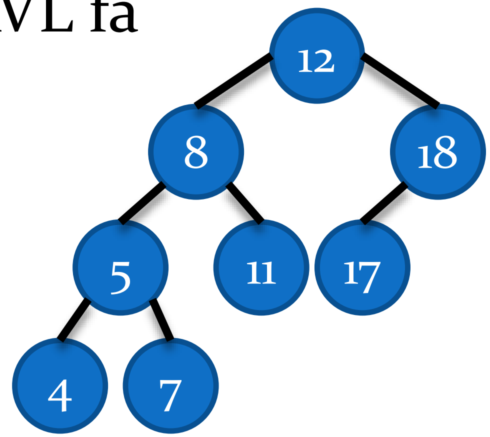
AVL fák

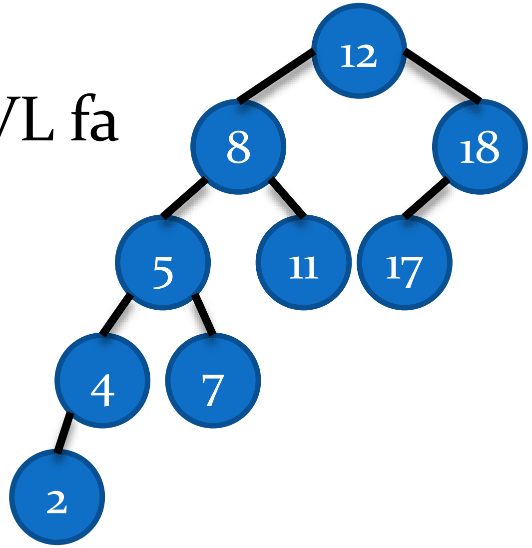
6. előadás

AVL fák

- Az első kiegyensúlyozott fa algoritmus
  - Kitalálói: Adelson-Velskii és Landis (1962)
- Tulajdonságok
  - Bináris rendezőfa
  - A bal és jobb részfák magassága legfeljebb 1-gyel különbözik egymástól
  - A részfák is AVL fák
  - AVL fa



Nem AVL fa



AVL fák

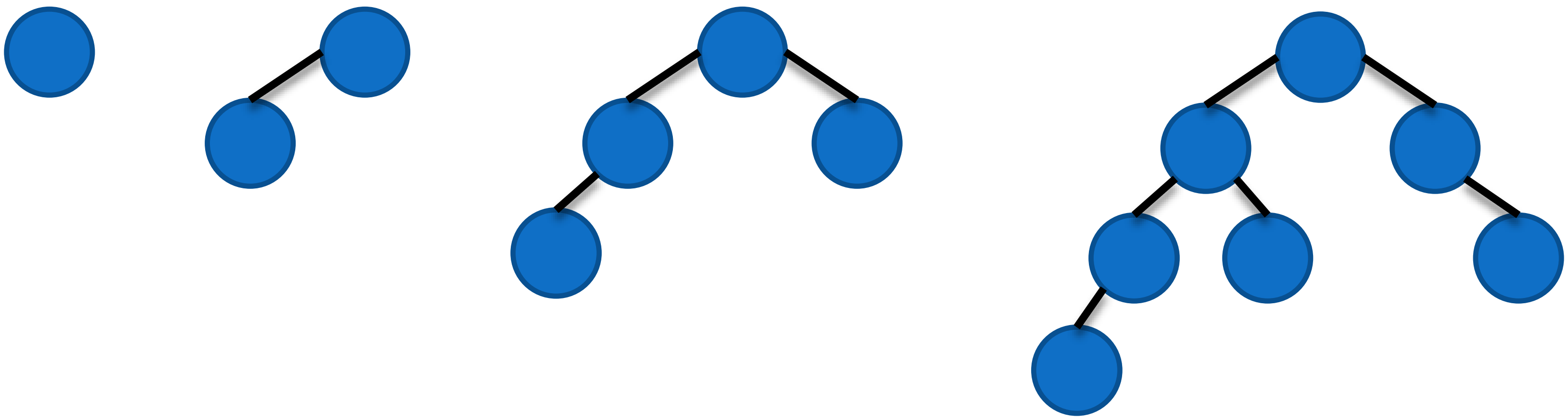
- Jelölje m(f) az f bináris fa magasságát (szintjeinek számát), ha x az f fa egy csúcsa: ekkor m(x) jelöli az x-gyökerű részfa magasságát
- Definíció (AVL-tulajdonság)
  - Egy bináris keresőfa AVL-fa, ha minden x csúcsára teljesül, hogy $|\mathrm{m}(\mathrm{bal}[x]) - \mathrm{m}(\mathrm{jobb}[x])| \leq 1$

AVL fák

- Mekkora a k-szintű AVL-fa minimális csúcsszáma?

$k = 1$ $\quad$ $k = 2$ $\quad$ $k = 3$ $\quad$ $k = 4$

$S_1 = 1$ $\quad$ $S_2 = 2$ $\quad$ $S_3 = 4$ $\quad$ $S_4 = 7$

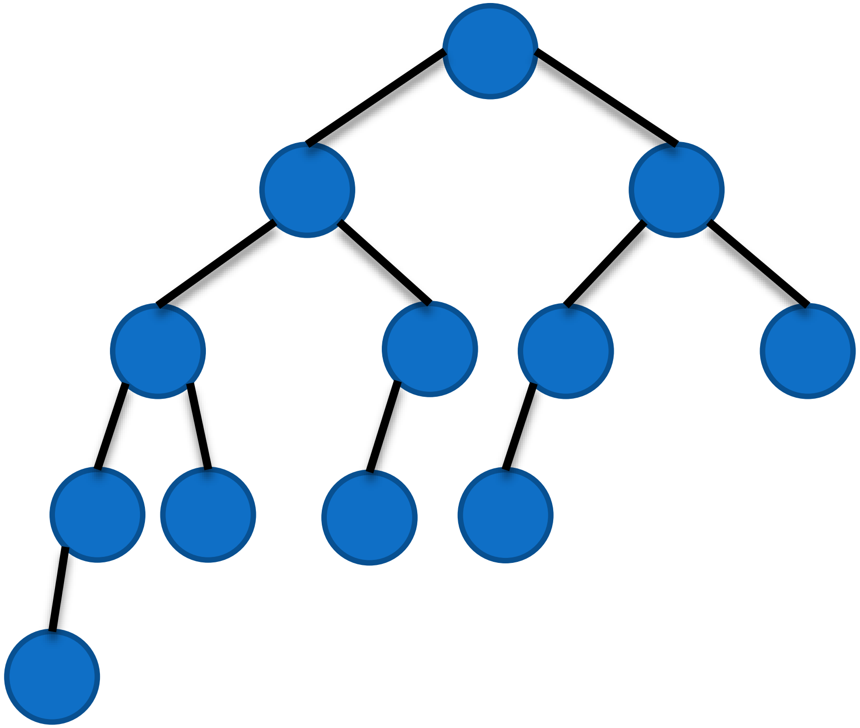


AVL fák

- Mekkora a k-szintű AVL-fa minimális csúcsszáma?

$$k = 5$$

$$S_5 = 12$$



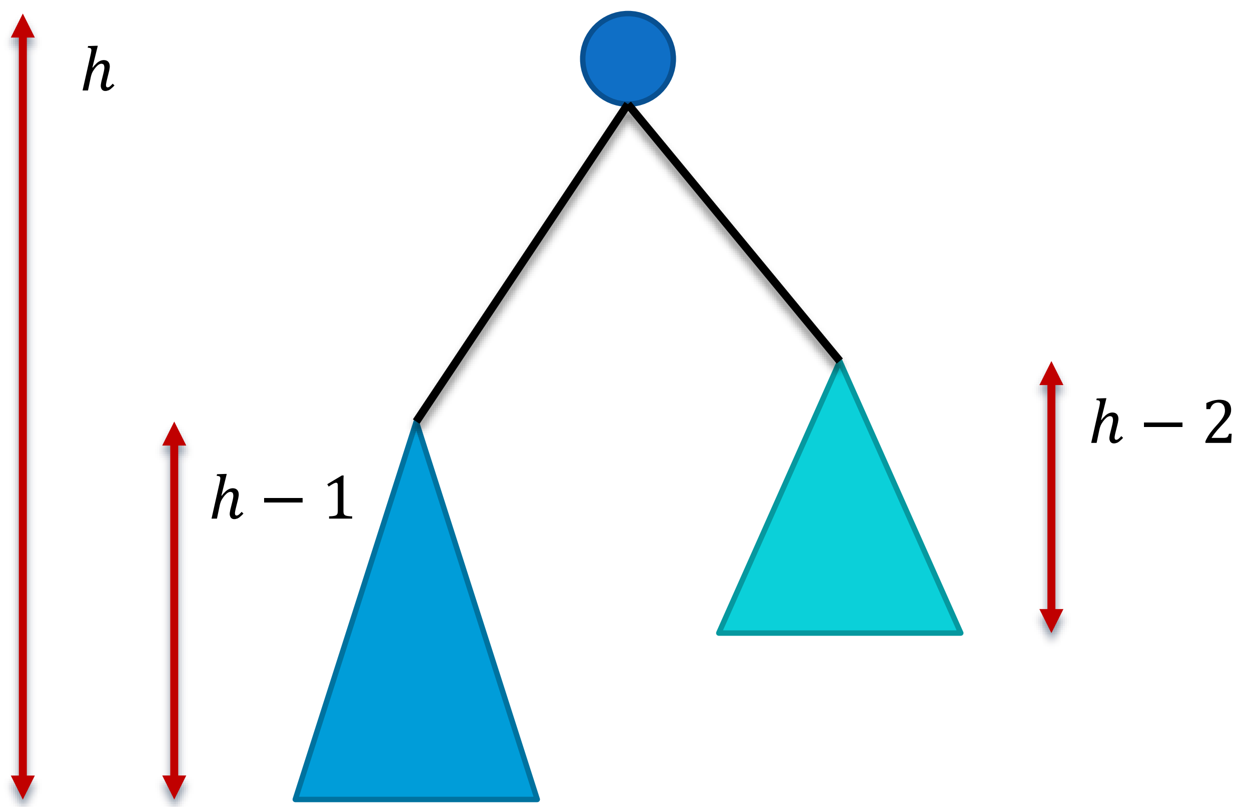
AVL fák

- Összefüggés az AVL-fa pontszáma és magassága között:
  - n adattal felépíthető fa minimális magassága?
    - Ez egy majdnem teljes bináris fa
  - n adattal felépíthető fa maximális magassága?
    - Ugyanez a kérdés: az adott h szintszámú AVL-fák közül mennyi a minimális pontszám?
  - Válasz
    - A h szintszámú minimális csúcsszámú AVL-fa gyökerének egyik részfája $h-1$, a másik $h-2$ szintű
    - Az eredeti fa minimalitása miatt pedig mindkét részfa minimális csúcsszámú

AVL fa maximális magassága

Rekurzió:

$$S_h = 1 + S_{h-1} + S_{h-2}$$



AVL fák – magasság

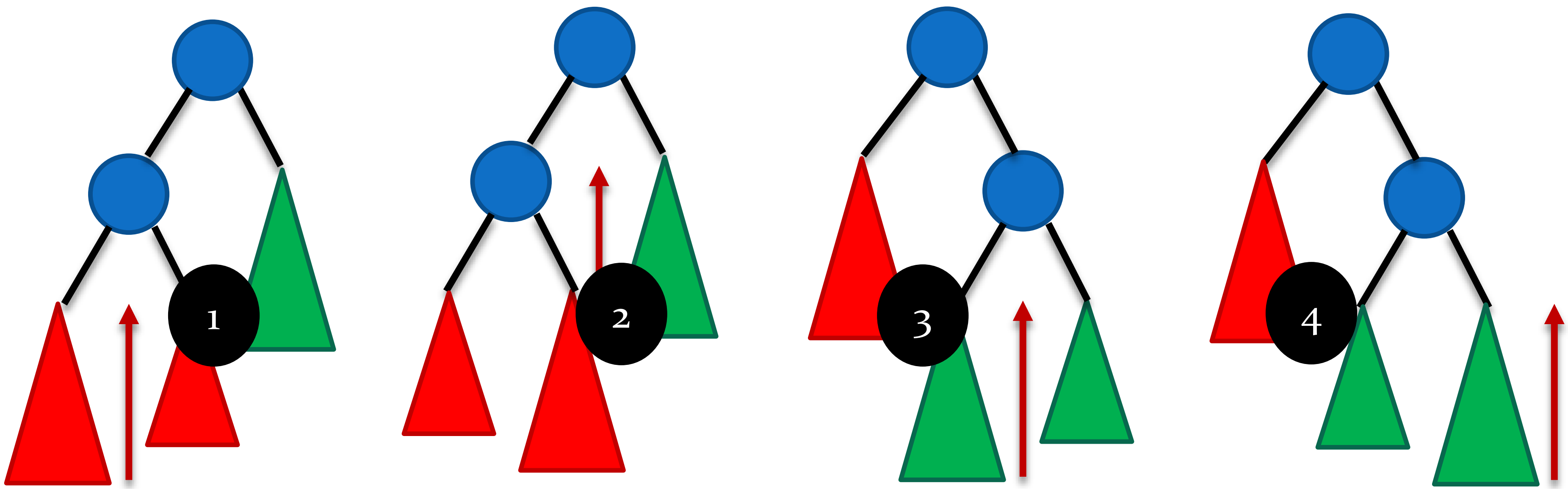
- Tétel
  Egy h magasságú AVL fának legalább $F_{h+3} - 1$ csúcsa van
- Bizonyítás
  - Legyen S_h a legkisebb h magasságú AVL fa mérete
    - ezt jelöljük majd n-nel
  - Ismert, hogy
    - $S_0 = 0$ és $S_1 = 1$, valamint $S_h = 1 + S_{h-1} + S_{h-2}$
  - Indukciót használva
    - $S_\mathrm{h} = F_{h+3} - 1$
      - Ez a „3-mal eltolt Fibonacci” szám
      - $F_{h+2} - 1 + F_{h+1} - 1 = F_{h+3} - 1$

AVL fák – magasság

- Tétel: Ha F AVL fa, akkor $h(F) \leq 1{,}44 * \log_2(n+1)$ ahol n az F fa pontjainak számát jelöli
- Bizonyítás: Legyen S_i az i magasságú, legkevesebb pontot tartalmazó AVL fa pontjainak száma, jelöljük $B_i = S_i + 1$
  - Ekkor $B_0 = 1$, $B_1 = 2$ és $B_m = B_{m-2} + B_{m-1}$ (ha $m > 1$)
  - Lemma: $\Phi^m \leq B_m$ ahol $\Phi = \left(1+\sqrt{5}\right)/2$
  - $1 = \Phi^0 \leq B_0\Phi \leq B_1$. Teljes indukcióval, a $2 \ldots m-1$-re igaz $B_m = B_{m-2} + B_{m-1} \leq \Phi^{m-2} + \Phi^{m-1} = \Phi^{m-2}(1+\Phi)$
  - Ugyanakkor $(1+\Phi) = \Phi^2$
- Tehát $\Phi^m \leq B_m = S_m + 1 \leq n + 1$ azaz $m * \log_2 \Phi \leq \log_2(n+1)$
- Ekkor $h(F) = m \leq \left(\frac{1}{\log_2 \Phi}\right) * \log_2(n+1) = 1{,}44 * \log_2(n+1)$

Újrakiegyensúlyozás beszúrásnál

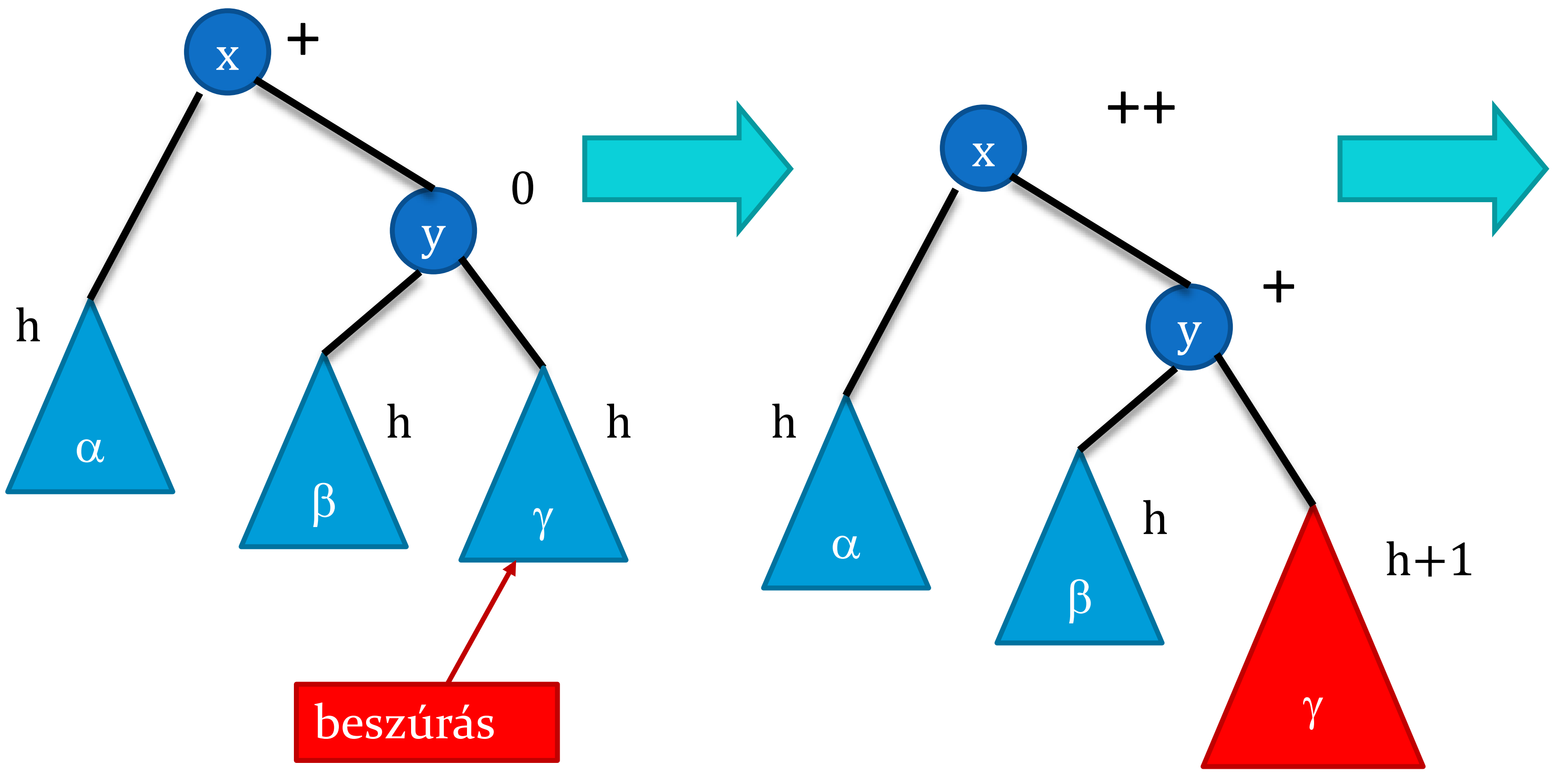
- Amikor beszúrunk egy elemet az AVL tulajdonság elromolhat
  - Ezt helyre kell hozni
    - 4 eset:
      - 1 és 4 tükörképek
      - 2 és 3 tükörképek



Újrakiegyensúlyozás beszúrásnál

- Egy új attribútumot vezetünk be, a kiegyensúlyozási tényezőt
  - -1 : bal részfa magasabb 1-gyel
  - 0 : egyforma magasak a részfák
  - +1: jobb részfa magasabb 1-gyel

A (++,+) szabály:

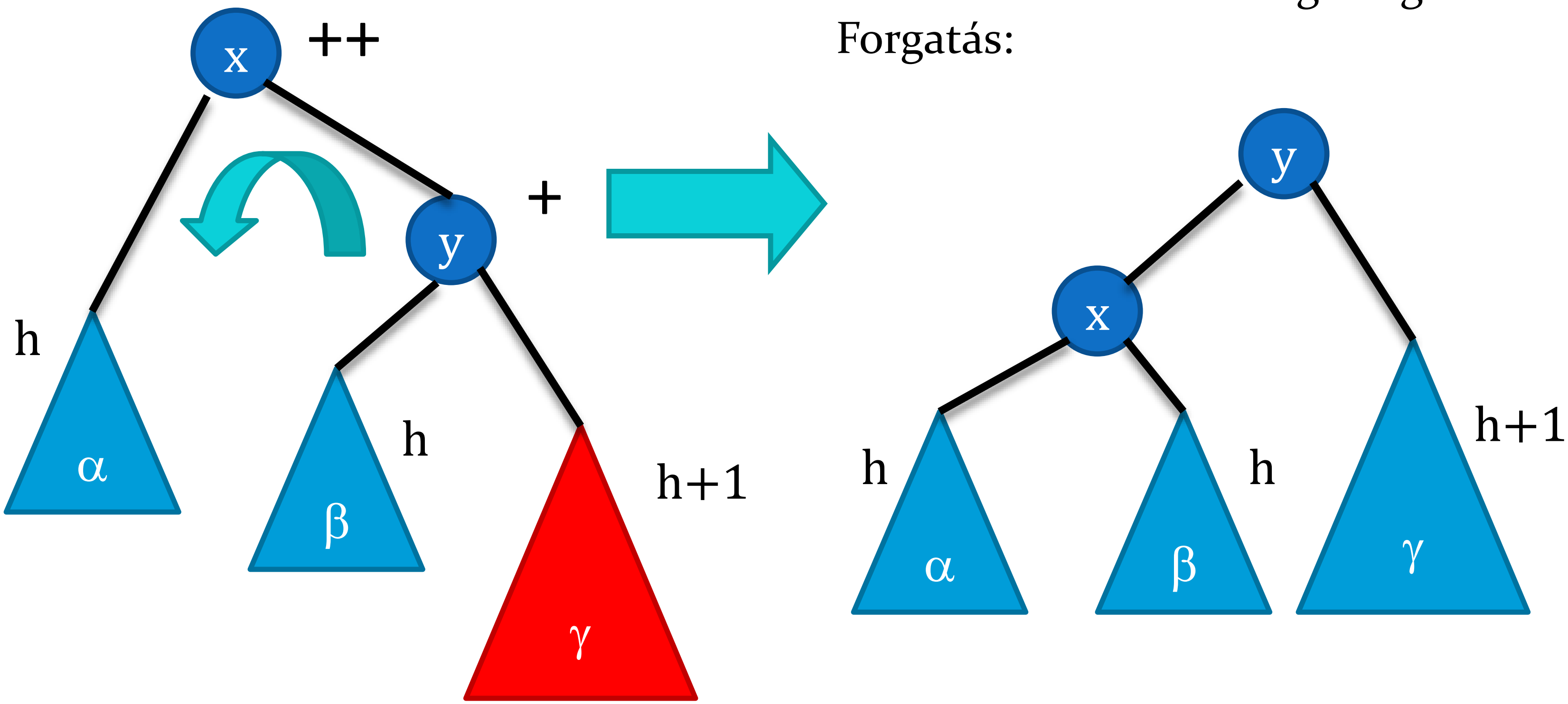


$\alpha < x < \beta < y < \gamma$

Az új levél a γ részfába került.
A beszúrás előtt a fa magassága h+2 volt.

A (++,+) szabály:

Az új levél a γ részfába került.
A beszúrás előtt a fa magassága h+2 volt.
Forgatás:

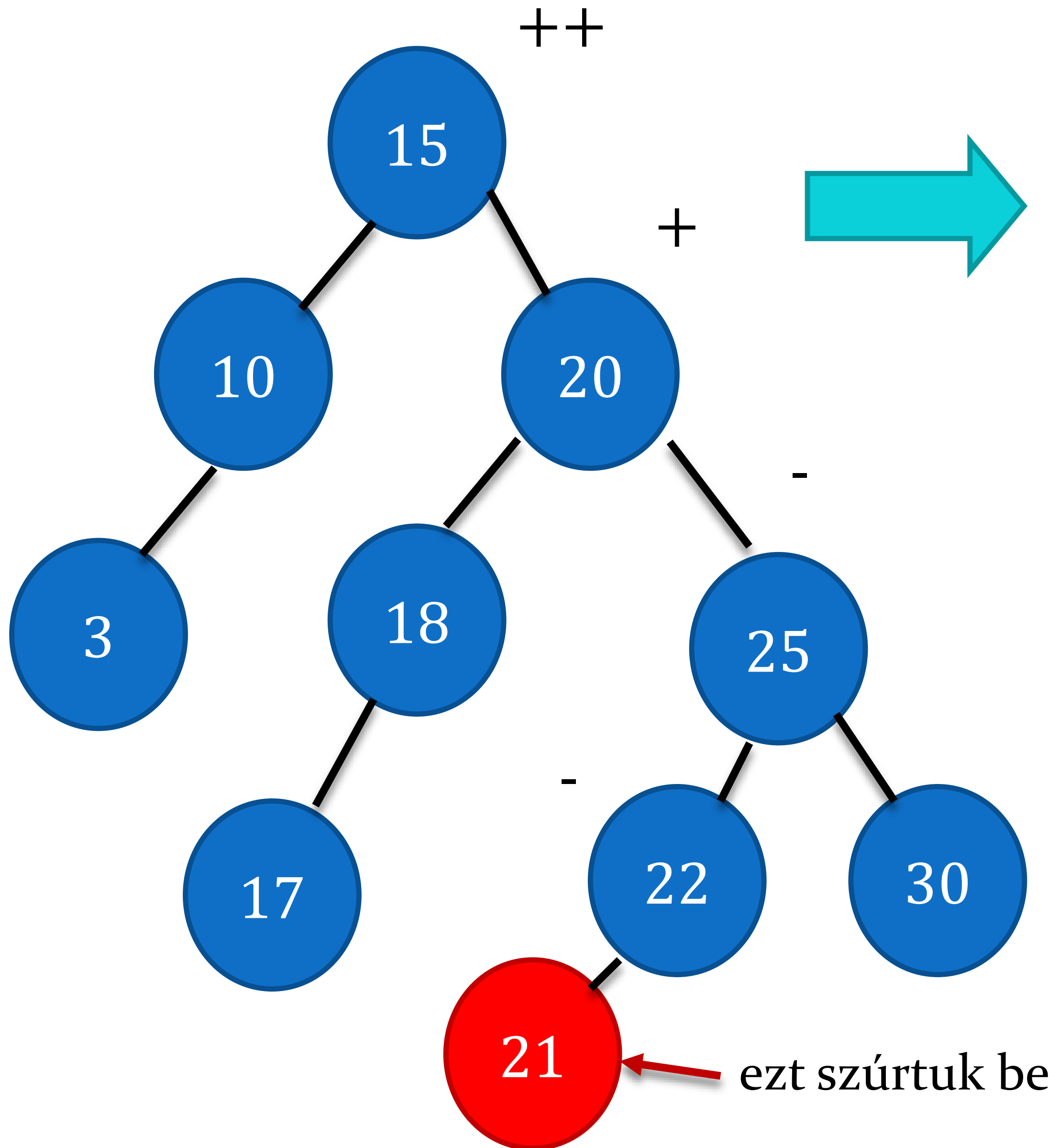


Ennek a tükörképe a (--,-) szabály! (1. eset)

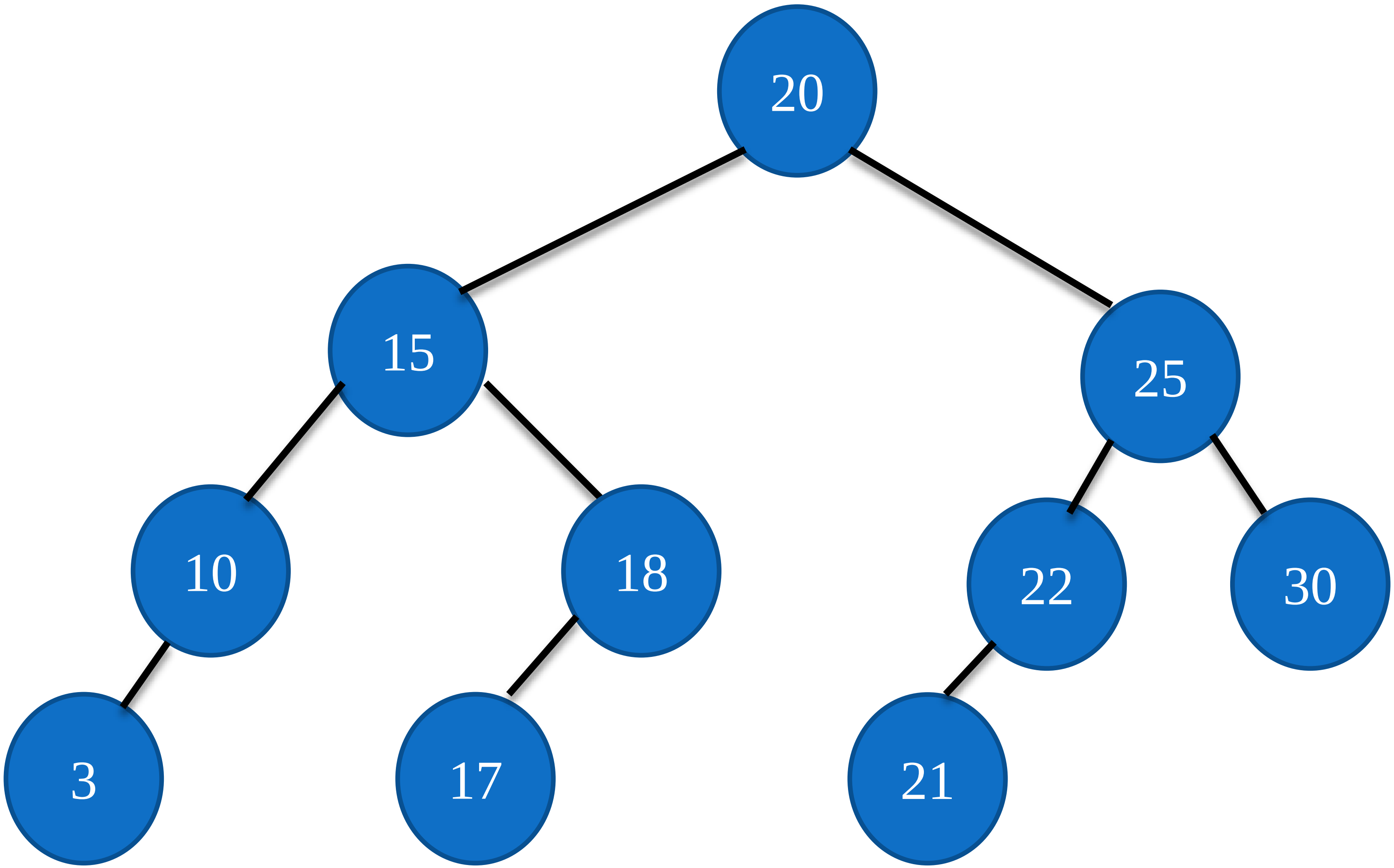
$\alpha < x < \beta < y < \gamma$

A forgatás után ismét h+2 a magasság. Ezért feljebb, a befoglaló fában (ha van), változatlanul érvényes az AVL tulajdonság, nem kell feljebb menni ellenőrizni.

Példa

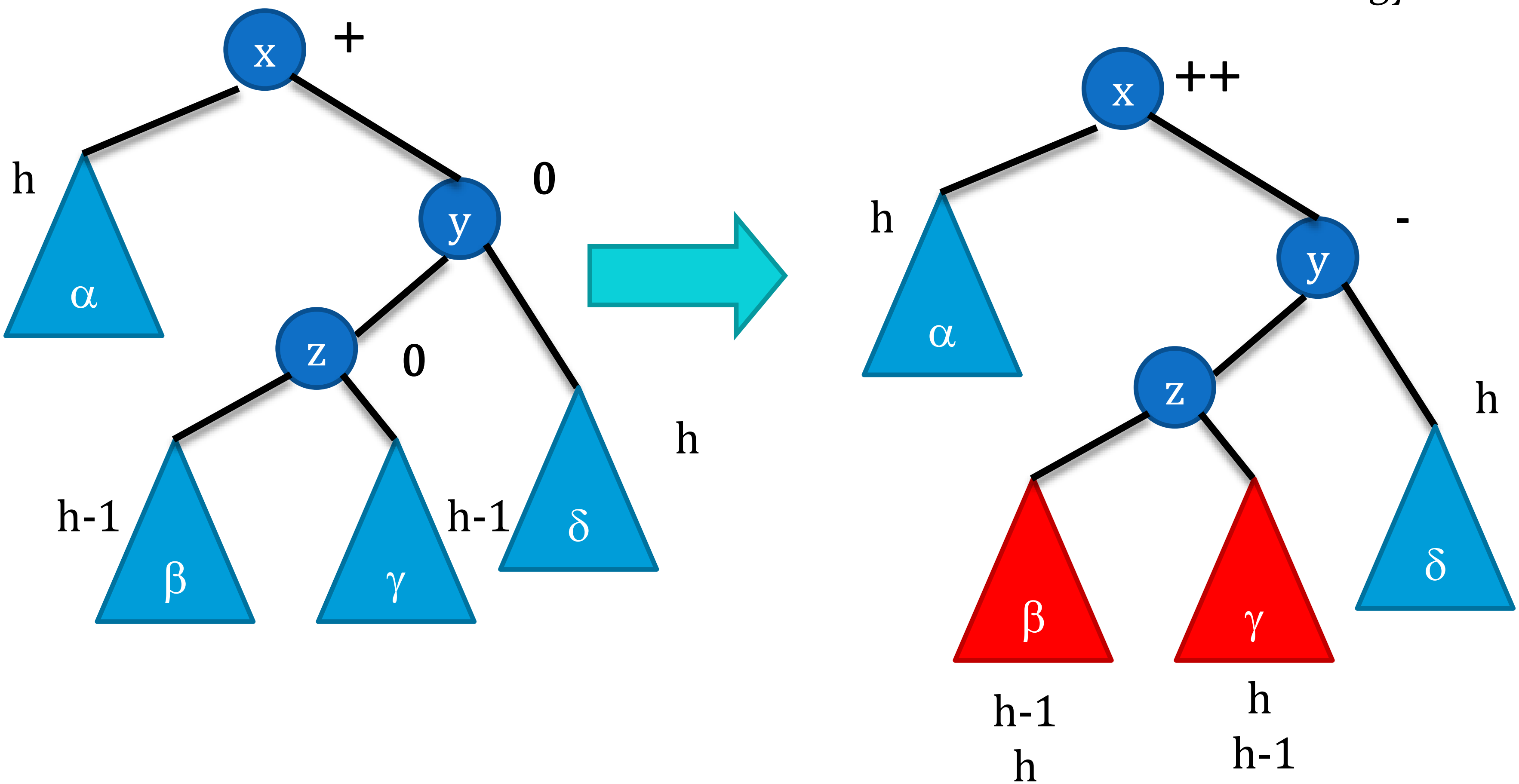


Példa



A (++,-) szabály:

Az új levél a z alatti β vagy γ részfába került.
A beszúrás előtt a z csúcs alatti fák egyformák:

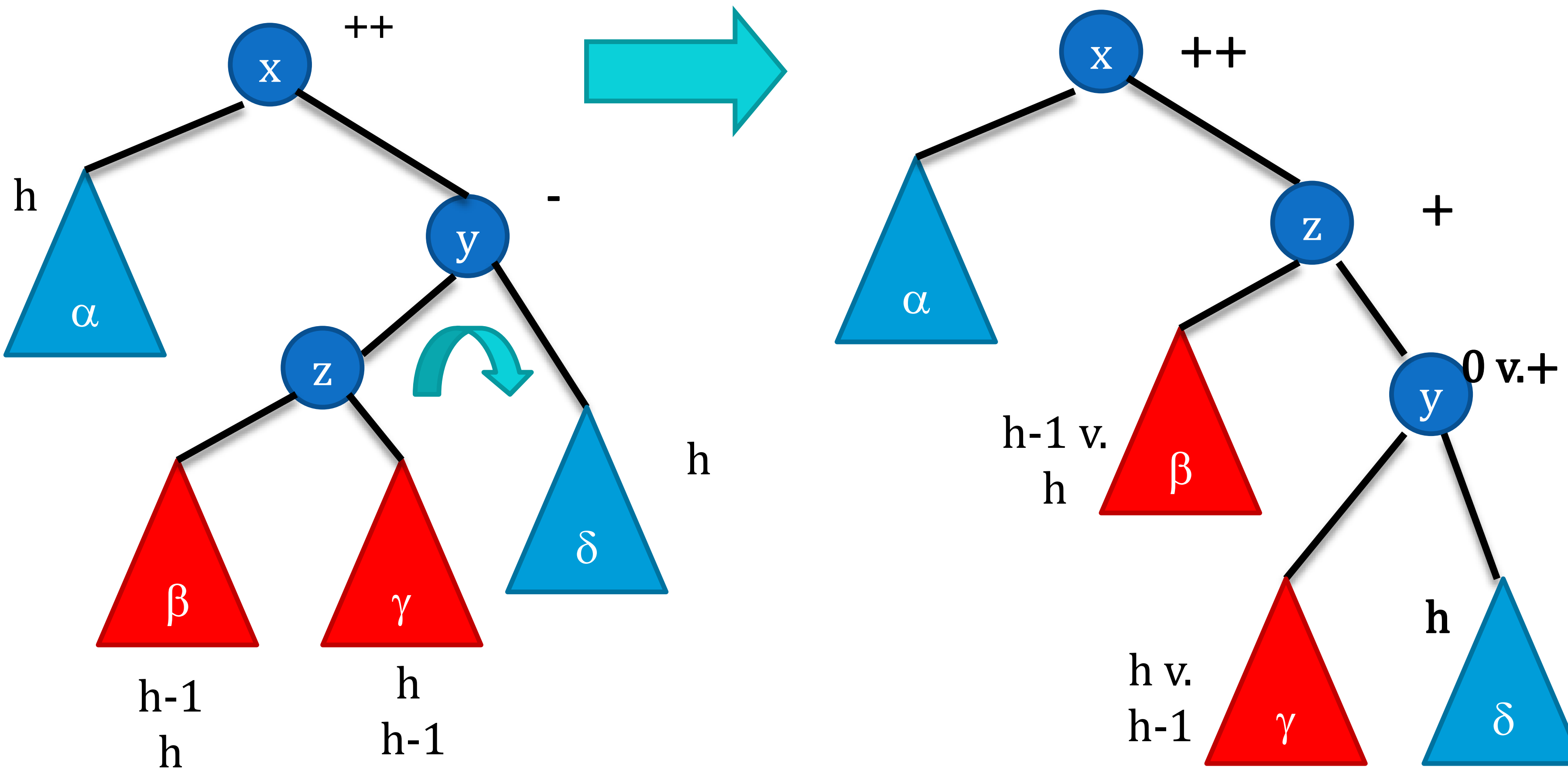


$\alpha < x < \beta < z < \gamma < y < \delta$

A beszúrás után az egyik részfa magassága h lett, a másik maradt h-1.

A (++,-) szabály:

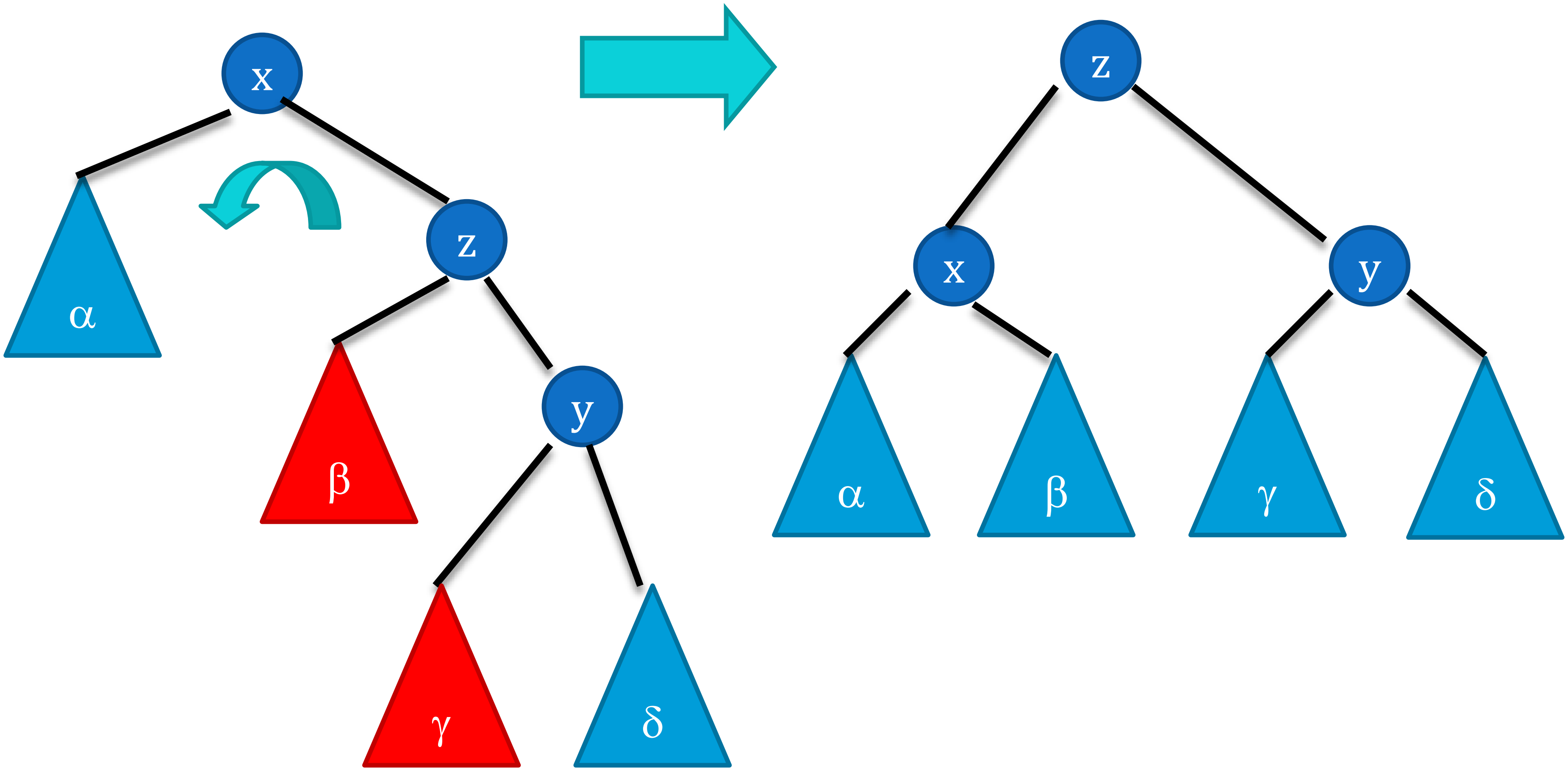
Dupla forgatás kell: először jobbra:



$\alpha < x < \beta < z < \gamma < y < \delta$

A (++,-) szabály:

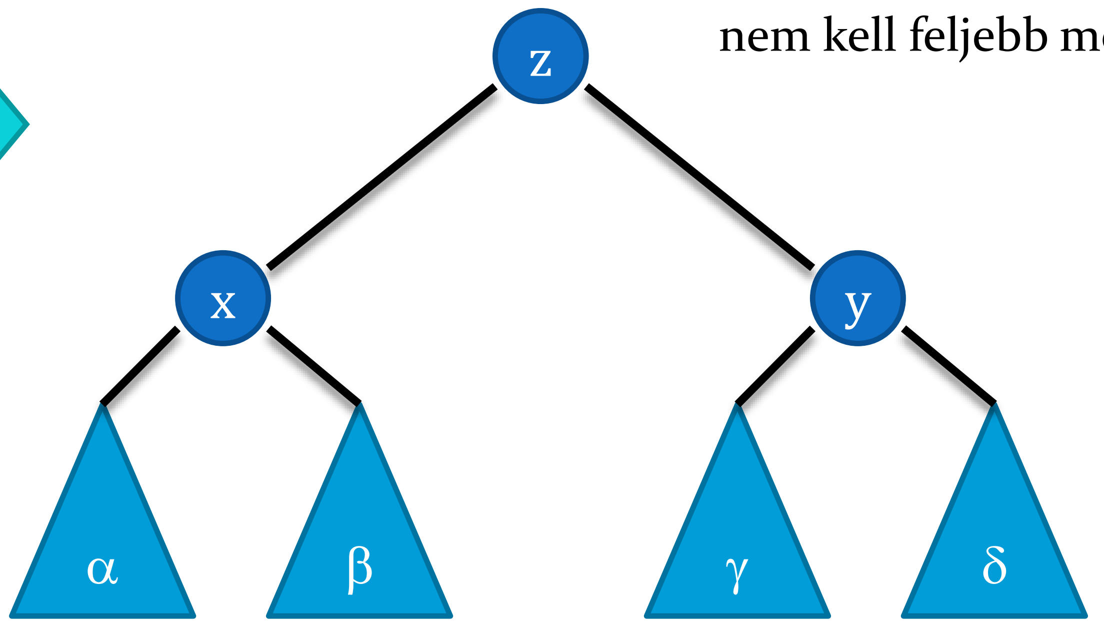
Dupla forgatás kell: azután balra:



A (++,-) szabály:

Végeredmény

A beszúrás előtt az x gyökerű fa magassága h+2 volt.
A forgatás után ismét h+2 a magasság.
Ezért feljebb, a befoglaló fában (ha van), változatlanul érvényes az AVL tulajdonság, nem kell feljebb menni ellenőrizni.



Továbbra is igaz:
$\alpha < x < \beta < z < \gamma < y < \delta$

Ennek a tükörképe a (--,+) szabály!

Újrakiegyensúlyozás beszúrásnál

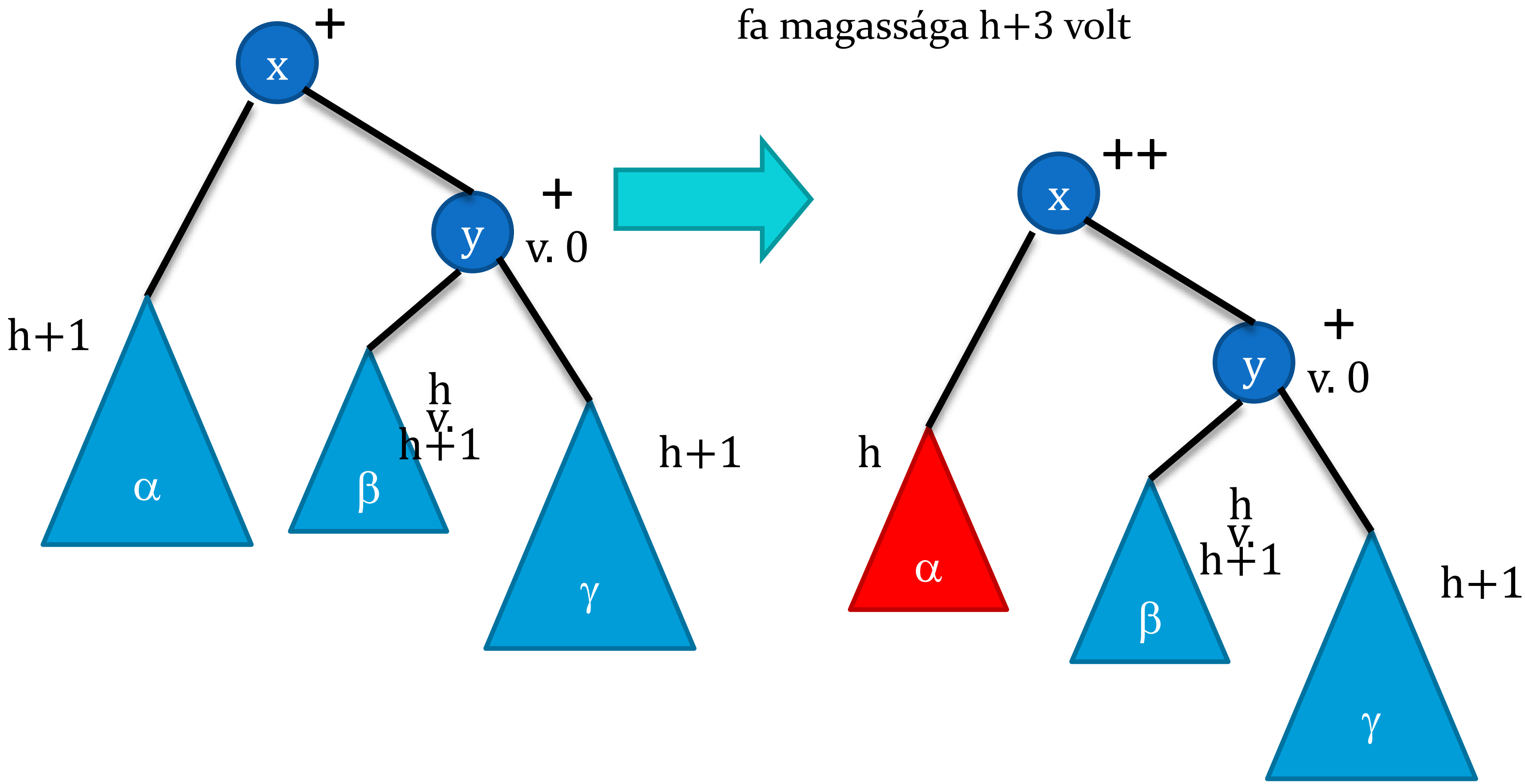
- Összefoglalva:
  - A beszúrás után az új levéltől felfelé haladva a gyökér felé újra számoljuk a csúcsok címkéit ezen az útvonalon. Ha egy x csúcs címkéje ++ vagy –– lesz, akkor az x gyökerű (rész)fa (esetleg dupla) forgatásával helyreállítható az AVL tulajdonság.
  - A tényleges helyreállítási lépés műveletigénye: $\mathcal{O}(1)$

Újrakiegyensúlyozás beszúrásnál

- Tétel
  - Legyen S egy n csúcsból álló AVL-fa. `BESZÚR(s; S)` után legfeljebb egy (esetleg dupla) forgatással helyreállítható az AVL-tulajdonság. A beszúrás költsége ezzel együtt is $\mathcal{O}(\log n)$.
- Bizonyítás
  - az előzőekből következik

AVL fák – újrakiegyensúlyozás törlésnél

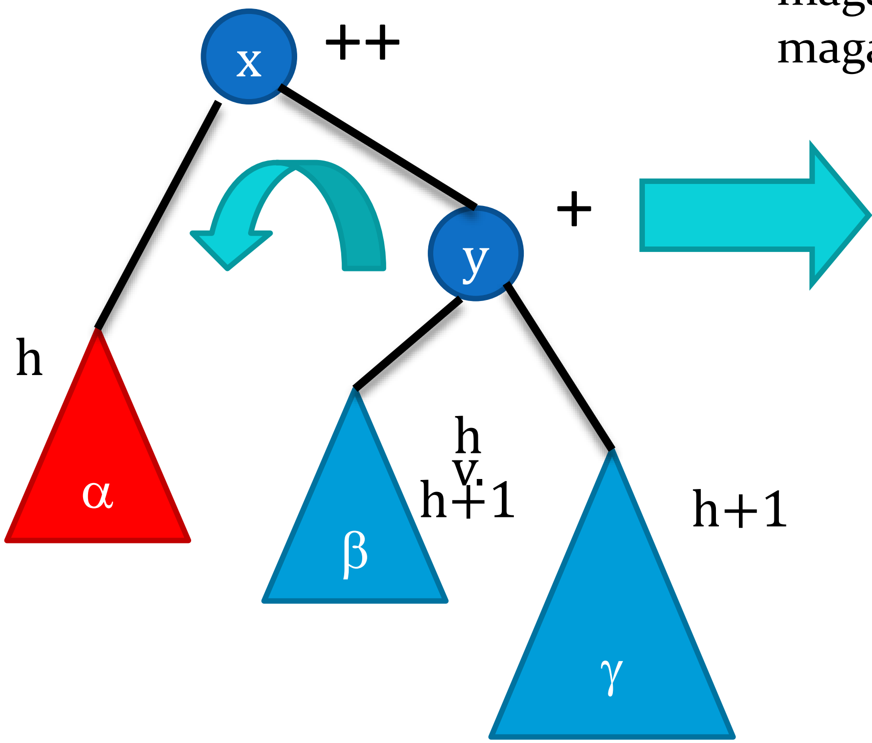
A törlés az α részfában történt. Ennek a magassága h+1 volt és h lett. Az x gyökerű fa magassága h+3 volt

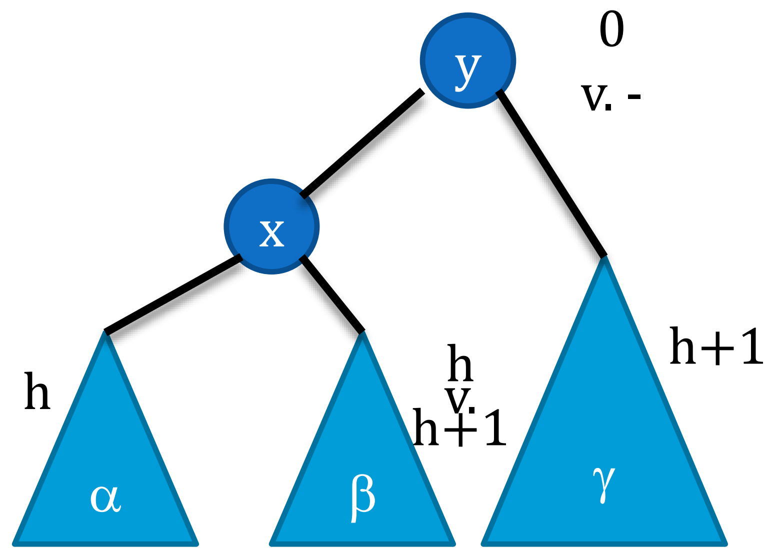


$\alpha < x < \beta < y < \gamma$

A (++,+) (++,0) szabályok

A törlés az α részfában történt. Ennek a magassága h+1 volt és h lett. Az x gyökerű fa magassága h+3 volt. Forgatás:



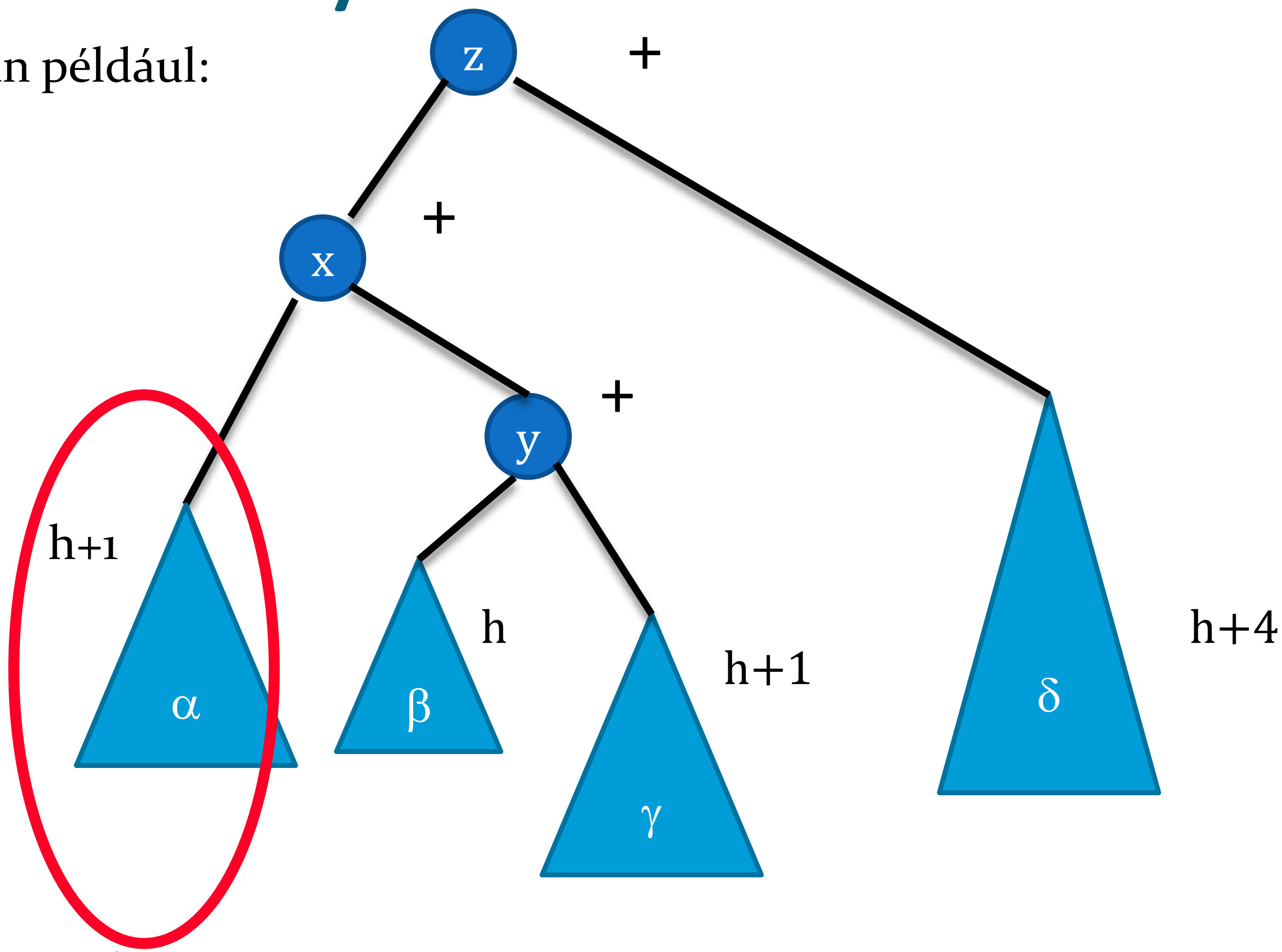


$\alpha < x < \beta < y < \gamma$

A forgatás után h+2 a magasság. Ezért feljebb, a befoglaló fában (ha van), nem biztos, hogy változatlanul érvényes az AVL tulajdonság, **feljebb kell menni** ellenőrizni, amíg a gyökérig nem jutunk.

A (++,+) szabály

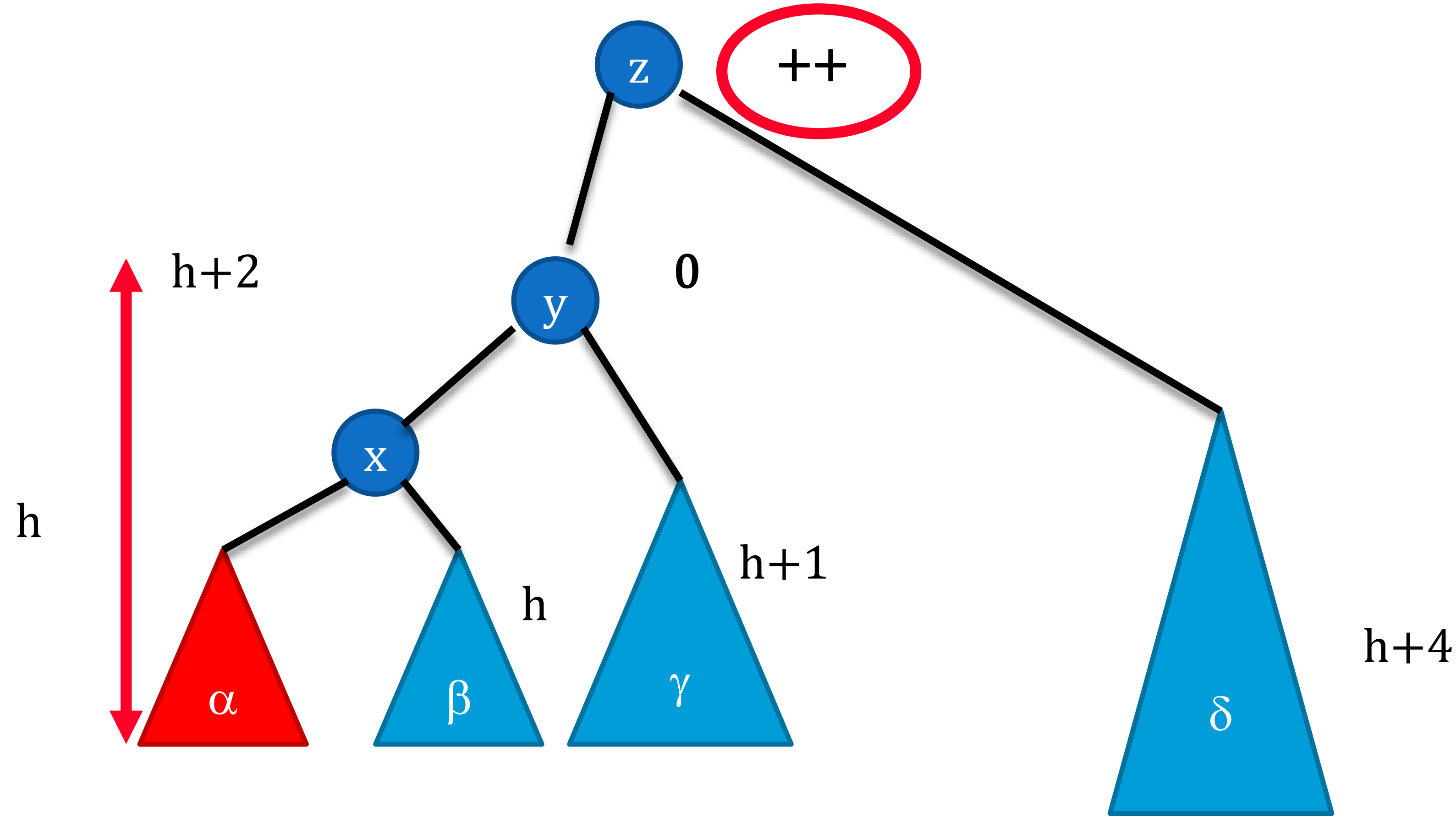
Ha az eredeti fában például:



$\alpha < x < \beta < y < \gamma < z < \delta$

A (++,+) szabály

Akkor az eredmény fa:

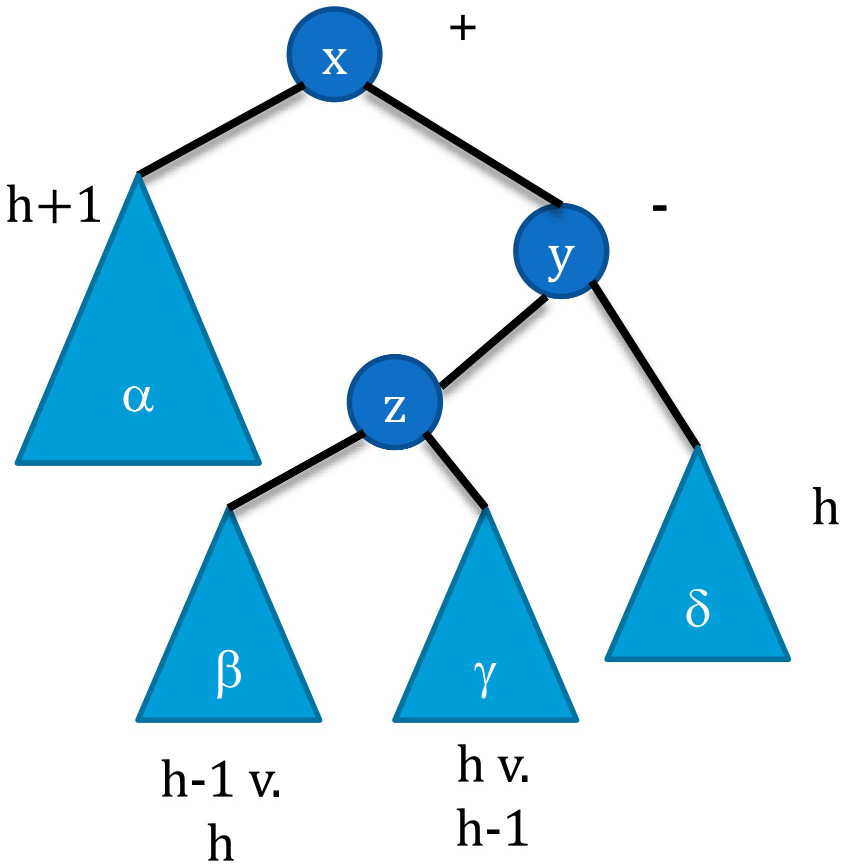


$\alpha < x < \beta < y < \gamma < z < \delta$

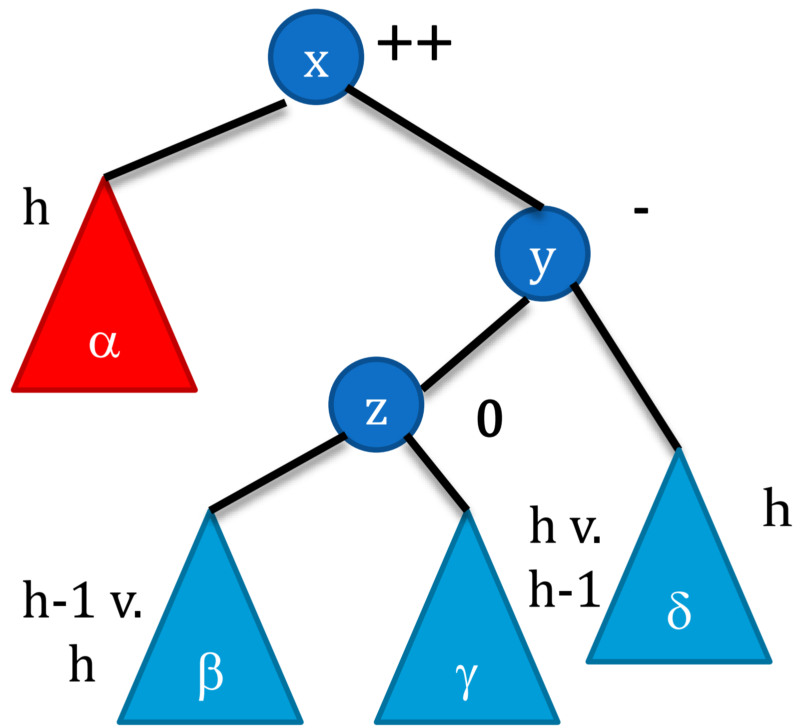
feljebb kell menni ellenőrizni, és szükség szerint helyreállítani, amíg a gyökérig nem jutunk.

A (++,-) szabály

A törlés az α részfában történik. Ennek a magassága h+1 volt és h lett. Az x gyökerű fa magassága h+3.



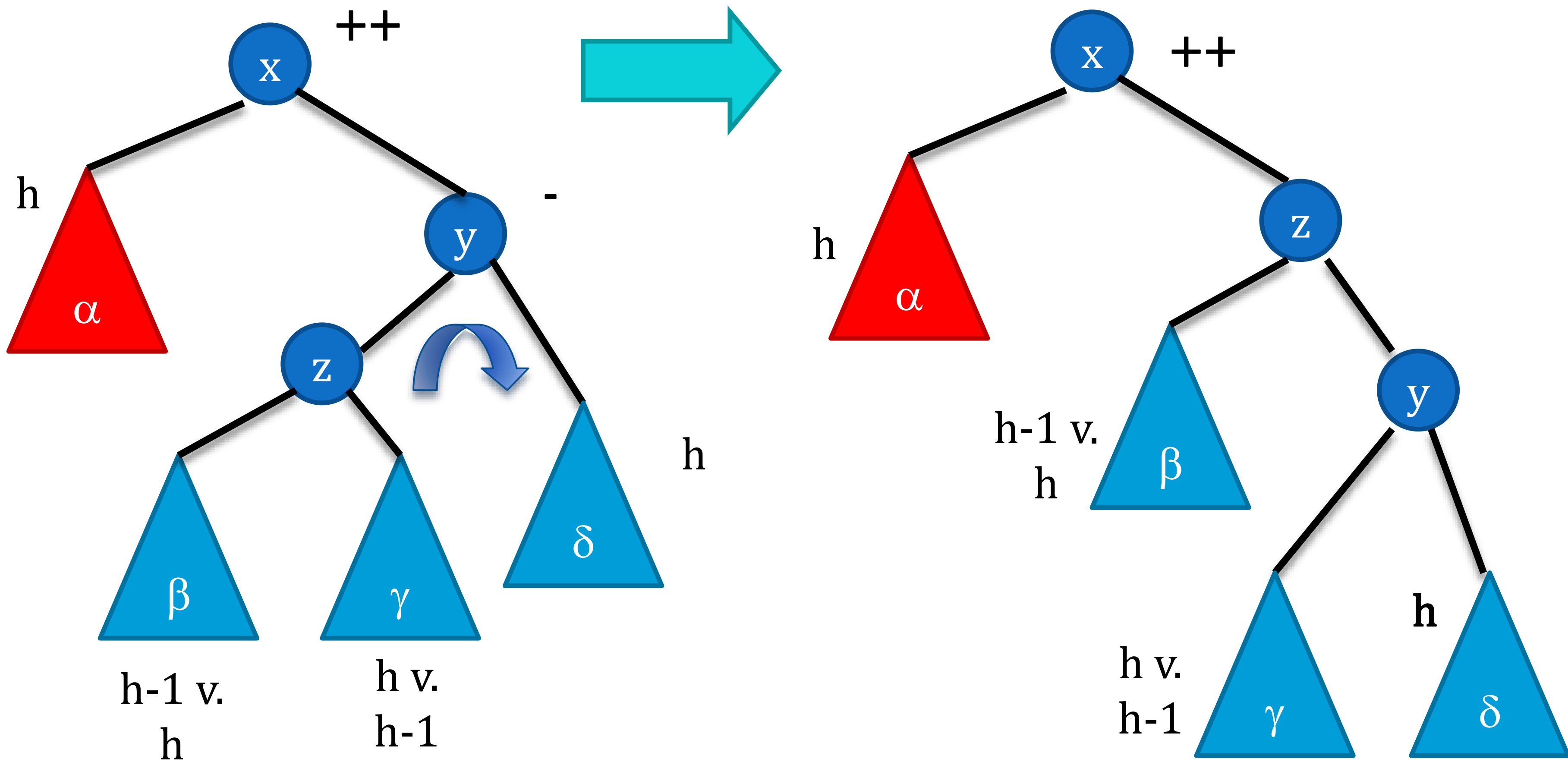
$\alpha < x < \beta < z < \gamma < y < \delta$



dupla forgatás

A (++,-) szabály

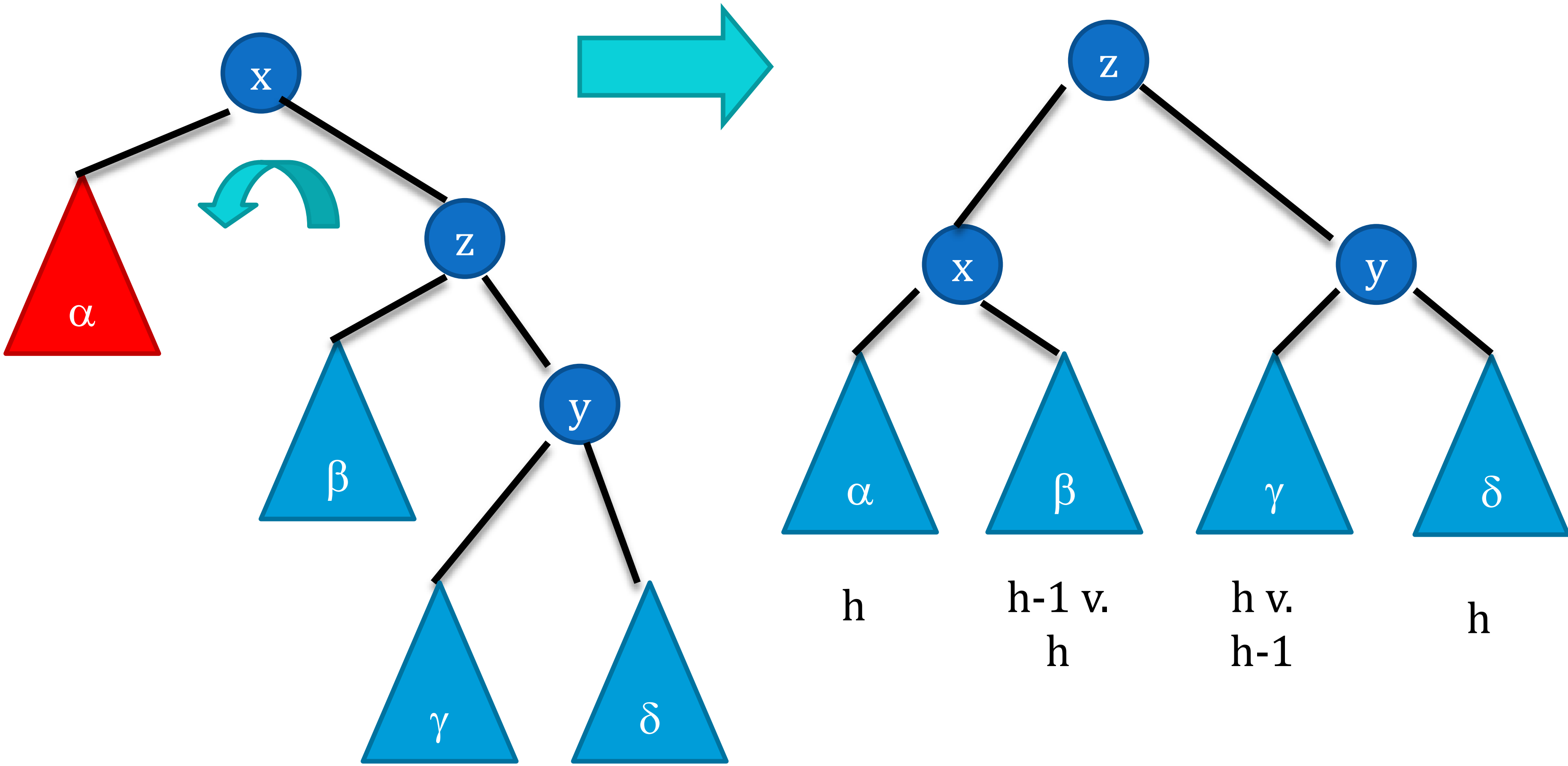
Dupla forgatás kell: először jobbra:



$\alpha < x < \beta < z < \gamma < y < \delta$

A (++,-) szabály

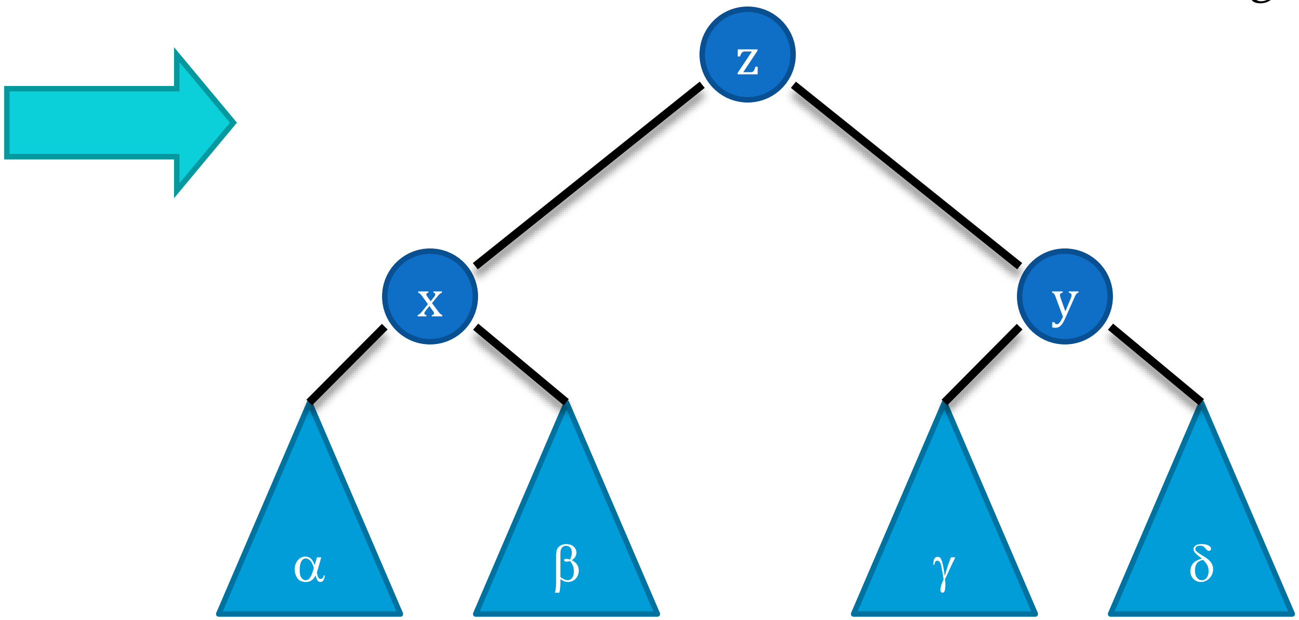
Dupla forgatás kell: azután balra:



A (++,-) szabály

Végeredmény

A forgatás után h+2 a magasság. Ezért feljebb, a befoglaló fában (ha van), nem biztos, hogy változatlanul érvényes az AVL tulajdonság, **feljebb kell menni** ellenőrizni, amíg a gyökérig nem jutunk.



Továbbra is igaz:

$\alpha < x < \beta < z < \gamma < y < \delta$

Újrakiegyensúlyozás törlésnél

- Összefoglalva:
  - Mivel az x gyökerű fa magassága csökkent a forgatással, ezért feljebb is, ha van befoglaló fa, elromolhatott az AVL tulajdonság
  - A törlés után a törölt elem szülőjétől kezdve felfelé haladva a gyökér felé újra számoljuk a csúcsok címkéit ezen az útvonalon
  - Ha egy x csúcs címkéje ++ vagy –- lesz, akkor az x gyökerű (rész)fa (esetleg dupla) forgatásával helyreállítjuk annak AVL tulajdonságát
  - Ha x nem a gyökér, akkor feljebb kell lépni és folytatni kell az ellenőrzést
  - Szélsőséges esetben az adott útvonal minden pontjában forgatni kell

Újrakiegyensúlyozás törlésnél

- Tétel
  - Az n pontú AVL-fából való törlés után legfeljebb $1{,}44\log_2 n$ (sima vagy dupla) forgatás helyreállítja az AVL-tulajdonságot.
- Bizonyítás
  - az előzőekből következik.

Összefoglalás

- AVL fák
  - Az első dinamikusan kiegyensúlyozott fák
  - A magasság az optimális 44%-án belül
  - Újrakiegyensúlyozás forgatásokkal
  - $\mathcal{O}(\log n)$