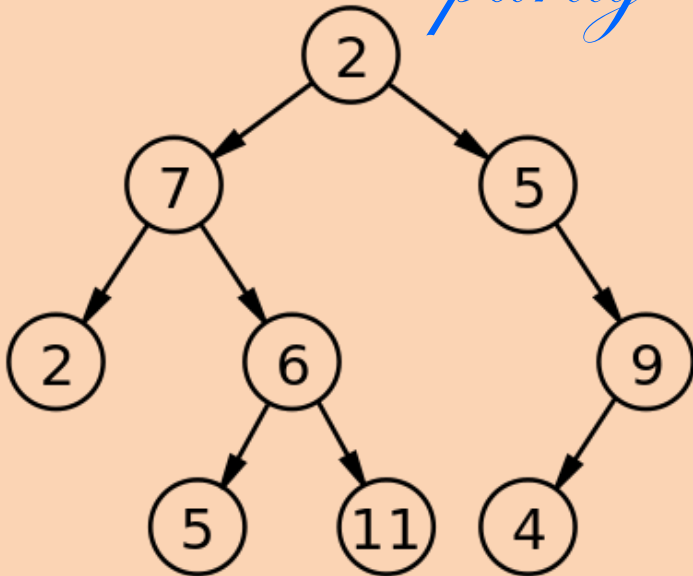




# ساختارهای داده

درخت‌های قرمز-سیاه ۲-۳ و الگوریتم حذف برابر-طلب

*2-3 Red-Black Trees and the  
parity-seeking delete algorithm*



مدرس:

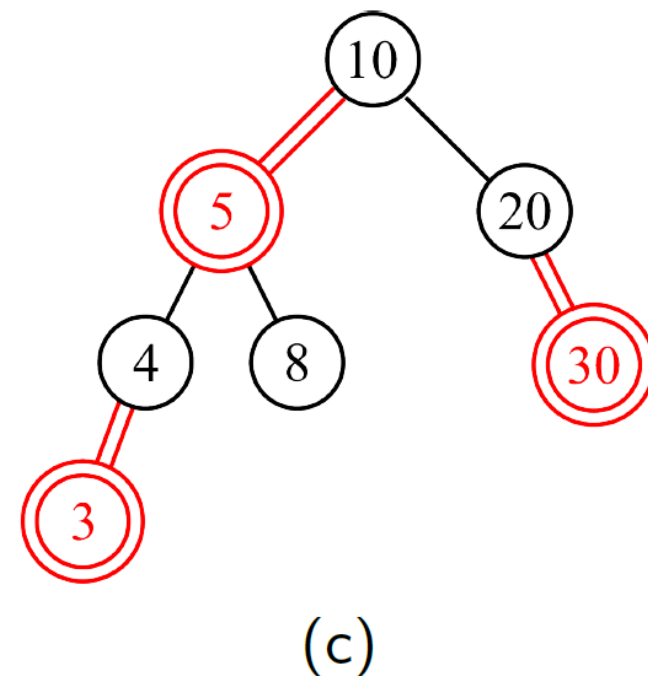
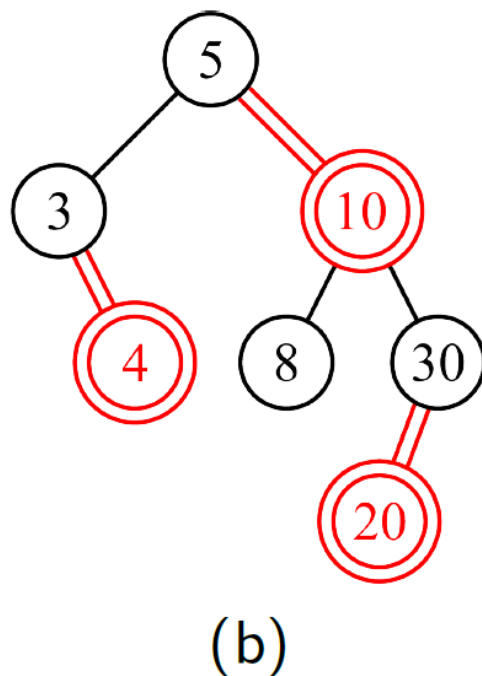
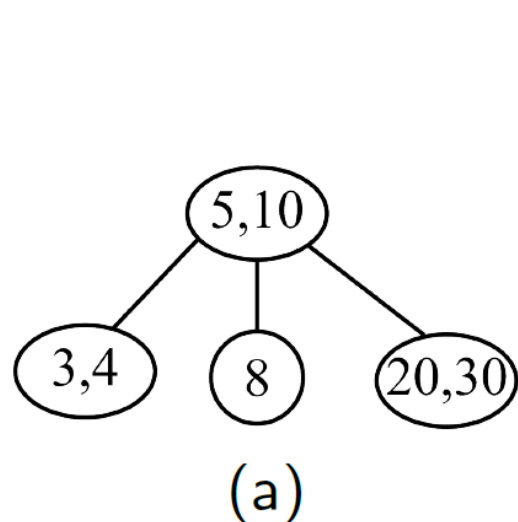
سید کمال الدین غیاثی شیرازی

## تعریف درخت‌های قرمز-سیاه 2-3

---

- یک درخت قرمز-سیاه 2-3 یک درخت قرمز-سیاه است که در آن هر گره حداکثر یک فرزند قرمز دارد.
- هر درخت قرمز-سیاه 2-3 دقیقاً با یک درخت ۲-۳ معادل است، اما متناظر با یک درخت ۲-۳ ممکن است چندین درخت قرمز-سیاه ۲-۳ وجود داشته باشد.

## مثالی از تناظر درخت‌های ۳-۲ با درخت‌های قرمز-سیاه ۳-۲



# الکوریتم درج

---

- محل گره جدید را طبق قانون درخت جستجوی دودویی به دست آور.
- گره جدید را با رنگ قرمز درج کن.
- با شروع از گره جاری تا زمانی که حداقل یکی از قوانین صفحه‌ی بعد قابل اجرا است، اصلاح را ادامه بده.

# قوانین اصلاح درخت پس از عمل درج



(a) Align red links.



(b) Convert red parent-child pairs to red siblings.



(c) Flip color on red siblings.

## کد عمل درج در درخت‌های قرمز-سیاه ۲-۳

```
void put_bottom_up_pass(Node* x) {
    Node* p;
    while (1) { // x->color is certainly RED
        p = x->parent;
        if (p->left->color == RED && p->right->color == RED) { // Rule (c)
            colorFlip(p);
            x = p;
        }
        else if (p->color == RED) {
            if (p == p->parent->left) {
                if (x == p->right) { // Rule (a)
                    x = p;
                    p = rotateLeft(p);
                }
                rotateRight(p->parent); // Rule (b)
            }
            else {
                if (x == p->left) { // Rule (a)
                    x = p;
                    p = rotateRight(p);
                }
                rotateLeft(p->parent); // Rule (b)
            }
        }
        else
            break;
    }
}
```

# مشاهده‌ی عملکرد الگوریتم درج در يك مثال جامع

---

- فایل psrbt-2-3-insert-images.pdf را ببینید.

# الگوریتم حذف

- حذف از گره‌های درجه‌ی دو به حذف از گره‌های برگ و درجه‌ی یک ساده می‌شود.

- حذف گره برگ قرمز

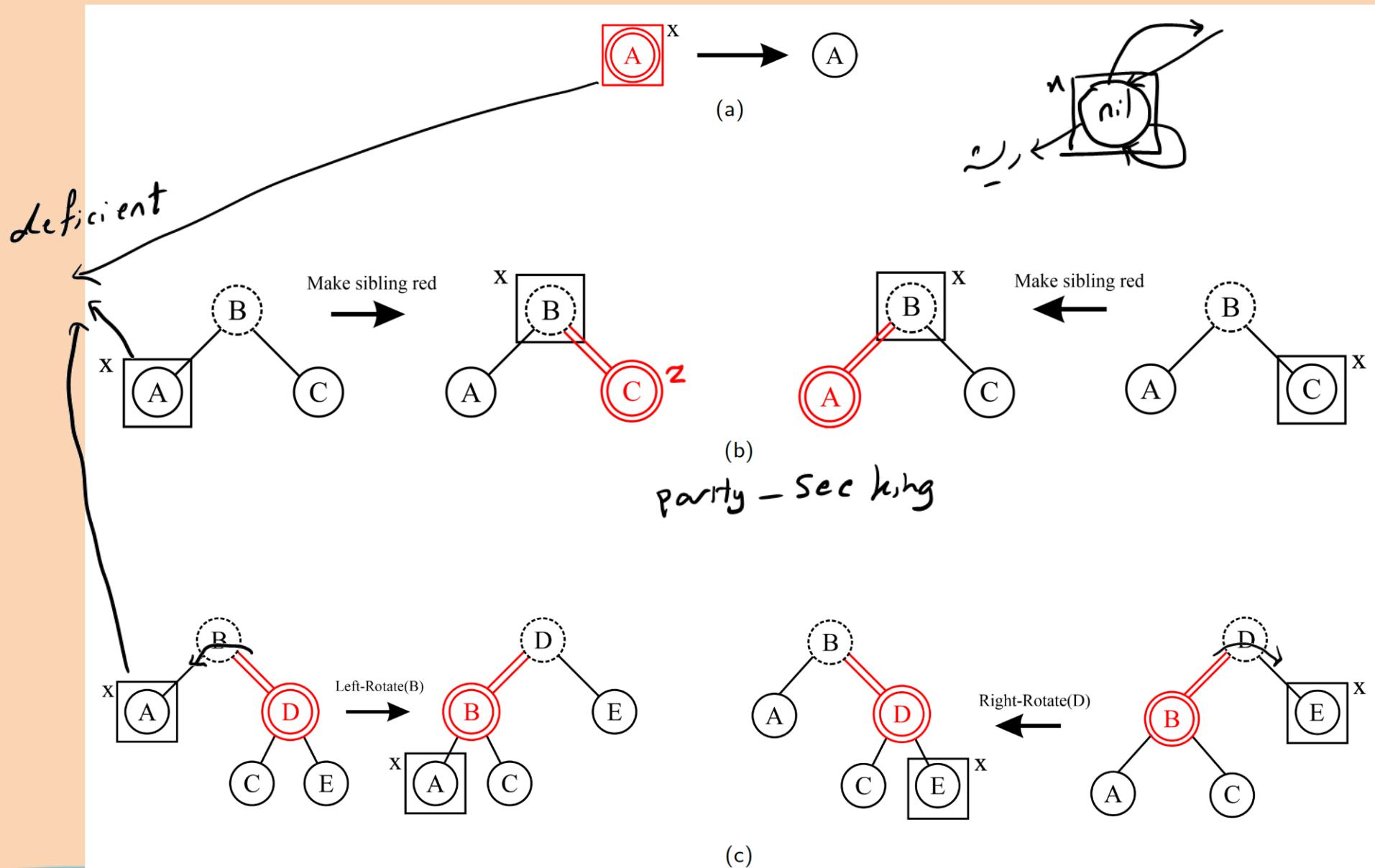


- حذف گره با درجه‌ی یک



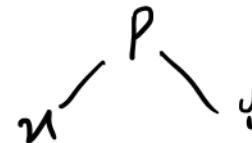
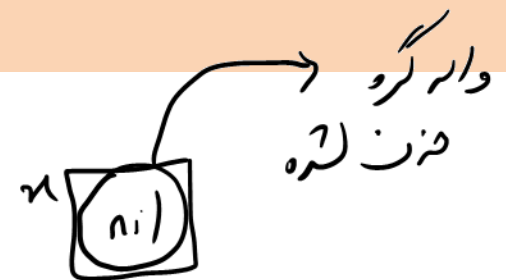


# الکوریتم حذف - قوانین اصلی (برابری-طلبی)



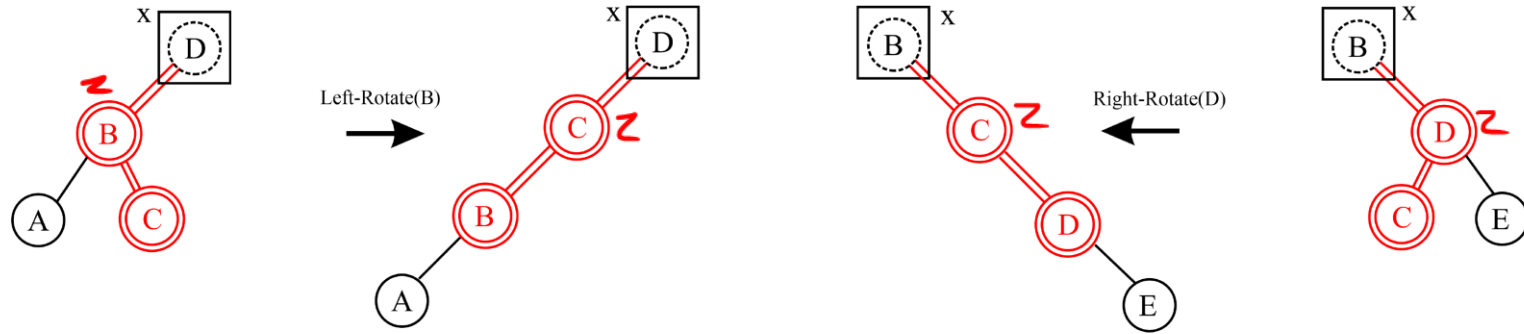
# کد قوانین اصلی (برابری-طلبی) الگوریتم حذف

```
void remove_bottom_up_pass(Node* x) {
    while (x != nil->left && x->color == BLACK) {
        x = applyParitySeekingRules(x);
    }
    x->color = BLACK; // Rule (a)
}
```

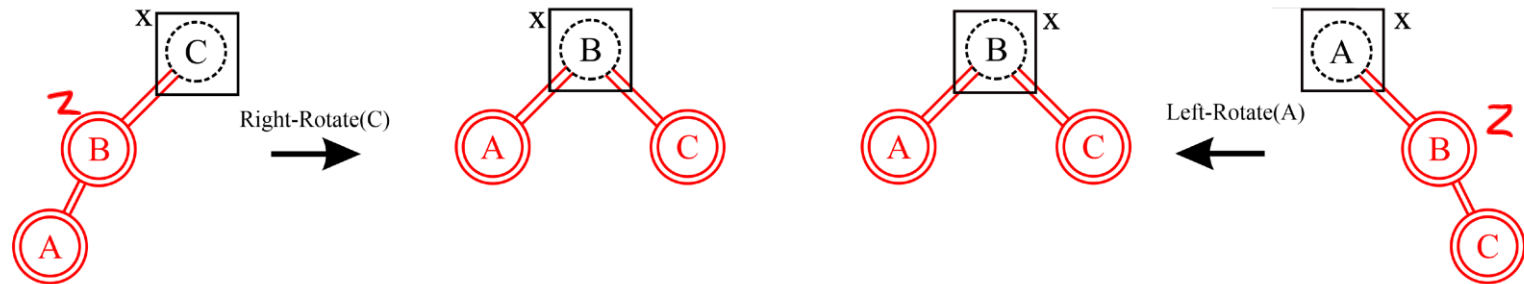


```
Node* applyParitySeekingRules(Node* x) {
    Node* p = x->parent;
    Node* y = x == x->parent->left? x->parent->right:x->parent->left;
    (c) {
        if (y->color == RED) // Rule (c): Rotate to have a black sibling
            if (x == x->parent->left)
                rotateLeft(p);
            else
                rotateRight(p);
        else { //Rule (b): Make the sibling red
            y->color = RED;
            x = p;
            x = fixUp(x);
        }
    }
    return x;
}
```

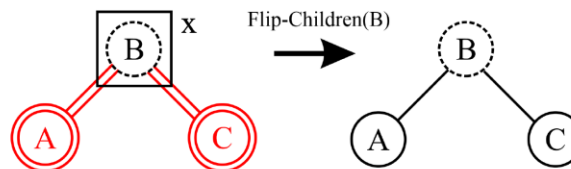
# الكوريتم حذف - قوانين اصلاح



(a) If red links are not aligned, rotate to have aligned red links.



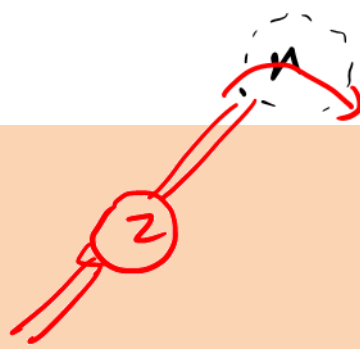

(b) Rotate aligned red links to have two red children.



(c) Flip color of children.

# كد الكوريتم حذف - قوانين اصلاح

```
Node* fixUp(Node* x) {
    Node* z = x->left->color == RED ? x->left : x->right;
    if (z->left->color == RED || z->right->color == RED) {
        if (z == z->parent->right) {
            if (z->right->color == BLACK) // Fixing rule (a)
                rotateRight(z);
            x = rotateLeft(x); // Fixing rule (b)
        }
        else {
            if (z->left->color == BLACK) // Fixing rule (a)
                rotateLeft(z);
            x = rotateRight(x); // Fixing rule (b)
        }
        x->left->color = BLACK; // Fixing rule (c)
        x->right->color = BLACK; // Fixing rule (c)
        x = nil->left; // terminate the bottom up pass
    }
    return x;
}
```



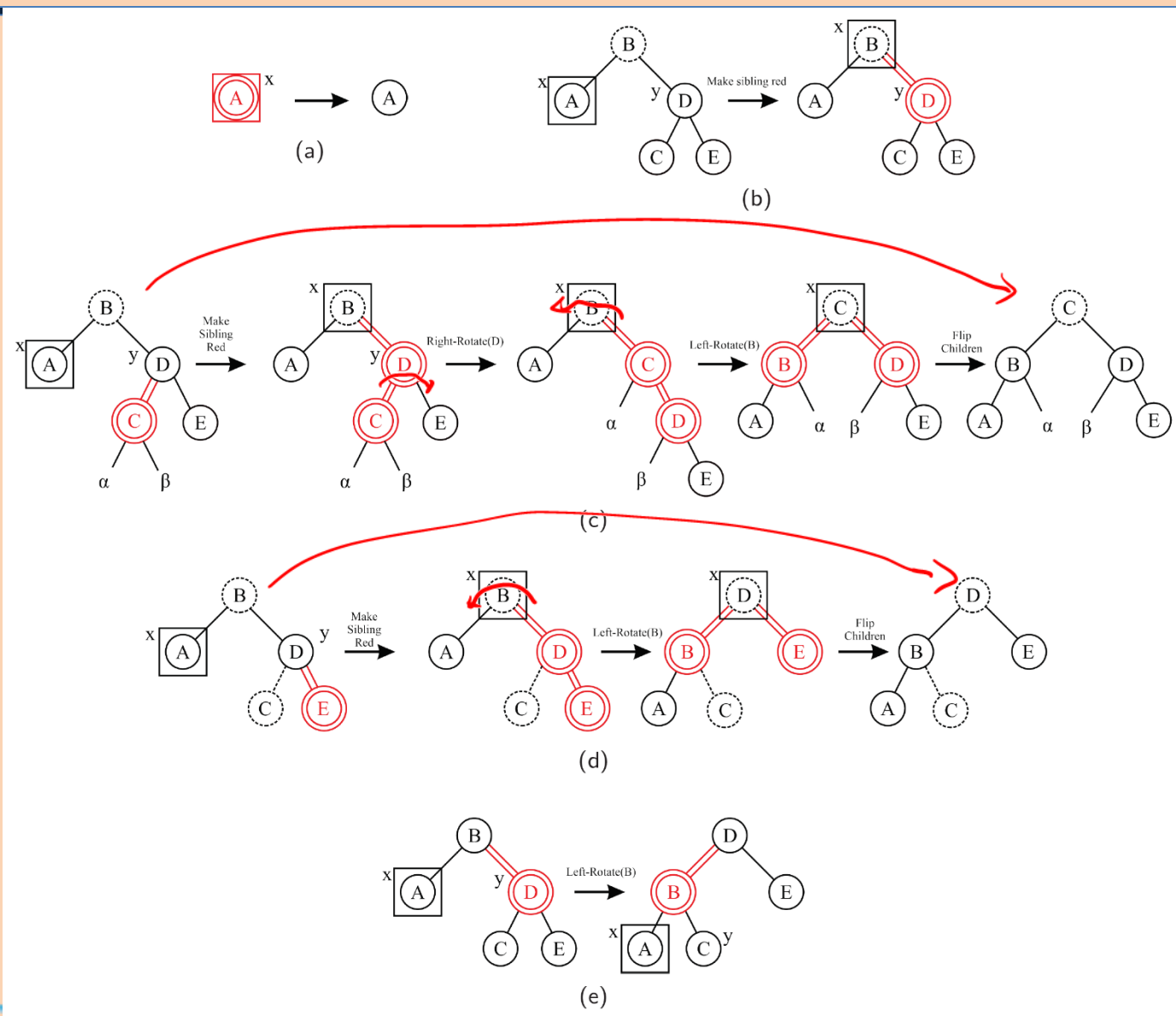
# مشاهده‌ی عملکرد الگوریتم حذف برابری-طلب برای درخت‌های

---

## قرمز-سیاه ۲-۳ در یک مثال جامع

• فایل [psrbt-2-3-delete-images.pdf](#) را ببینید.

# الکوریتم حذف برابری-طلب با الگوریتم حذف CLRS معادل است



# مشاهده‌ی عملکرد الگوریتم حذف برابری-طلب برای درخت‌های قرمز-سیاه ۲-۳-۴ در یک مثال جامع

---

• فایل `psrbt-2-3-4-delete-images.pdf` را ببینید.

# پیاده‌سازی

---

● پیاده‌سازی ما از این الگوریتم‌ها به زبان C++ را می‌توانید اینجا مشاهده کنید.

- <https://github.com/k-ghiasi/RedBlackTrees/blob/main/ParitySeeking-23RedBlackBST.h>
- <https://github.com/k-ghiasi/RedBlackTrees/blob/main/ParitySeeking-234RedBlackBST.h>