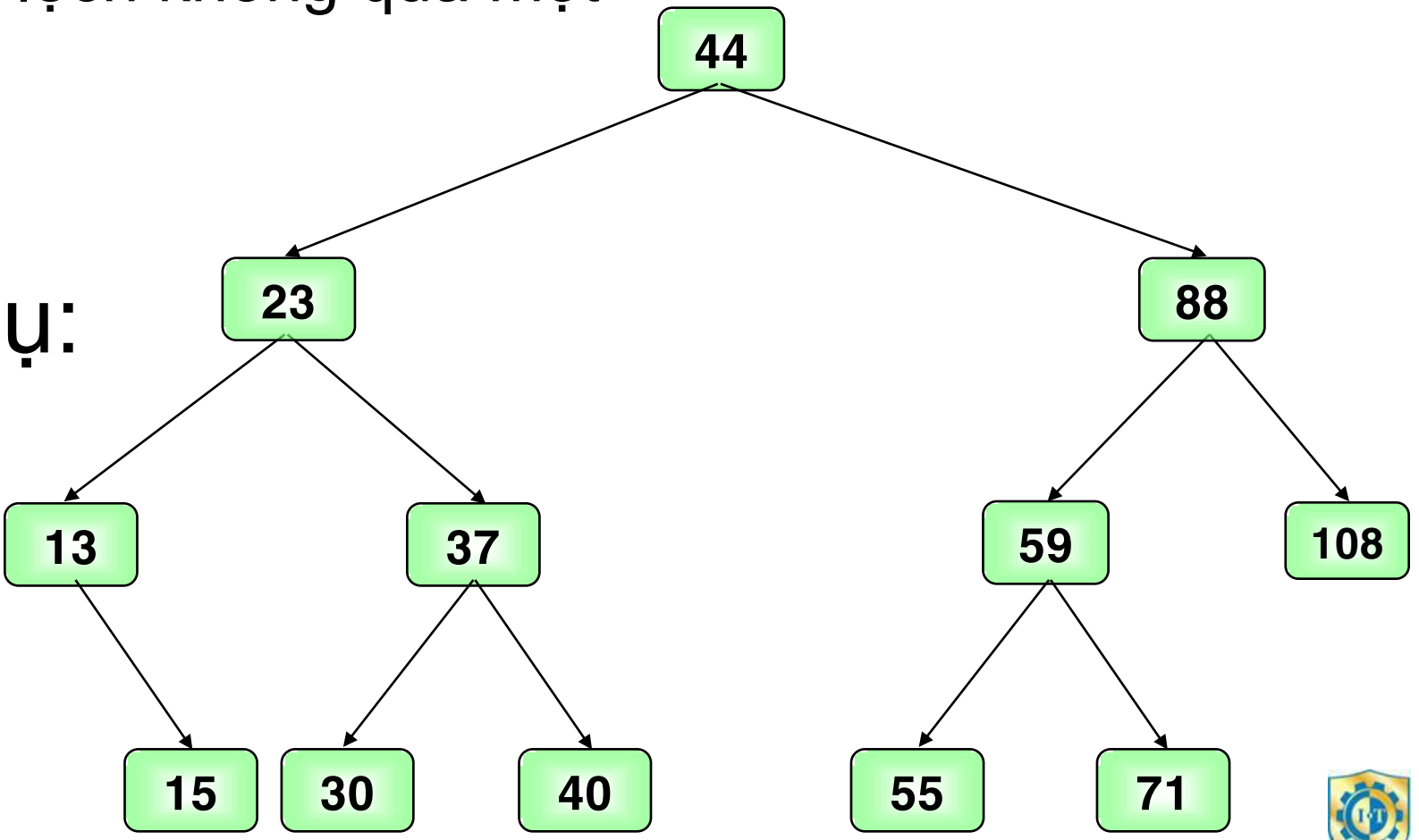
NỘI DUNG

CÂY NHỊ PHÂN TÌM KIẾM CÂN BẰNG

Định nghĩa

- Cây nhị phân tìm kiếm cân bằng là cây mà tại mỗi nút của nó độ cao của cây con trái và của cây con phải chênh lệch không quá một

Ví dụ:



Tổ chức dữ liệu

- **Chỉ số cân bằng = độ lệch giữa cây trái và cây phải của một nút**
- **Các giá trị hợp lệ :**
  - **CSCB(p) = 0 ⇔ Độ cao cây trái (p) = Độ cao cây phải (p)**
  - **CSCB(p) = 1 ⇔ Độ cao cây trái (p) < Độ cao cây phải (p)**
  - **CSCB(p) = -1 ⇔ Độ cao cây trái (p) > Độ cao cây phải (p)**

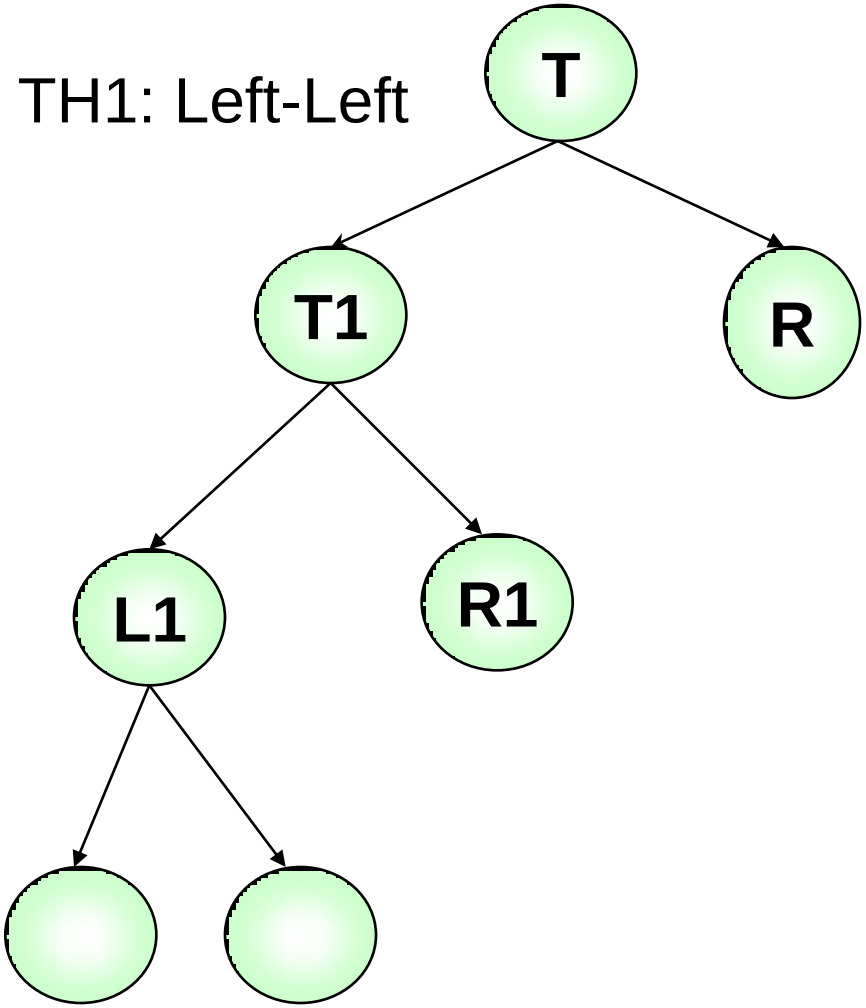
Tổ chức dữ liệu(tt)

```
#define LH -1 //cây con trái cao hơn
#define EH 0 //cây con trái bằng cây con phải
#define RH 1 //cây con phải cao hơn
typedef struct tagAVLNode
{  char    balFactor; //chỉ số cân bằng
   Data    key;
   struct tagAVLNode*       pLeft;
   struct tagAVLNode*       pRight;
}AVLNode;
typedef AVLNode    *AVLTree;
```

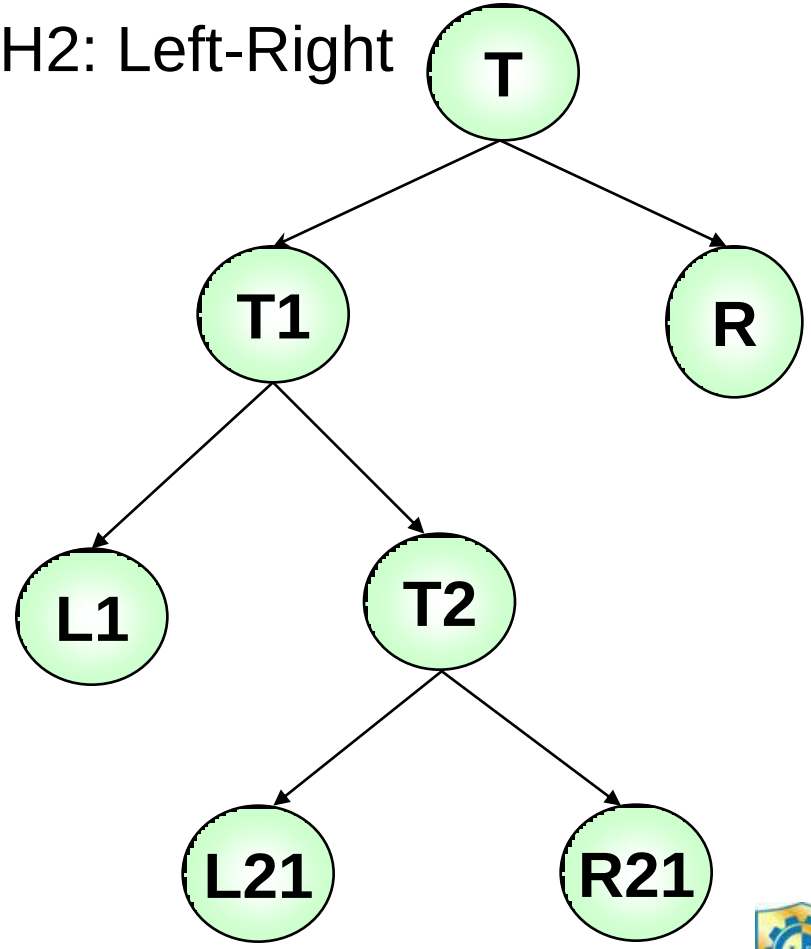
Các trường hợp mất cân bằng do lệch trái

- Cây mất cân bằng tại nút T

TH1: Left-Left



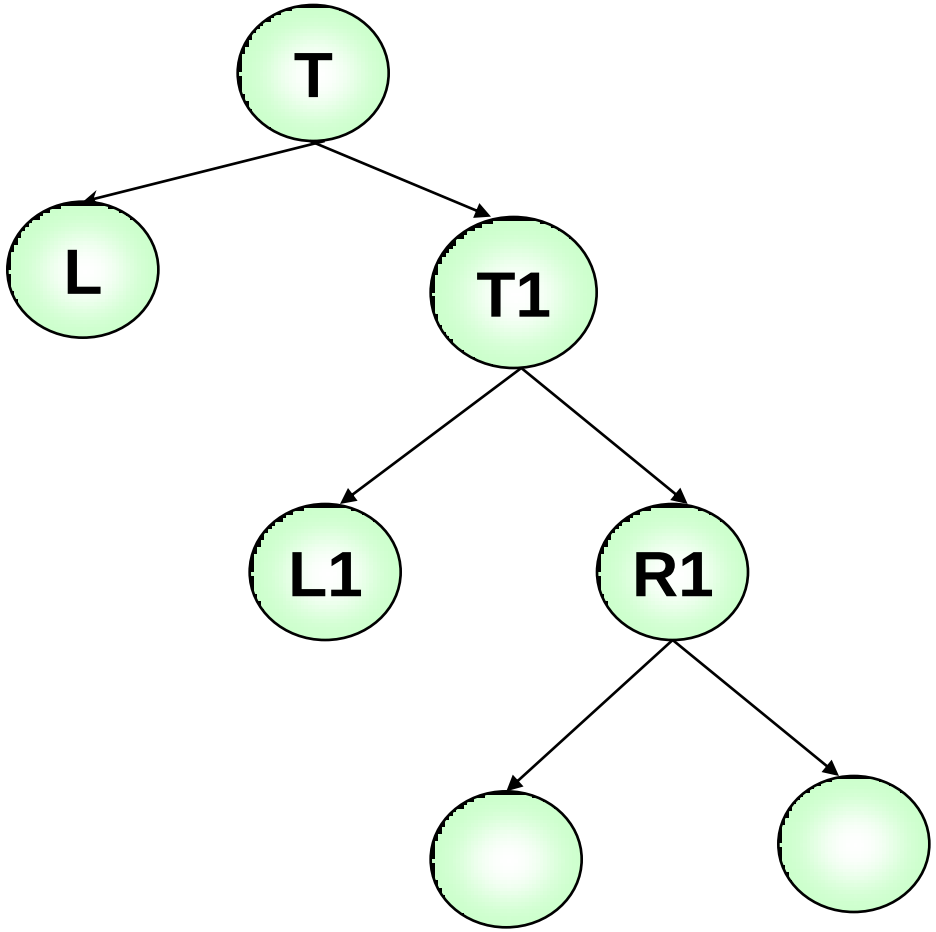
TH2: Left-Right



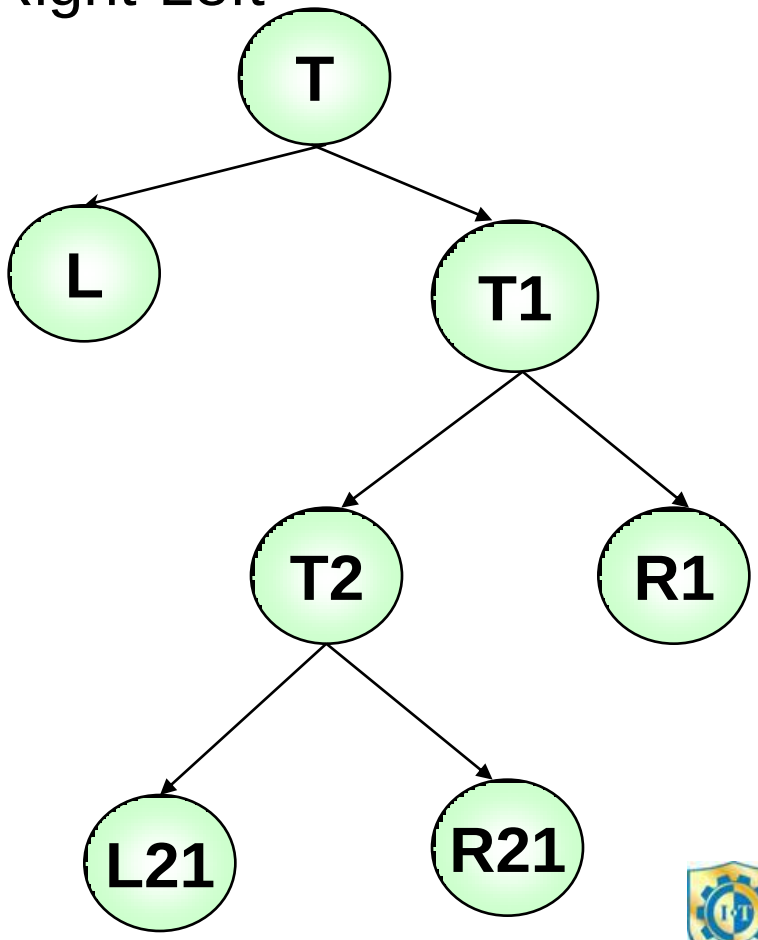
Các trường hợp mất cân bằng do lệch phải

- Cây mất cân bằng tại nút T

TH3: Right-Right



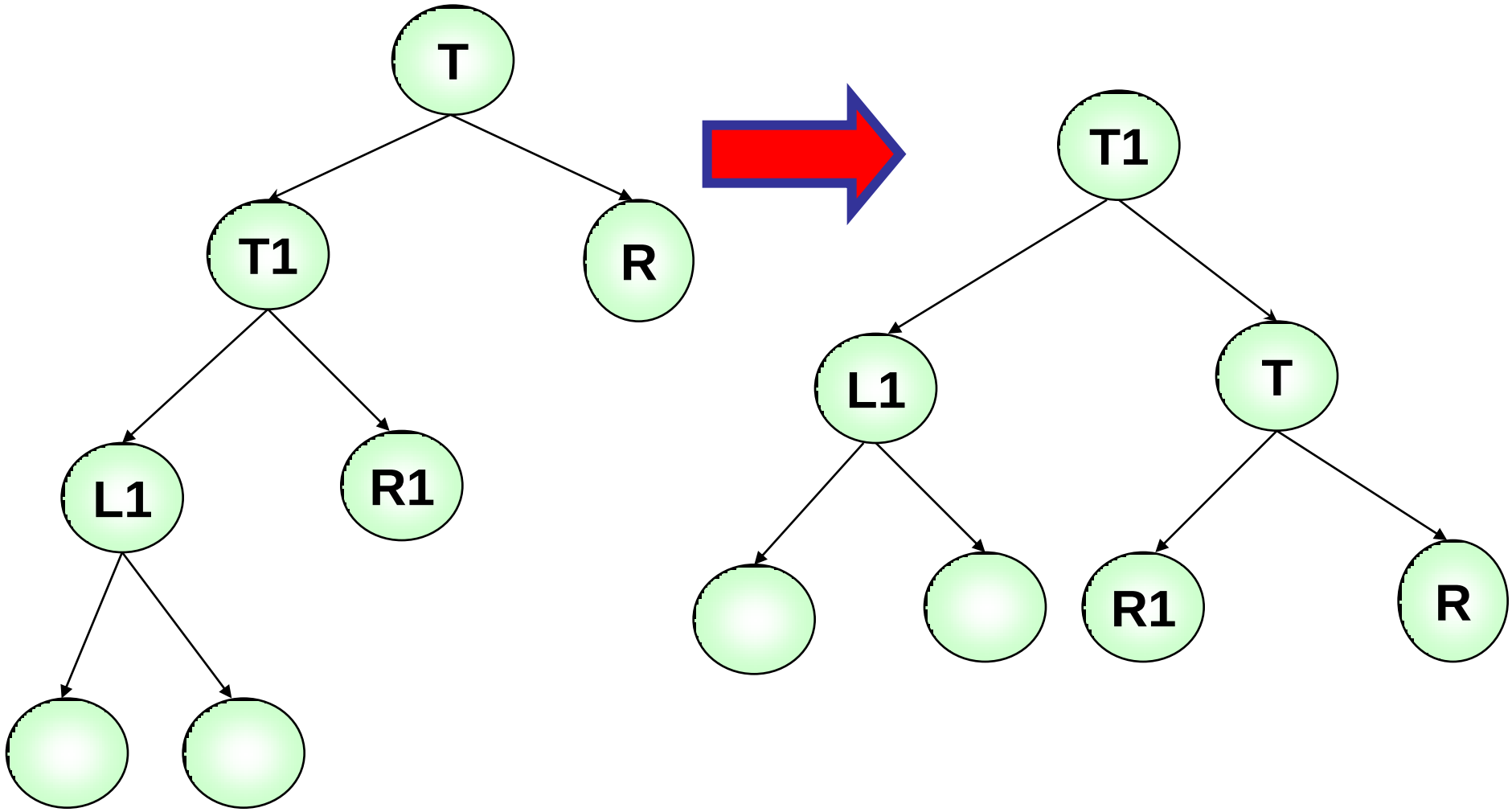
TH4: Right-Left



Các thao tác trên cây cân bằng

- Khi thêm hay xoá 1 nút trên cây, cĩ thể làm cho cây mất tính cân bằng, khi ấy ta phải tiến hành cân bằng lại.
- Cây có khả năng mất cân bằng khi thay đổi chiều cao:
  - Lệch nhánh trái, thêm bên trái
  - Lệch nhánh phải, thêm bên phải
  - Lệch nhánh trái, hủy bên phải
  - Lệch nhánh phải, hủy bên trái
- Cân bằng lại cây : tìm cách bố trí lại cây sao cho chiều cao 2 cây con cân đối:
  - Kéo nhánh cao bù cho nhánh thấp
  - Phải bảo đảm cây vẫn là Nhị phân tìm kiếm

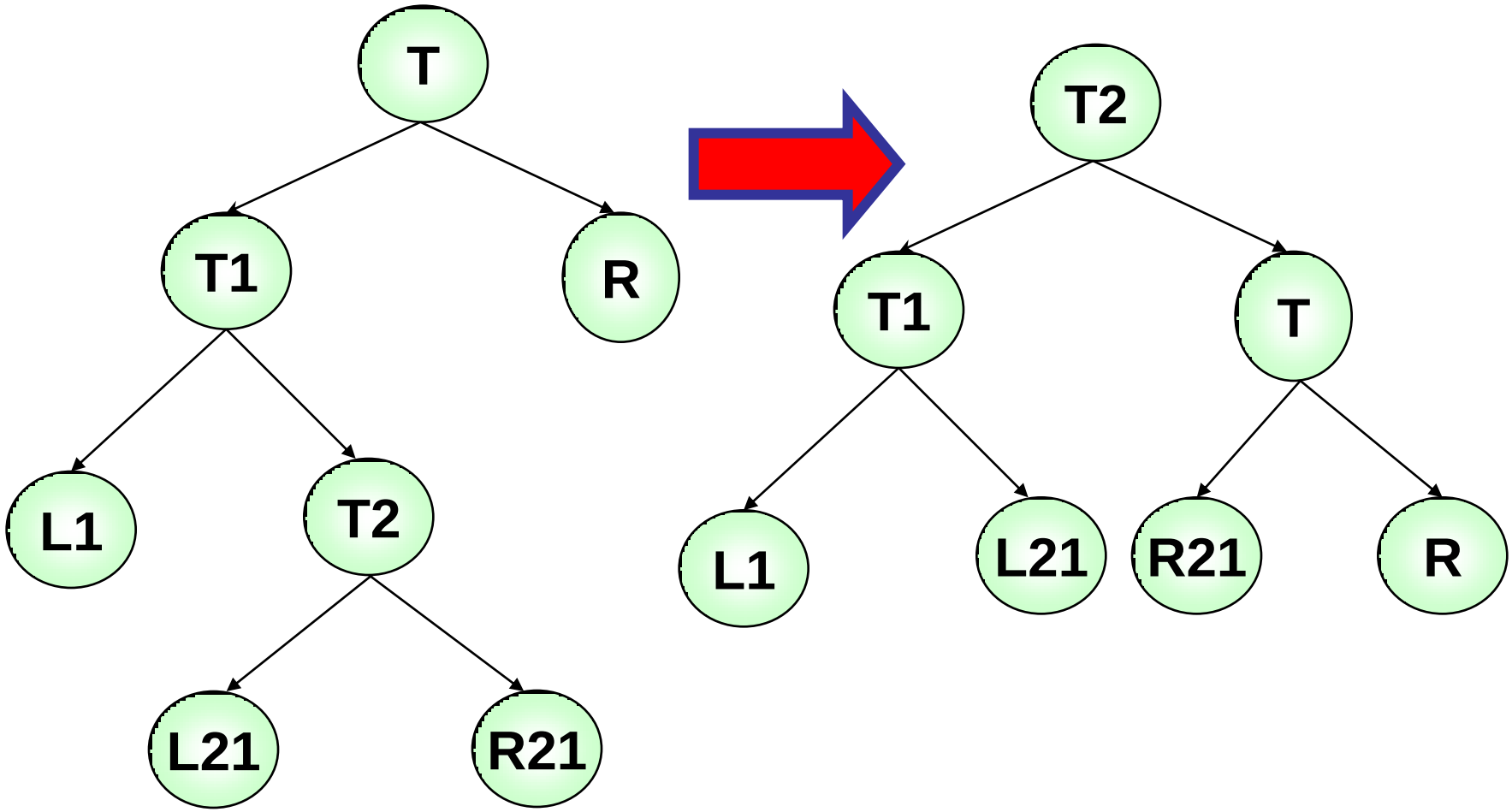
Cân bằng lại trường hợp 1



Cài đặt cân bằng lại cho trường hợp 1

```
void LL(AVLTree &T)
{
   AVLNode *T1=T->pLeft;
   T->pLeft = T1->pRight;
   T1->pRight=T;
   switch(T1-> balFactor)
   { case LH:    T-> balFactor =EH;
                 T1->balFactor=EH; break;
     case EH:    T->balFactor=LH;
                 T1->balFactor =RH; break;
   }
   T=T1;
}
```

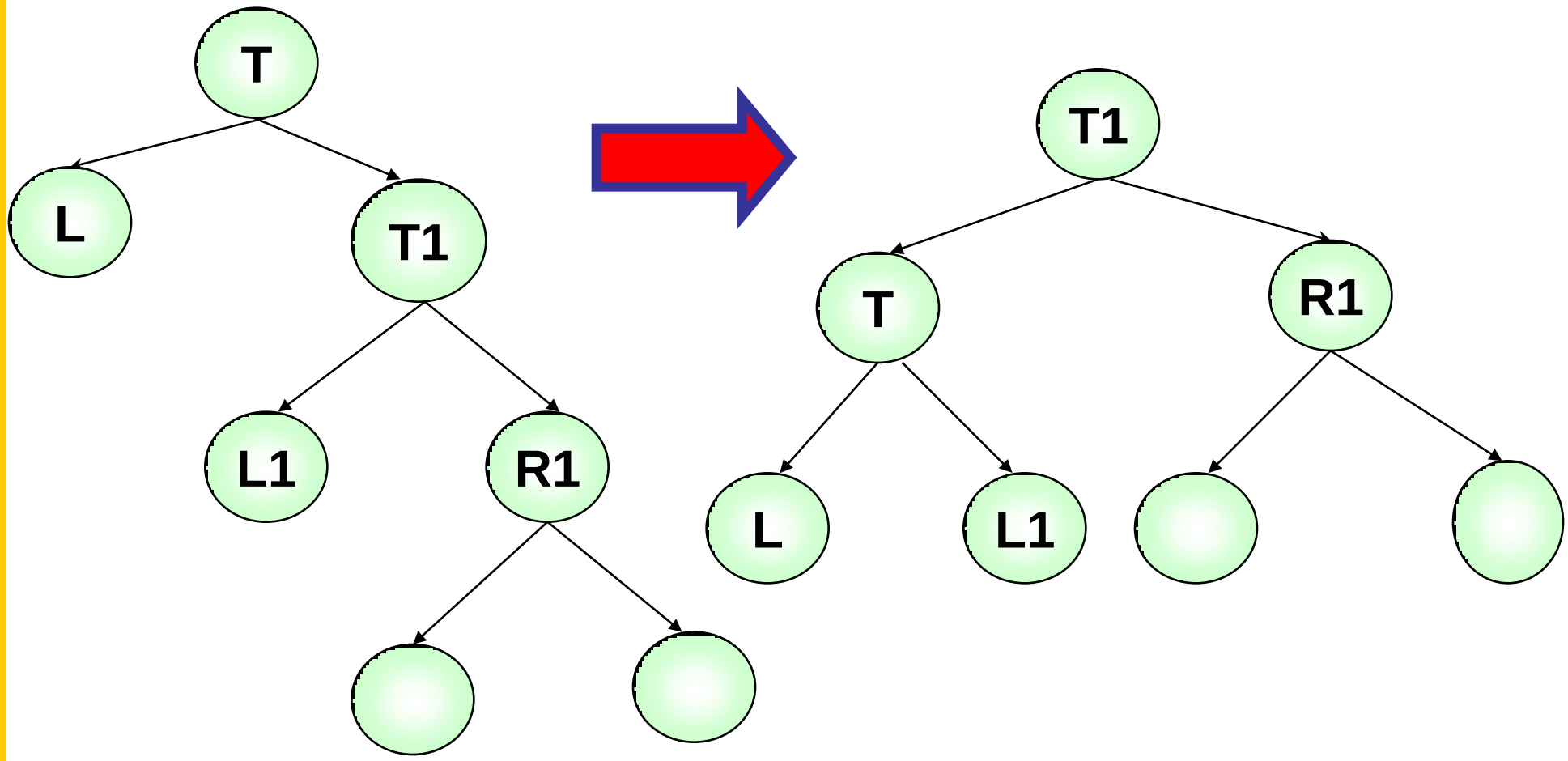
Cân bằng lại trường hợp 2



Cài đặt cân bằng lại cho trường hợp 2

```
void LR(AVLTree &T)
{   AVLNode *T1=T->pLeft;
    AVLNode *T2=T1->pRight;
    T->pLeft=T2->pRight;
    T2->pRight=T;
    T1->pRight= T2->pLeft;
    T2->pLeft = T1;
    switch(T2->balFactor)
    {     case LH:      T->balFactor=RH;
                        T1->balFactor=EH; break;
          case EH:      T->balFactor = EH;
                        T1->balFactor=EH; break;
          case RH:      T->balFactor =EH;
                        T1->balFactor= LH; break;
    }T2->balFactor =EH; T=T2}
```

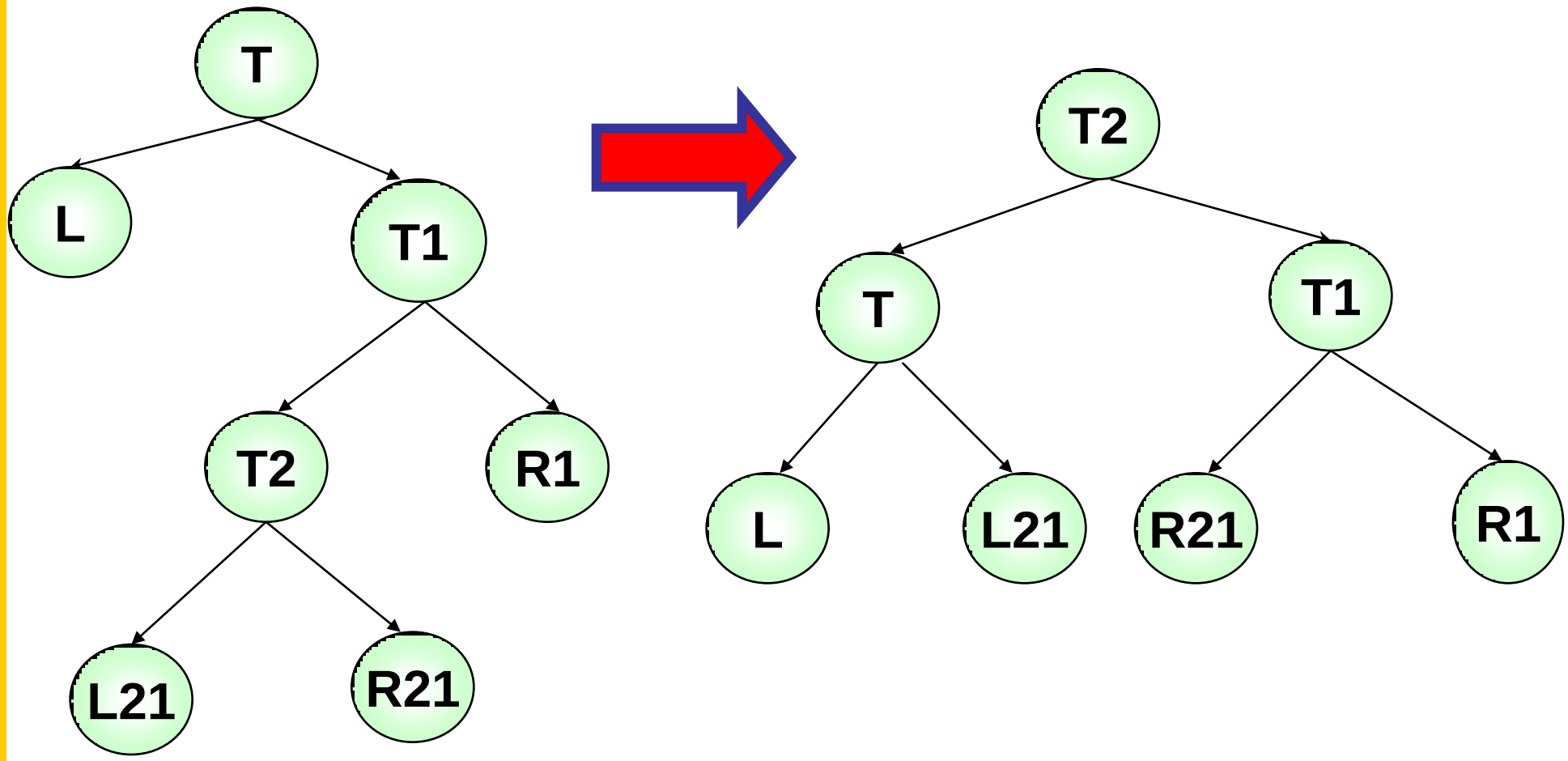
Cân bằng lại trường hợp 3



Cài đặt cân bằng lại cho trường hợp 3

```
void RR(AVLTree &T)
{  AVLNode  *T1= T->pRight;
   T->pRight=T1->pLeft;
   T1->pLeft=T;
   switch(T1-> balFactor)
   {
        case RH:    T-> balFactor = EH;
                    T-> balFactor = EH; break;
        case EH:    T-> balFactor = RH;
                    T1-> balFactor = LH; break;
   }
   T=T1
}
```

Cân bằng lại trường hợp 4



Cài đặt cân bằng lại cho trường hợp 4

```
void RR(AVLTree &T)
{   AVLNode  *T1= T->pRight;
    AVLNode *T2=T1->pLeft;
    T->pRight = T2->pLeft;
    T2->pLeft = T;
    T1->pLeft = T2->pRight;
    T2->pRight = T1;
    switch(T2-> balFactor)
    {      case RH:        T-> balFactor = LH;
                           T1-> balFactor = EH; break;
           case EH:        T-> balFactor = EH;
                           T1-> balFactor = EH; break;
           case LH:        T-> balFactor = EH;
                           T1-> balFactor = RH; break;
    }
    T2-> balFactor =EH; T=T2;}
```

Thêm 1 nút

- Thêm bình thường như trường hợp cây NPTK
- Nếu cây tăng trưởng chiều cao
  - Lần ngược về gốc để phát hiện nút bị mất cân bằng
  - Tiến hành cân bằng lại nút đó bằng thao tác cân bằng thích hợp
- Việc cân bằng lại chỉ cần thực hiện 1 lần nơi mất cân bằng

Hủy 1 nút

- Hủy bình thường như trường hợp cây NPTK
- Nếu cây giảm chiều cao:
  - Lần ngược về gốc để phát hiện nút bị mất cân bằng
  - Tiến hành cân bằng lại nút đó bằng thao tác cân bằng thích hợp
  - Tiếp tục lần ngược lên nút cha…
- Việc cân bằng lại co thể lan truyền lên tận gốc