Algoritmusok és adatszerkezetek II. 1. Előadás

AVL fa I.

fogalma, miértje, láncolt és szöveges ábrázolása, kulcs beszúrása, legkisebb kulcsú csúcs kivétele.

Tartalom

AVL fa fogalma
(Adelszon-Velszkij és Landisz, 1962)

- **Definíció:** Az AVL fák magasság szerint kiegyensúlyozott bináris keresőfák.
- Egy bináris *fa magasság szerint kiegyensúlyozott* (height-balanced BST)
  - ha minden csúcsa kiegyensúlyozott.
    - kiegyensúlyozott ~ magasság szerint kiegyensúlyozott

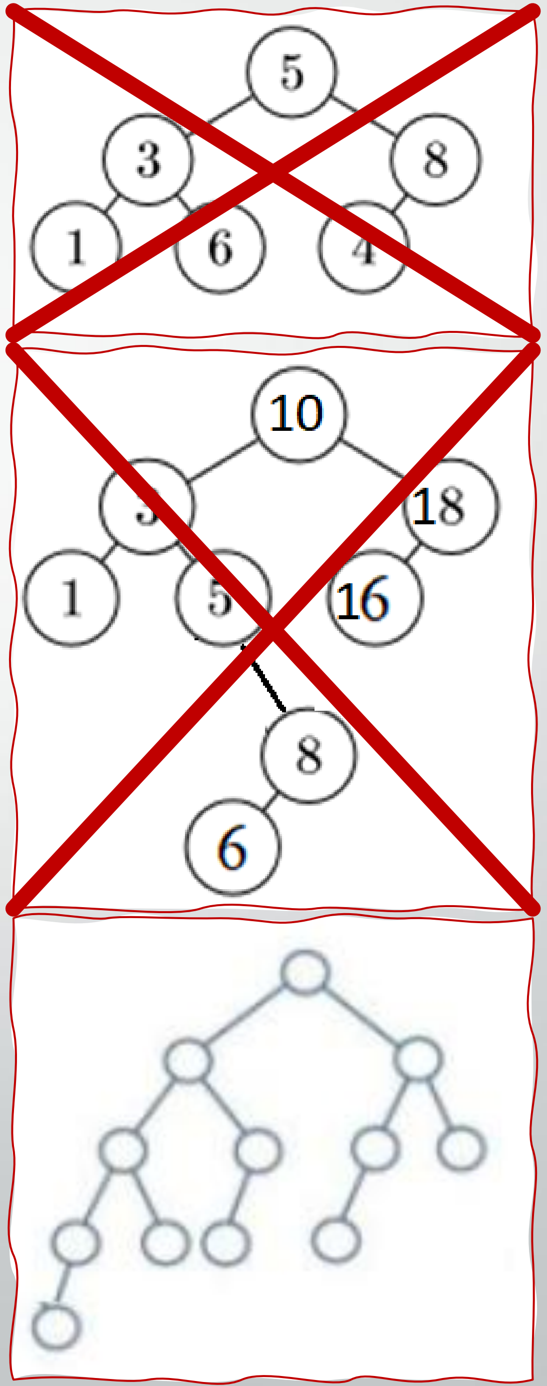
AVL fa fogalma

(Adelszon-Velszkij és Landisz, 1962)

- Egy bináris fa egy (*p) csúcsa *kiegyensúlyozott (*balanced node*):*
  - ha a csúcs (p -> b) egyensúlyára (balance) $|p \text{ -> } b| \leq 1$.

- **Definíció:** A (*p) csúcs egyensúlya (the (*p) node's balance):

$$p \text{ -> } b = h(p \text{ -> } right) - h(p \text{ -> } left)$$



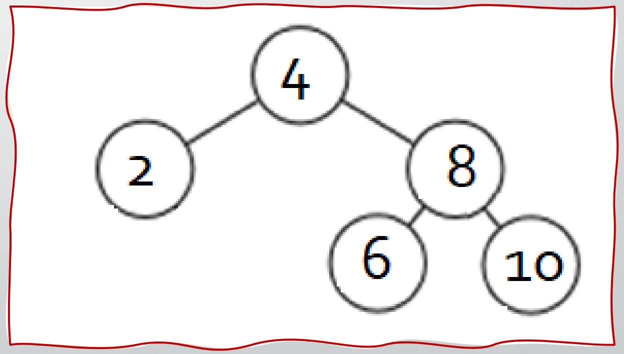
Az AVL fák láncolt reprezentálása

Node
$+ \; key : \mathcal{T}$ // $\mathcal{T}$ is some known type $+ \; b : -1..1$ // the balance of the node $+ \; left, right$: Node*
$+$ Node() $\{ \; left := right := \emptyset \; ; \; b := 0 \; \}$ // create a tree of a single node $+$ Node(x:$\mathcal{T}$) $\{ \; left := right := \emptyset \; ; \; b := 0 \; ; \; key := x \; \}$

- A csúcsokban a *b* egyensúly attribútumot expliciten tároljuk
- //Más lehetőségek:
  - a két részfa magasságainak tárolása
  - csak az aktuális részfa magasságának tárolása

Az AVL fák szöveges reprezentálása

- Keresőfák
  - (bal_részfa gyökér jobb_részfa) jelölés
  - az üres részfák elhagyása
  - a könnyebb olvashatóság kedvéért [] és {} zárójelek alkalmazása
  - ➢ Az így ábrázolt fák inorder bejárása a zárójelek elhagyásával adódik
- AVL fák
  - A belső csúcsok egyensúlyainak jelölése *kb* formában
    - *k:* a csúcsot azonosító kulcs
    - *b:* a csúcs egyensúlya: **0:**◦**, 1:+, 2:++, -1:, -2**
  - a leveleknél nem jelöljük az egyensúlyt
    - a levélcsúcsok egyensúlya mindig nulla



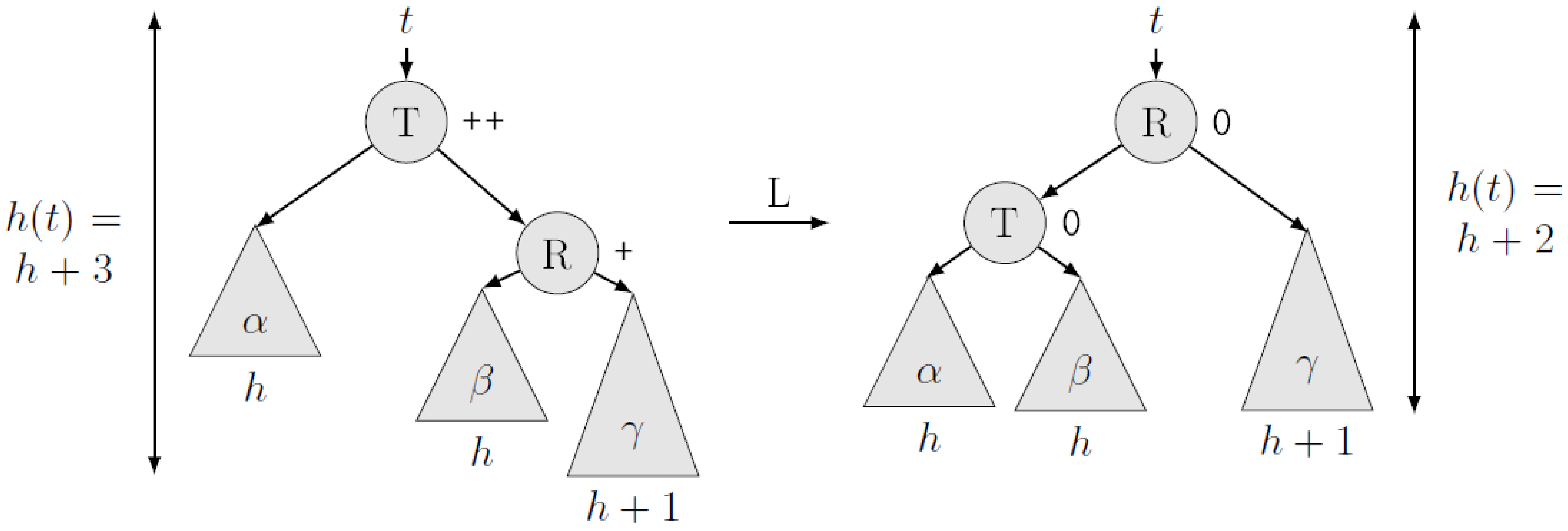
{ [2] 4 [(6) 8 (10)] }

{ [2] 4+ [(6) 8◦ (10)] }

AVL fa kiegyensúlyozó forgatások (balancing rotation)

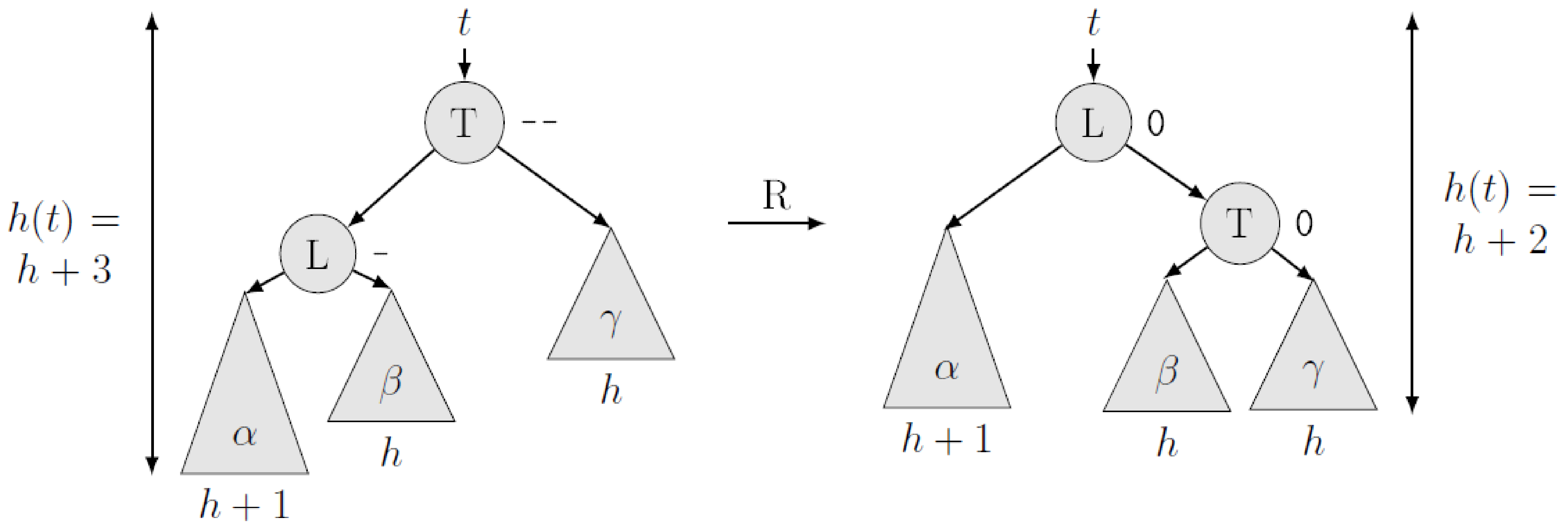
- Jelölések
  - görög kisbetűk: részfák (ezek mindig AVL fák)
  - latin nagybetűk: csúcsok kulcsait (egyensúlyok nélkül)
- Forgatások:
  - Balra forgatás (Left rotation): [α T (β R γ)] -> [(α T β) R γ]
  - Jobbra forgatás (Right rotation): [(α L β) T γ] -> [α L (β T γ)]
- A bináris keresőfa tulajdonságot a kiegyensúlyozások is megtartják
  - a fa inorder bejárását egyik forgatás sem változtatja -> a bináris keresőfa tulajdonságot is megtartják
  - a kiegyensúlyozatlan részfák kiegyensúlyozása minden esetben egy vagy két forgatásból áll

AVL fa forgatások: (++,+) forgatás



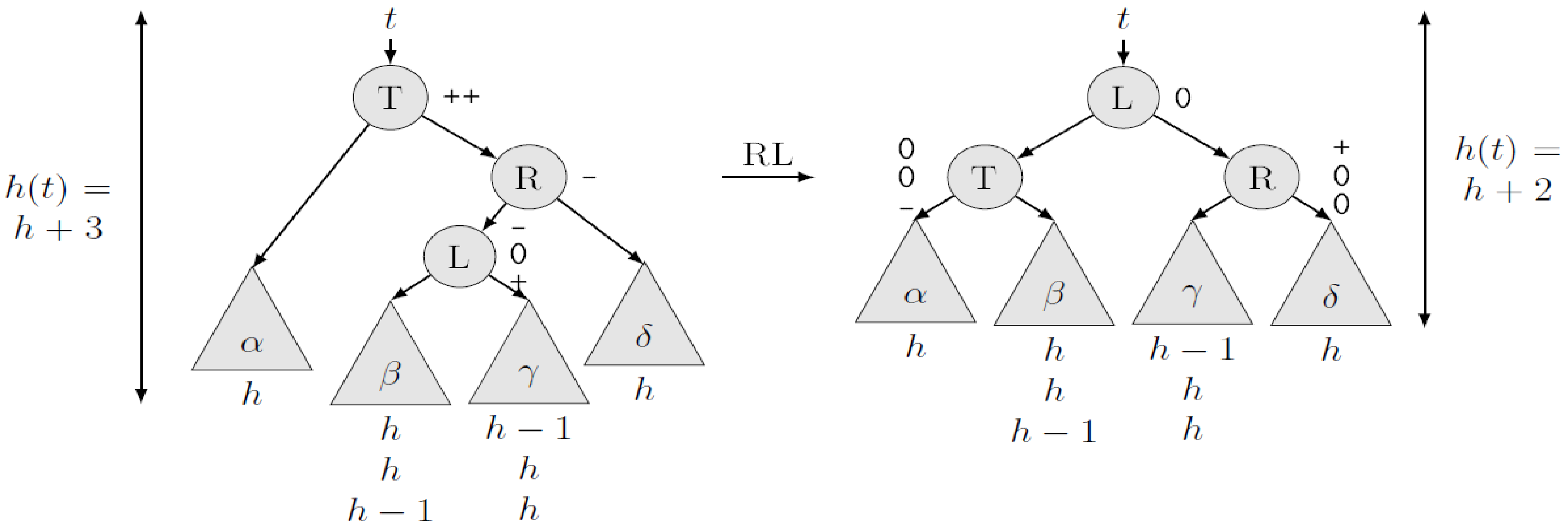
[α T (β R γ)] -> [(α T β) R γ]

AVL fa forgatások: (--,-) forgatás



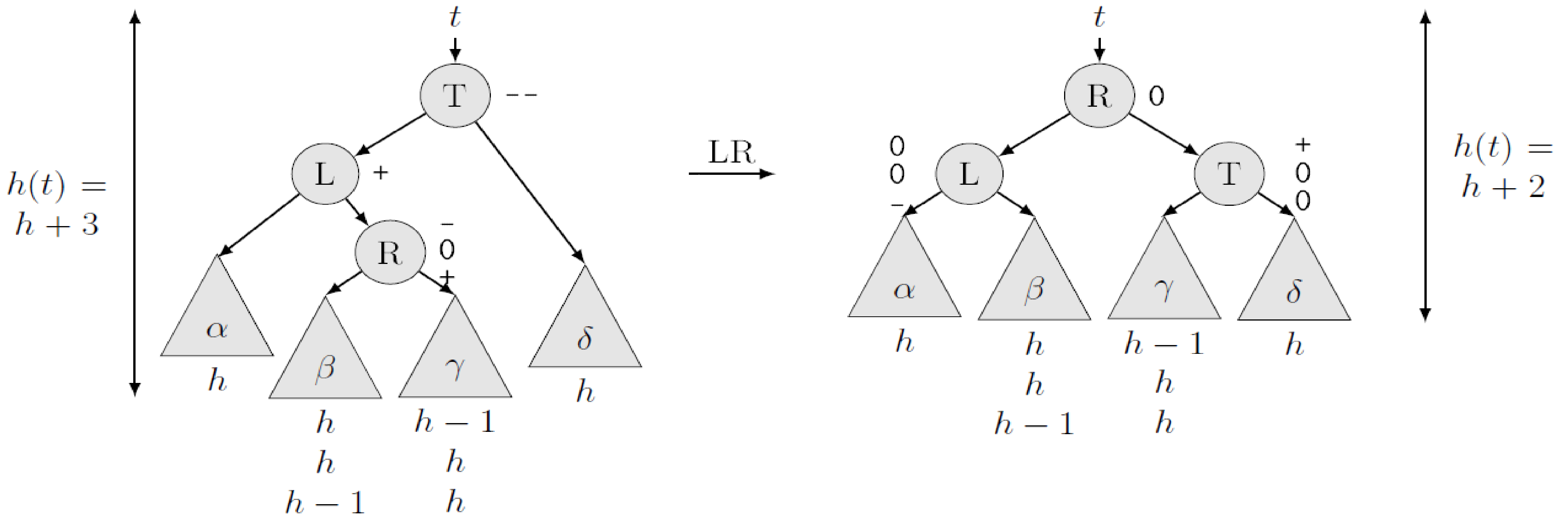
[(α L β) T γ] -> [α L (β T γ)]

AVL fa forgatások: (++,-) forgatás



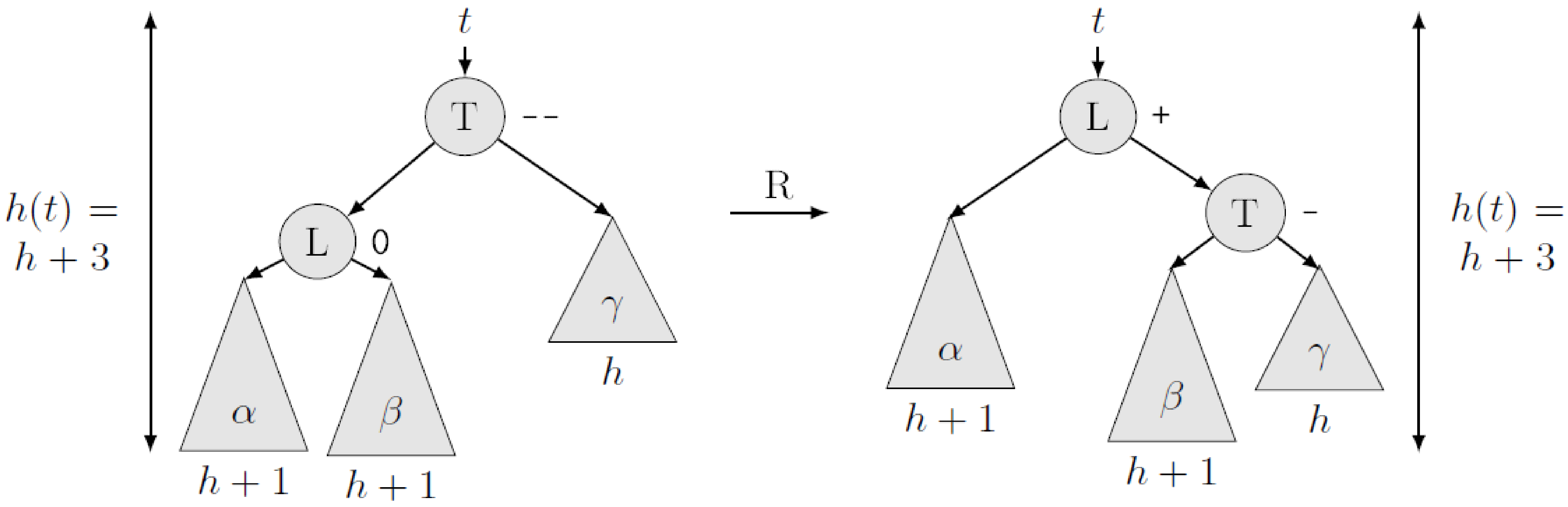
{ α T [(β L γ) R δ] } -> { [α T β] L [γ R δ] }

AVL fa forgatások: (--,+) forgatás



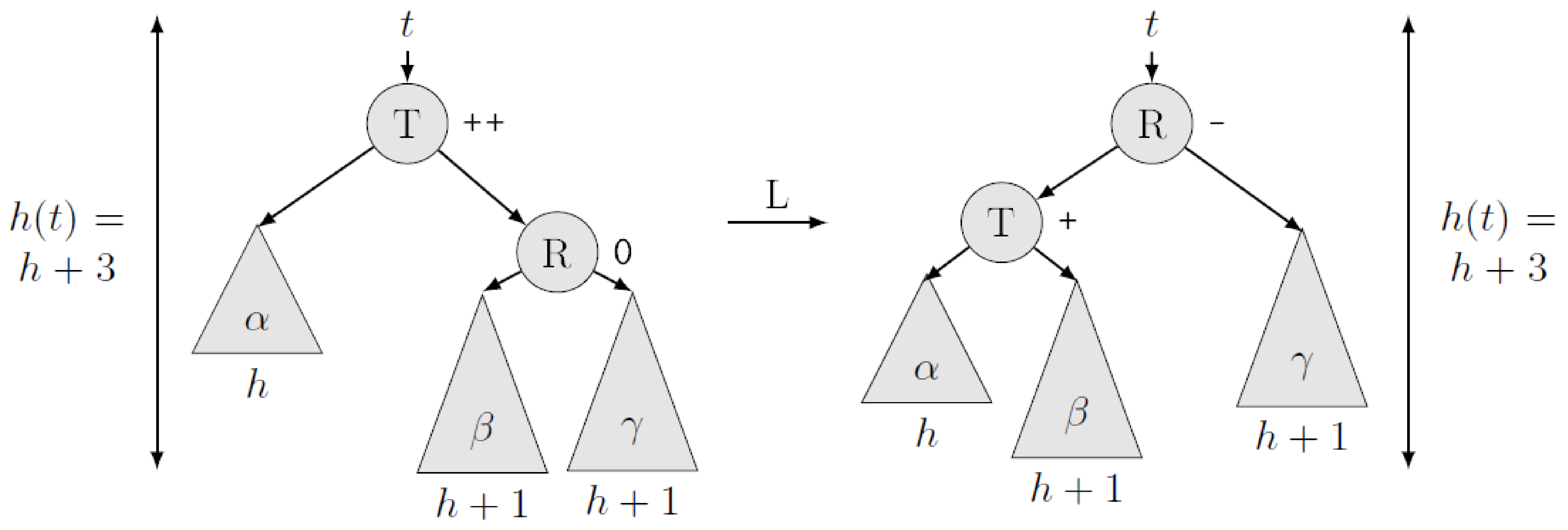
{ [α L (β R γ)] T δ } -> { [α L β] R [γ T δ] }

AVL fa forgatások: (--,o) forgatás



[(α L β) T γ] -> [α L (β T γ)]

AVL fa forgatások: (++,o) forgatás



[α T (β R γ)] -> [(α T β) R γ]

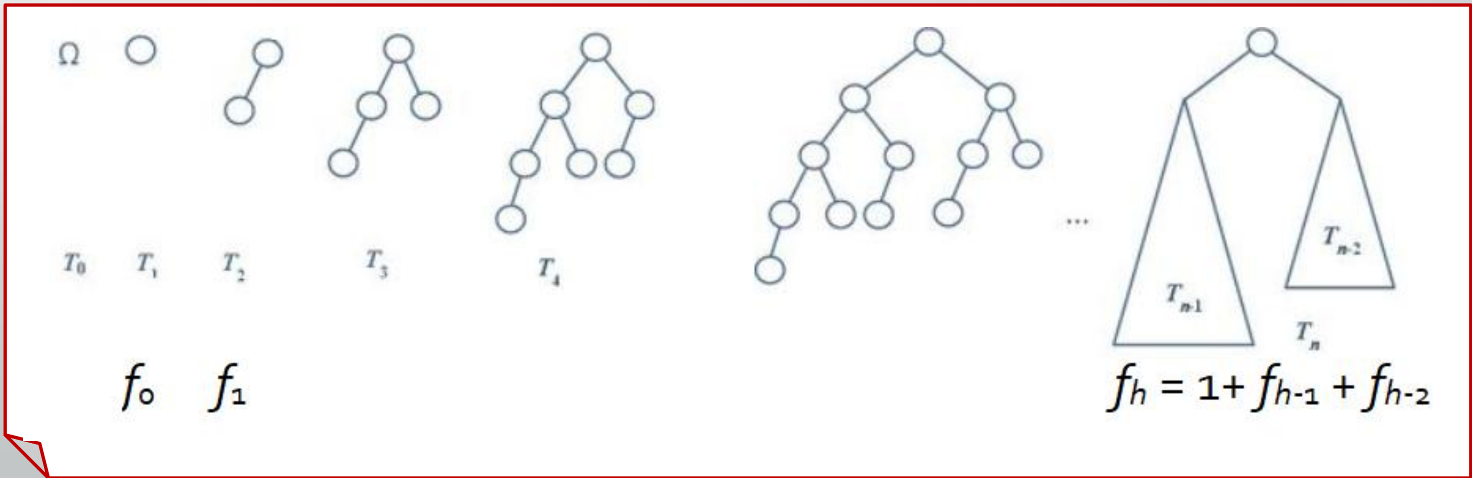
AVL fa magassága

- **Tétel:** Tetszőleges *n* csúcsú nemüres AVL fa *h* magasságára:

  $\lfloor \log n \rfloor \leq h \leq 1{,}45 \log n$, azaz $h \in \Theta(\log n)$

- **Bizonyítás** vázlata: 1. 2.

  1. alsó és felső becslés a *h* magasságú, nemüres KBF-ek *n* méretére
     - KBF: kiegyensúlyozott, bináris fák
     - $n < 2^{h+1} \rightarrow \lfloor \log n \rfloor \leq h$
  2. a *h* mélységű, legkisebb méretű KBF-ek csúcsainak f_h számának meghatározása:
     - $f_0 = 1$, $f_1 = 2$, $f_h = 1 + f_{h-1} + f_{h-2}$. (**Fibonacci fák**)
     - tetszőleges *h* magasságú KBF *n* méretére: $n \geq f_h$
     - ➢ $h \leq 1{,}45 \log n$

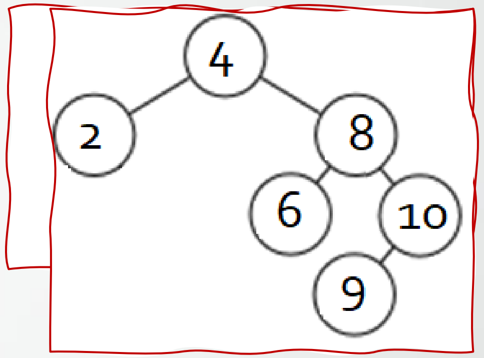


Műveletigény

- **search(*t*; *k*), min(*t*), max(*t*)**
  - $MT \in \Theta(\log n)$**, ahol *n* = |*t*|**
  - az AVL fák magassága: $\lfloor \log n \rfloor$
- **insert(*t*; *k*), del(*t*; *k*), remMin(*t*; *minp*), remMax(*t*; *maxp*)**
  - változtatják a fa alakját -> elromolhat a kiegyensúlyozottság
    - már nem garantált a fenti műveletigény
  - Elkerülésére:
    - Minden rekurzív eljáráshívás után ellenőrizés: hogyan változott a megfelelő részfa magassága
    - Ez hogyan befolyásolta a felette levő csúcs kiegyensúlyozottságát
    - Szükség esetén: helyreállítás
    - minden szinten **legfeljebb konstans** mennyiségű **extra műveletet** fog jelenteni
  - $MT \in \Theta(\log n)$**, ahol *n* = |*t*|**

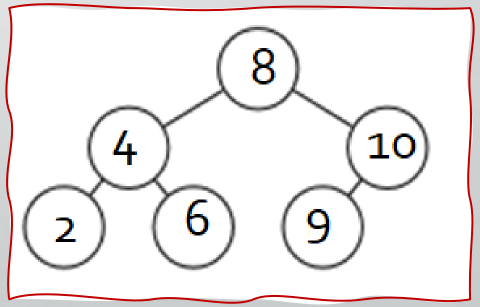
AVL fák: beszúrás (balra forgatás)

- A transzformáció helyességének belátása
  - $h = h(\alpha)$ jelölés
  - a kiinduló fa:
    - T++ és R+ egyensúlyok
    - ➢ $h((\beta \text{ R } \gamma)) = h+2$
    - ➢ $h(\gamma) = h+1$
    - ➢ $h(\beta) = h$
  - ➢ az eredmény fára:
    - $h(\alpha) = h = h(\beta)$ -> T∘
    - $h((\alpha \text{ T∘ } \beta)) = h + 1 = h(\gamma)$ -> R∘



{ [2] 4++ [(6) 8+ ({9} 10-)] }

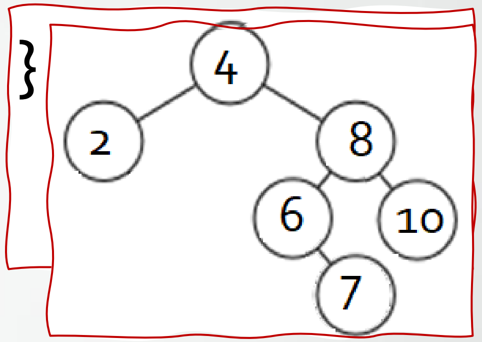
Balra forgatás (Left rotation):
[α T (β R γ)] -> [(α T β) R γ]



{ [(2) 4 (6)] 8 ∘ [(9) 10-] }

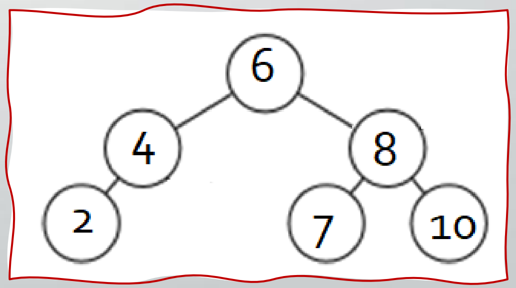
AVL fák: beszúrás (kettős forgatás)

- $\{ \alpha\, T{+}{+}\, [(\beta\, L{-}{\circ}{+}\, \gamma)\, R{-}\, \delta] \} \rightarrow \{ [\alpha\, T{\circ}{\circ}{-}\, \beta]\, L{\circ}\, [\gamma\, R{+}{\circ}{\circ}\, \delta] \}$
  - T=4, α = [2], R=8, δ = (10), L=6+, $\beta = \emptyset$ γ = {7}
- Kettős forgatás (8. 9. dia)
  - $\{ \alpha\, T\, [(\beta\, L\, \gamma)\, R\, \delta] \}$ fára az R csúcsnál alkalmaztunk egy jobbra forgatást
  - $\{ \alpha\, T\, [\beta\, L\, (\gamma\, R\, \delta)] \}$ eredményfát balra forgatjuk
  - Eredményfa: $\{ [\alpha\, T\, \beta]\, L\, [\gamma\ \ R \circ \delta] \}$



{ [2] 4++ [(6 + {7}) 8- (10)] }

Kettős (RL) forgatás:
$\{ \alpha\, T\, [(\beta\, L\, \gamma)\, R\, \delta] \} \rightarrow \{ [\alpha\, T\, \beta]\, L\, [\gamma\ \ R\ \ \delta] \}$



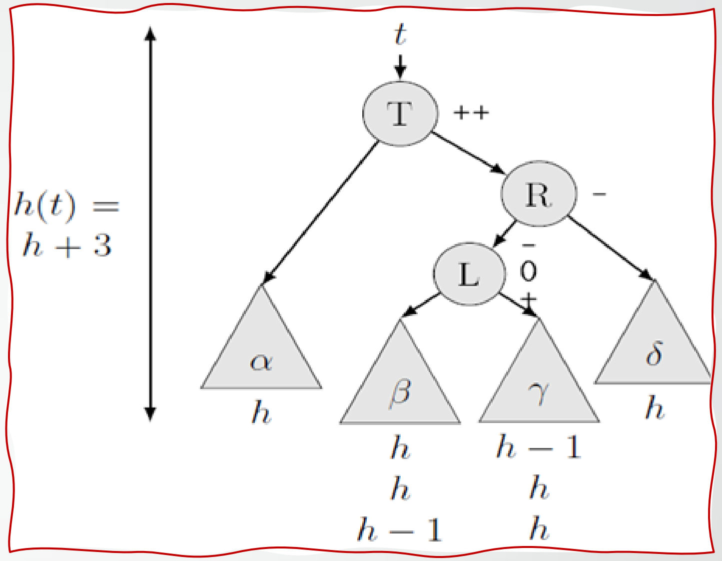
{ [(2) 4-] 6∘ [(7) 8∘ (10)] }

Kettős forgatás helyessége:

- $h = h(\alpha)$ jelölés
- T++ -> $h((\beta \text{ L } \gamma) \text{ R } \delta) = h+2$
- R- -> $h(\beta \text{ L } \gamma) = h+1$ *és* $h(\delta) = h$
- L lehetséges egyensúlyai szerint:
  - L- -> $h(\beta) = h$ és $h(\gamma) = h-1$
  - L∘ -> $h(\beta) = h$ és $h(\gamma) = h$
  - L+ -> $h(\beta) = h-1$ és $h(\gamma) = h$
- ➢ Eredményfában: T- és R∘
- Mindhárom esetben:

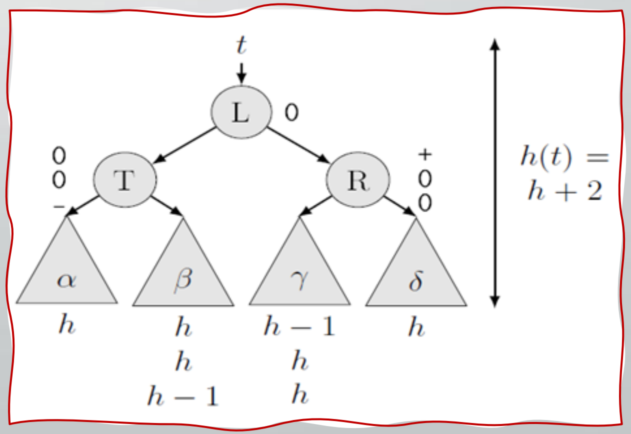
  $h([\alpha \text{ T } \beta]) = h+1 = h([\gamma \text{ R } \delta])$

  - ➢ L∘



Kettős (RL) forgatás:

$\{ \alpha \text{ T } [(\beta \text{ L } \gamma) \text{ R } \delta] \} \rightarrow \{ [\alpha \text{ T } \beta] \text{ L } [\gamma \text{ R } \delta] \}$



Beszúrás

- Összefüggések a kettős forgatás utáni egyensúlyokra
  - Jelölések:
    - az L csúcsnak a kettős forgatás előtti egyensúlya: *s*
    - a T és a R csúcsoknak a kettős forgatás utáni egyensúlyai: s_t, s_r
  - $s_t = -\left\lfloor \frac{s+1}{2} \right\rfloor$ és $s_r = \left\lfloor \frac{1-s}{2} \right\rfloor$
- A fent említett két kiegyensúlyozási séma mellett igazak a tükörképei:
  - a forgatások előtt a T-- csúccsal a gyökérben
  - $[(\alpha\ L-\beta)T--\gamma] \rightarrow [\alpha L\circ\ (\beta T\circ\gamma)\ ]$
  - $\{\ [\alpha\ L+(\beta\ R-\circ\ \ +\ \gamma)\ ]\ T--\delta \rightarrow [\ (\alpha\ L\circ\circ\beta)\ R\circ\ (\gamma\ T+\circ\circ\delta)\ ]$
  - $s_l = -\left\lfloor \frac{s+1}{2} \right\rfloor$ és $s_t = \left\lfloor \frac{1-s}{2} \right\rfloor$

Beszúrás

Kérdések:

- Az AVL fában az új csúcs beszúrása után **mely csúcsoknak és milyen sorrendben** számoljuk újra az egyensúlyát?
- Mikor ütemezzük be a kiegyensúlyozást?

Válaszok

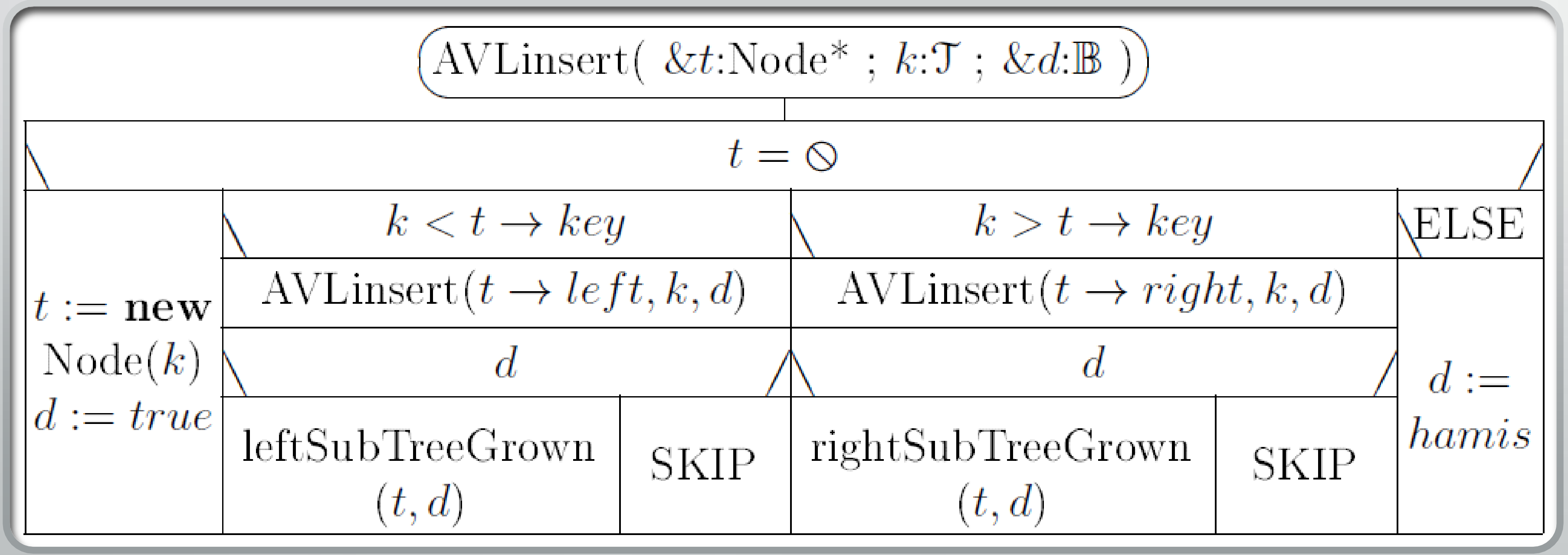
- Csak a beszúrás nyomvonalán visszafelé haladva kell újra számolnunk az egyensúlyokat
- Csak itt kell kiegyensúlyoznunk, ha kiegyensúlyozatlan csúcsot találunk

➢ Futási idő a fa magasságával arányos marad

- AVL fákra: $O(\log n)$

Beszúró és kiegyensúlyozó algoritmus működése

- A fa magassága:
  - eggyel növekszik (d = true)
  - ugyanannyi marad (d = false).



Beszúró és kiegyensúlyozó algoritmusok II.

leftSubTreeGrown(&t:Node* ; &d:$\mathbb{B}$)

$t \to b = -1$

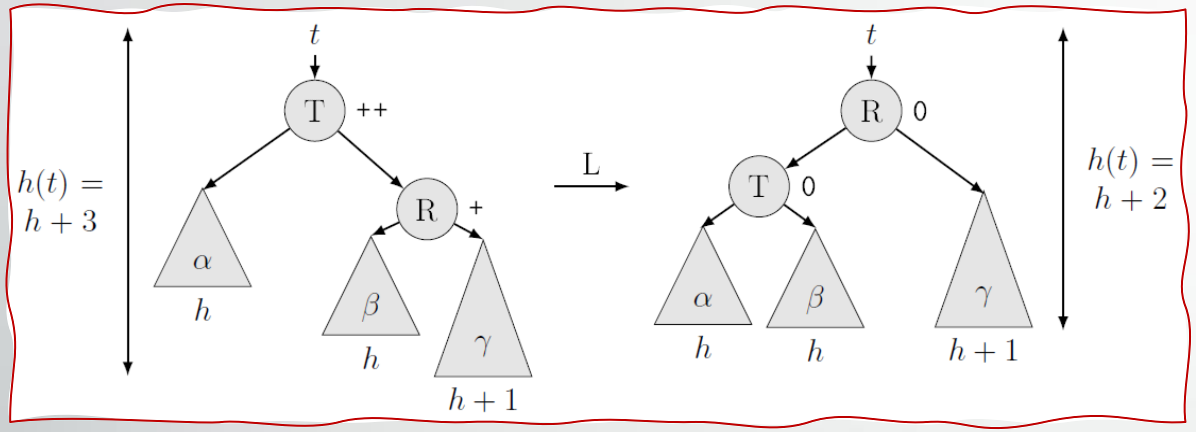
- igaz ág:
  - $l := t \to left$
  - $l \to b = -1$
    - igaz ág: **(R)** balanceMMm(t, l)
    - hamis ág: **(LR)** balanceMMp(t, l)
  - $d := false$
- hamis ág:
  - $t \to b := t \to b - 1$
  - $d := (t \to b < 0)$

rightSubTreeGrown(&t:Node* ; &d:$\mathbb{B}$)

$t \to b = 1$

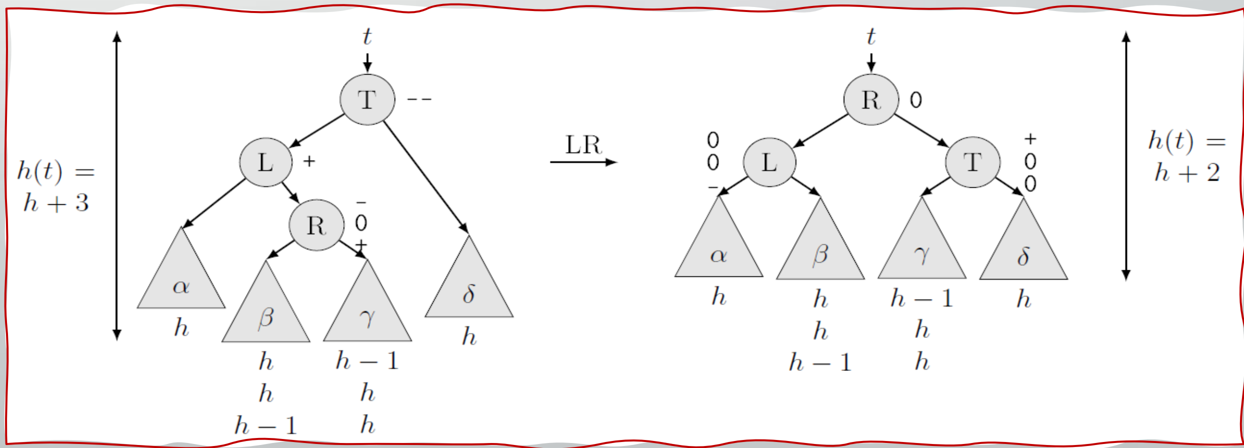
- igaz ág:
  - $r := t \to right$
  - $r \to b = 1$
    - igaz ág: **(L)** balancePPp(t, r)
    - hamis ág: **(RL)** balancePPm(t, r)
  - $d := false$
- hamis ág:
  - $t \to b := t \to b + 1$
  - $d := (t \to b > 0)$

Beszúró és kiegyensúlyozó algoritmusok III.



balancePPp($\&t, r$: Node*)
$t \to right := r \to left$
$r \to left := t$
$r \to b := t \to b := 0$
$t := r$

balanceMMp($\&t, l$: Node*)
$r := l \to right$
$l \to right := r \to left$
$t \to left := r \to right$
$r \to left := l$
$r \to right := t$
$l \to b := -\lfloor (r \to b + 1)/2 \rfloor$
$t \to b := \lfloor (1 - r \to b)/2 \rfloor$
$r \to b := 0$
$t := r$



Beszúró és kiegyensúlyozó algoritmusok IV. (tükörképek)

balanceMMm($\&t, l$: Node*)
$t \to left := l \to right$
$l \to right := t$
$l \to b := t \to b := 0$
$t := l$

~balancePPp (R forgatás)

balancePPm($\&t, r$: Node*)
$l := r \to left$
$t \to right := l \to left$
$r \to left := l \to right$
$l \to left := t$
$l \to right := r$
$t \to b := -\lfloor (l \to b + 1)/2 \rfloor$
$r \to b := \lfloor (1 - l \to b)/2 \rfloor$
$l \to b := 0$
$t := l$

~balanceMMp (RL forgatás)

Az AVL fába való beszúrás rövid összefoglalása

1. Megkeressük a kulcs helyét a fában.
2. Ha a kulcs benne van a fában
   - STOP.
3. Ha a kulcs helyén egy üres részfa található,
   - Beszúrunk az üres fa helyére egy új, a kulcsot tartalmazó levélcsúcsot.
   - Ez a részfa eggyel magasabb lett.

Az AVL fába való beszúrás rövid összefoglalása

4. Ha a gyökércsúcsnál vagyunk,
   - STOP.
   - Különben egyet fölfelé lépünk a keresőfában.
     - Mivel az a részfa, amiből fölfele léptünk, eggyel magasabb lett:
       - Az aktuális csúcs egyensúlyát megfelelőképp módosítjuk:
         - Ha a jobb részfa lett magasabb: +1, ha a bal: -1
5. Ha az aktuális csúcs egyensúlya 0 lett
   - az aktuális csúcshoz tartozó részfa alacsonyabb ága hozzá nőtt a magasabbikhoz
   - az aktuális részfa magassága = a beszúrás előtti állapottal
   - egyetlen más csúcs egyensúlyát sem kell módosítani: STOP.

Az AVL fába való beszúrás rövid összefoglalása

6. Ha az aktuális csúcs új egyensúlya 1 vagy -1
    - előtte 0 volt -> az aktuális részfa magasabb lett eggyel. -> a 4. ponttól folytatjuk.
7. Ha az aktuális csúcs új egyensúlya 2 vagy -2
    - a hozzá tartozó részfát ki kell egyensúlyozni.
        - A kiegyensúlyozás után az aktuális részfa visszanyeri a beszúrás előtti magasságát
        - már egyetlen más csúcs egyensúlyát sem kell módosítani: STOP.

- A 2 és -2 eseteket a struktogramban nem számoltuk ki expliciten, hogy az egyensúly tárolására elég legyen két bit.

Az algoritmusban a kiegyensúlyozás után az aktuális részfa visszanyeri a beszúrás előtti magasságát

- **Állítás:** Az algoritmusban a kiegyensúlyozás után az aktuális részfa visszanyeri a beszúrás előtti magasságát

- **Bizonyítás:** A kiegyensúlyozandó részfa gyökere T++ eset: (~ T -- eset)
  - A T++ esethez tartozó kiegyensúlyozási sémák:
    - [α T++ (β R+ γ)] -> [(α T ∘ β) R ∘ γ]
    - { α T++ [(β L-∘+ γ) R- δ] } -> { [α T∘∘- β] L∘ [γ R+∘∘ δ] }

Bizonyítás folytatása

1. Ha a T csúcs jobboldali R gyereke + vagy - súlyú -> a fenti sémák közül a megfelelő alkalmazható:
   - a beszúró algoritmus a fában a beszúrás helyétől egyesével fölfele lépked
   - az első kiegyensúlyozatlan csúcsnál azonnal kiegyensúlyoz
   - ez alatt nincs kiegyensúlyozatlan csúcs (azaz az $\alpha\beta\gamma\delta$ részfák is kiegyensúlyozottak)
   - ez a fenti forgatások feltétele,
   - Keresőfa marad: a kiegyensúlyozás nélküli beszúró algoritmus garantál
2. A fenti sémák minden esetet lefednek, azaz R+ vagy R -:
   - R nem lehet a beszúrás által létrehozott új csúcs
     - mert különben T-nek a beszúrás előtti jobboldali részfája üres lett volna -> most nem lehetne T++
   - Ha a fölfele lépkedés során nulla egyensúlyú csúcs áll elő, akkor a fölötte levő csúcsok egyensúlya már nem módosul, így kiegyensúlyozatlan csúcs sem állhat elő. Márpedig most T++. Így tehát az új csúcstól T-ig fölfelé vezető úton minden csúcs, azaz R egyensúlya is + vagy -.

Bizonyítás folytatása

3. A kiegyensúlyozások visszaállítják a részfa beszúrás előtti magasságát.
   - A T++ kiegyensúlyozatlan csúcs a beszúrás előtt kiegyensúlyozott volt
   - A beszúrás, a beszúrás helyétől kezdve T-ig fölfelé vezető úton mindegyik részfa magasságát pontosan eggyel növelte ->, a beszúrás előtt T+ volt
   - ➢ A beszúrás előtt, $h = h(\alpha)$ jelöléssel, a T gyökerű részfa h+2 magas volt
   - ➢ A beszúrás után T++ lett -> a T gyökerű részfa h+3 magas lett
   - A beszúrás utáni, de még a kiegyensúlyozás előtti állapotot tekintve a T jobboldali gyerekére megkülönböztetjük:
     - R+ eset
     - R- eset

Bizonyítás folytatása

- R+ eset:
  - $h(\alpha) = h = h(\beta)$ és $h(\gamma) = h + 1$
  - ➢ A kiegyensúlyozás után: $h([(\alpha \text{ T } \beta) \text{ R } \gamma\ ]) = h + 2$
- R- eset:
  - $h(\alpha) = h = h(\delta)$ és $h(\beta \text{ L } \gamma) = h + 1$
  - ➢ $h(\beta), h(\gamma) \leq h$
  - ➢ A kiegyensúlyozás után: $h(\{[\alpha \text{ T } \beta] \text{ L } [\gamma \text{ R } \delta]\}) = h + 2$

➢ Ezzel beláttuk, hogy a kiegyensúlyozások mindkét esetben visszaállítják a részfa beszúrás előtti magasságát

- úgy, hogy a beszúrás által eggyel megnövelt magasságot eggyel csökkentik.

- Szimmetria okokból ez hasonlóan látható be T-- esetén is
  - figyelembe véve a L- és a L+ eseteket.

Ellenőrző kérdések

- Mit jelent az, hogy egy bináris fa magasság szerint kiegyensúlyozott?
- Hogyan számoljuk ki egy csúcs egyensúlyát?
- Hogyan változik a fa magassága beszúrás után?
- Miért csak a beszúrási útvonalon kell ellenőrizni a kiegyensúlyozottságot?
- Milyen sorrendben számoljuk újra az egyensúlyokat beszúrás után?
- Rajzoljuk le a következő AVL fát a belső csúcsok egyensúlyaival együtt!
  - { [() 1 (2)] 4 [(5) 6 (⟨7⟩ 8 ⟨⟩)] }
  - Szemléltessük a 3 beszúrását és a 4 törlését,
    - mindkét esetben az eredeti fára!
    - Jelöljük, ha ki kell egyensúlyozni, a kiegyensúlyozás helyét, és a kiegyensúlyozás után is rajzoljuk újra fát! A rajzokon jelöljük a belső csúcsok egyensúlyait is, a szokásos módon!
  - Rajzoljuk le a hat általános kiegyensúlyozási séma közül azokat, amiket alkalmaztunk!

Köszönöm a figyelmet!

Pusztai Kinga

A bemutató Ásványi Tibor: Algoritmusok és adatszerkezetek II. Előadásjegyzete (Fák) és Fekete István: Algoritmusok és adatszerkezetek / AVL-Fák előadásjegyzete alapján készült.