

The following notation is used in these slides

Splitting a node cases

$R_0 R_1 R_2$

Keys in the root being split

$C_0 C_1 C_2$

Keys in a child node being split

$P P_0 P_1$

Keys in the parent of a node being split

$T_0 T_1 T_2 T_3 T_4 T_5$

Subtrees of node being split (can be empty)

Inserting into a leaf cases

$E E_0 E_1$

Existing keys in a leaf being inserted into

N

New key being inserted

SPLIT ROOT

BEFORE

AFTER

2-3-4 Tree

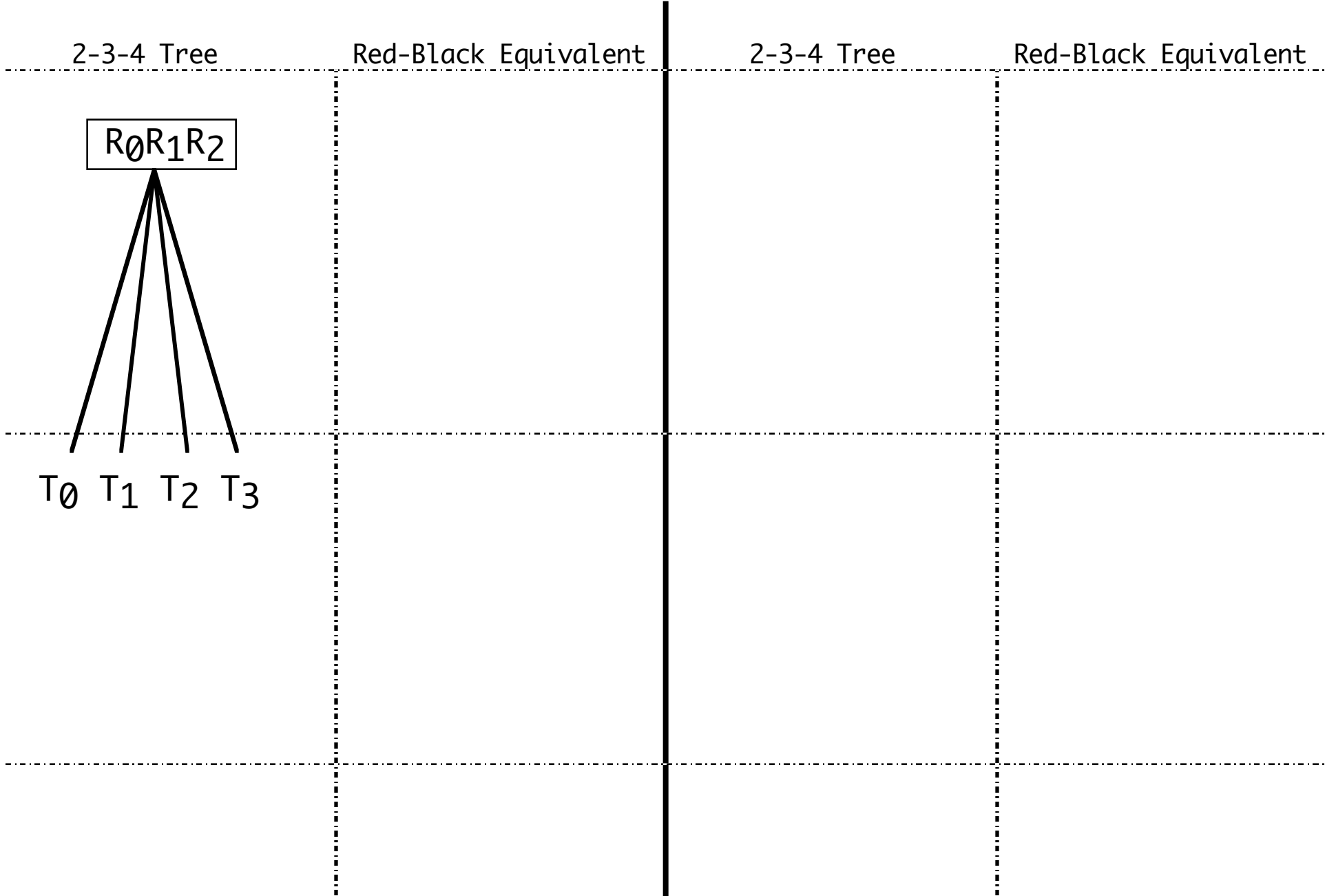
Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent

$R_0 R_1 R_2$

T_0 T_1 T_2 T_3



SPLIT ROOT

BEFORE

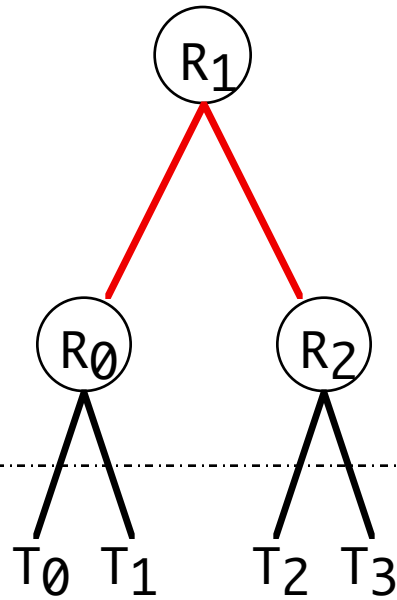
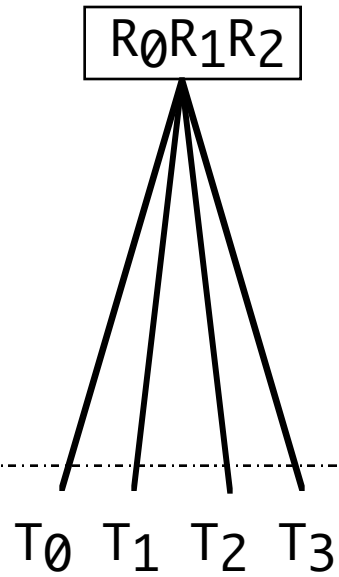
AFTER

2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent



SPLIT ROOT

BEFORE

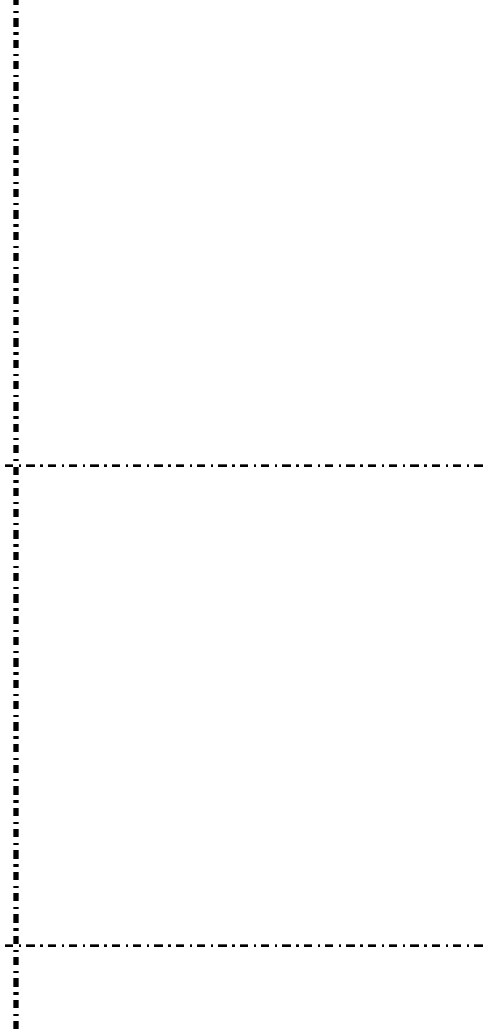
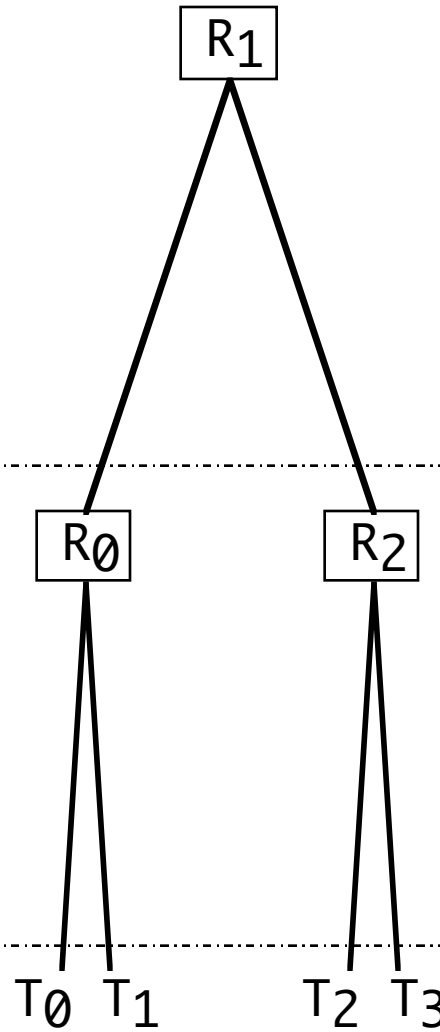
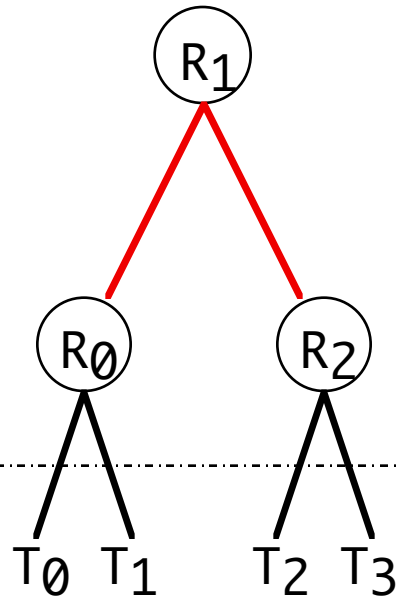
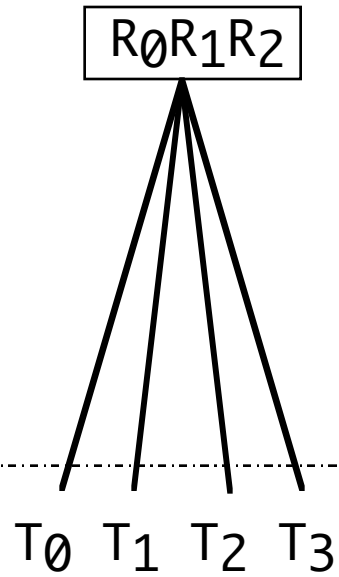
AFTER

2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent



SPLIT ROOT

BEFORE

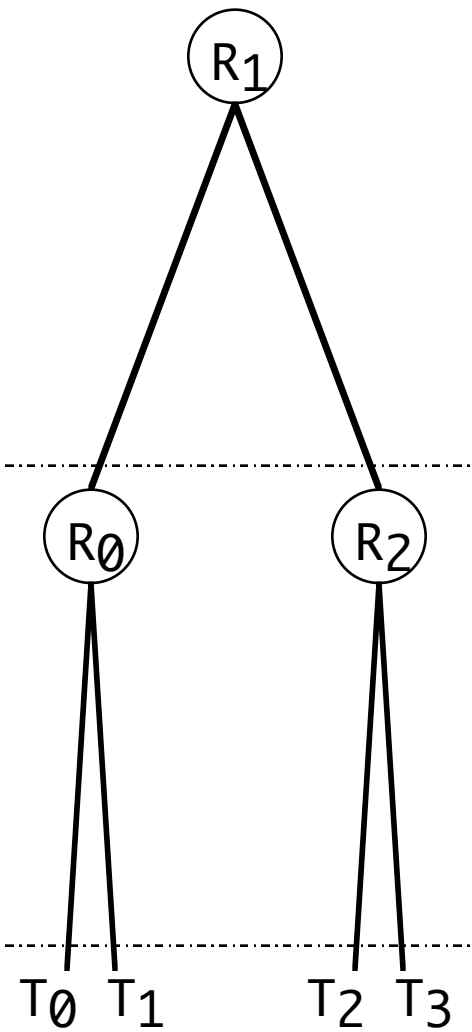
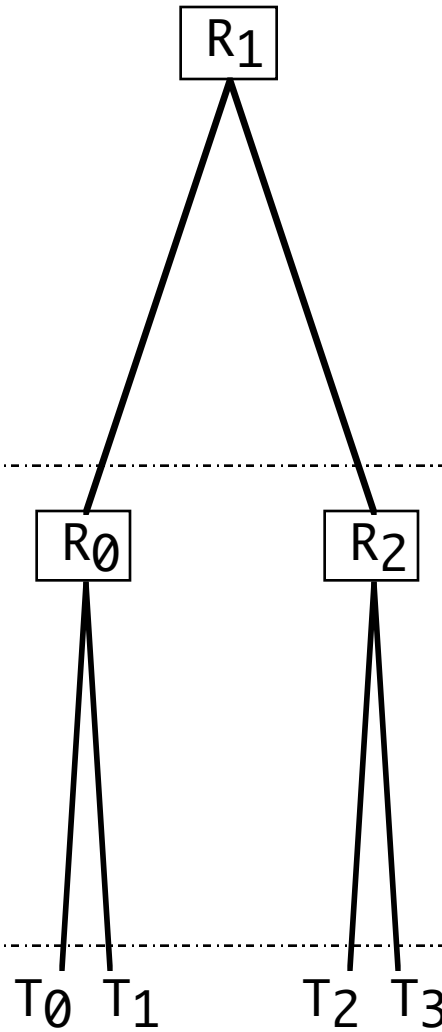
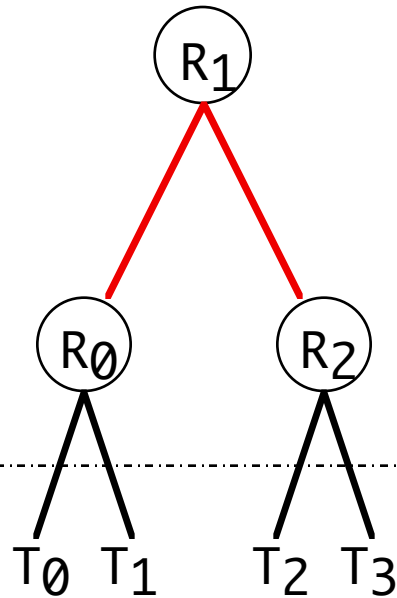
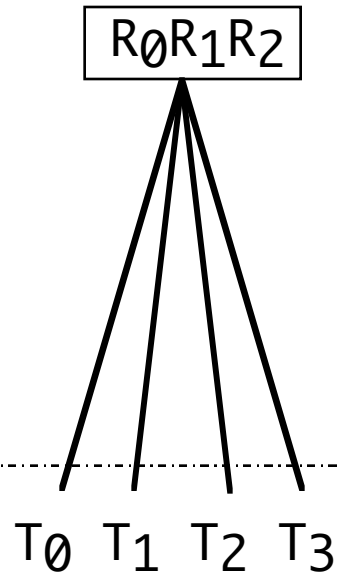
AFTER

2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent



SPLIT LEFT CHILD OF 2-NODE

BEFORE

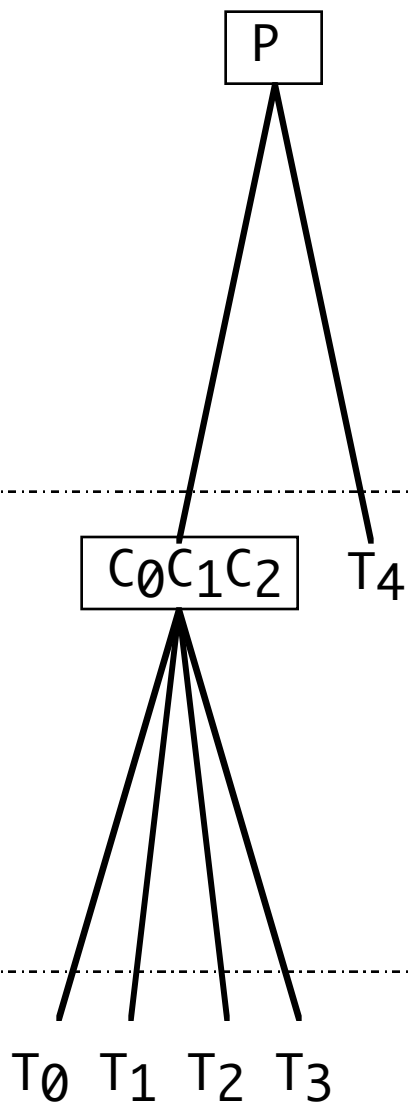
AFTER

2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

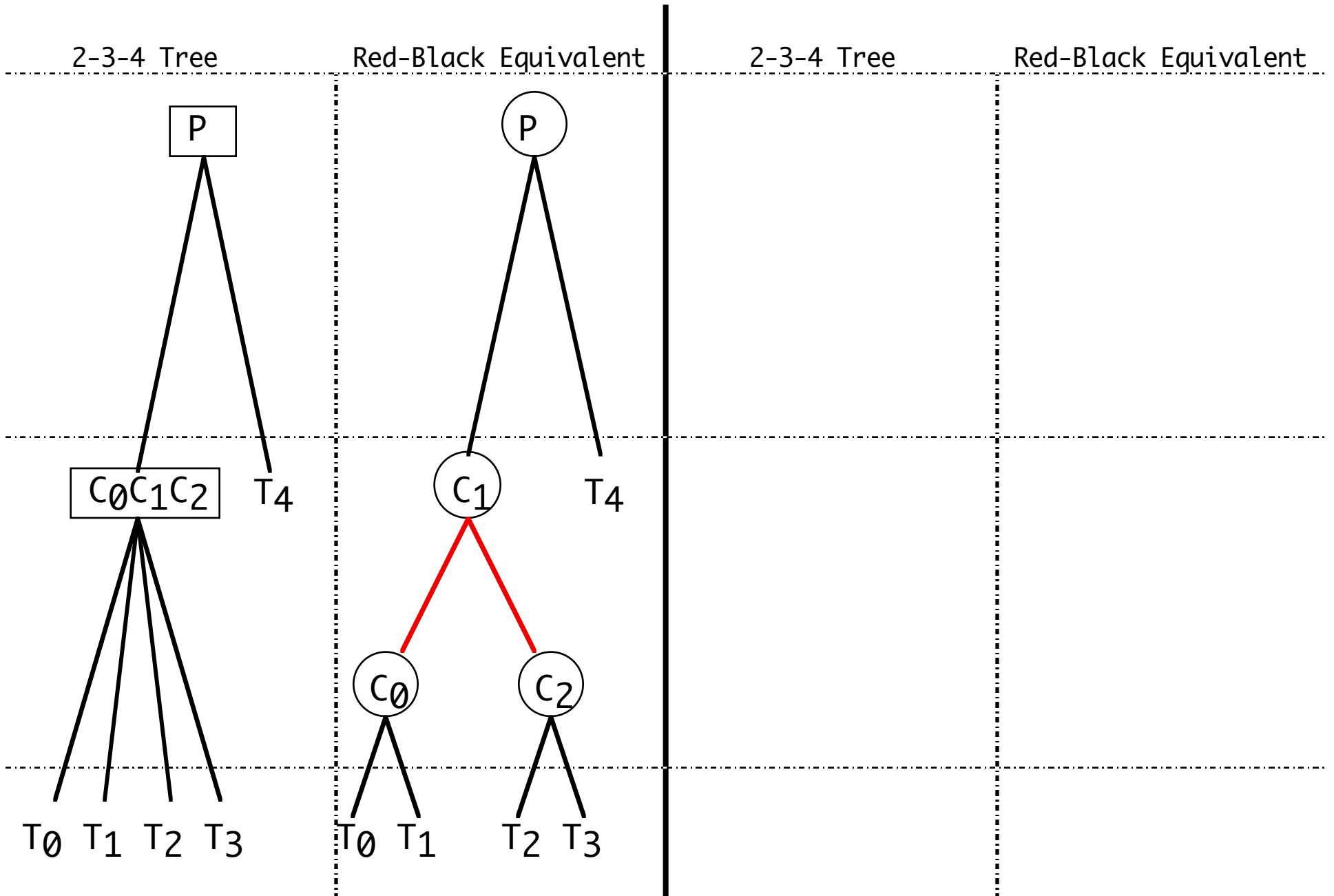
Red-Black Equivalent



SPLIT LEFT CHILD OF 2-NODE

BEFORE

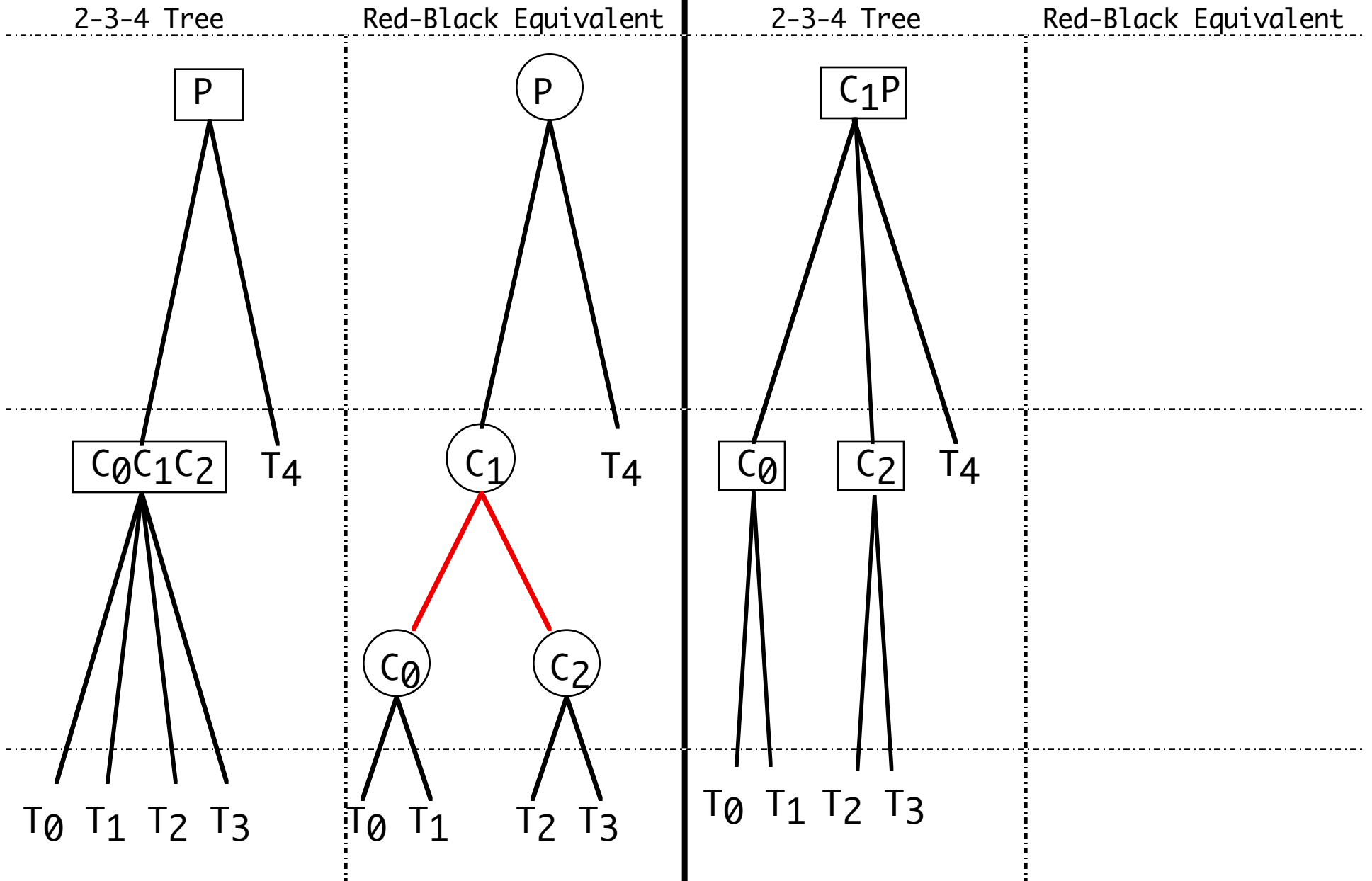
AFTER



SPLIT LEFT CHILD OF 2-NODE

BEFORE

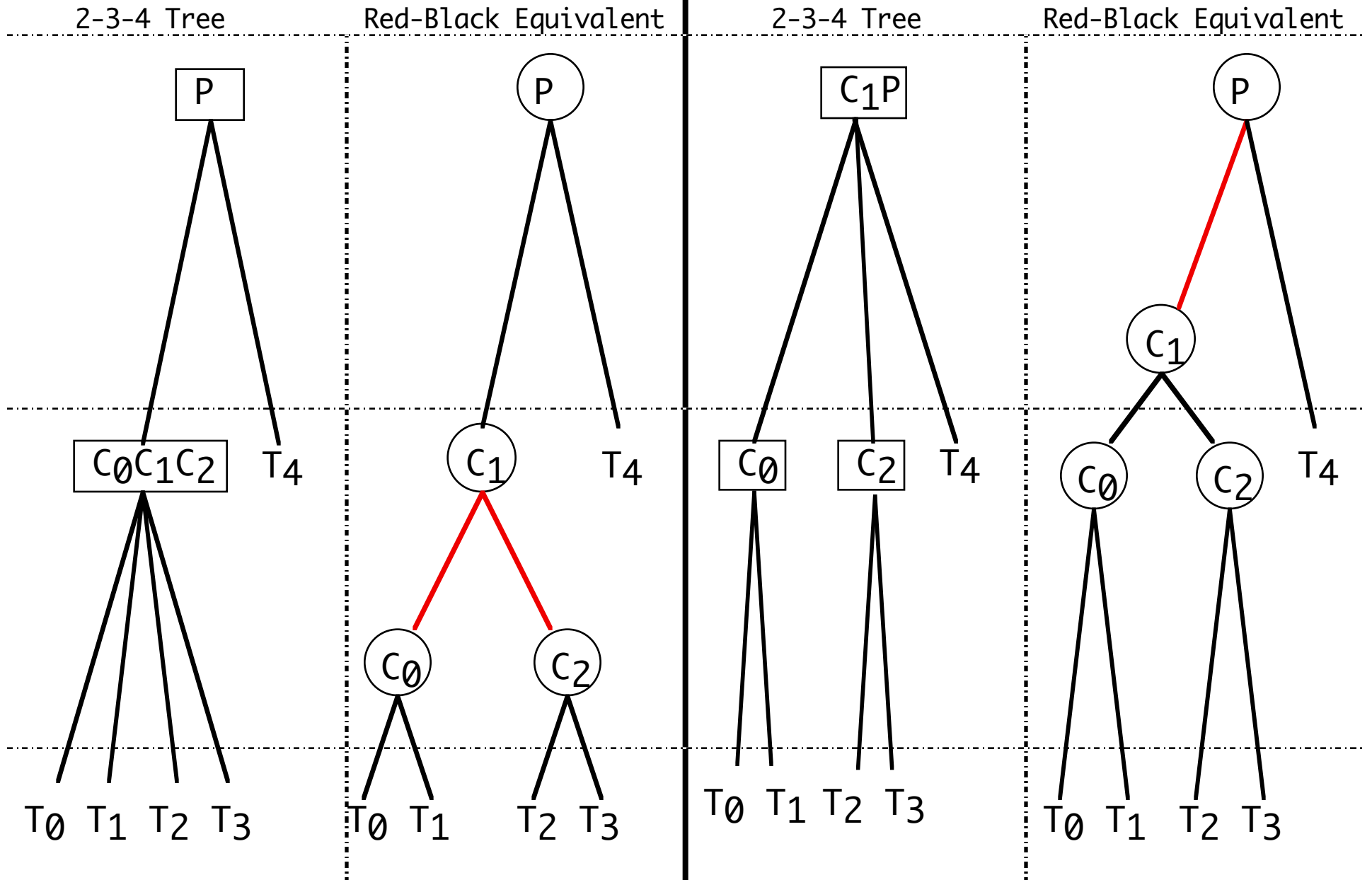
AFTER



SPLIT LEFT CHILD OF 2-NODE

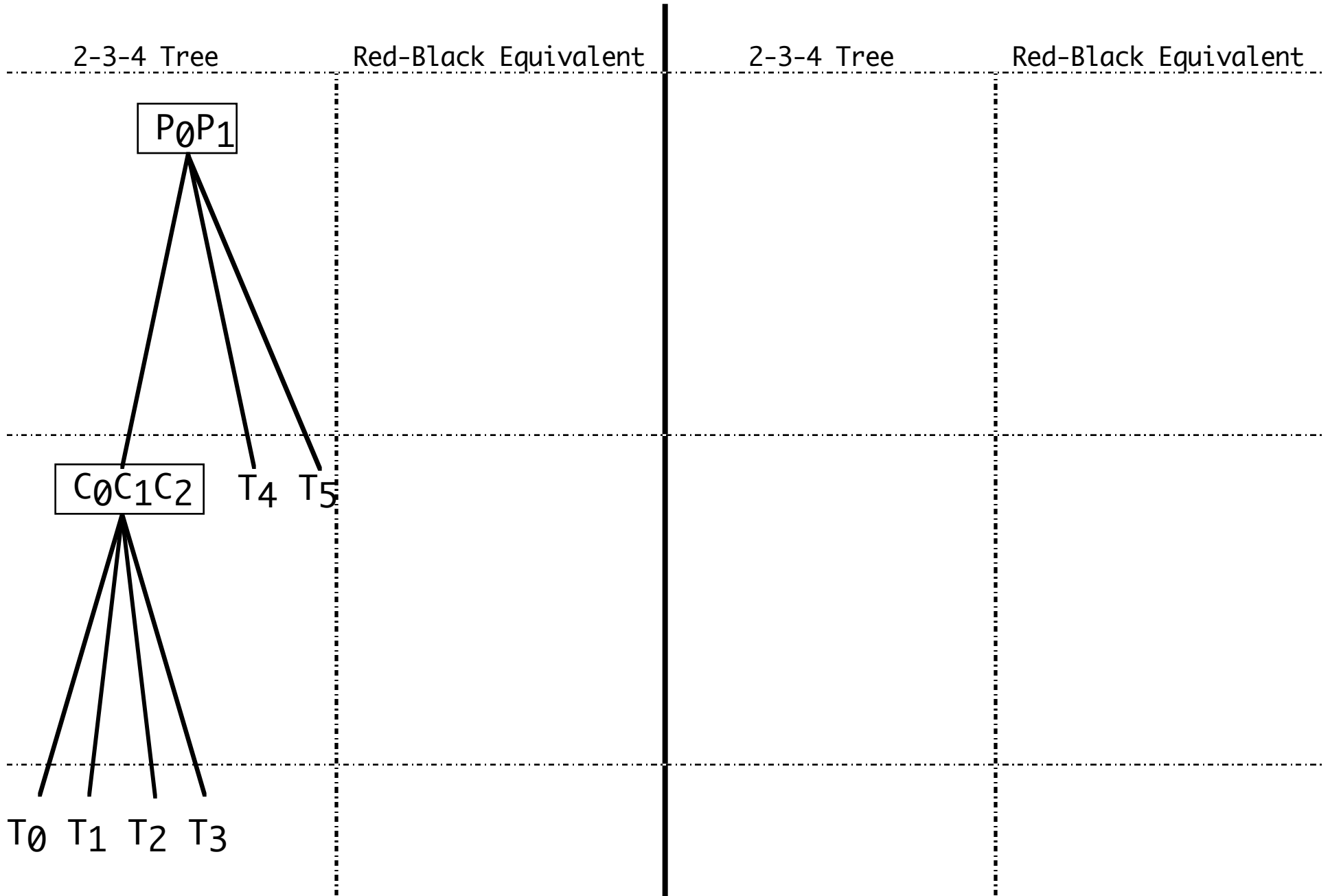
BEFORE

AFTER



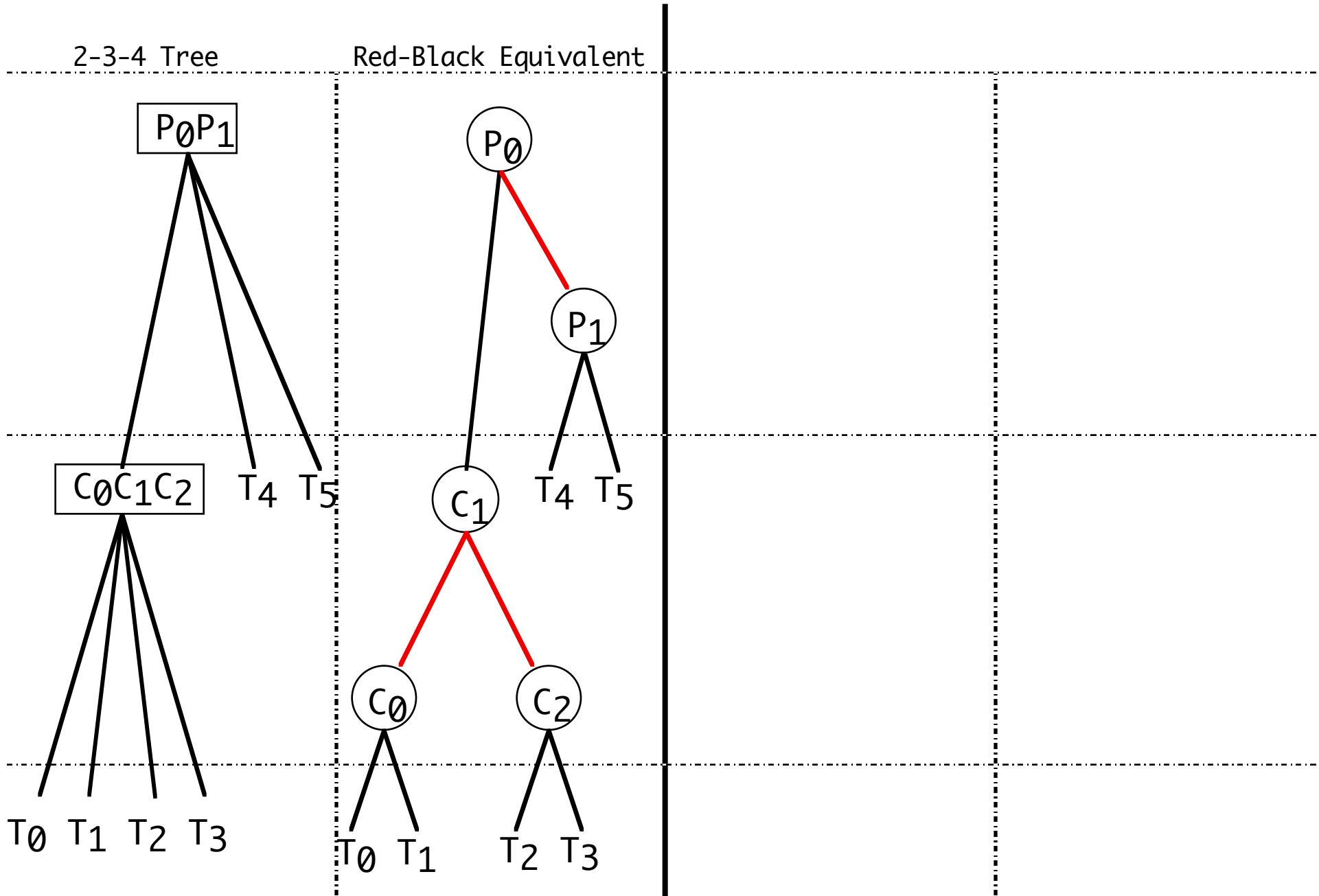
SPLIT LEFTMOST CHILD OF 3-NODE - RED ON RIGHT

BEFORE **AFTER**



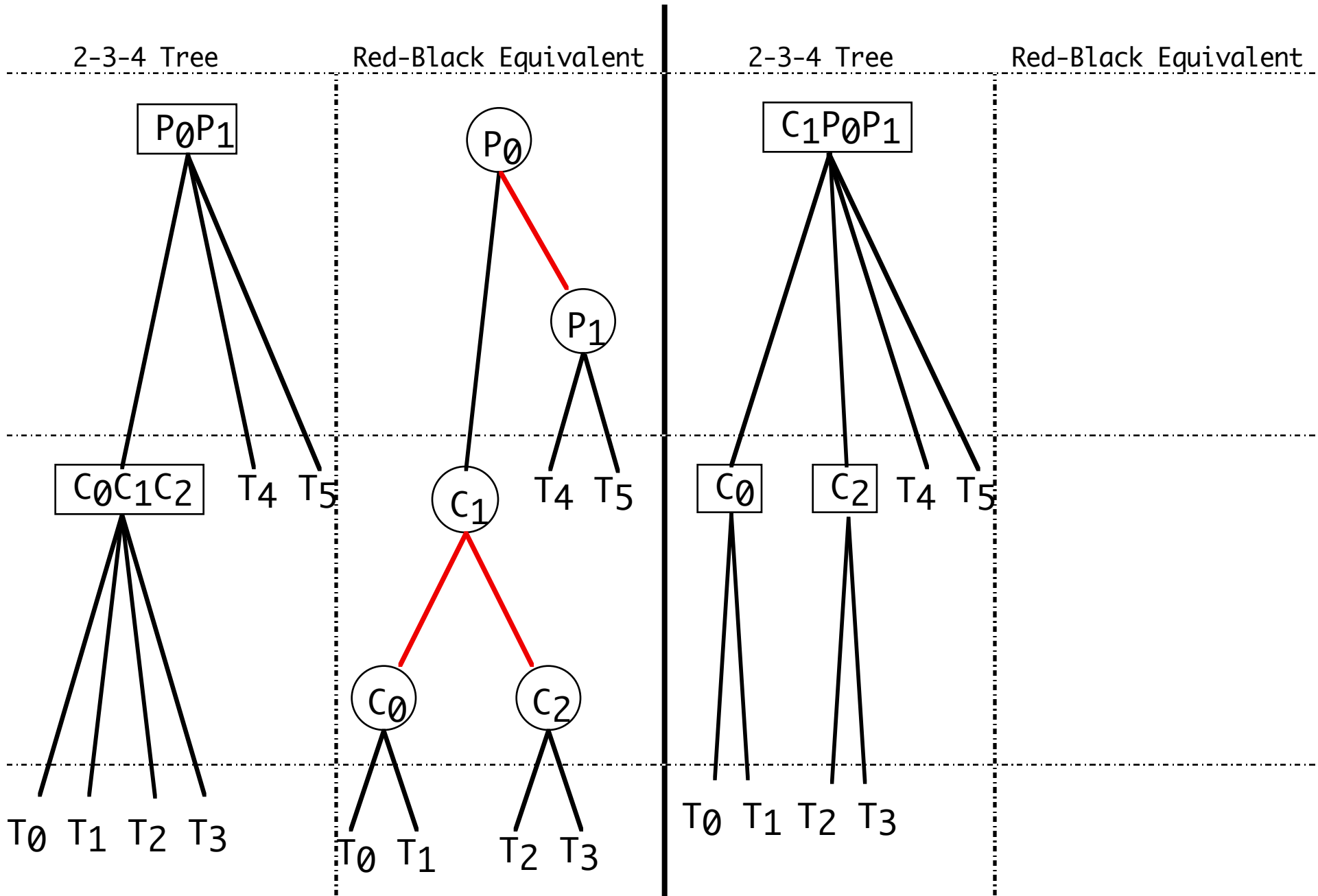
SPLIT LEFTMOST CHILD OF 3-NODE - RED ON RIGHT

BEFORE **AFTER**



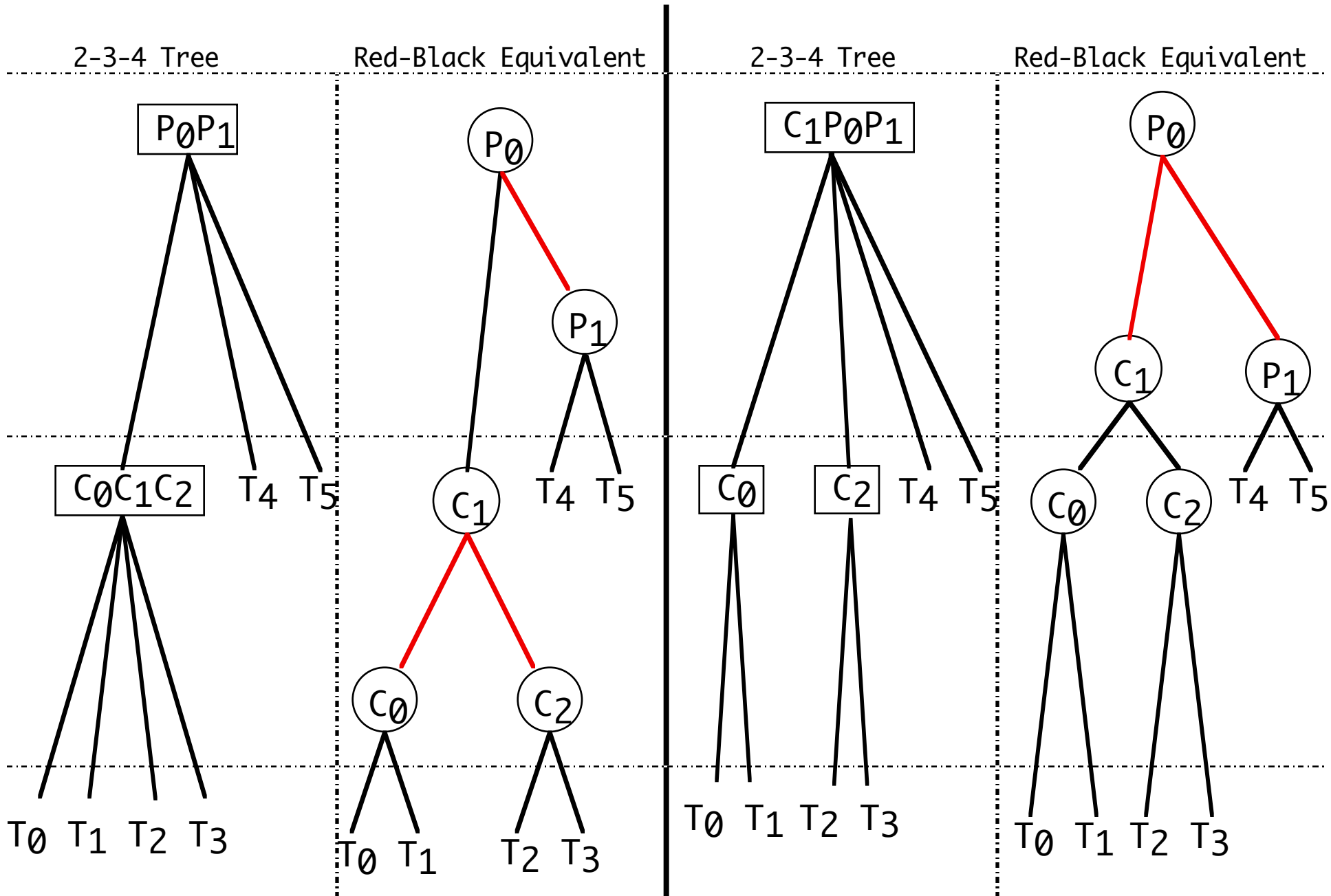
SPLIT LEFTMOST CHILD OF 3-NODE - RED ON RIGHT

BEFORE **AFTER**



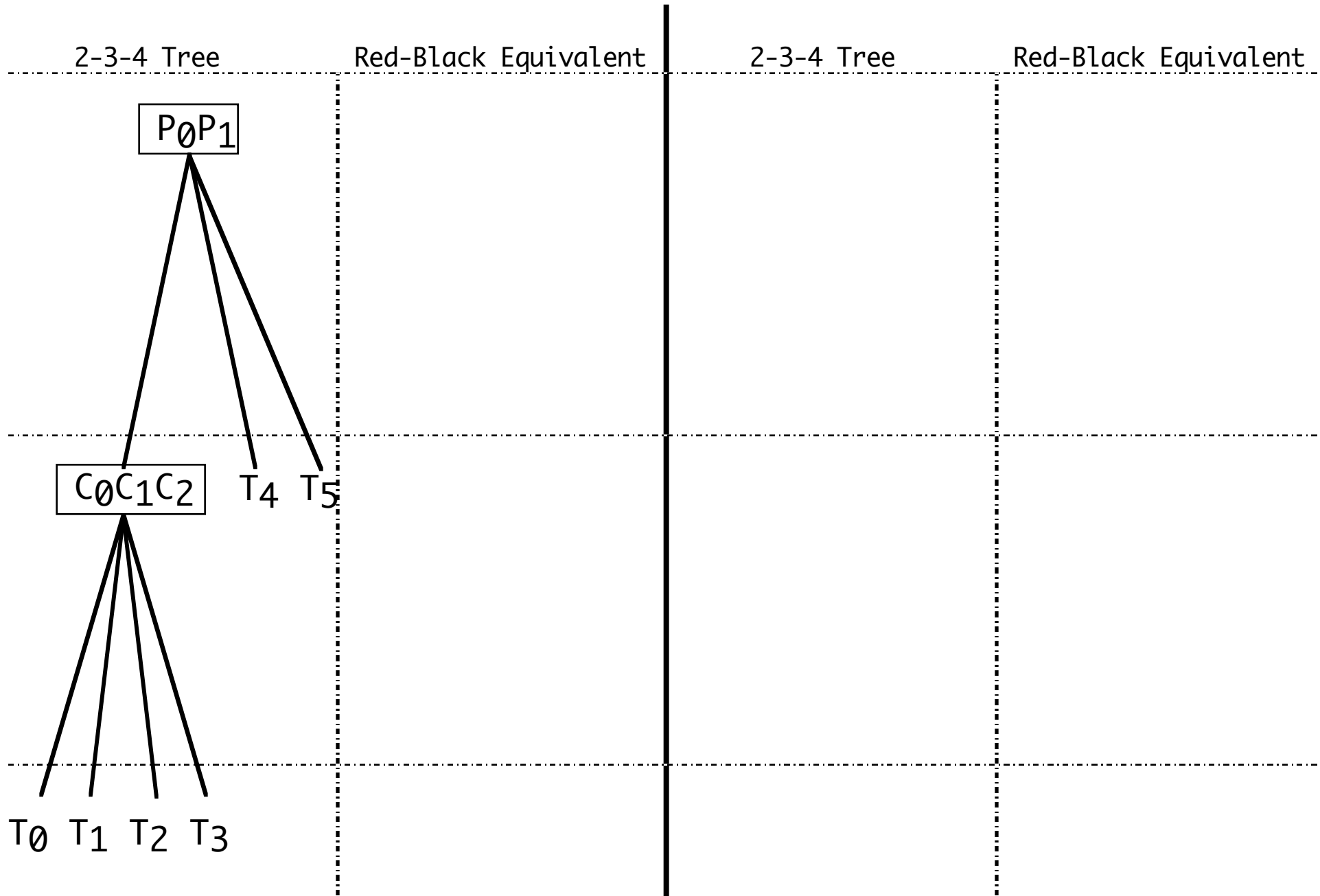
SPLIT LEFTMOST CHILD OF 3-NODE - RED ON RIGHT

BEFORE **AFTER**



SPLIT LEFTMOST CHILD OF 3-NODE - RED ON LEFT

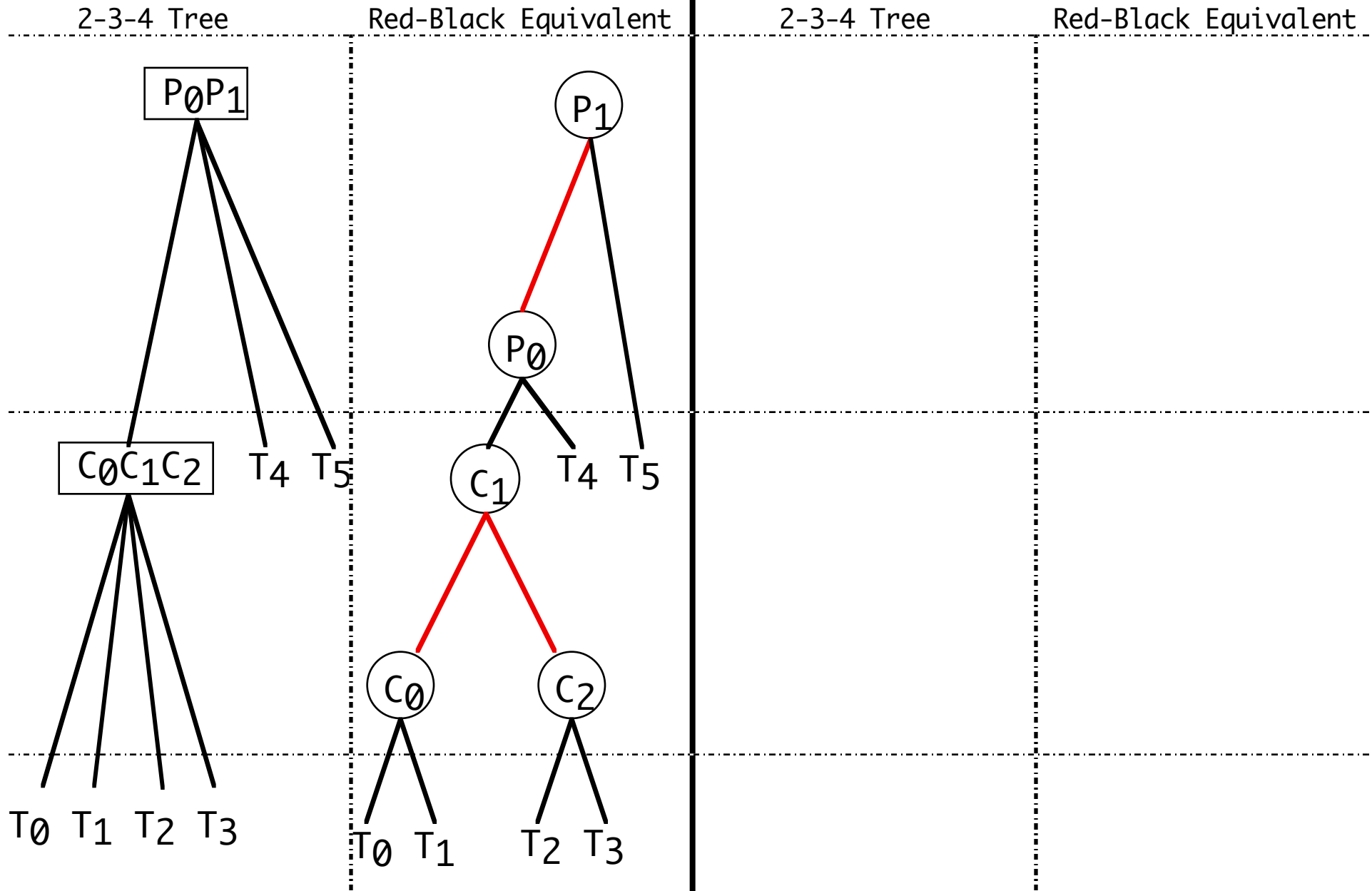
BEFORE **AFTER**



SPLIT LEFTMOST CHILD OF 3-NODE - RED ON LEFT

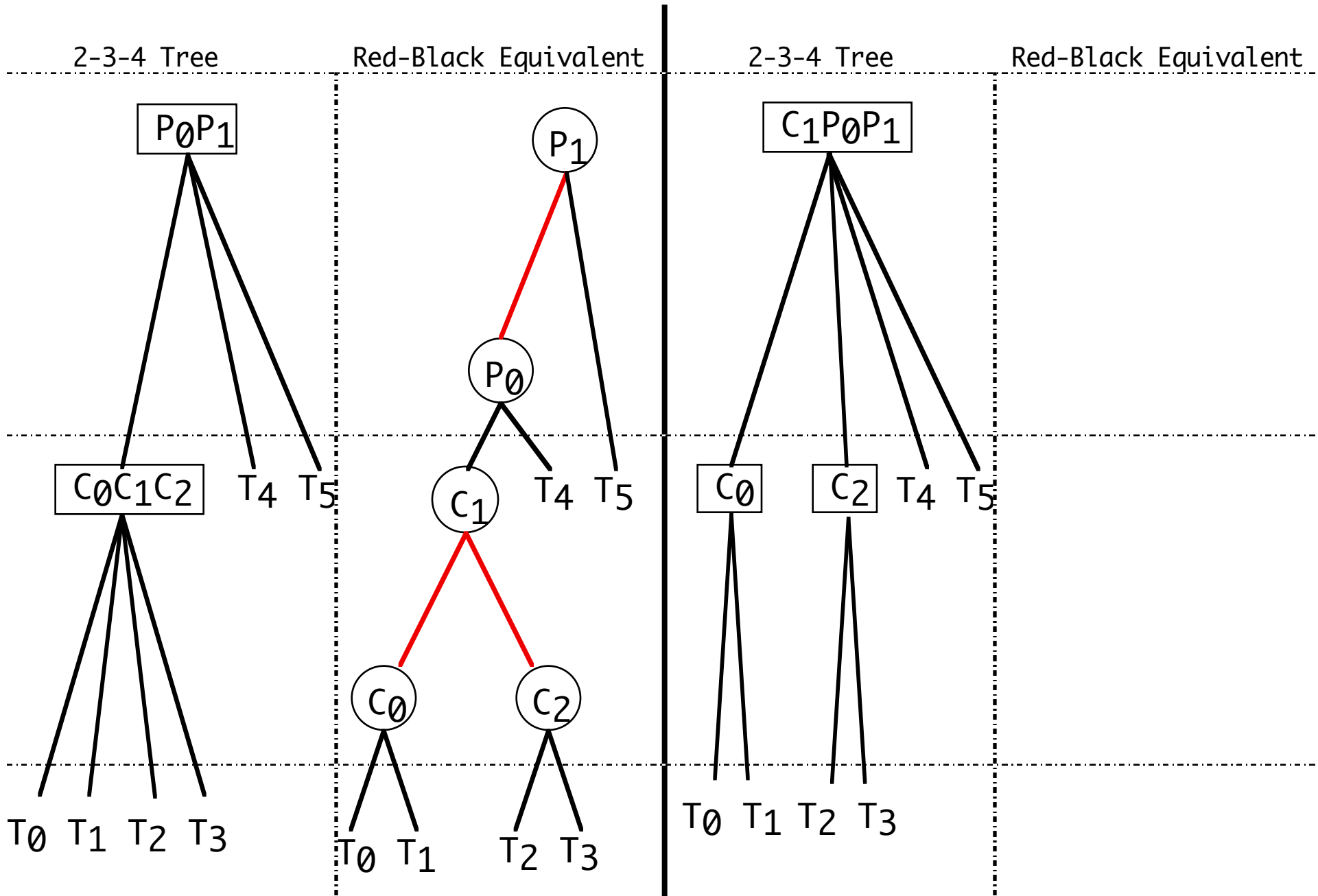
BEFORE

AFTER



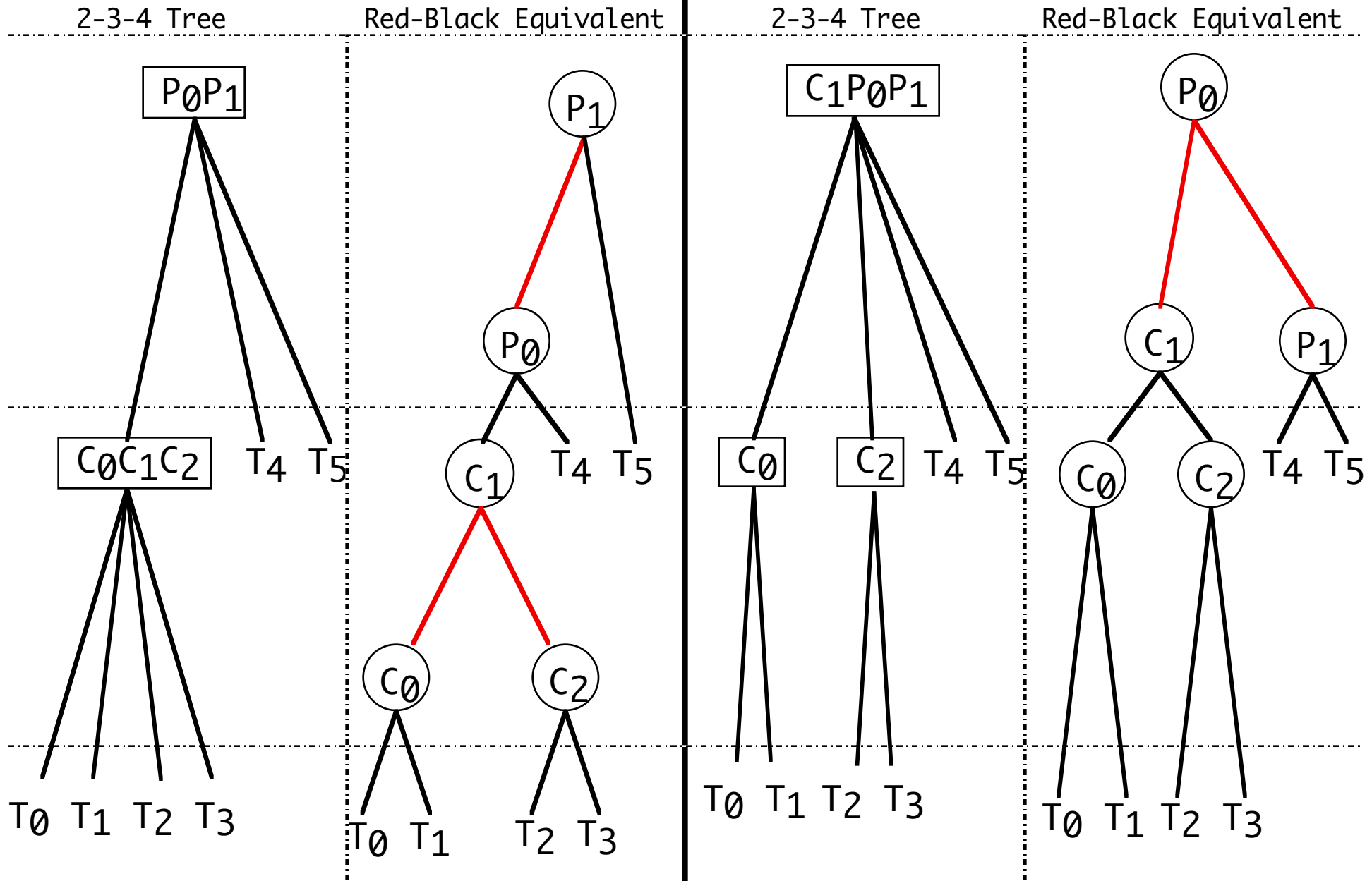
SPLIT LEFTMOST CHILD OF 3-NODE - RED ON LEFT

BEFORE **AFTER**



SPLIT LEFTMOST CHILD OF 3-NODE - RED ON LEFT

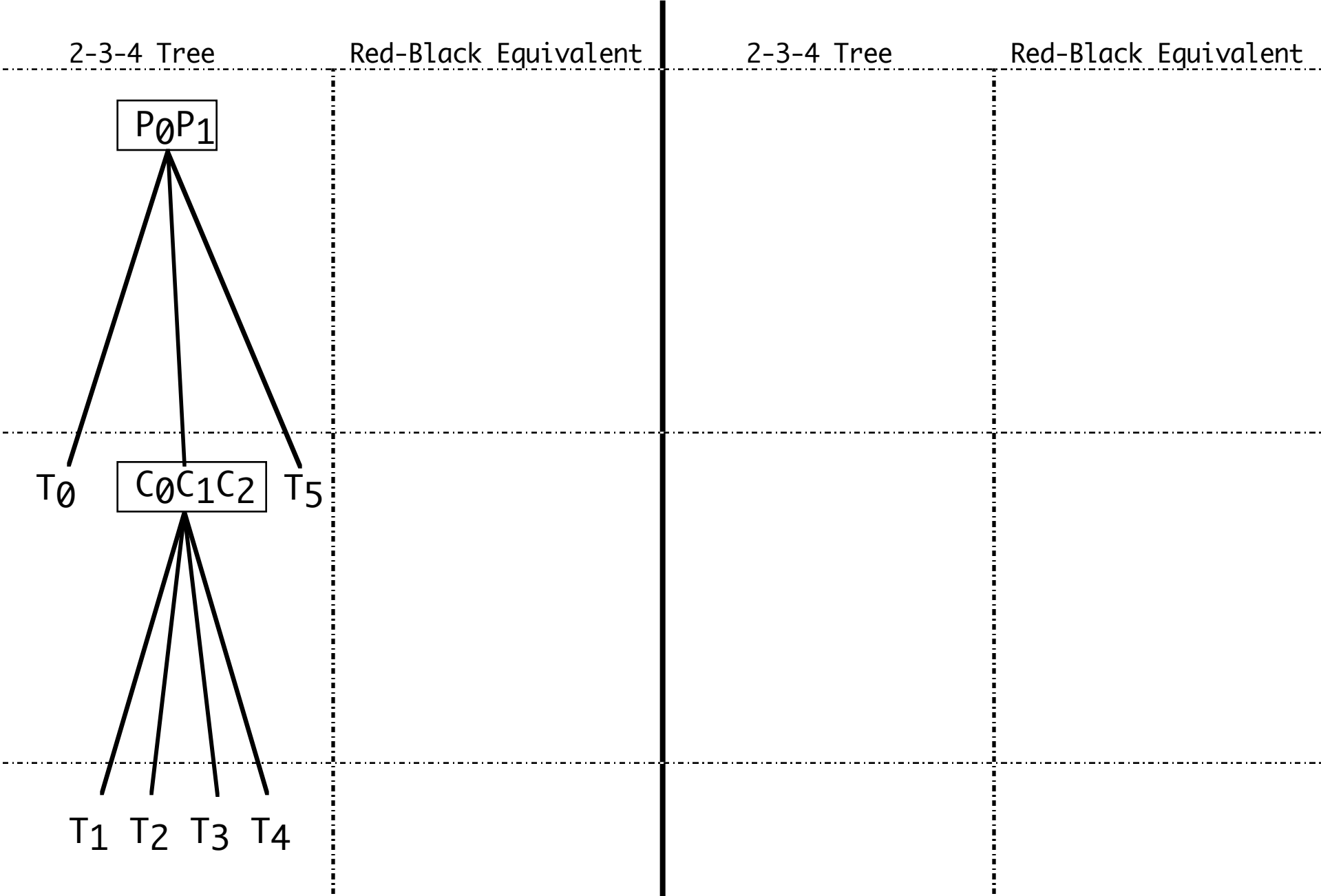
BEFORE **AFTER**



SPLIT MIDDLE CHILD OF 3-NODE - RED ON LEFT

BEFORE

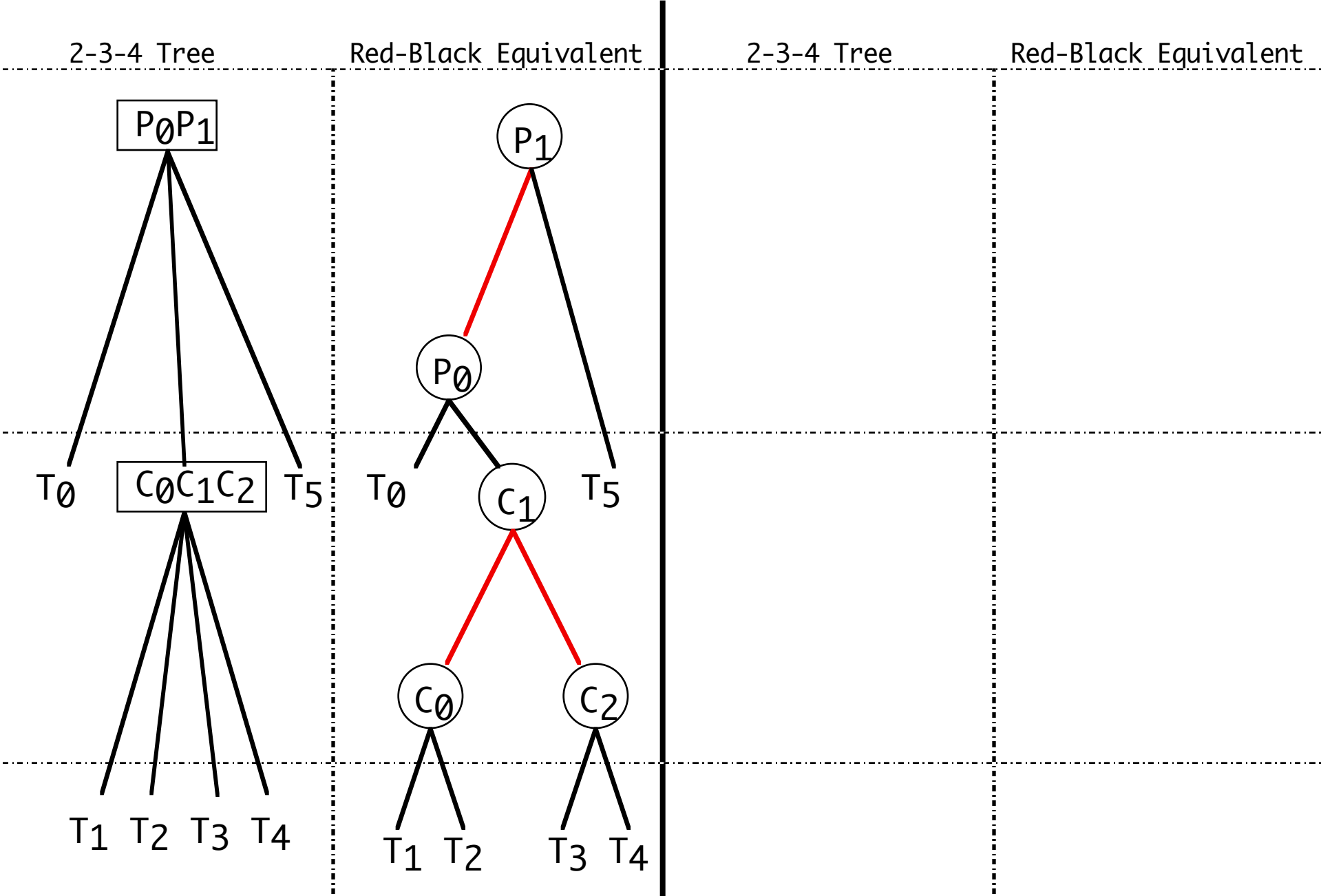
AFTER



SPLIT MIDDLE CHILD OF 3-NODE - RED ON LEFT

BEFORE

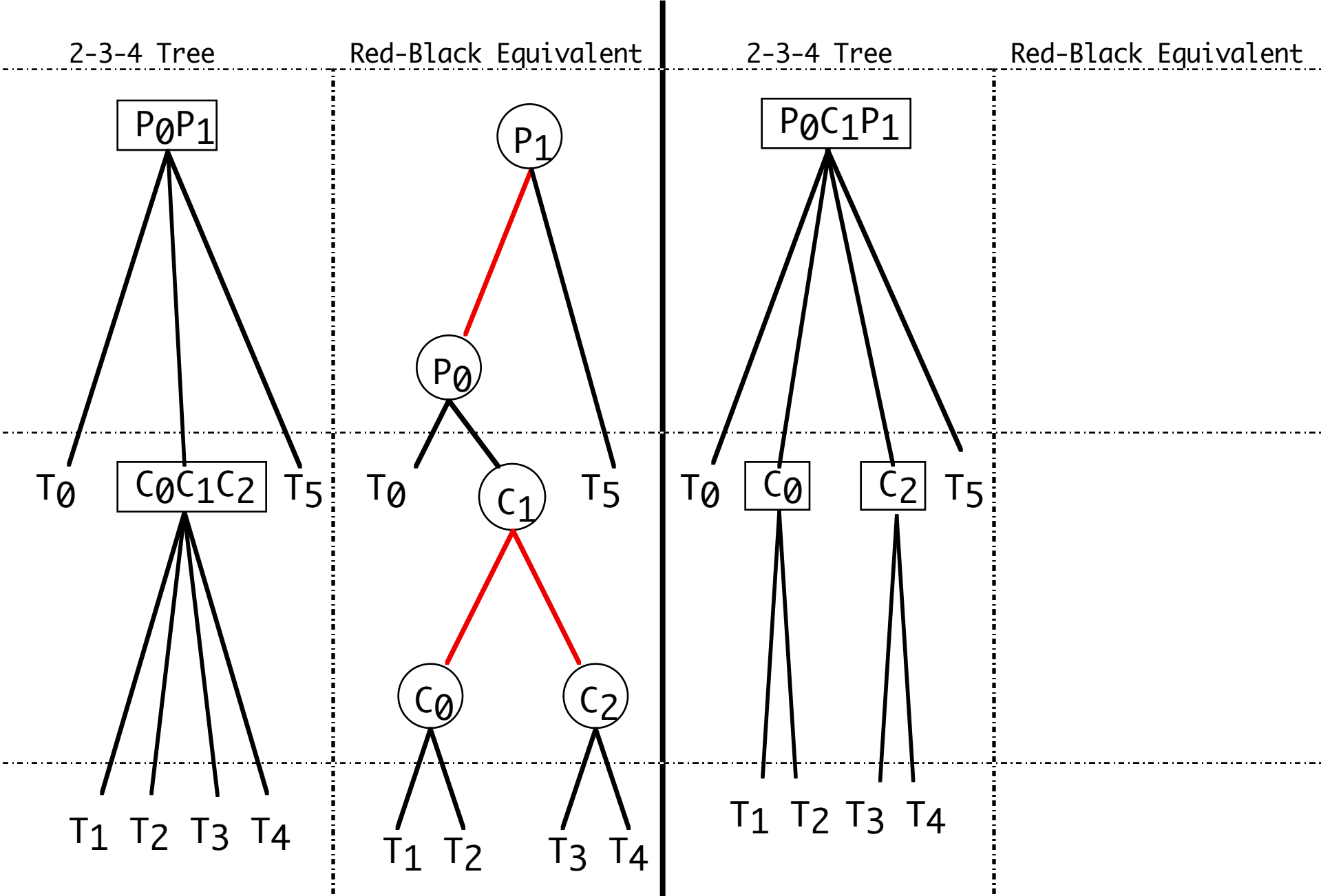
AFTER



SPLIT MIDDLE CHILD OF 3-NODE - RED ON LEFT

BEFORE

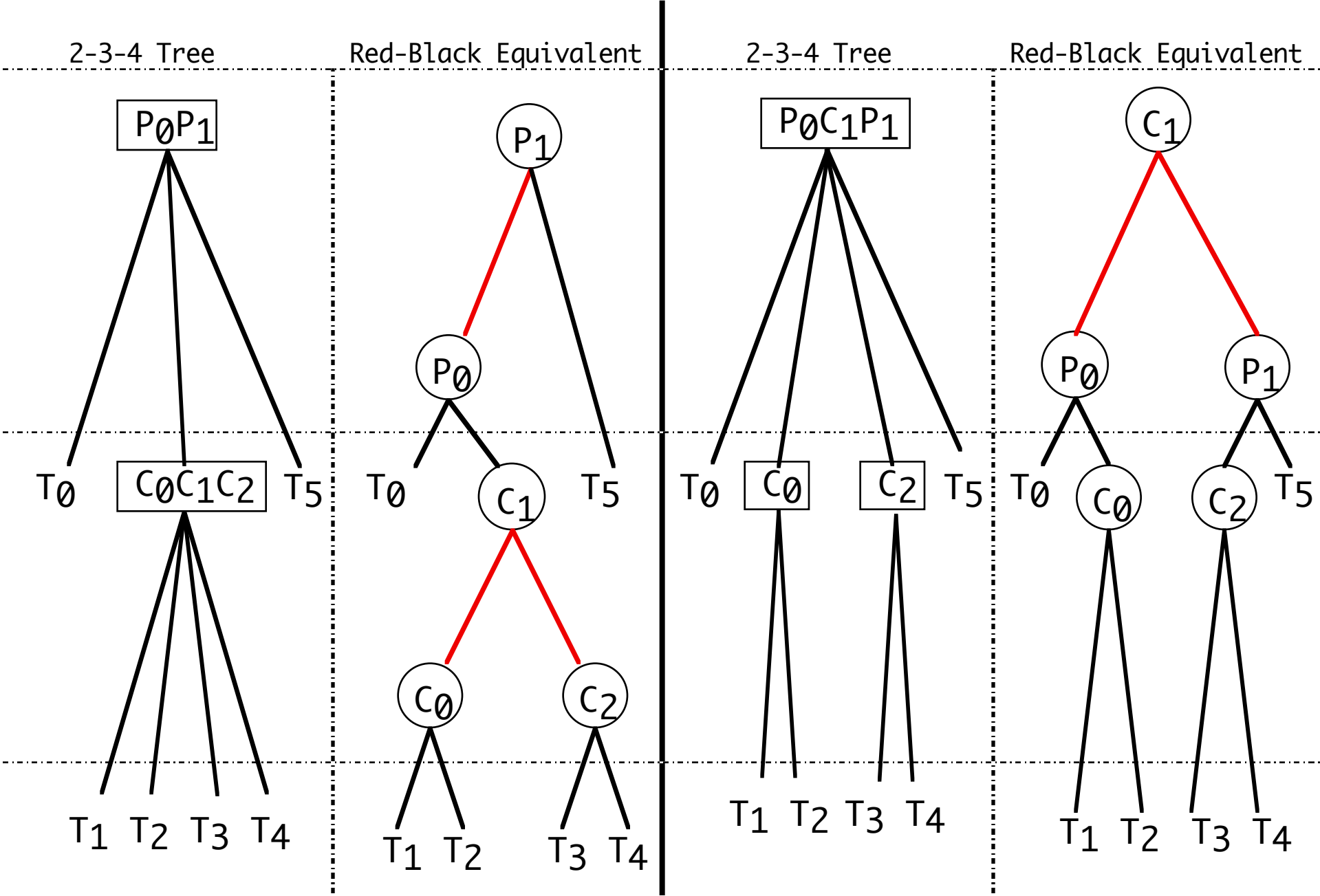
AFTER



SPLIT MIDDLE CHILD OF 3-NODE - RED ON LEFT

BEFORE

AFTER



-

There are three more cases that arise when splitting a node that is a child of a 3-Node.

Each is symmetrical (mirror-image) with one of the cases we have just considered

- 1) Split right child of a 3-Node whose red is on the left - symmetrical with split left child of a 3-Node whose red is on the right.
- 2) Split right child of a 3-Node whose red is on the right - symmetrical with split left child of a 3-Node whose red is on the left.
- 3) Split middle child of a 3-Node whose red is on the right - symmetrical with split middle child of a 3-Node whose red is on the left.

INSERT NEW LEFT KEY INTO 2-NODE

BEFORE **AFTER**



INSERT NEW LEFT KEY INTO 2-NODE

BEFORE **AFTER**

2-3-4 Tree

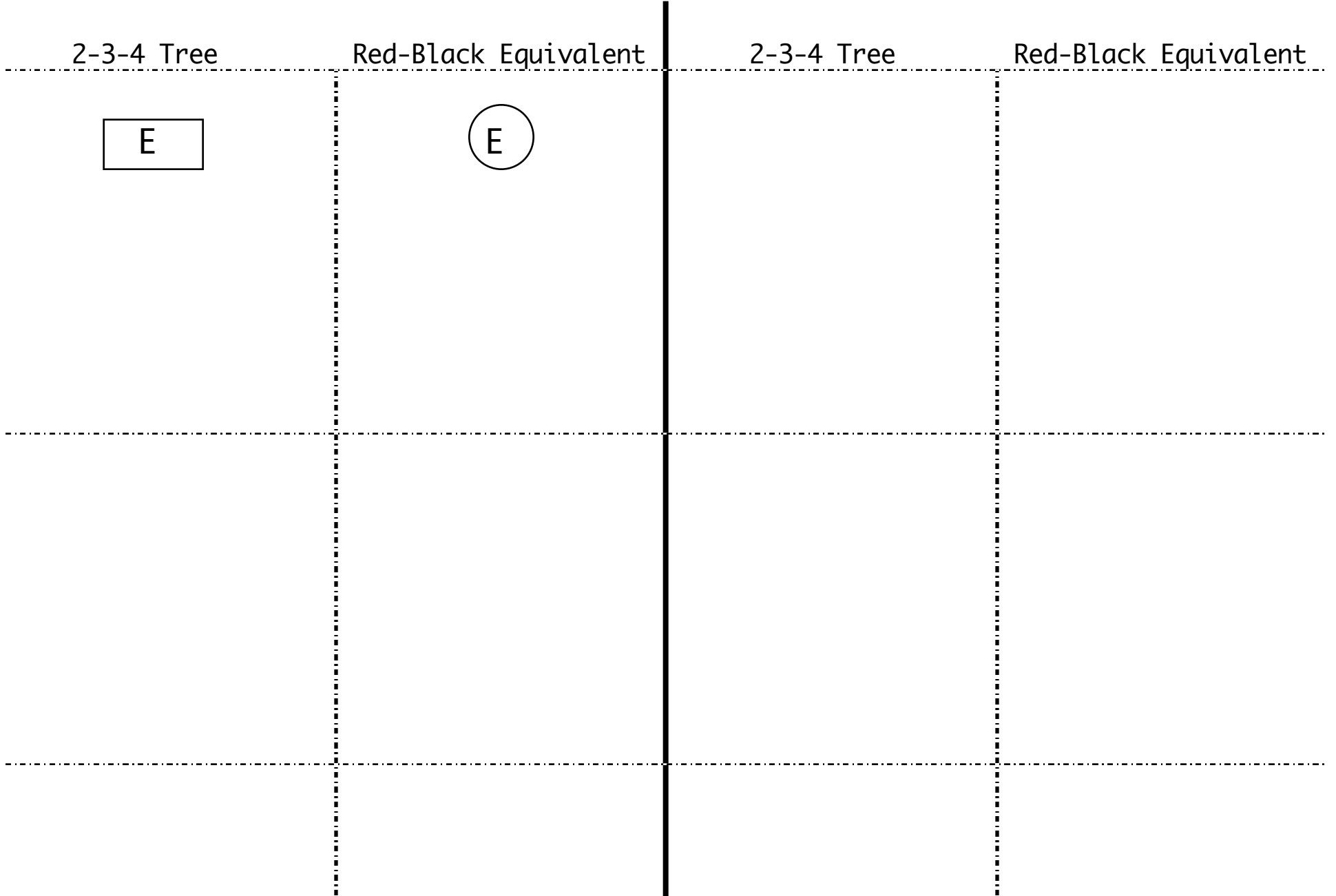


Red-Black Equivalent



2-3-4 Tree

Red-Black Equivalent



INSERT NEW LEFT KEY INTO 2-NODE

BEFORE **AFTER**

2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent

E

E

N E



INSERT NEW LEFT KEY INTO 2-NODE

BEFORE **AFTER**

2-3-4 Tree



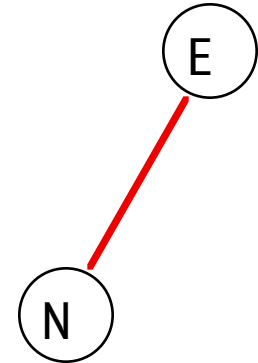
Red-Black Equivalent



2-3-4 Tree



Red-Black Equivalent



INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON RIGHT

BEFORE **AFTER**



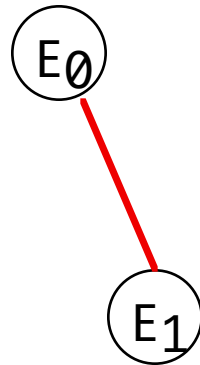
INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON RIGHT

BEFORE **AFTER**

2-3-4 Tree

E_0E_1

Red-Black Equivalent

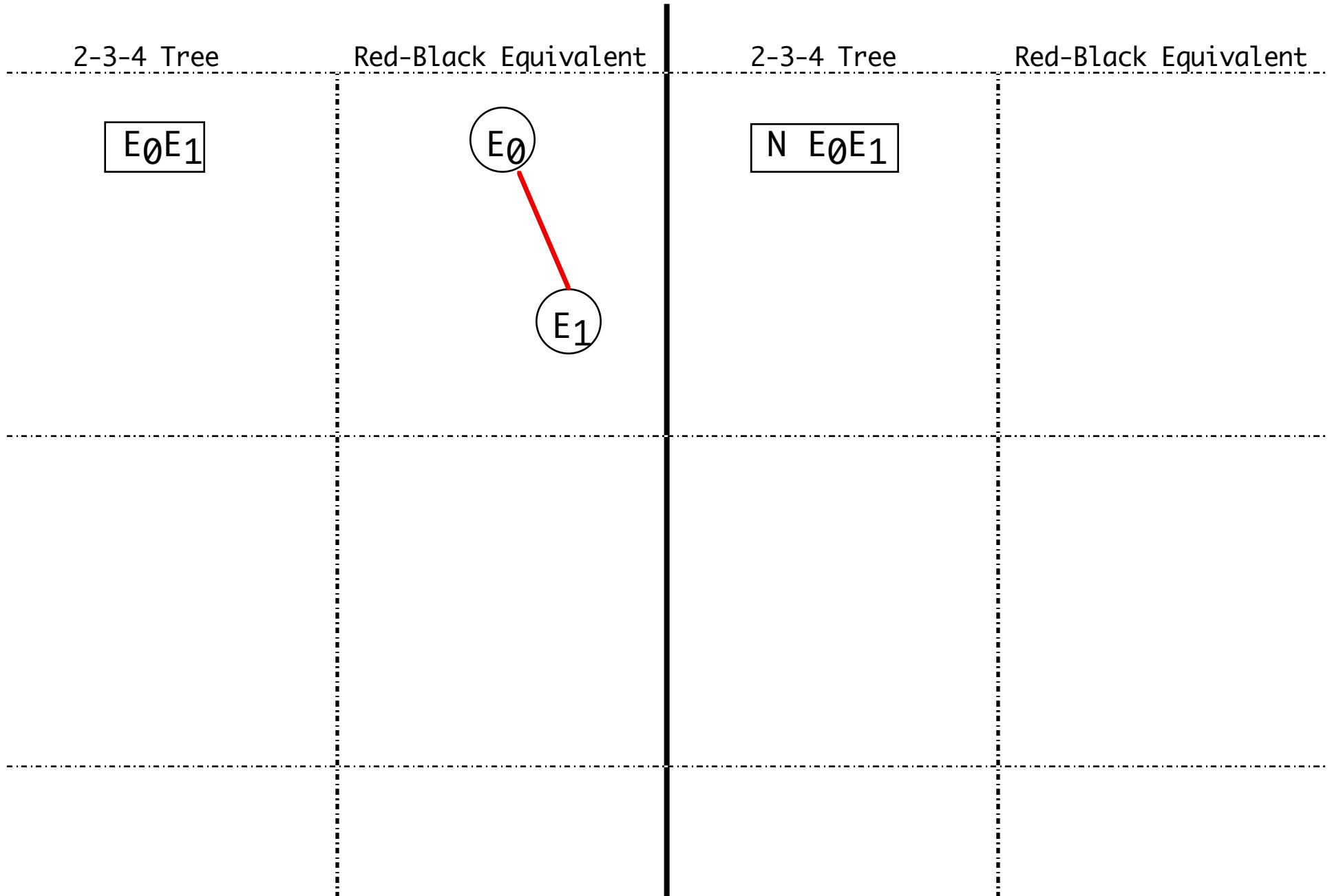


2-3-4 Tree

Red-Black Equivalent

INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON RIGHT

BEFORE **AFTER**



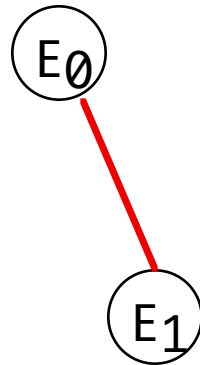
INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON RIGHT

BEFORE **AFTER**

2-3-4 Tree

$E_0 E_1$

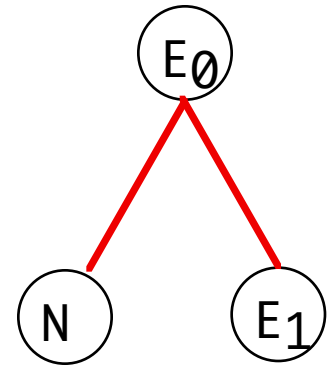
Red-Black Equivalent



2-3-4 Tree

$N E_0 E_1$

Red-Black Equivalent



INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON LEFT

BEFORE

AFTER

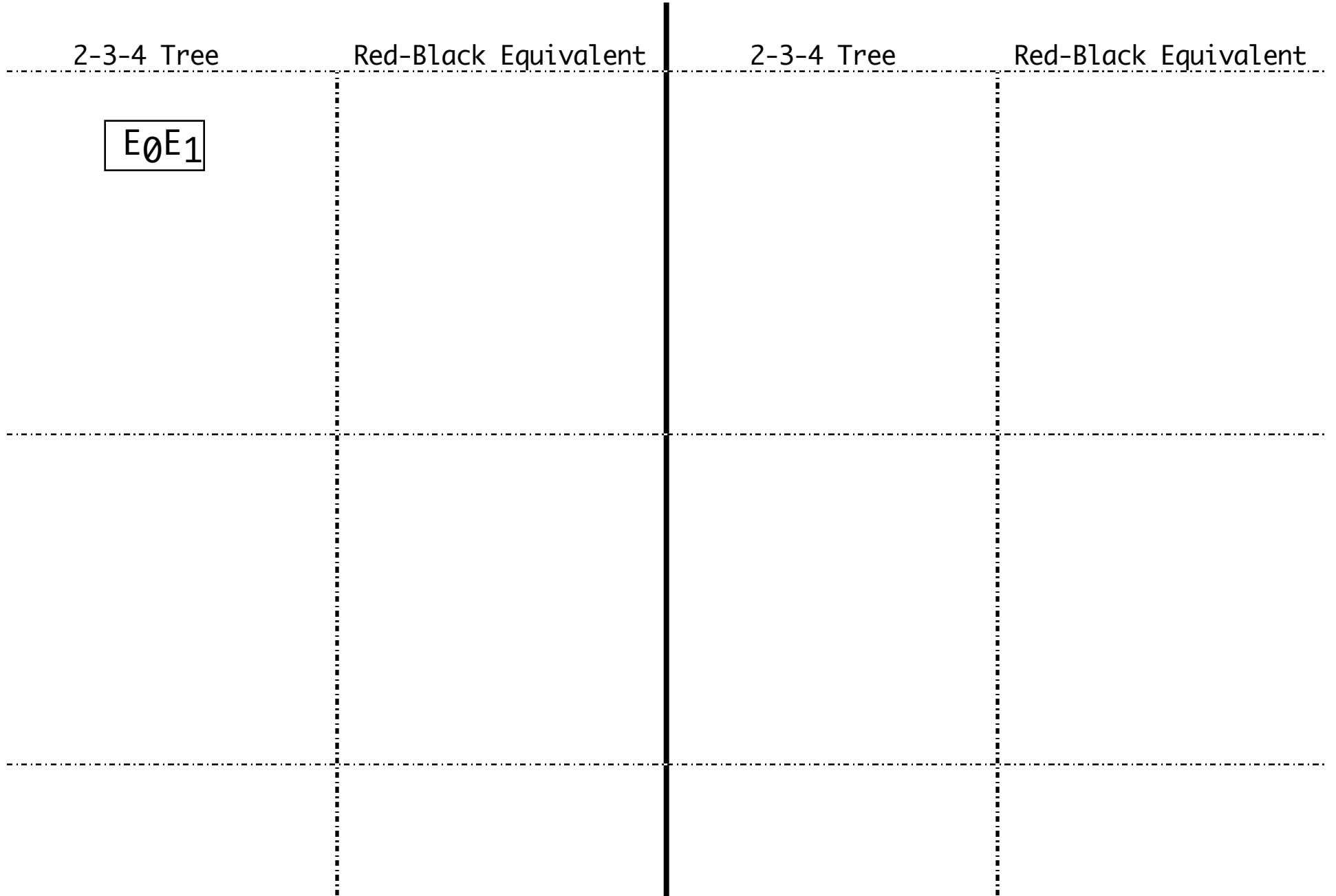
2-3-4 Tree

Red-Black Equivalent

2-3-4 Tree

Red-Black Equivalent

$E_0 E_1$



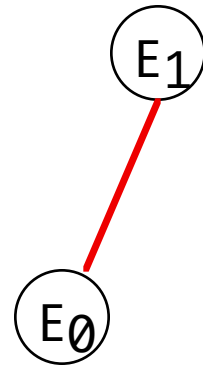
INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree

E_0E_1

Red-Black Equivalent



2-3-4 Tree

Red-Black Equivalent

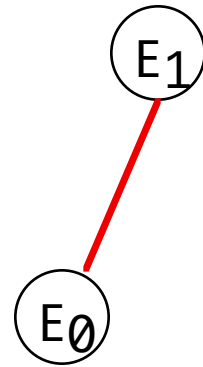
INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree

$E_0 E_1$

Red-Black Equivalent



2-3-4 Tree

$N E_0 E_1$

Red-Black Equivalent

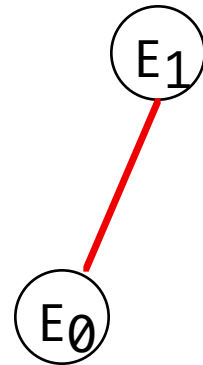
INSERT NEW LEFTMOST KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree

$E_0 E_1$

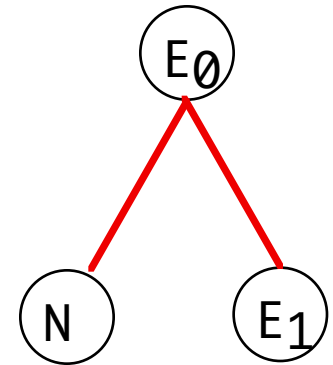
Red-Black Equivalent



2-3-4 Tree

$N E_0 E_1$

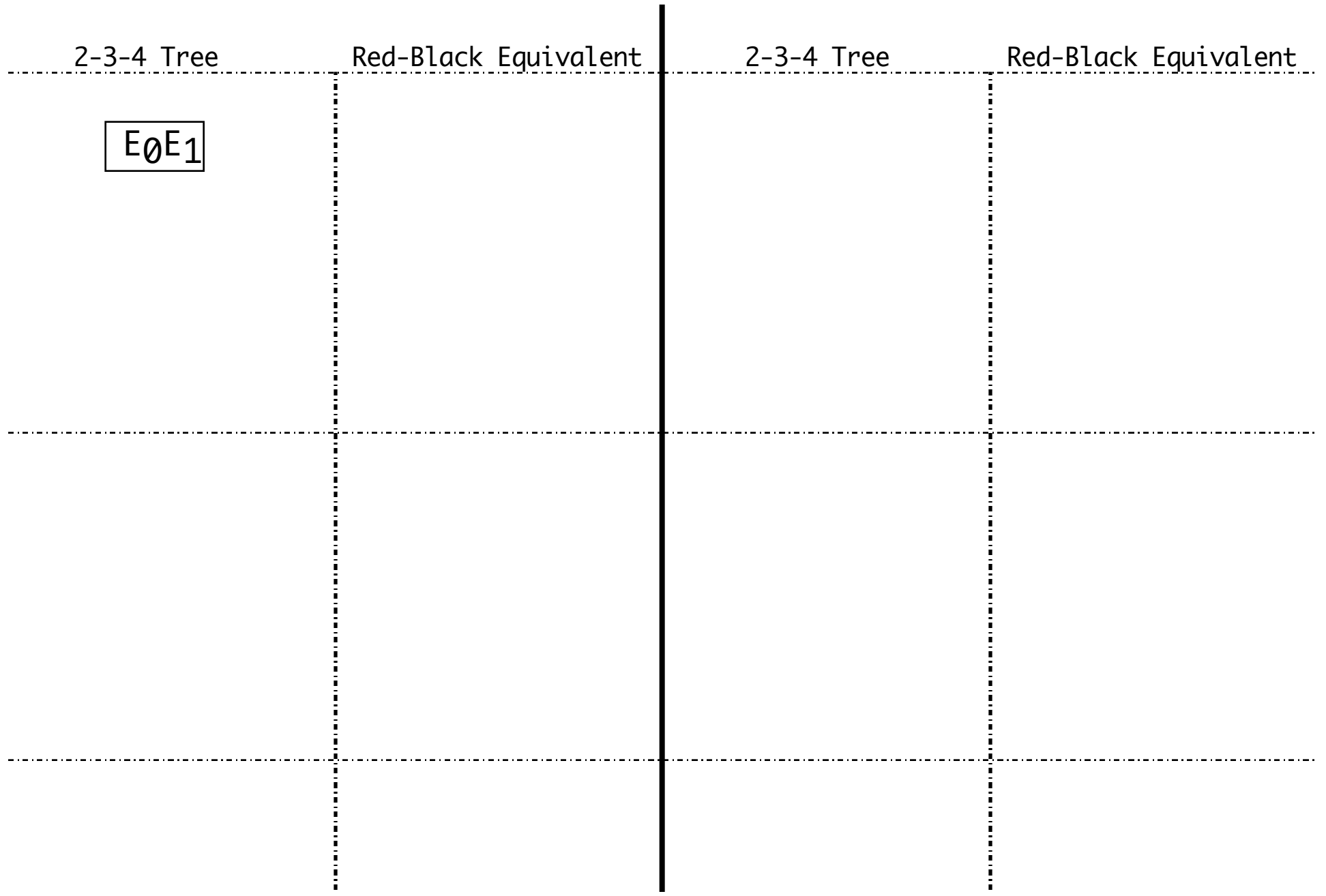
Red-Black Equivalent



INSERT NEW MIDDLE KEY INTO 3-NODE - RED ON LEFT

BEFORE

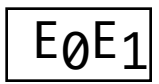
AFTER



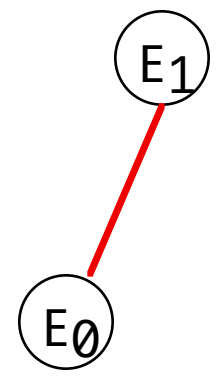
INSERT NEW MIDDLE KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree



Red-Black Equivalent



2-3-4 Tree

Red-Black Equivalent



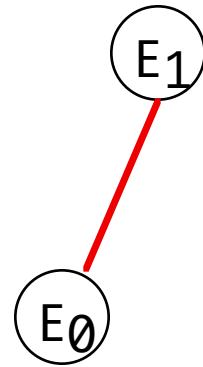
INSERT NEW MIDDLE KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree

$E_0 E_1$

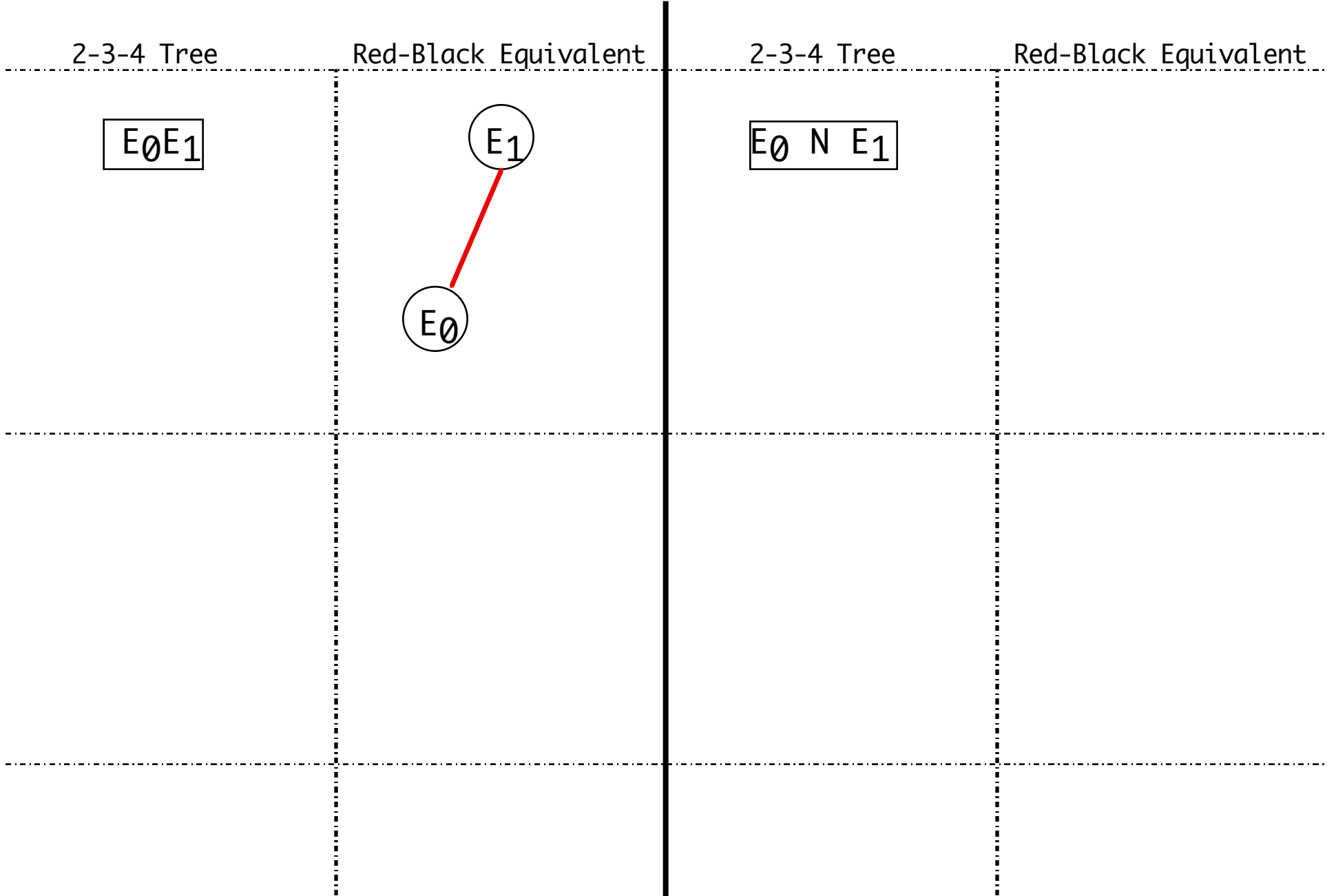
Red-Black Equivalent



2-3-4 Tree

$E_0 N E_1$

Red-Black Equivalent



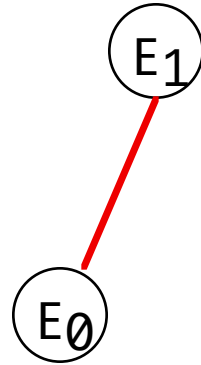
INSERT NEW MIDDLE KEY INTO 3-NODE - RED ON LEFT

BEFORE **AFTER**

2-3-4 Tree

$E_0 E_1$

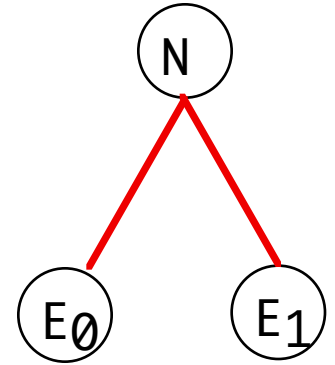
Red-Black Equivalent



2-3-4 Tree

$E_0 N E_1$

Red-Black Equivalent



There are three more cases that arise when inserting a new key into a 3-Node.

Each is symmetrical (mirror-image) with one of the cases we have just considered

- 1) Insert new rightmost child into a 3-Node whose red is on the left - symmetrical with insert new leftmost child into a 3-Node whose red is on the right.
- 2) Insert new rightmost child into a 3-Node whose red is on the right - symmetrical with insert new leftmost child of a 3-Node whose red is on the left.
- 3) Insert new middle child into a 3-Node whose red is on the right - symmetrical with insert new middle child into a 3-Node whose red is on the left.