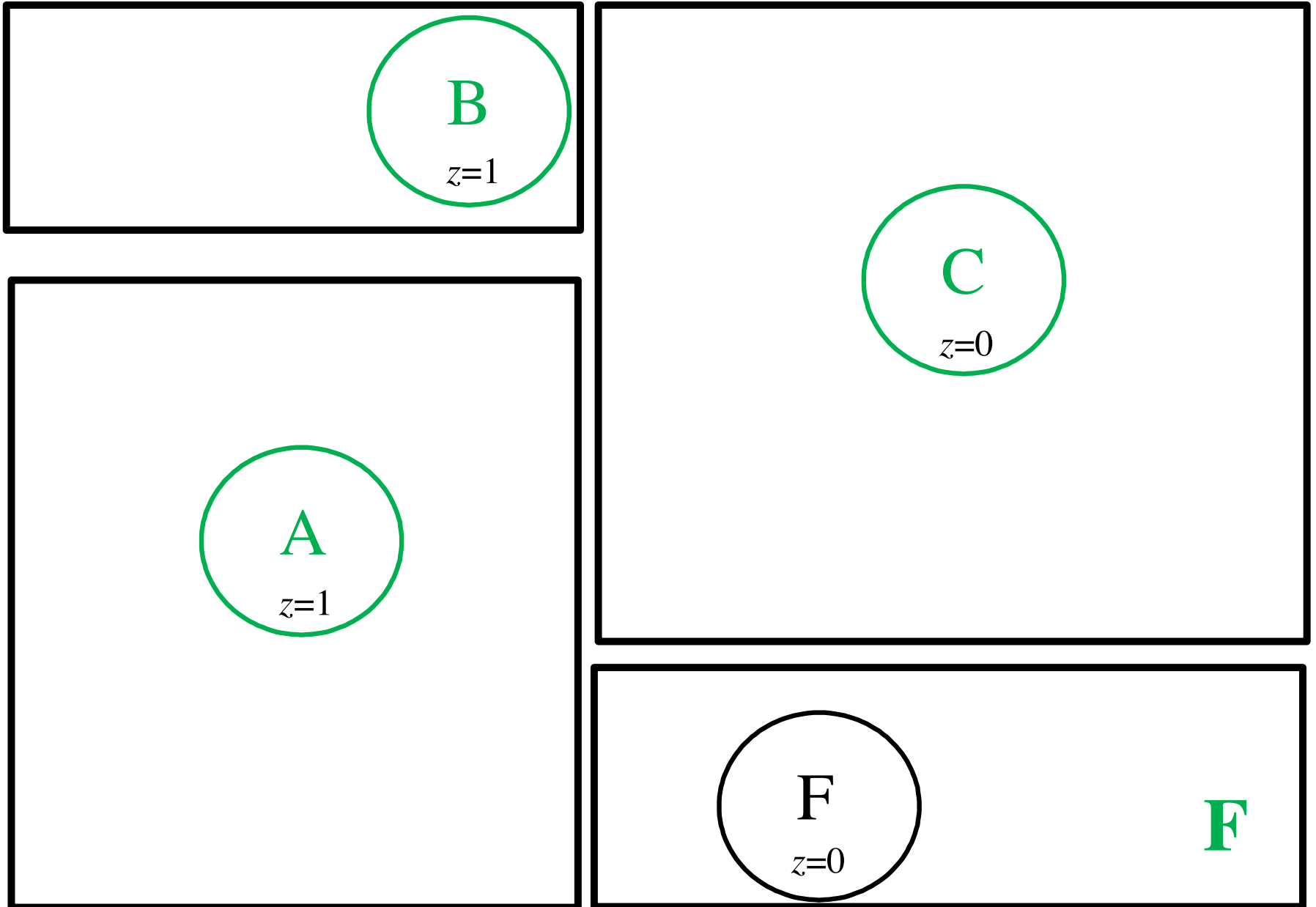
Merge the states in the same partition



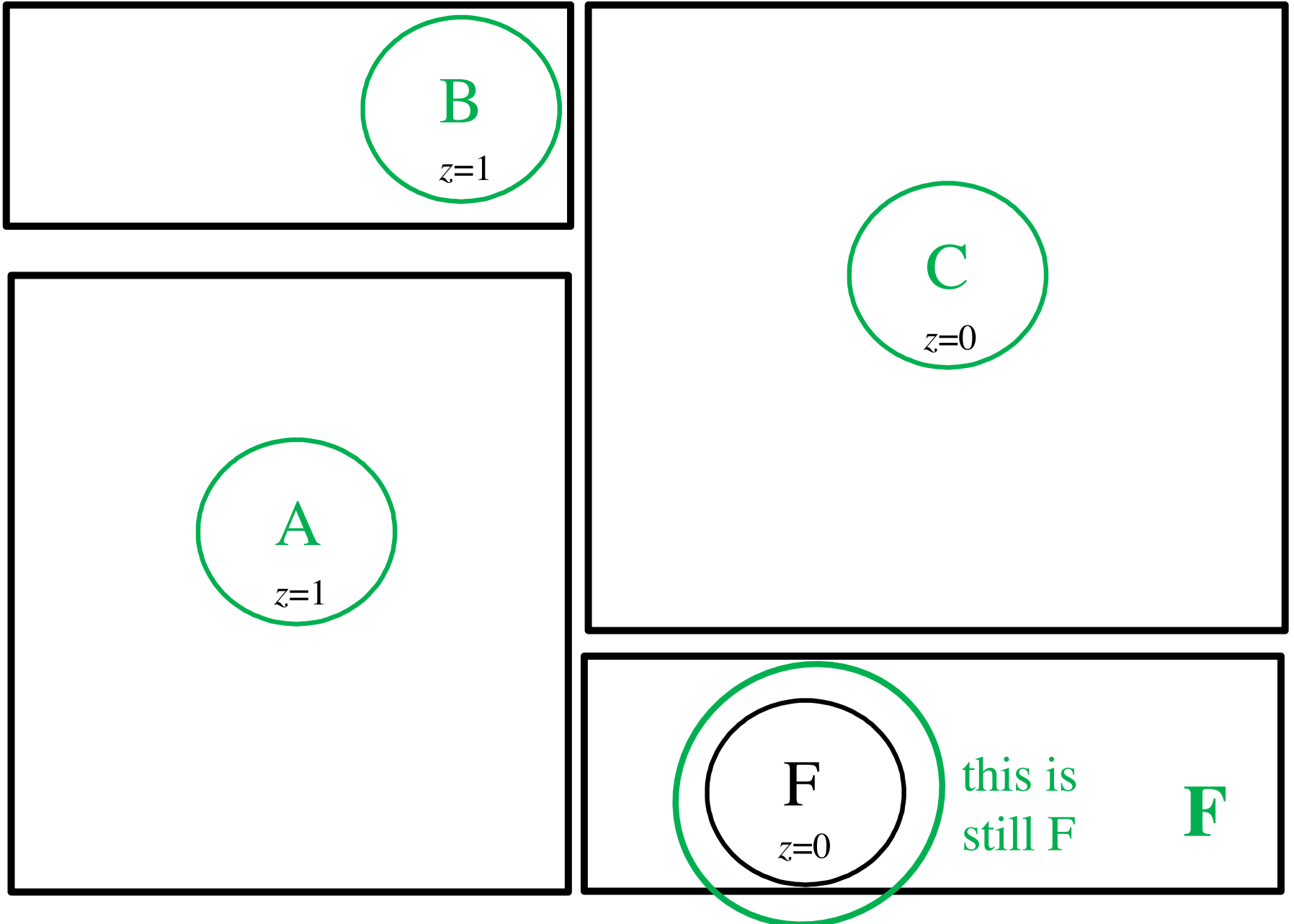
Merge the states in the same partition



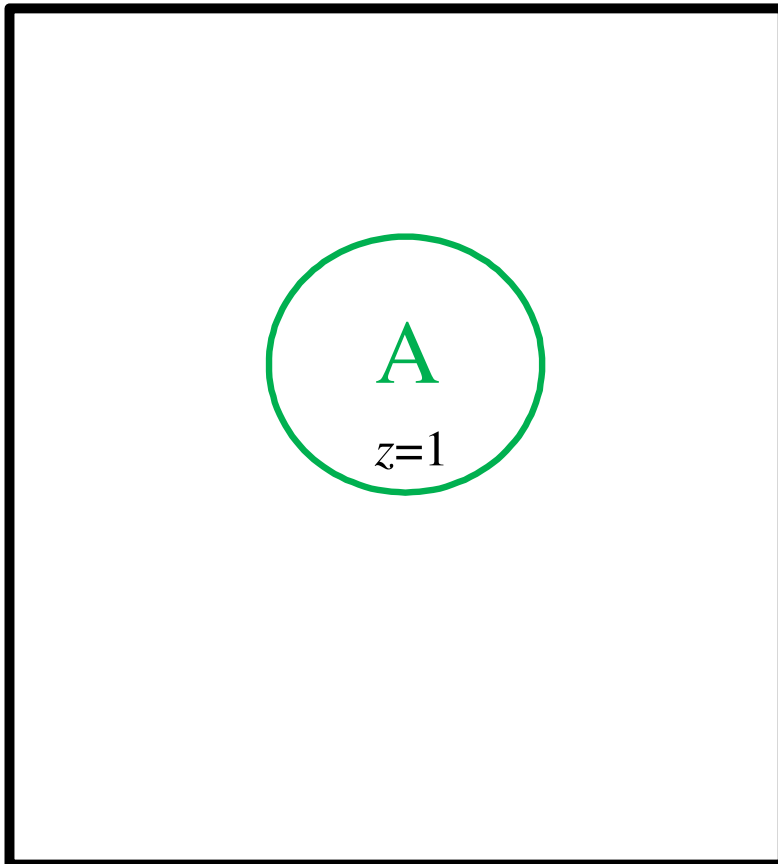
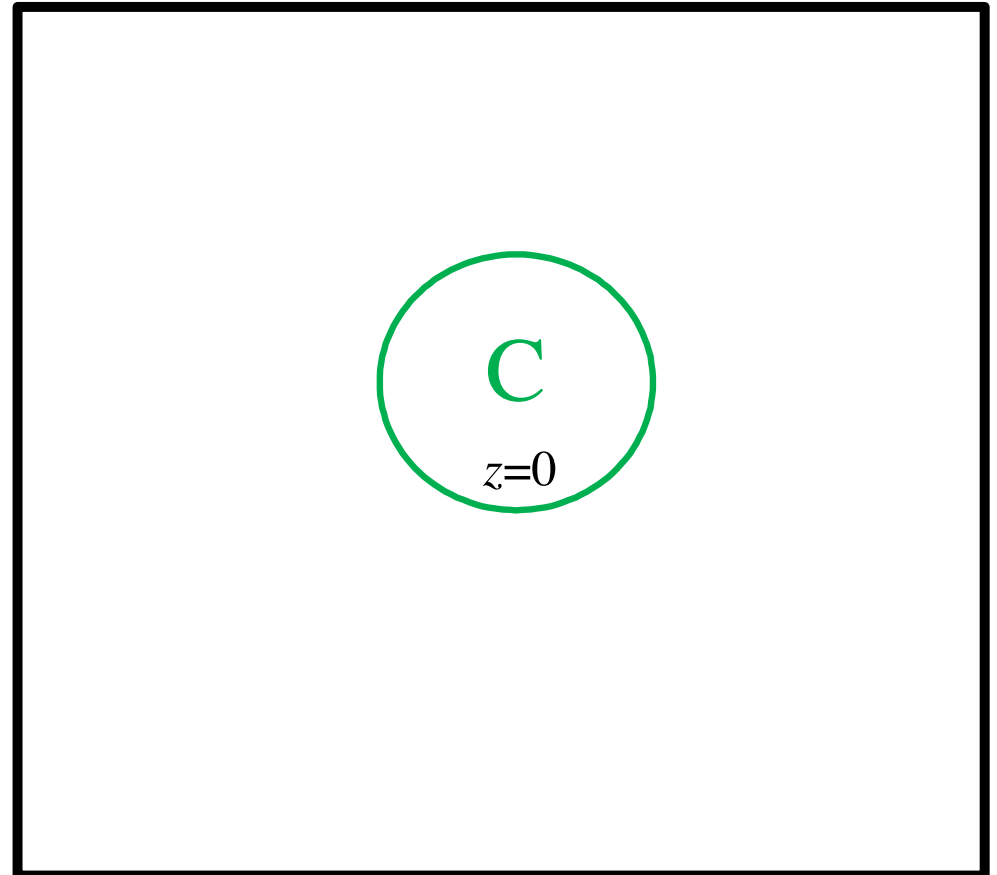
Merge the states in the same partition



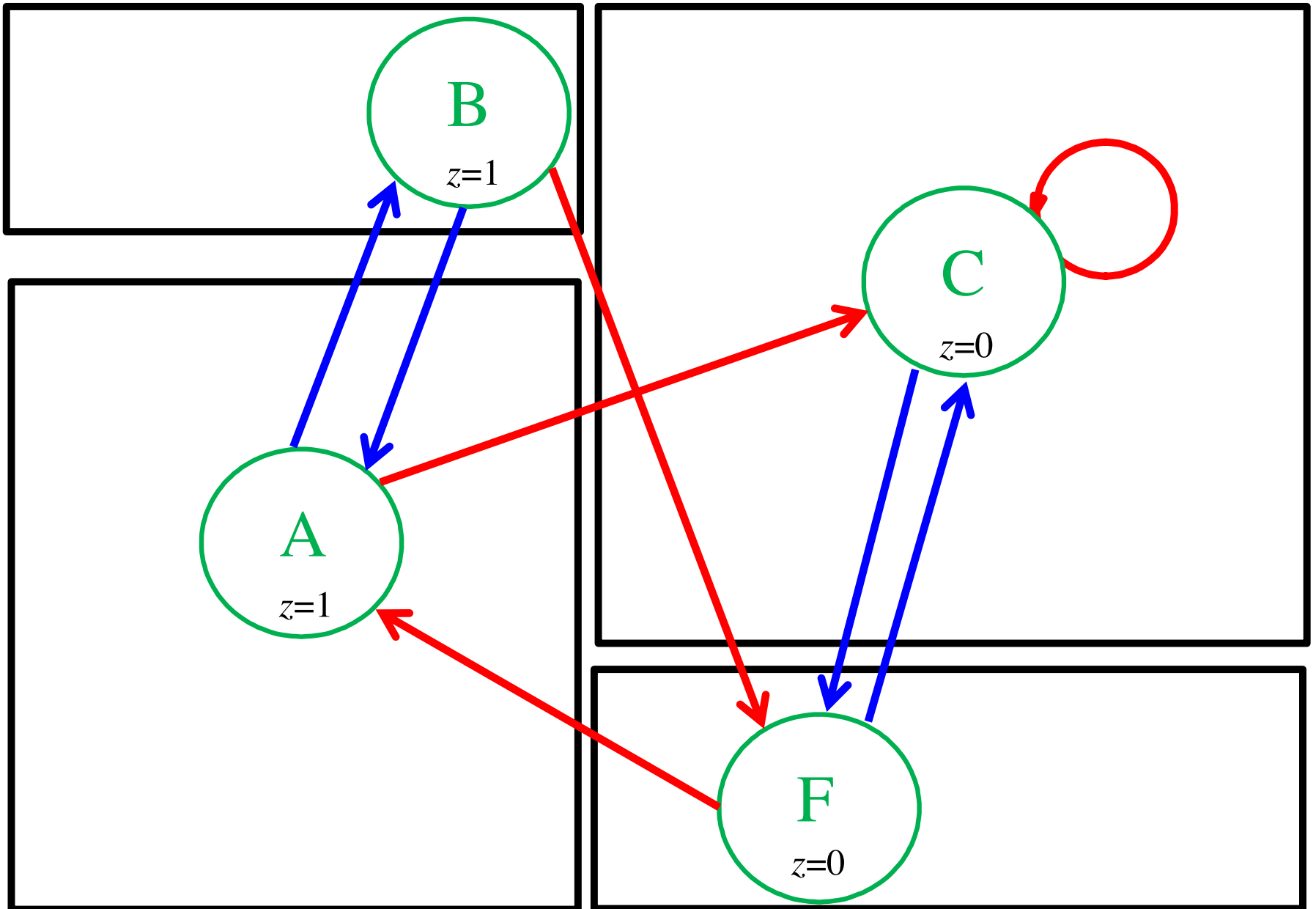
Merge the states in the same partition



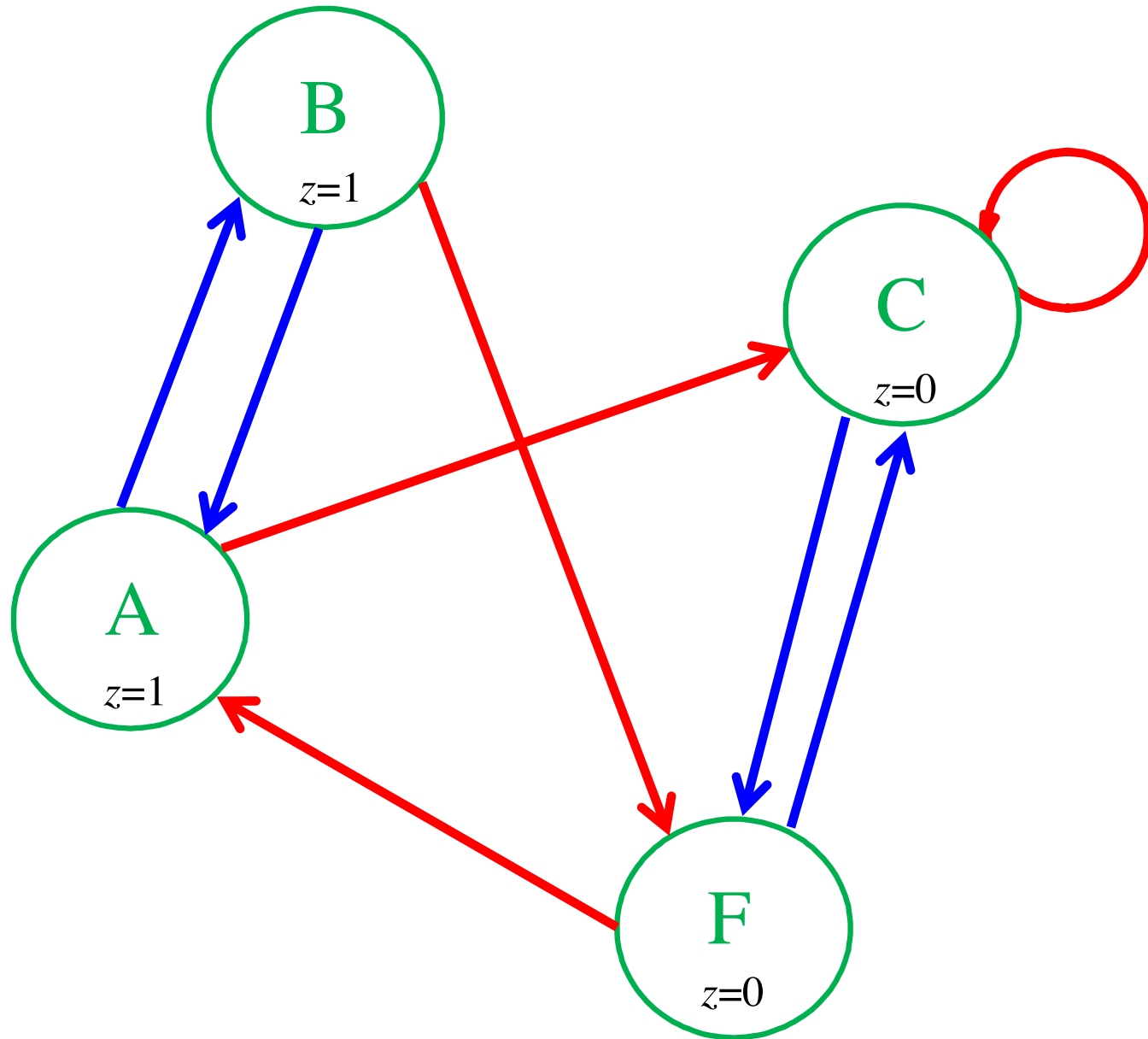
Merge the states in the same partition



Merge the transitions too



The Minimized Graph



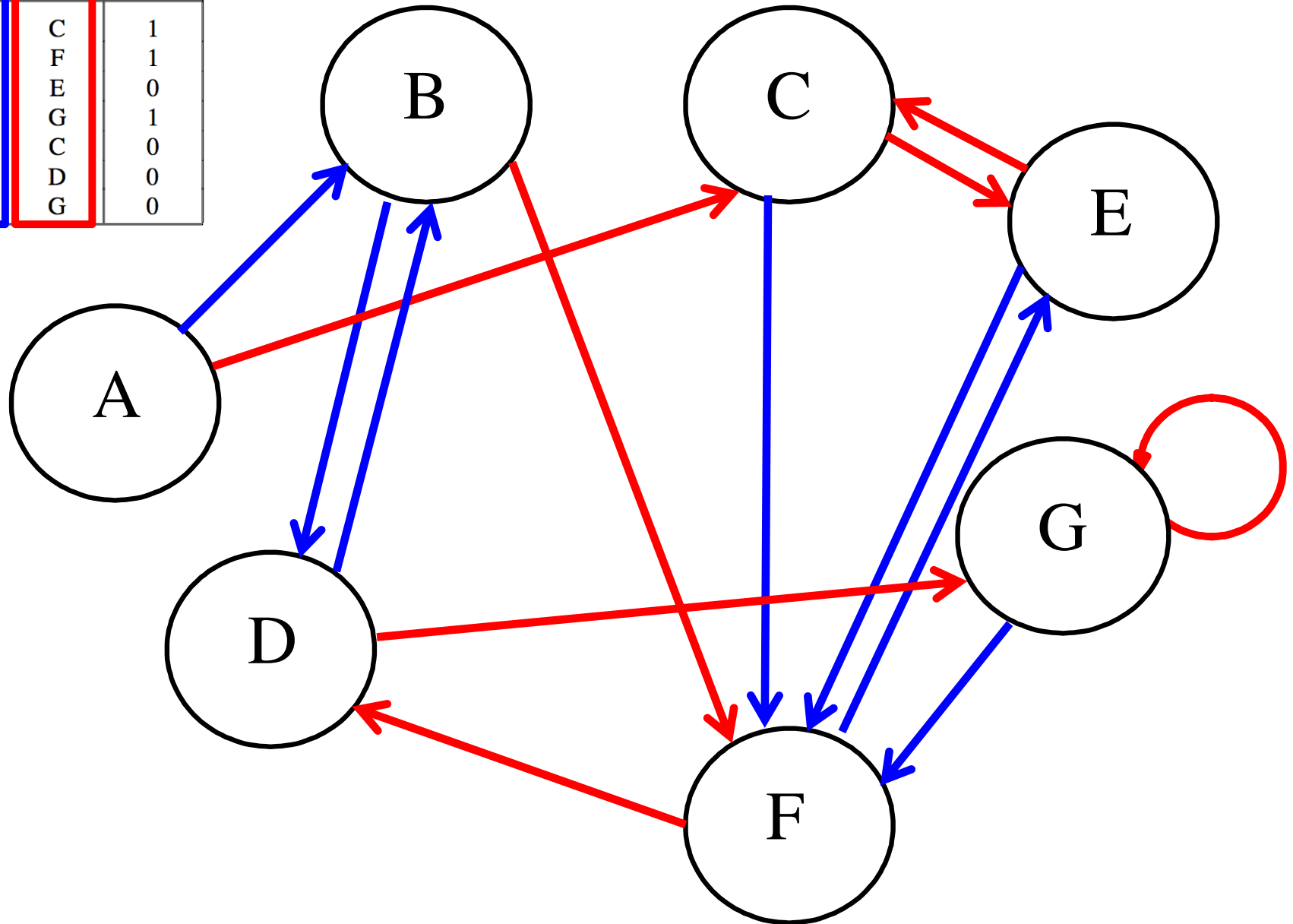
Minimized state table

Present state	Nextstate		Output z
	w = 0	w = 1	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0

To Summarize

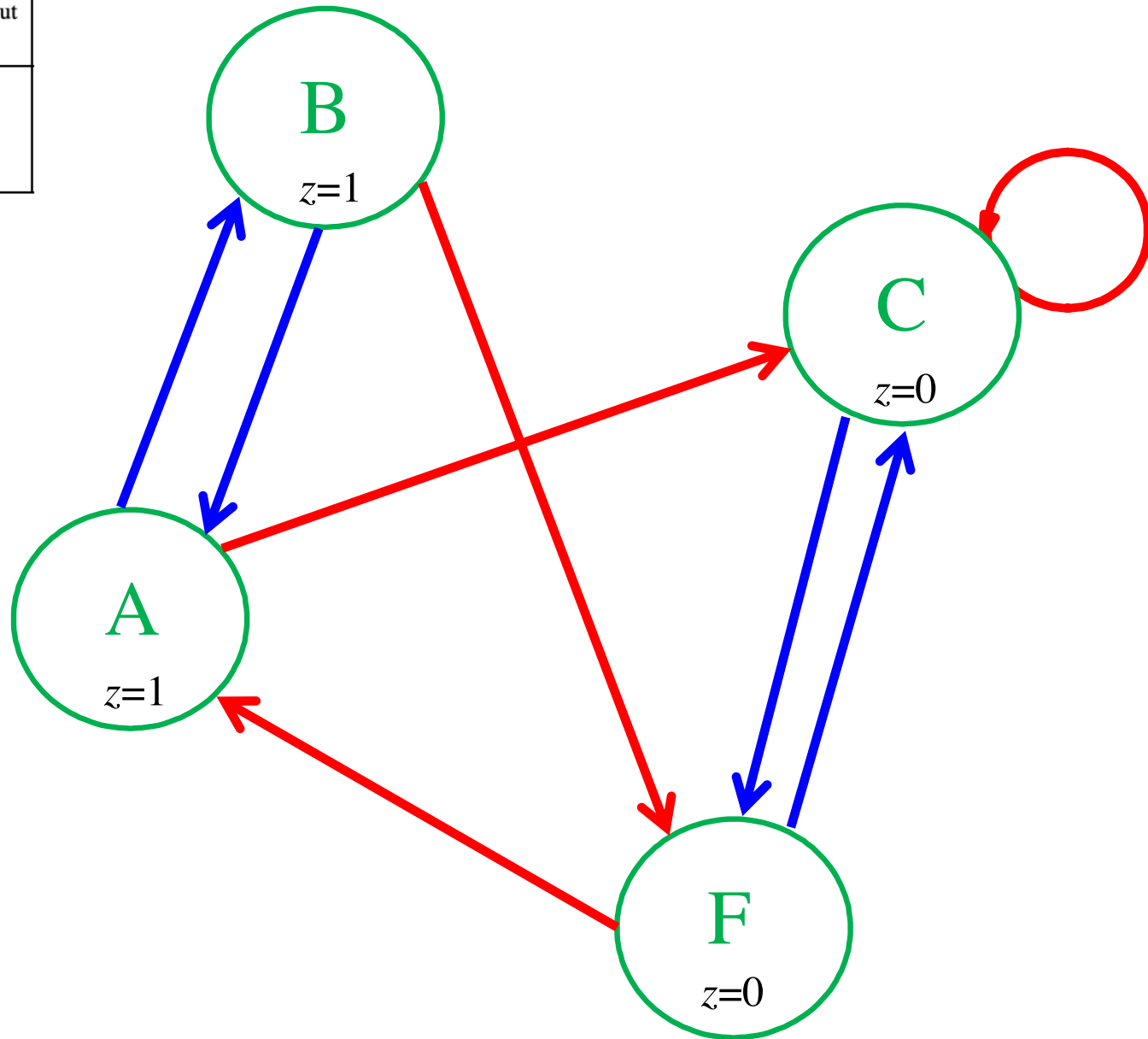
Original State Diagram

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0



Minimized State Diagram

Present state	Nextstate		Output z
	w = 0	w = 1	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0



Minimized state table

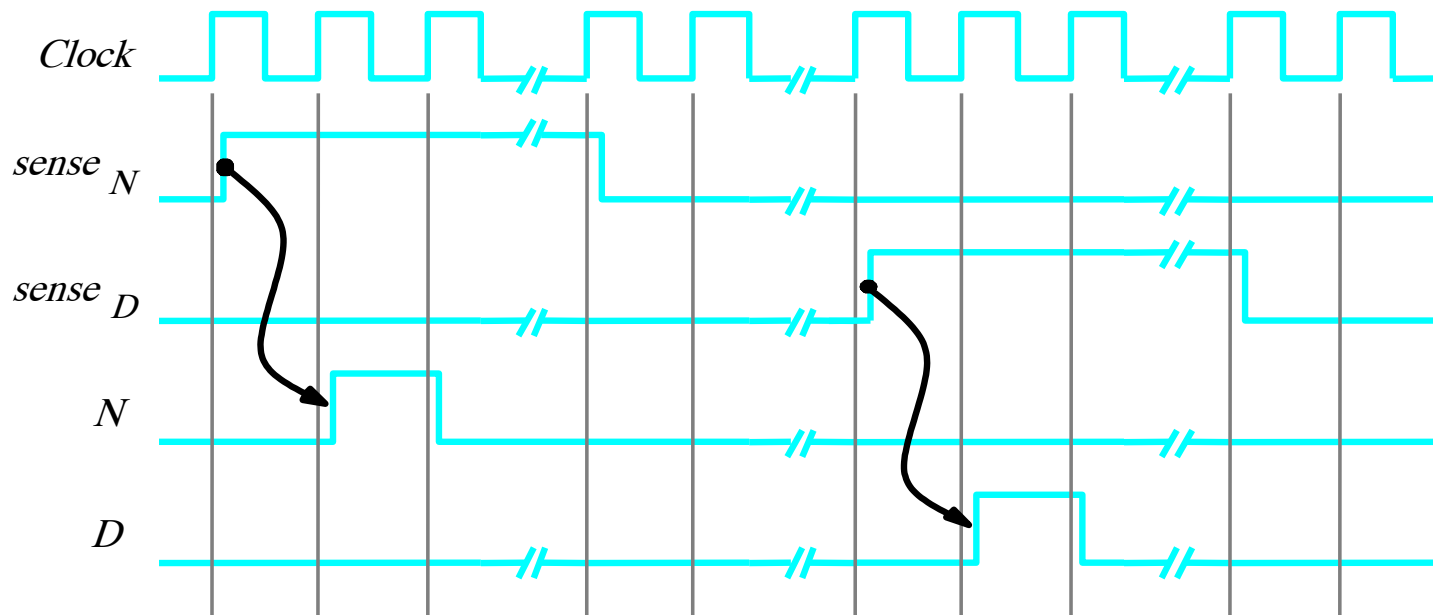
Present state	Nextstate		Output z
	w = 0	w = 1	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0

Vending Machine Example

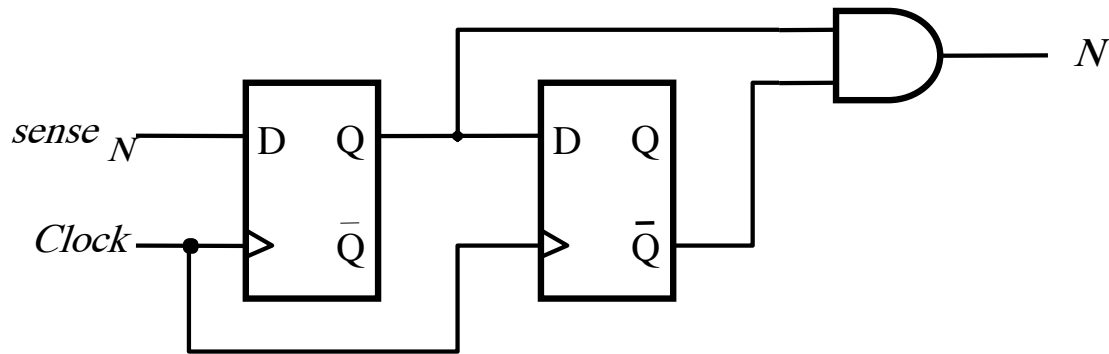
Vending Machine Example

- The machine accepts nickels and dimes
- It takes 15 cents for a piece of candy to be released from the machine
- If 20 cents is deposited, the machine will not return the change, but it will credit the buyer with 5 cents and wait for the buyer to make a second purchase.

Signals for the vending machine

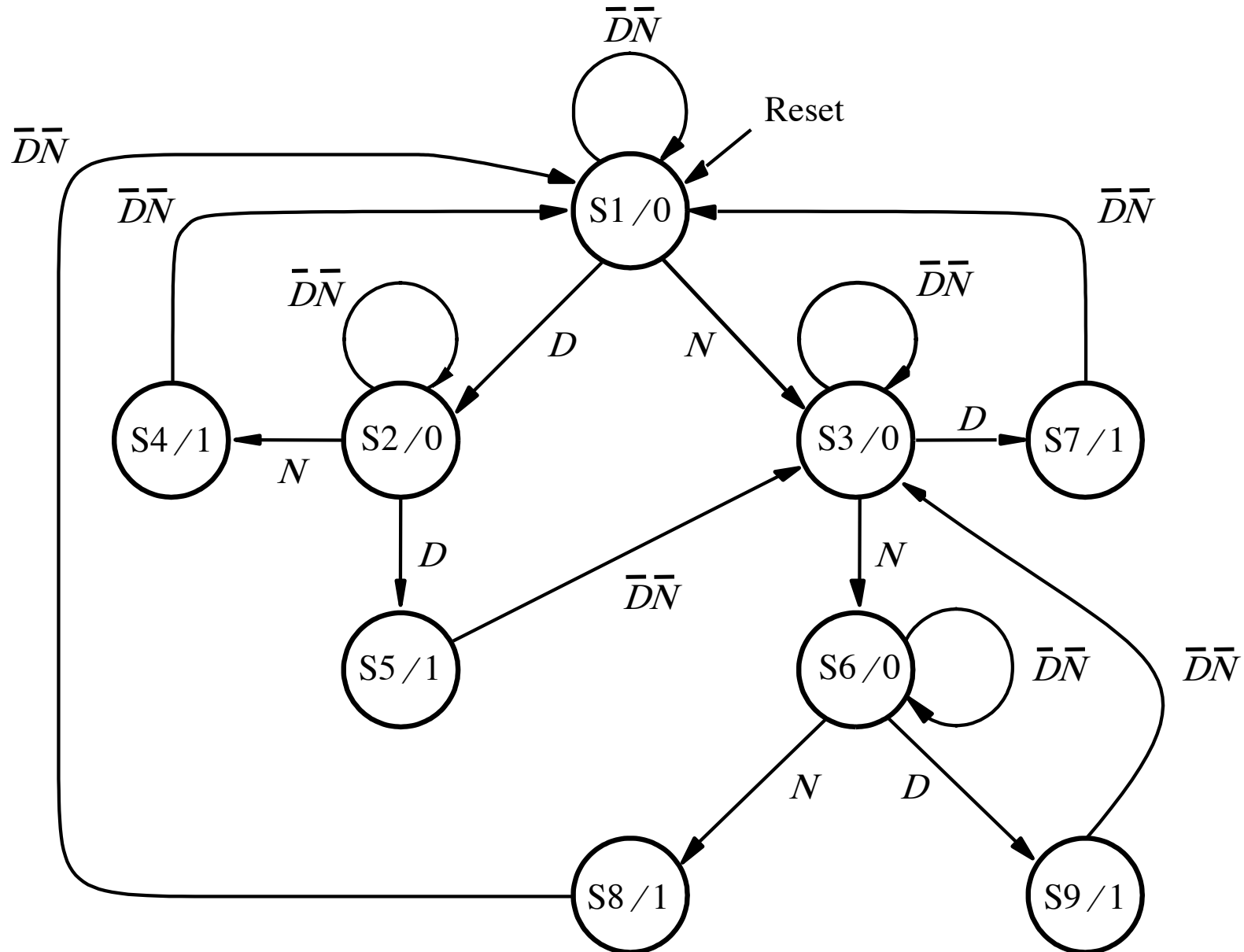


(a) Timing diagram



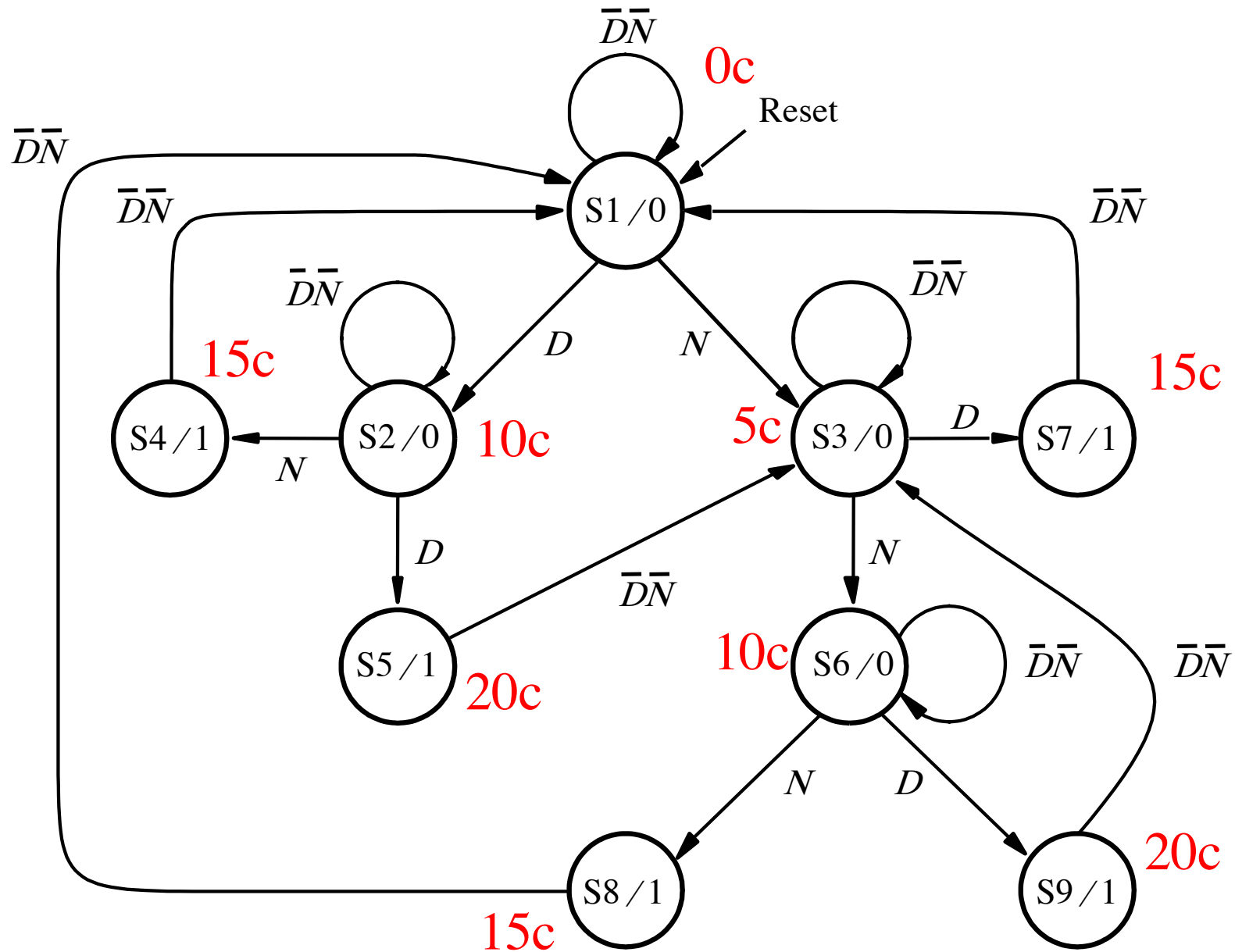
(b) Circuit that generates N

State Diagram for the vending machine



[Figure 6.54 from the textbook]

State Diagram for the vending machine



[Figure 6.54 from the textbook]

State Table for the vending machine

Present state	Next state				Output z
	DN	$=00$	01	10	
S1	S1	S3	S2	–	0
S2	S2	S4	S5	–	0
S3	S3	S6	S7	–	0
S4	S1	–	–	–	1
S5	S3	–	–	–	1
S6	S6	S8	S9	–	0
S7	S1	–	–	–	1
S8	S1	–	–	–	1
S9	S3	–	–	–	1

Incompletely specified state table

State Table for the vending machine

Present state	Next state				Output z
	DN	$=00$	01	10	
S1	S1	S3	S2	—	0
S2	S2	S4	S5	—	0
S3	S3	S6	S7	—	0
S4	S1	—	—	—	1
S5	S3	—	—	—	1
S6	S6	S8	S9	—	0
S7	S1	—	—	—	1
S8	S1	—	—	—	1
S9	S3	—	—	—	1

Incompletely specified state table

We cannot insert both a nickel and a dime at the same time.

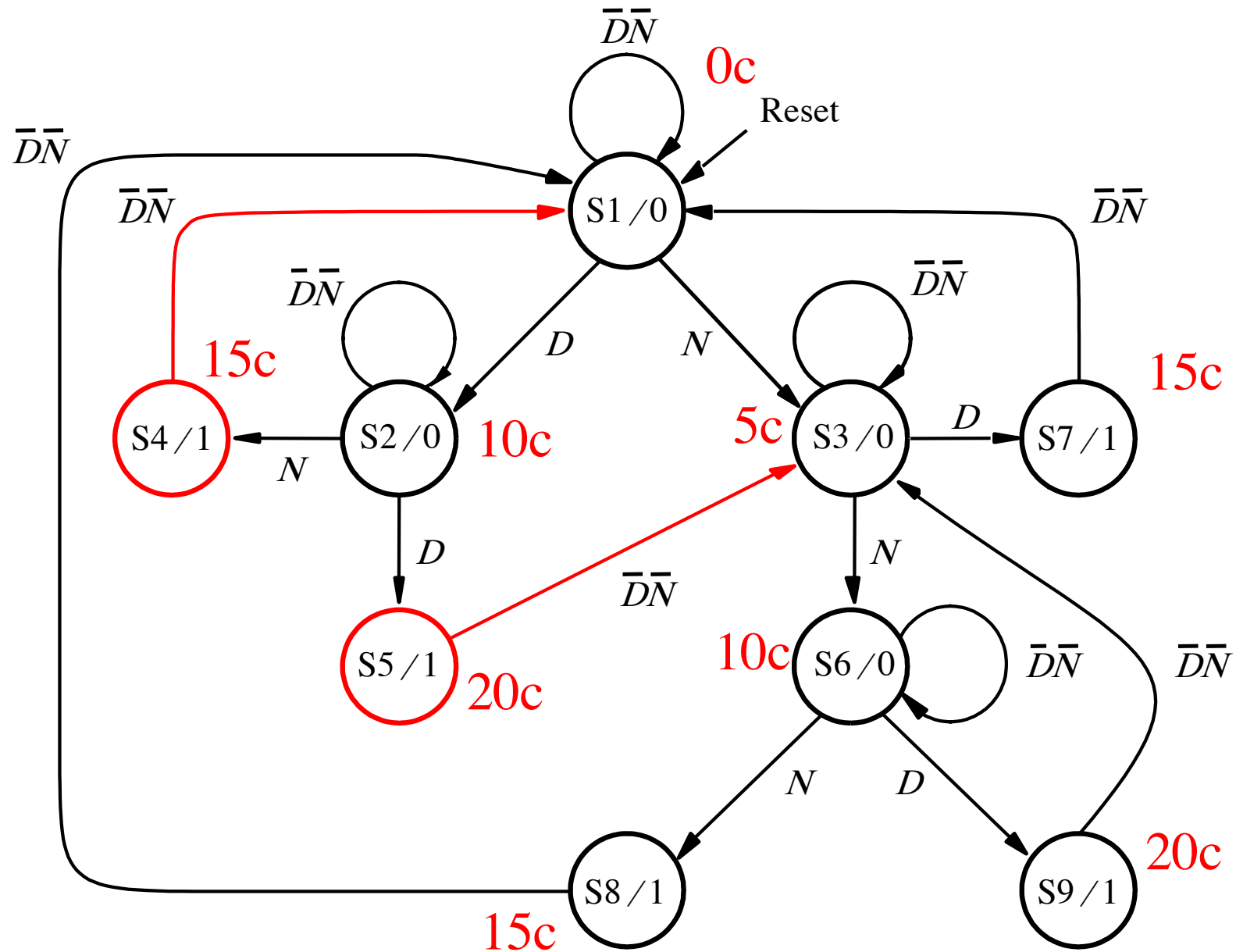
State Table for the vending machine

Present state	Next state				Output z
	DN	$=00$	01	10	
S1	S1	S3	S2	—	0
S2	S2	S4	S5	—	0
S3	S3	S6	S7	—	0
S4	S1	—	—	—	1
S5	S3	—	—	—	1
S6	S6	S8	S9	—	0
S7	S1	—	—	—	1
S8	S1	—	—	—	1
S9	S3	—	—	—	1

Incompletely specified state table

The machine is in S4 and S5 for only 1 clock cycle. Which is shorter than the time it takes for the coin to fall down. It is physically impossible for another coin to be inserted at that time.

State Diagram for the vending machine



[Figure 6.54 from the textbook]

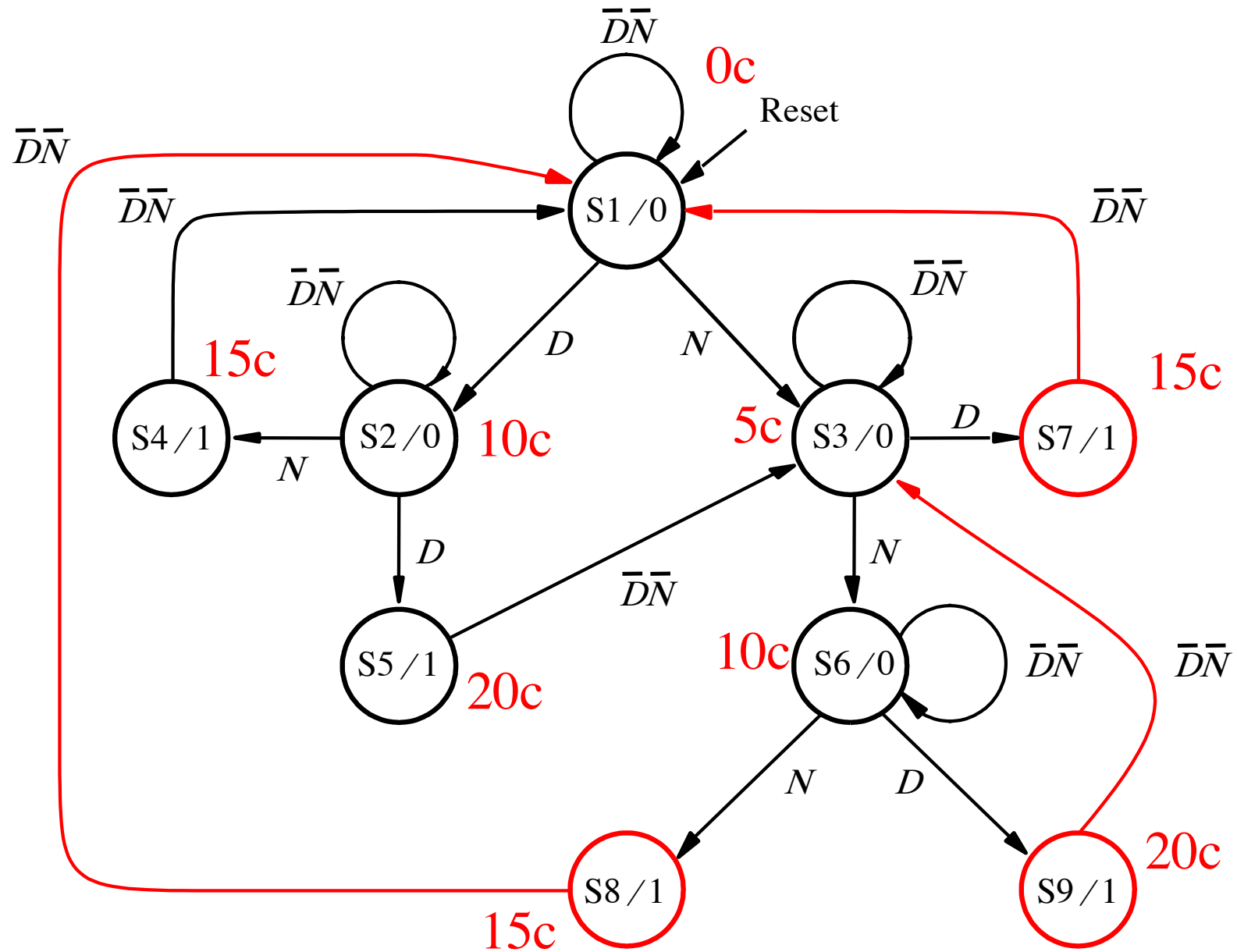
State Table for the vending machine

Present state	Next state				Output z
	DN	$=00$	01	10	
S1	S1	S3	S2	—	0
S2	S2	S4	S5	—	0
S3	S3	S6	S7	—	0
S4	S1	—	—	—	1
S5	S3	—	—	—	1
S6	S6	S8	S9	—	0
S7	S1	—	—	—	1
S8	S1	—	—	—	1
S9	S3	—	—	—	1

Incompletely specified state table

The machine is in states S7, S8, and S9 for only 1 clock cycle. Which is shorter than the time it takes for the coin to fall down.

State Diagram for the vending machine



[Figure 6.54 from the textbook]

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

$P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)$

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)

P2=(S1,S2,S3,S6) (S4,S5,S7,S8,S9)

partition based on
common output

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)

P2=(S1,S2,S3,S6) (S4,S5,S7,S8,S9)

P3=(S1) (S3) (S2,S6) (S4,S5,S7,S8,S9)

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)

P2=(S1,S2,S3,S6) (S4,S5,S7,S8,S9)

P3=(S1) (S3) (S2,S6) (S4,S5,S7,S8,S9)

P4=(S1) (S3) (S2,S6) (S4,S7,S8) (S5,S9)

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)

P2=(S1,S2,S3,S6) (S4,S5,S7,S8,S9)

P3=(S1) (S3) (S2,S6) (S4,S5,S7,S8,S9)

P4=(S1) (S3) (S2,S6) (S4,S7,S8) (S5,S9)

P5=(S1) (S3) (S2,S6) (S4,S7,S8) (S5,S9)

Partition for Vending Machine FSM

Present state	Next state				Output z
	00	01	10	11	
S1	S1	S3	S2	-	0
S3	S3	S6	S7	-	0
S2	S2	S4	S5	-	0
S6	S6	S8	S9	-	0
S4	S1	-	-	-	1
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S5	S3	-	-	-	1
S9	S3	-	-	-	1

P1=(S1,S2,S3,S4,S5,S6,S7,S8,S9)

P2=(S1,S2,S3,S6) (S4,S5,S7,S8,S9)

P3=(S1) (S3) (S2,S6) (S4,S5,S7,S8,S9)

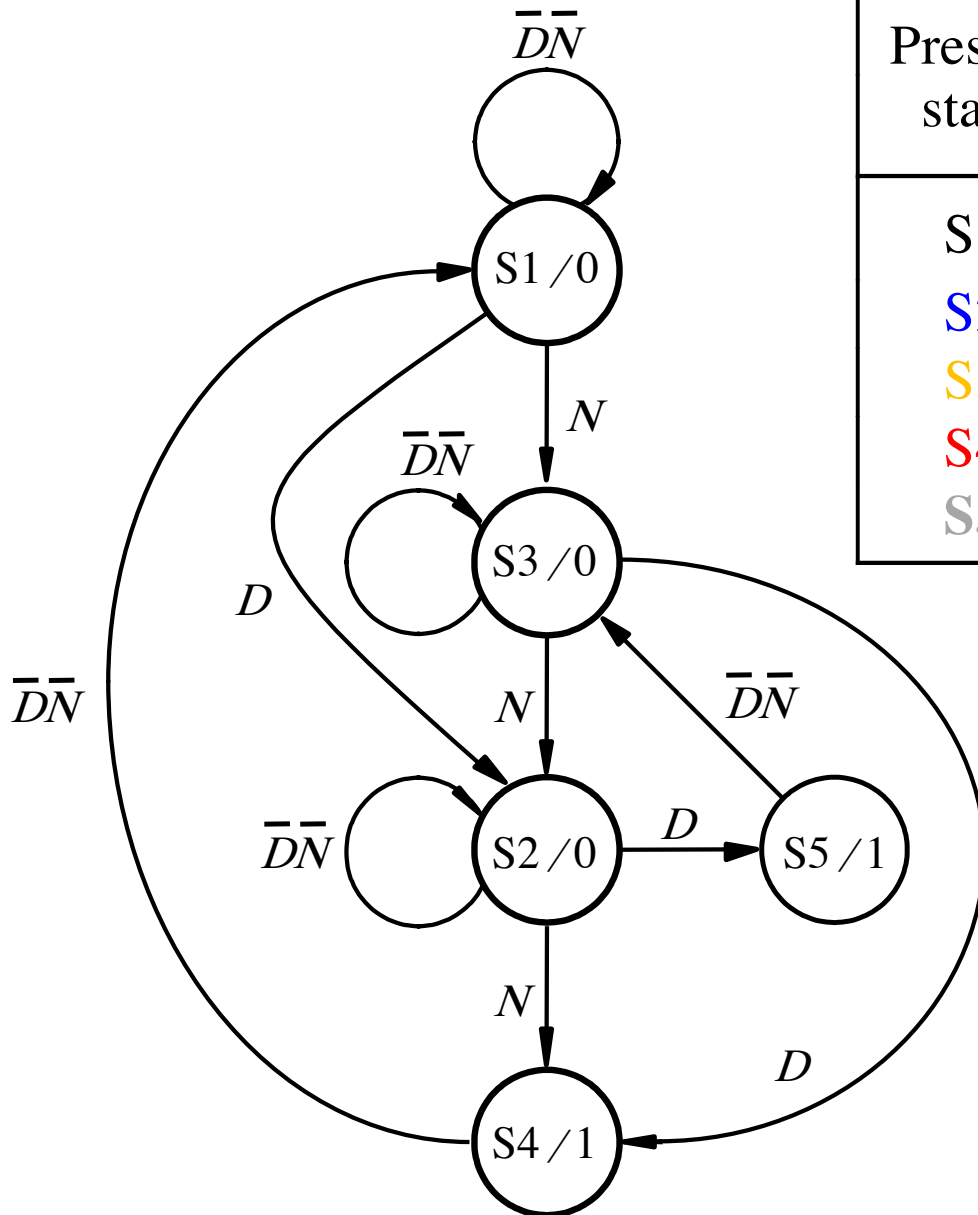
P4=(S1) (S3) (S2,S6) (S4,S7,S8) (S5,S9)

P5=(S1) (S3) (S2,S6) (S4,S7,S8) (S5,S9)

Minimized State Table for the vending machine

Present state	Next state				Output z
	DN	$=00$	01	10	
S1	S1	S3	S2	—	0
S2	S2	S4	S5	—	0
S3	S3	S2	S4	—	0
S4	S1	—	—	—	1
S5	S3	—	—	—	1

Minimized State Table for the vending machine

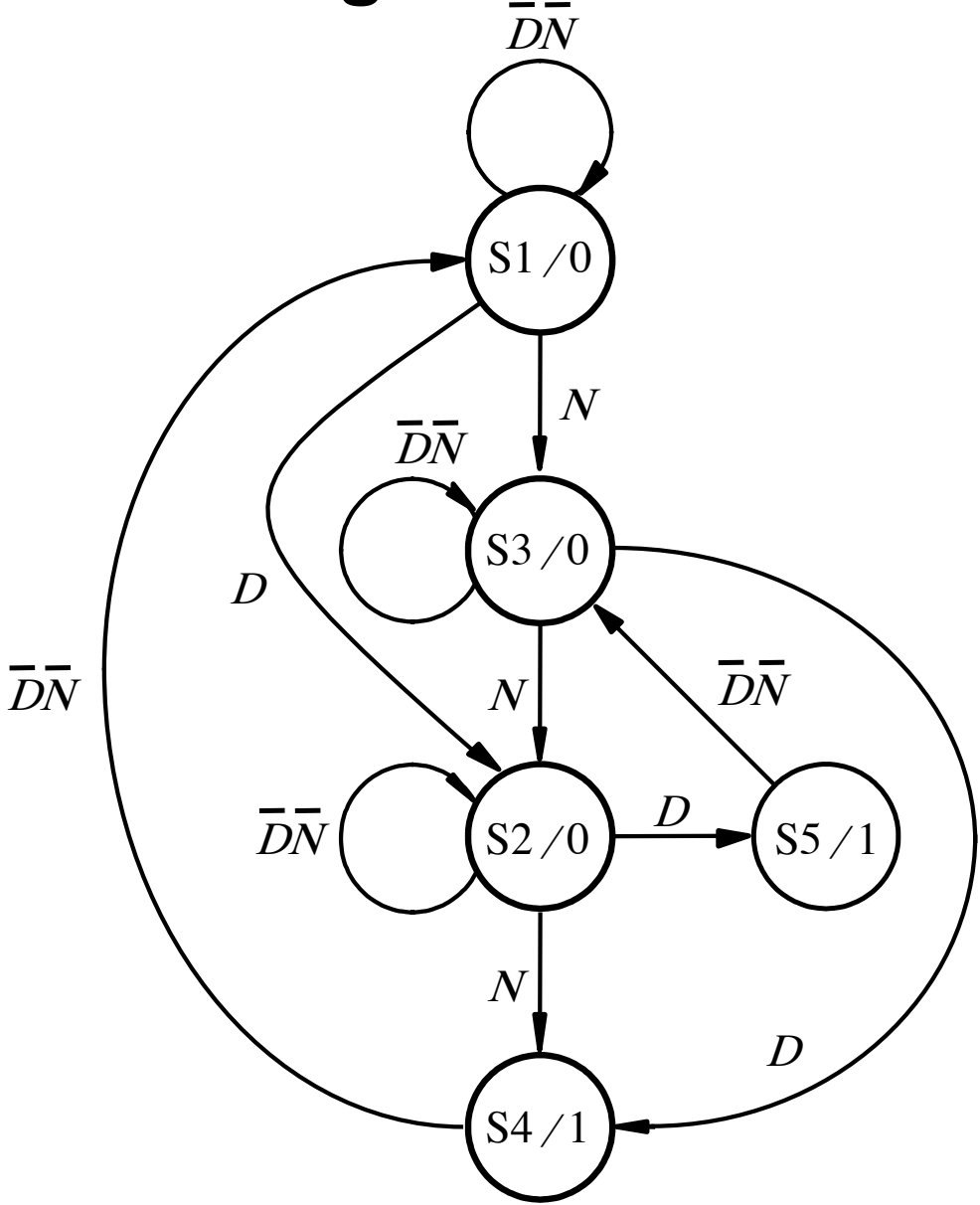


Present state	Next state				Output z
	$DN = 00$	01	10	11	
S1	S1	S3	S2	—	0
S2	S2	S4	S5	—	0
S3	S3	S2	S4	—	0
S4	S1	—	—	—	1
S5	S3	—	—	—	1

[Figure 6.57 from the textbook]

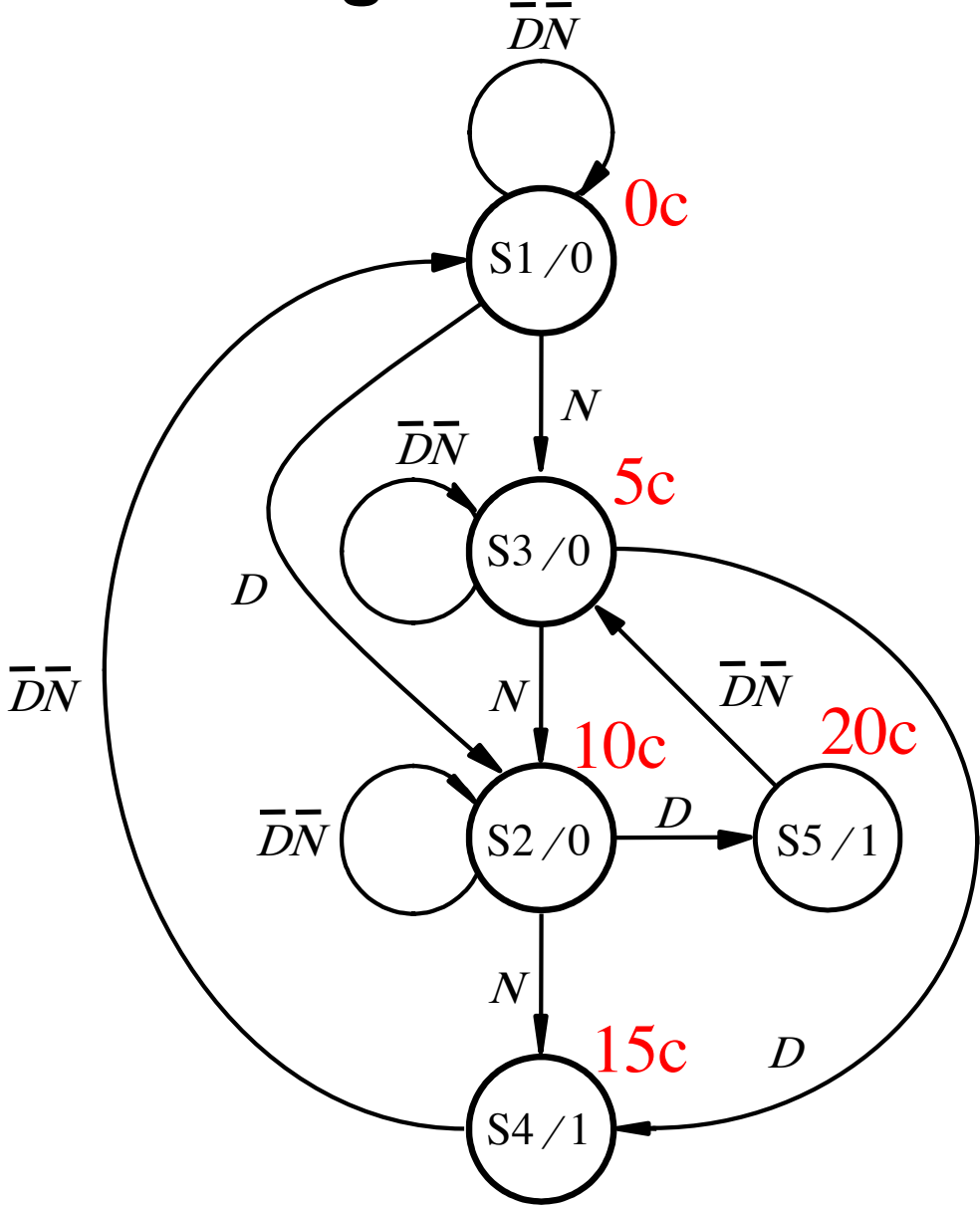
[Figure 6.56 from the textbook]

Minimized State Diagram for the vending machine



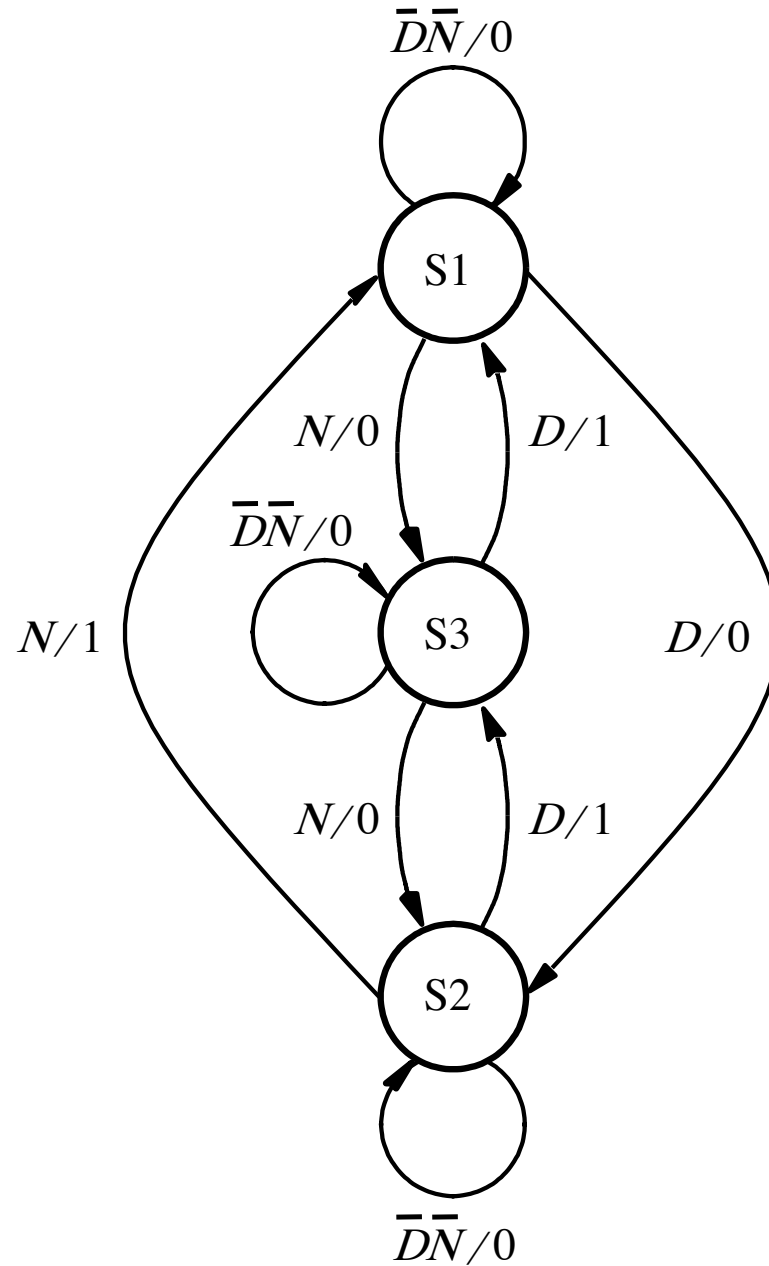
[Figure 6.57 from the textbook]

Minimized State Diagram for the vending machine



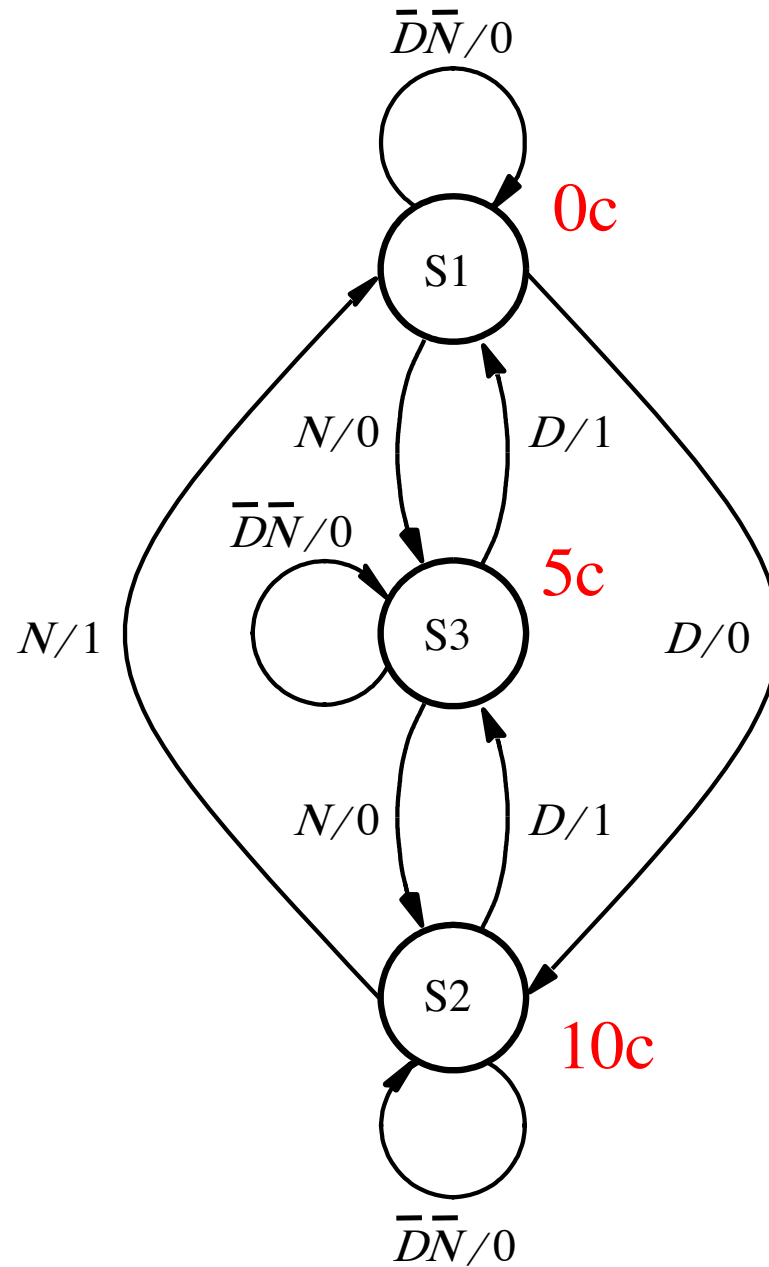
[Figure 6.57 from the textbook]

Mealy-type FSM for the vending machine



[Figure 6.58 from the textbook]

Mealy-type FSM for the vending machine



[Figure 6.58 from the textbook]

Another Example of Incompletely specified state table

Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	B	C	0	0
B	D	—	0	—
C	F	E	0	1
D	B	G	0	0
E	F	C	0	1
F	E	D	0	1
G	F	—	0	—

Questions?

THE END