Kavics Makró Fatranszformátorok és Típus Ellenőrzésük

Fülöp Zoltán, **Gazdag Zsolt** és **Muzamel Loránd**

`fulop@inf.u-szeged.hu, gazdag@inf.u-szeged.hu, muzamel@inf.u-szeged.hu`

Szegedi Tudományegyetem Informatikai Tanszékcsoport

Számítástudomány Alapjai Tanszék

H-6720 Szeged, Árpád tér 2.

Tartalom

- Fogalmak és jelölések
- (Kavics) fabejárás
- Kavics makró fatranszformátorok
- Típus ellenőrzés
- Eldönthetőségi eredmények

Fogalmak és jelölések

<u>kétváltozós relációk:</u> $\rho \subseteq H_1 \times H_2$, ρ^+, ρ^*

<u>ábécé:</u> A $A = \{a, b, c\}$

<u>A feletti szavak:</u> A^*, üres szó: ε

<u>szavak feletti szavak:</u> Ha A elemei maguk is szavak, pl.

$A = \{ac, acc, ab, ac\}$

$[ab; acc; ab; ac] \in A^*$, $[\,] \in A^*$

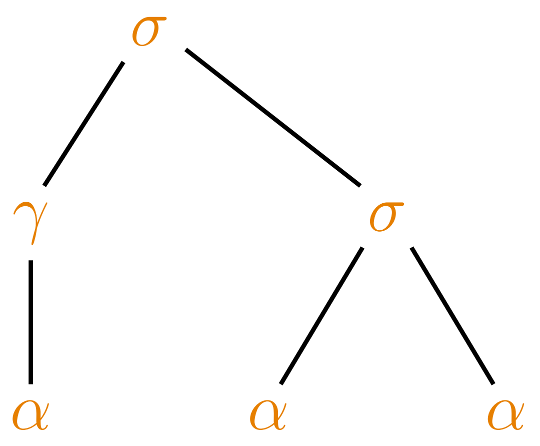
<u>A feletti legfeljebb n hosszú szavak:</u> $A^{\leq n}$

Fogalmak és jelölések

rangolt ábécé: $(\Sigma, rank)$, ahol $rank : \Sigma \to \mathbb{N}$

$$\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$$

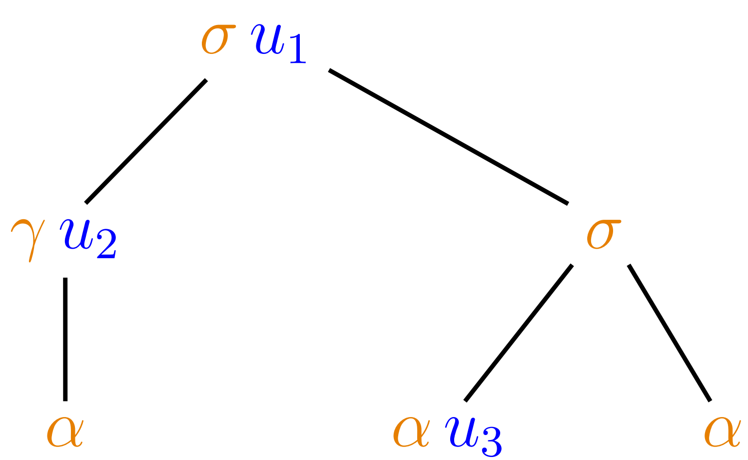
$$maxrank(\Sigma) = max\{rank(\sigma) \mid \sigma \in \Sigma\}$$



Σ feletti fák: T_Σ

Fogalmak és jelölések

<u>csúcsok címzése:</u>



$$u_1 = \varepsilon$$

$$u_2 = 1$$

$$u_3 = 21$$

Fogalmak és jelölések

fanyelvek: $L \subseteq T_\Sigma$

fatranszformációk: $\tau \subseteq T_\Sigma \times T_\Delta$

fatranszformációk output nyelvei:

$$\tau(L) = \{t \in T_\Delta \mid \exists\, s \in L : (s,t) \in \tau\}$$

fatranszformációk inverzei:

$$\tau^{-1} = \{(t,s) \in T_\Delta \times T_\Sigma \mid (s,t) \in \tau\}$$

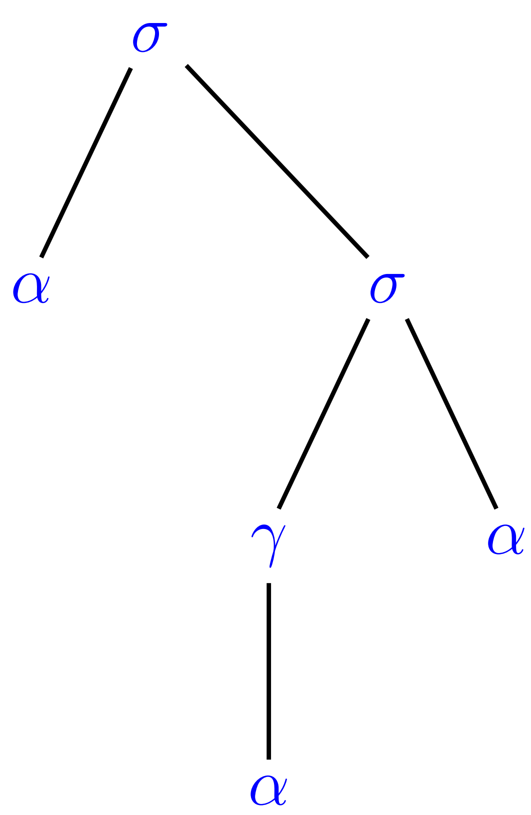
Faautomaták [GS84, GS97]

$$A = (P, \Sigma, p_0, R)$$

$$R: \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

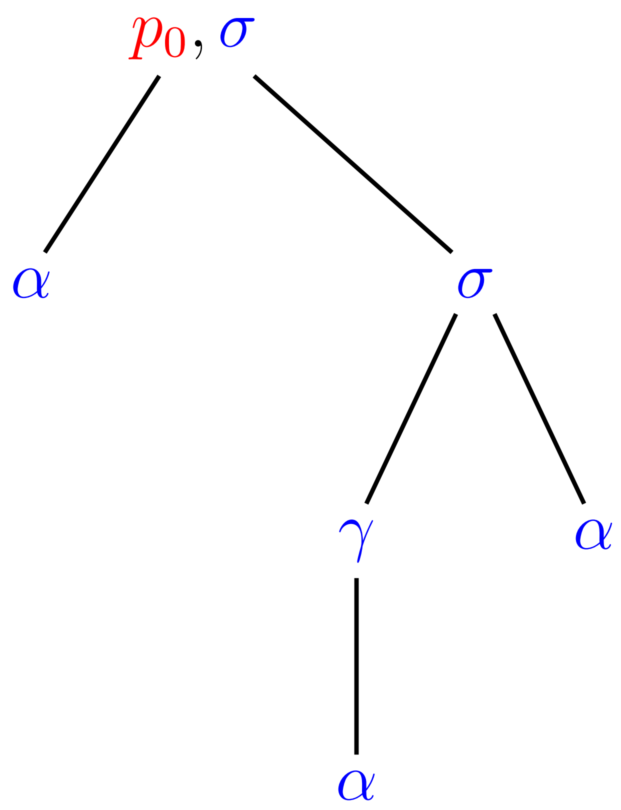
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{aligned} \langle p_0, \sigma \rangle &\longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle &\longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle &\longrightarrow (p_2) \\ \langle p_2, \alpha \rangle &\longrightarrow () \end{aligned}$$

Faautomaták [GS84, GS97]

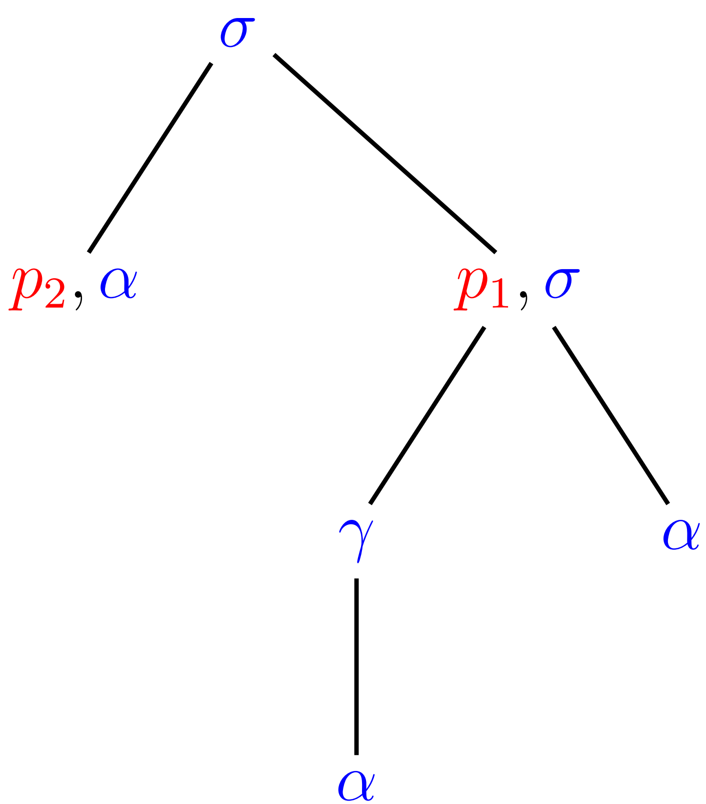
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

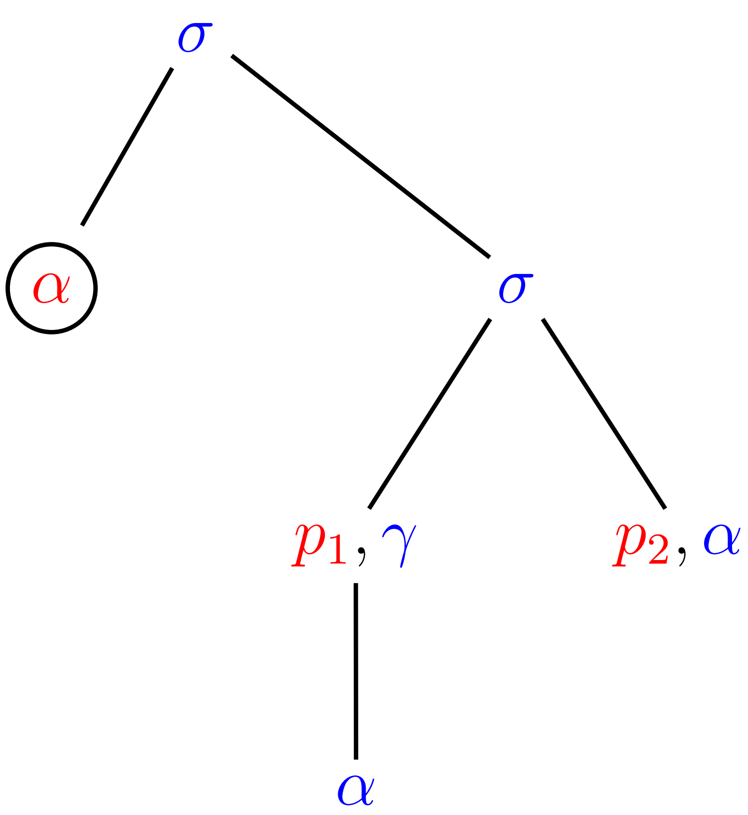
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

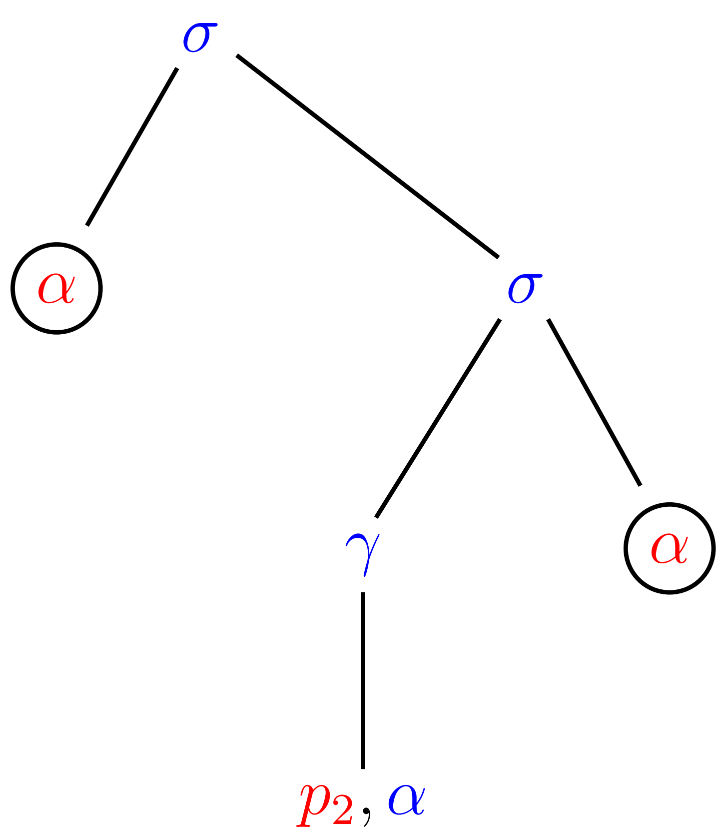
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

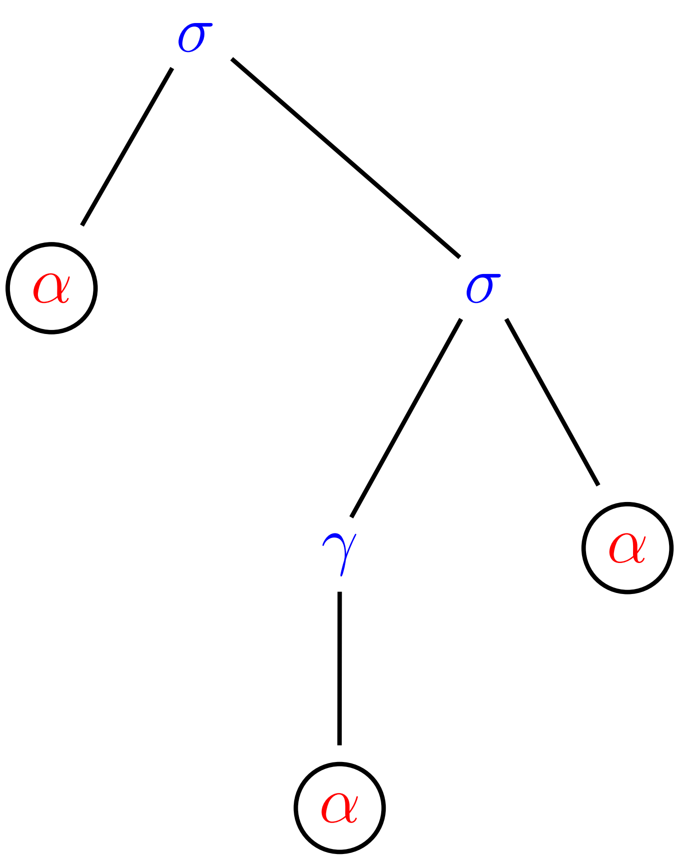
$$A = (P, \Sigma, p_0, R)$$



$$R : \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

$$A = (P, \Sigma, p_0, R)$$



$$R : \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

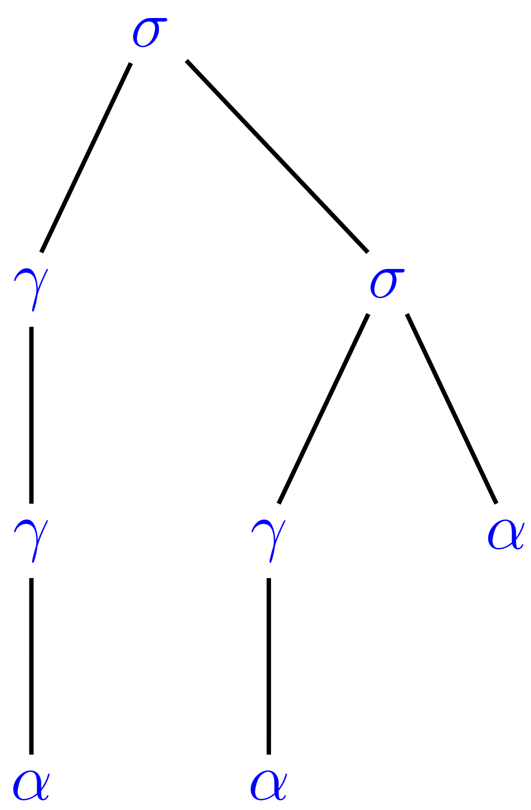
Faautomaták [GS84, GS97]

$$A = (P, \Sigma, p_0, R)$$

$$R : \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

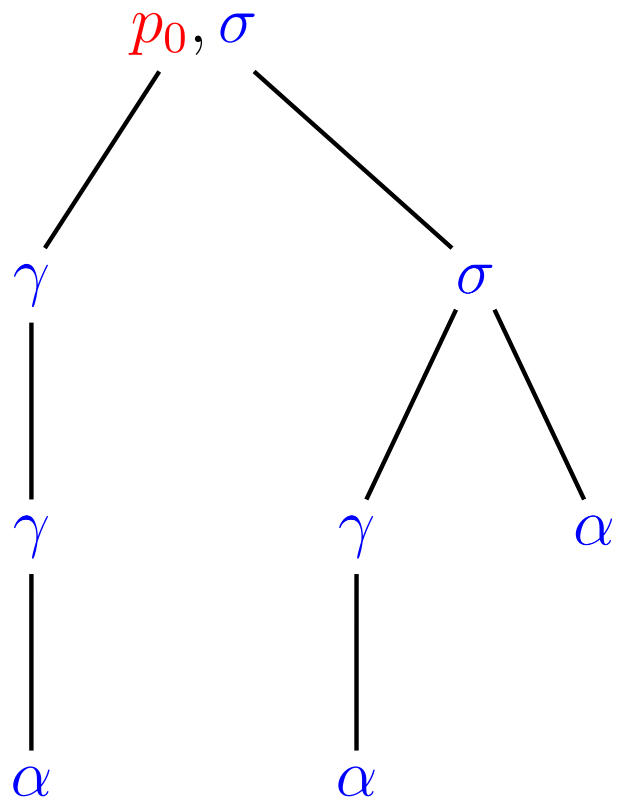
$$A = (P, \Sigma, p_0, R)$$



$$R: \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

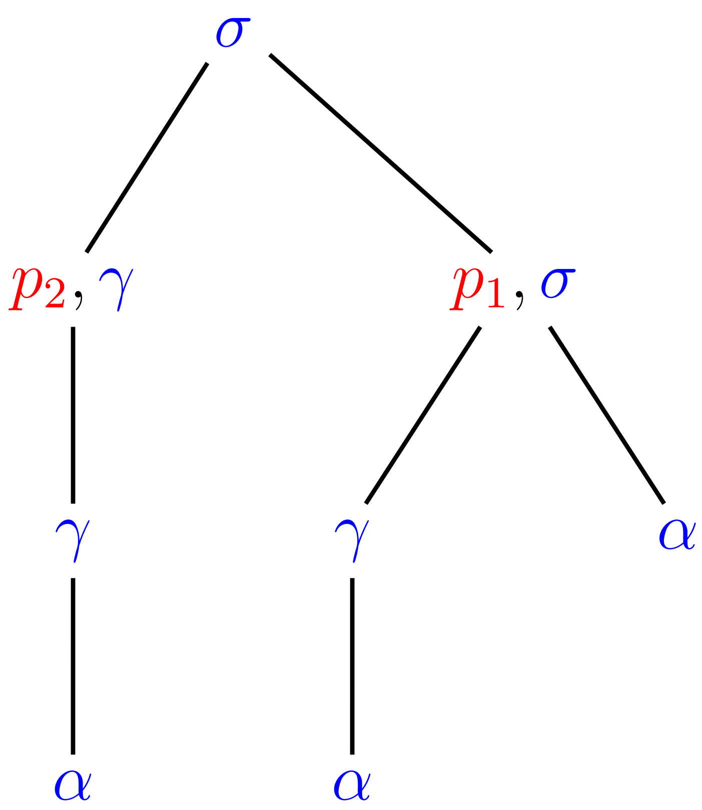
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{aligned} \langle p_0, \sigma \rangle &\longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle &\longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle &\longrightarrow (p_2) \\ \langle p_2, \alpha \rangle &\longrightarrow () \end{aligned}$$

Faautomaták [GS84, GS97]

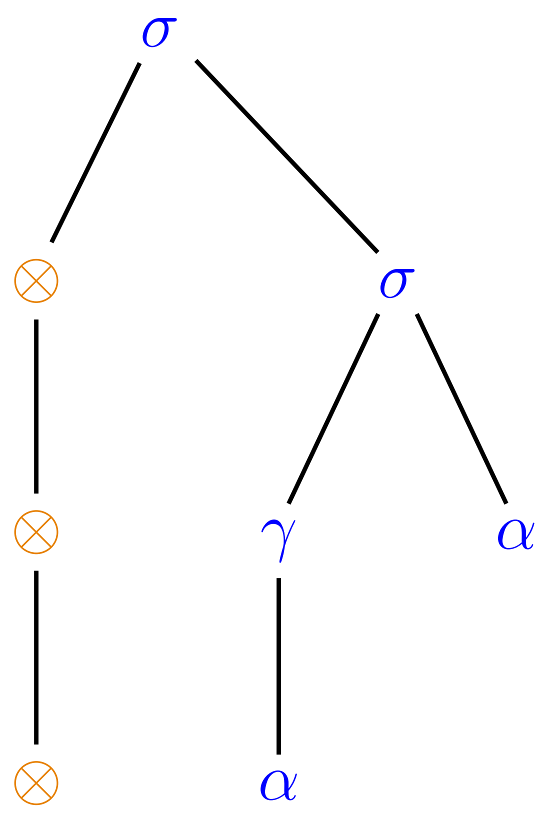
$$A = (P, \Sigma, p_0, R)$$



$$R: \quad \begin{array}{l} \langle p_0, \sigma \rangle \longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle \longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle \longrightarrow (p_2) \\ \langle p_2, \alpha \rangle \longrightarrow () \end{array}$$

Faautomaták [GS84, GS97]

$$A = (P, \Sigma, p_0, R)$$



$$R : \begin{aligned} \langle p_0, \sigma \rangle &\longrightarrow (p_2, p_1) \\ \langle p_1, \sigma \rangle &\longrightarrow (p_1, p_2) \\ \langle p_1, \gamma \rangle &\longrightarrow (p_2) \\ \langle p_2, \alpha \rangle &\longrightarrow () \end{aligned}$$

Faautomaták [GS84, GS97]

A által felismert fanyelv: $L(A)$

reguláris fanyelvek: REG, $dREG$

Lemma. **([GS84, GS97])**

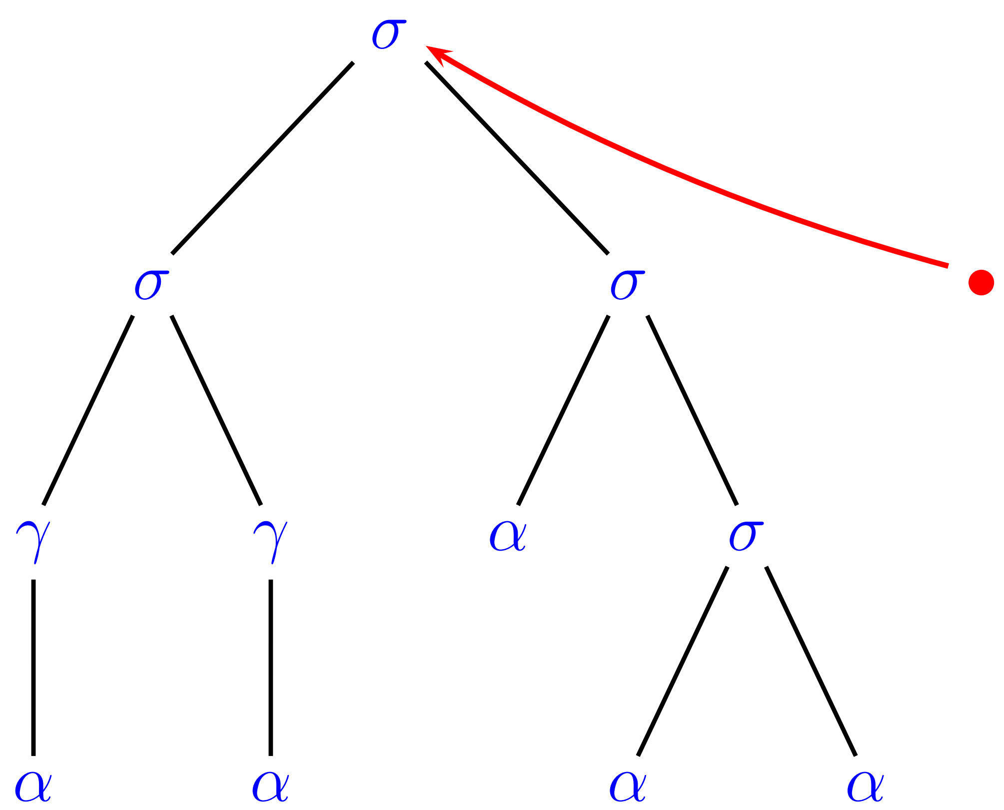
a) $dREG \subset REG$

b) A reguláris fanyelvek zártak a $\cap, \cup$ és $^{-}$ műveletekre.

c) Van olyan algoritmus, amely minden A faautomatára eldönti, hogy $L(A) = \emptyset$ teljesül-e.

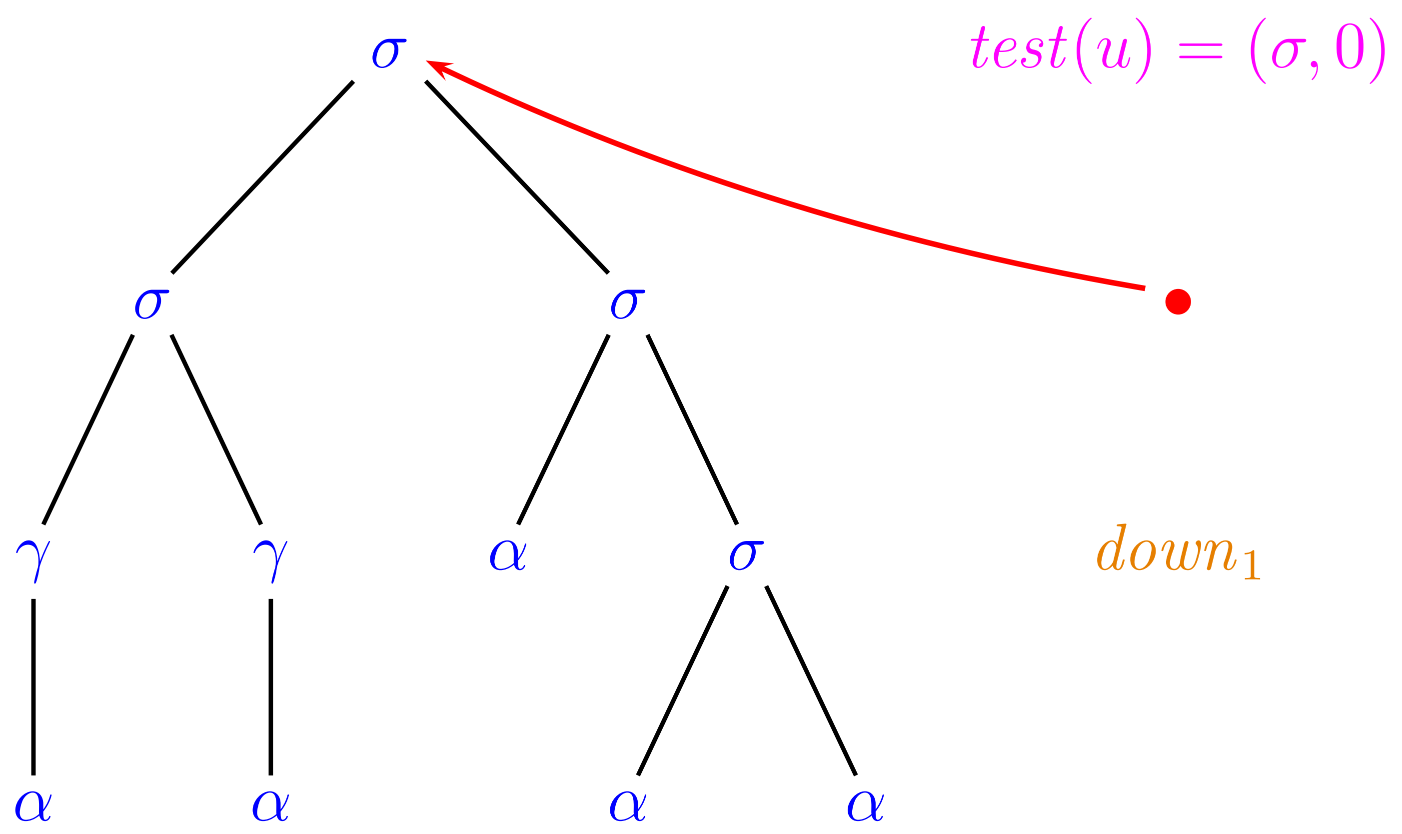
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



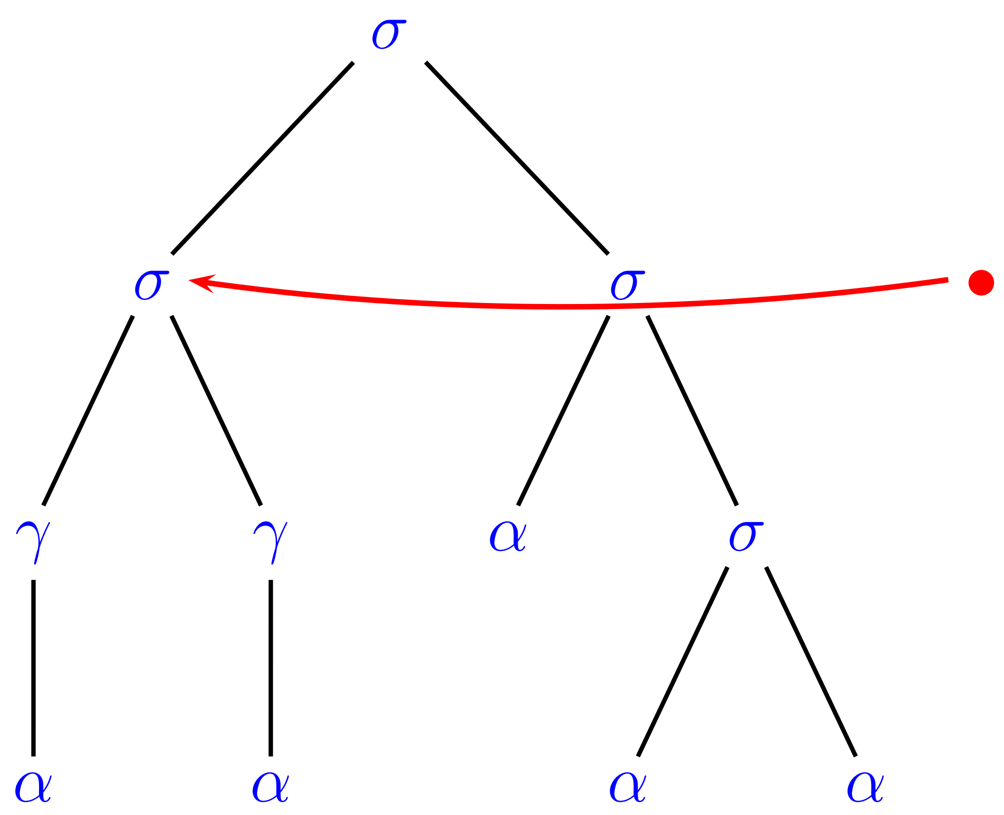
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



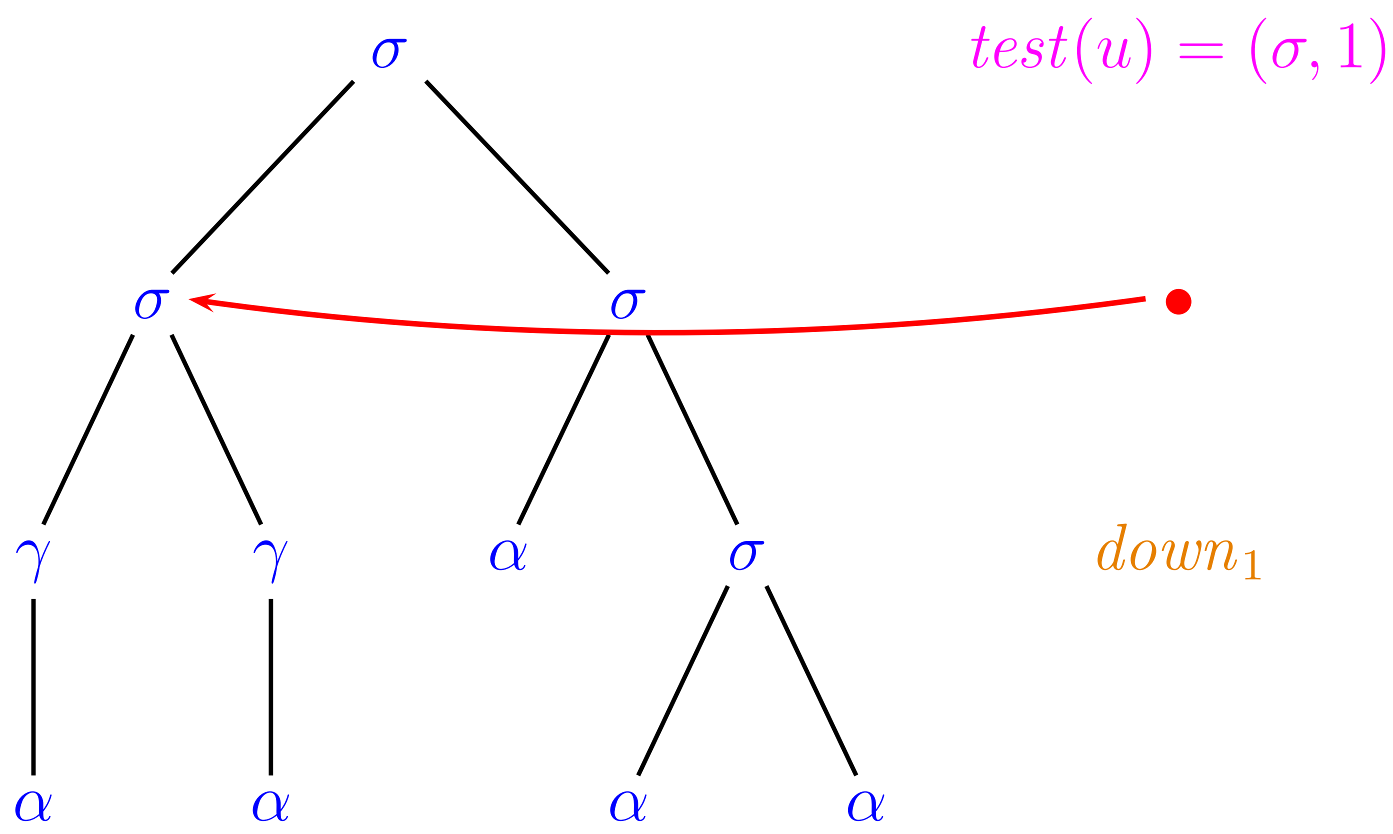
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



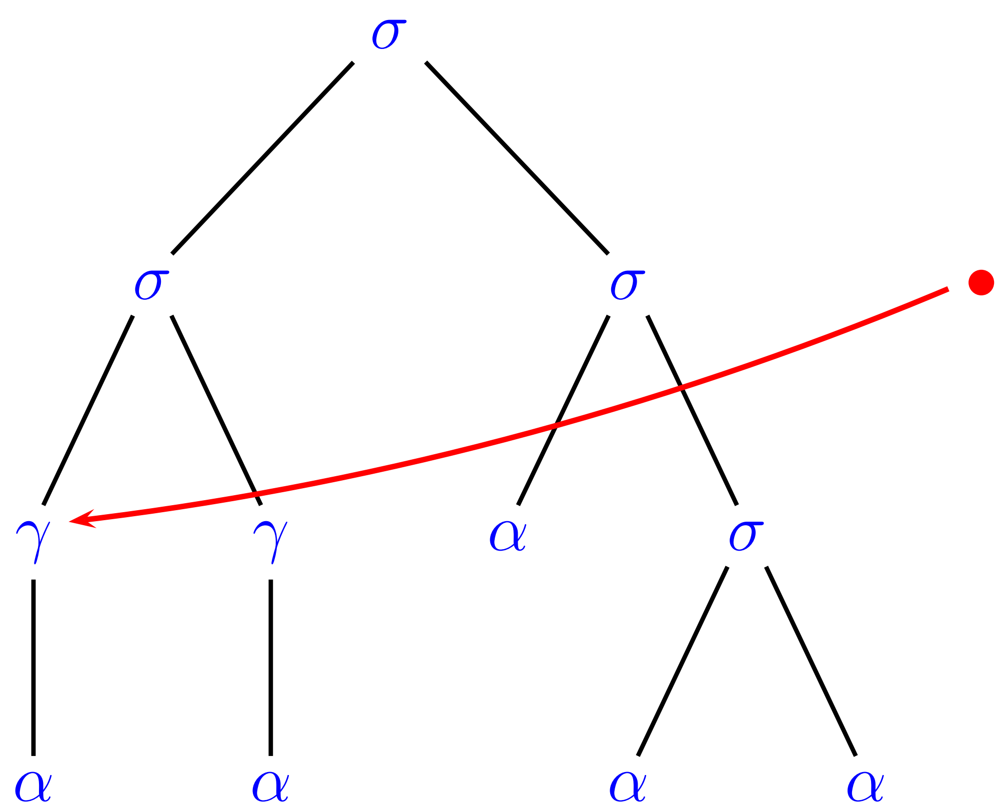
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



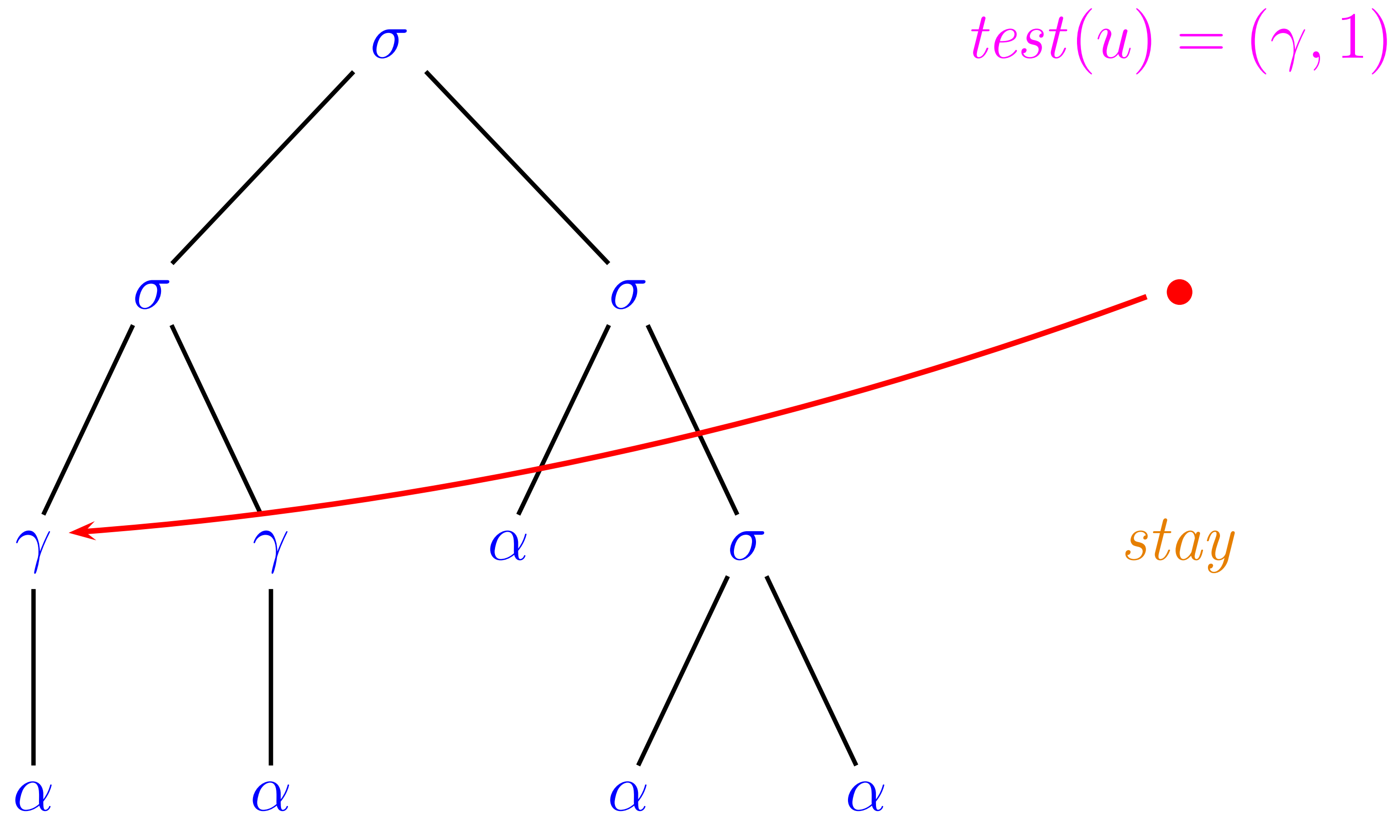
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



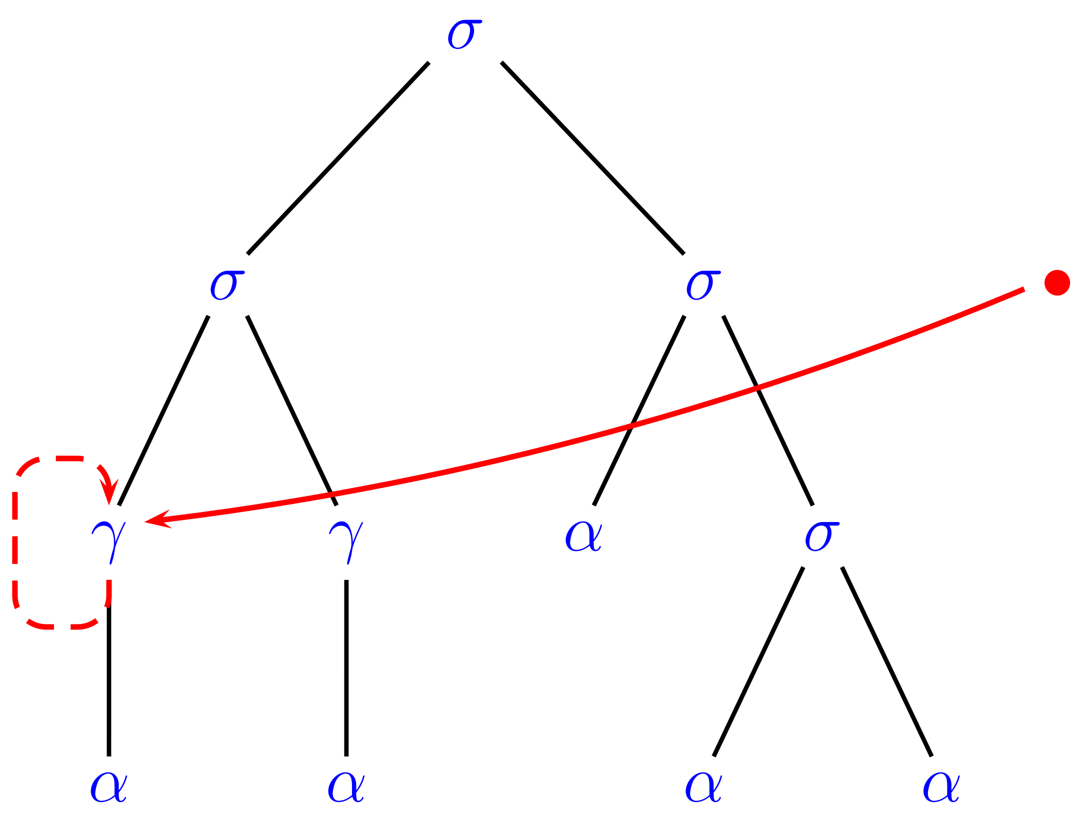
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



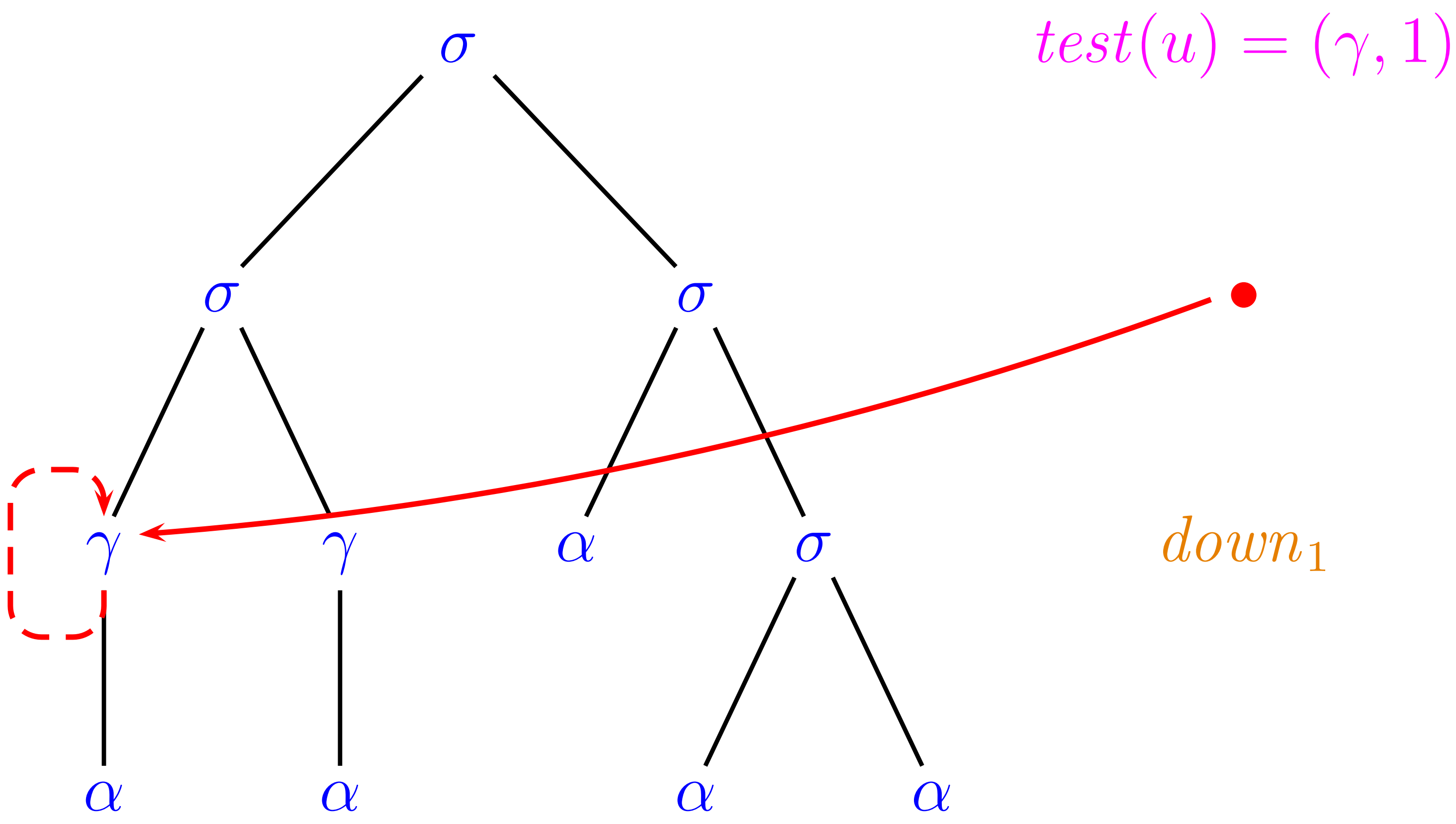
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



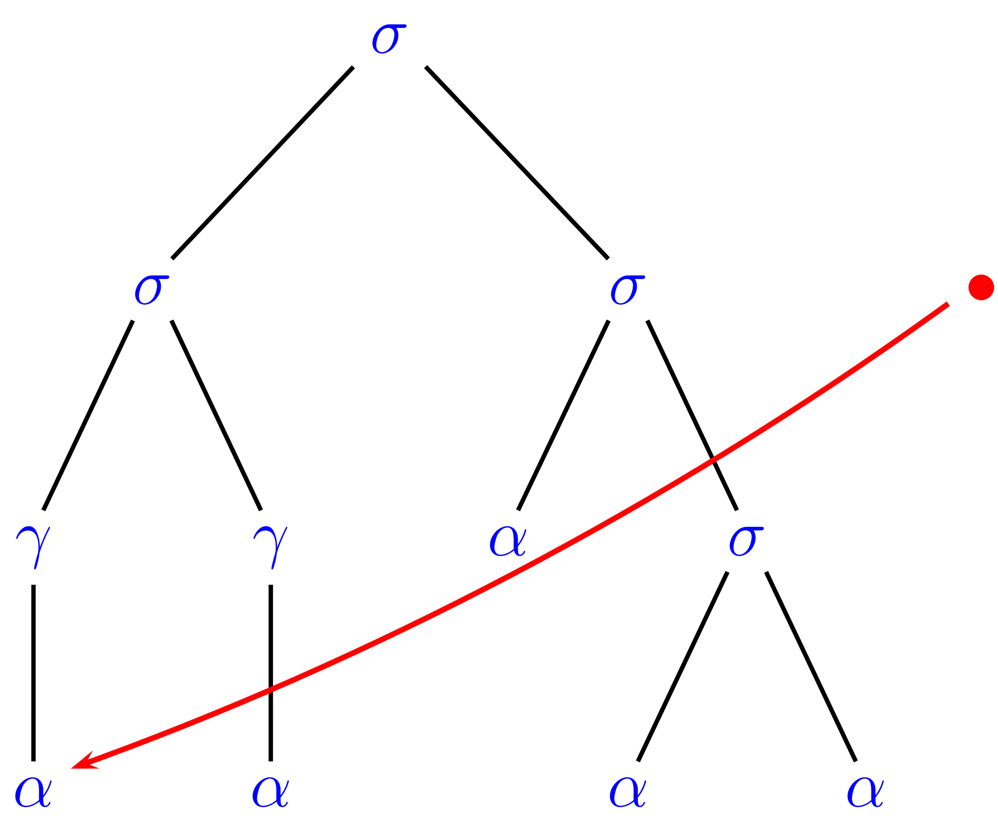
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



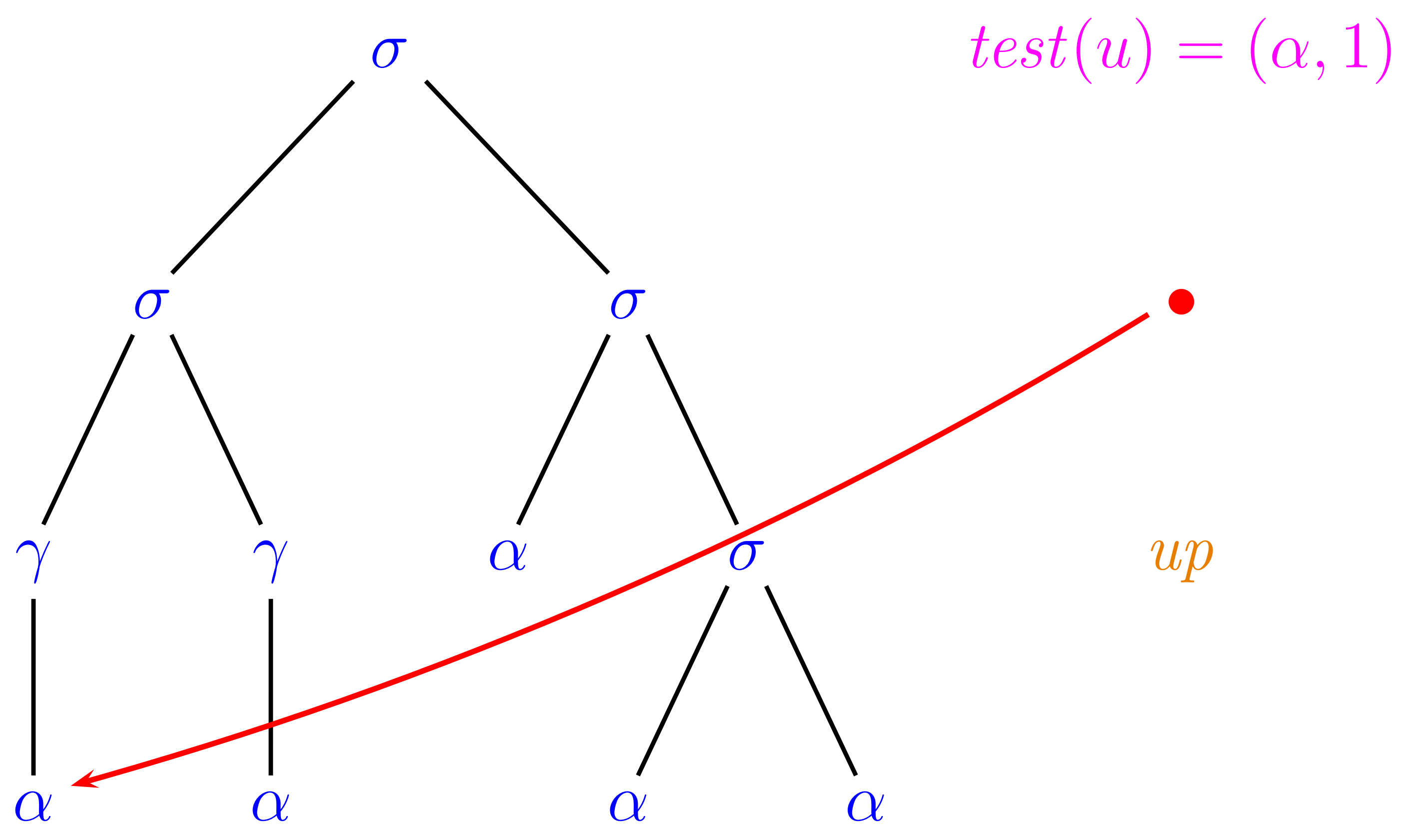
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



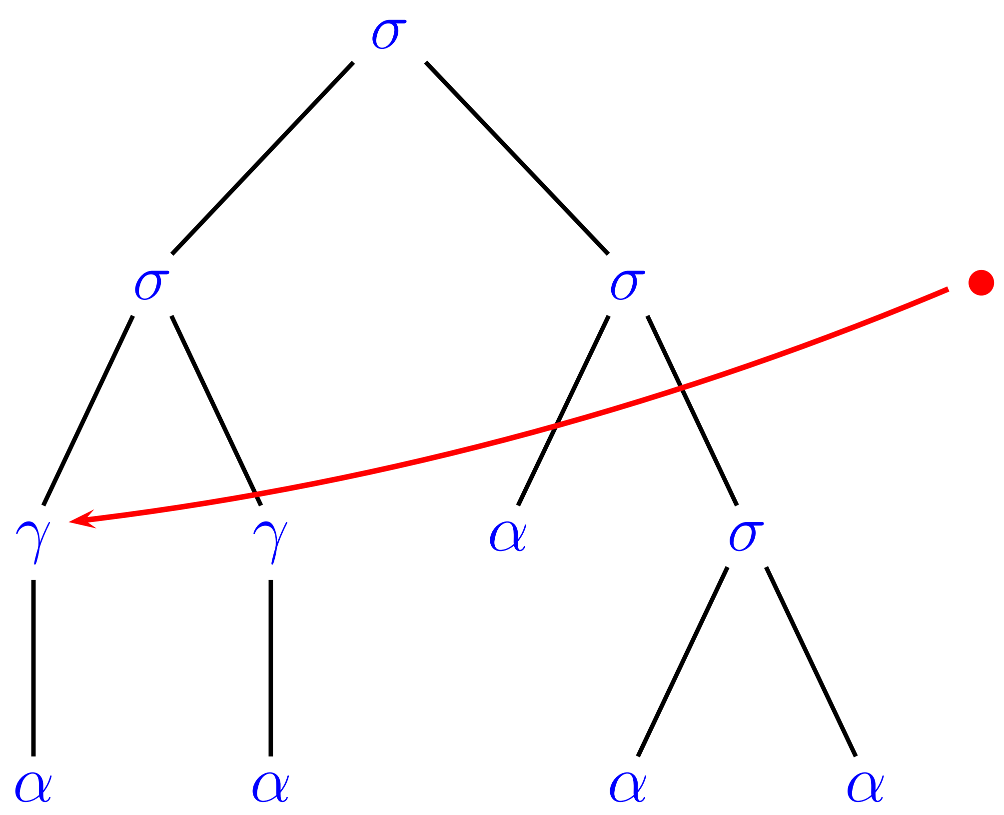
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



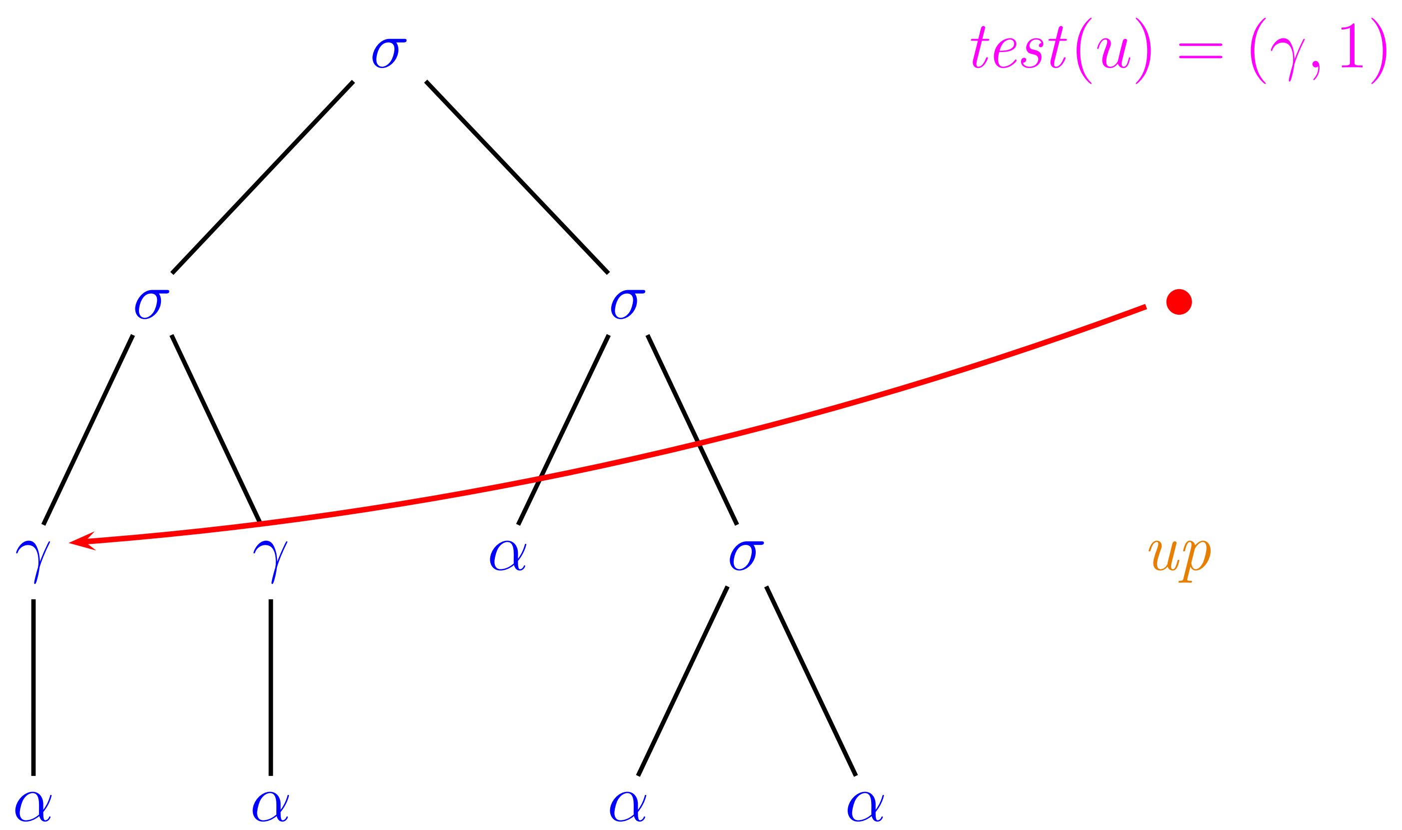
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



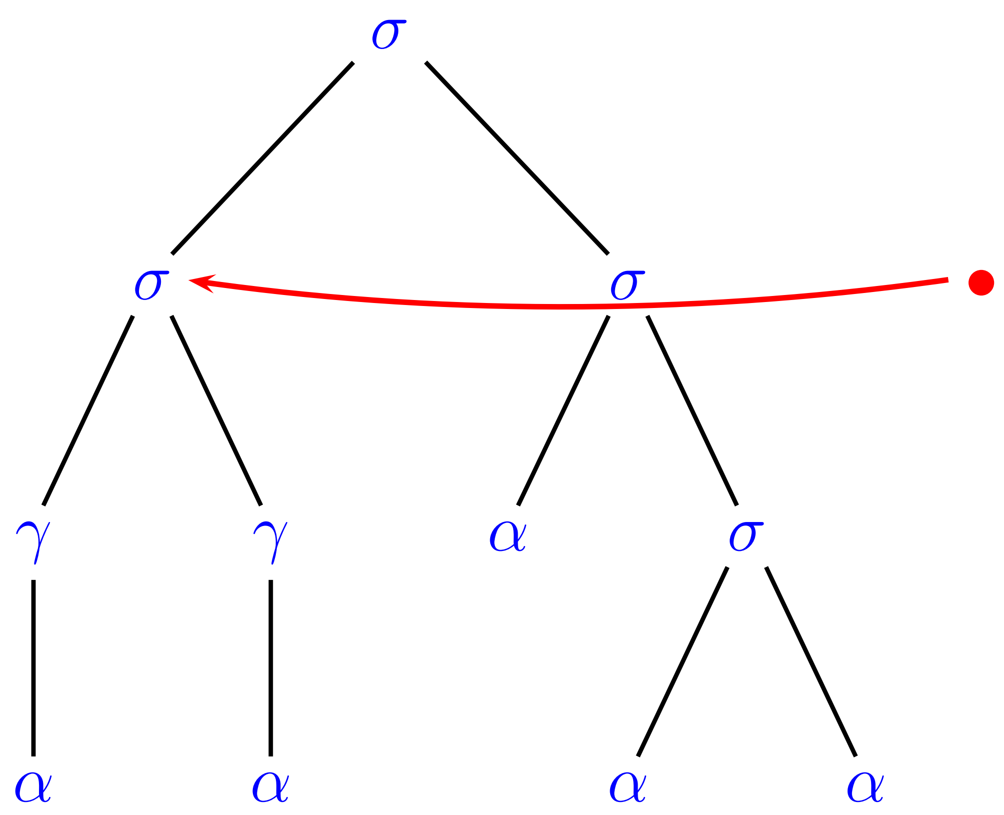
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



Fabejárás

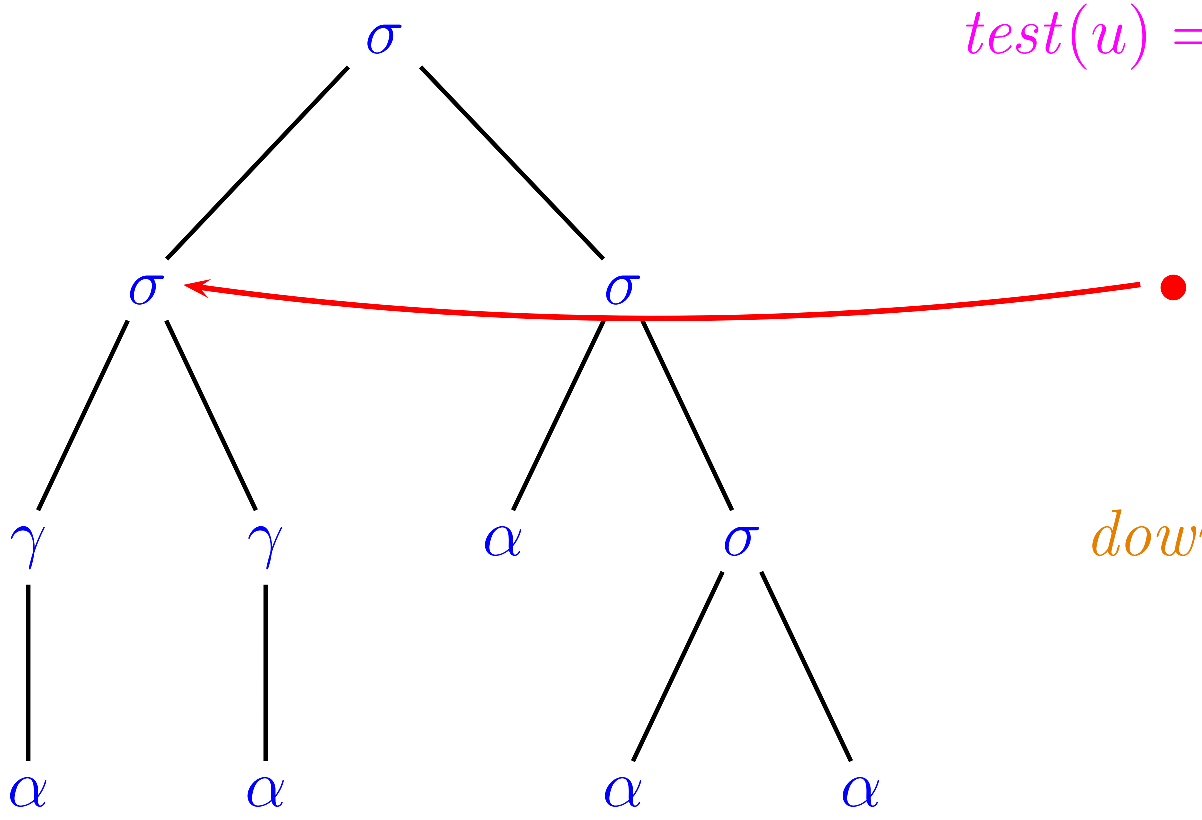
utasítások: $\{stay, up, down_1, \ldots, down_d\}$



Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$

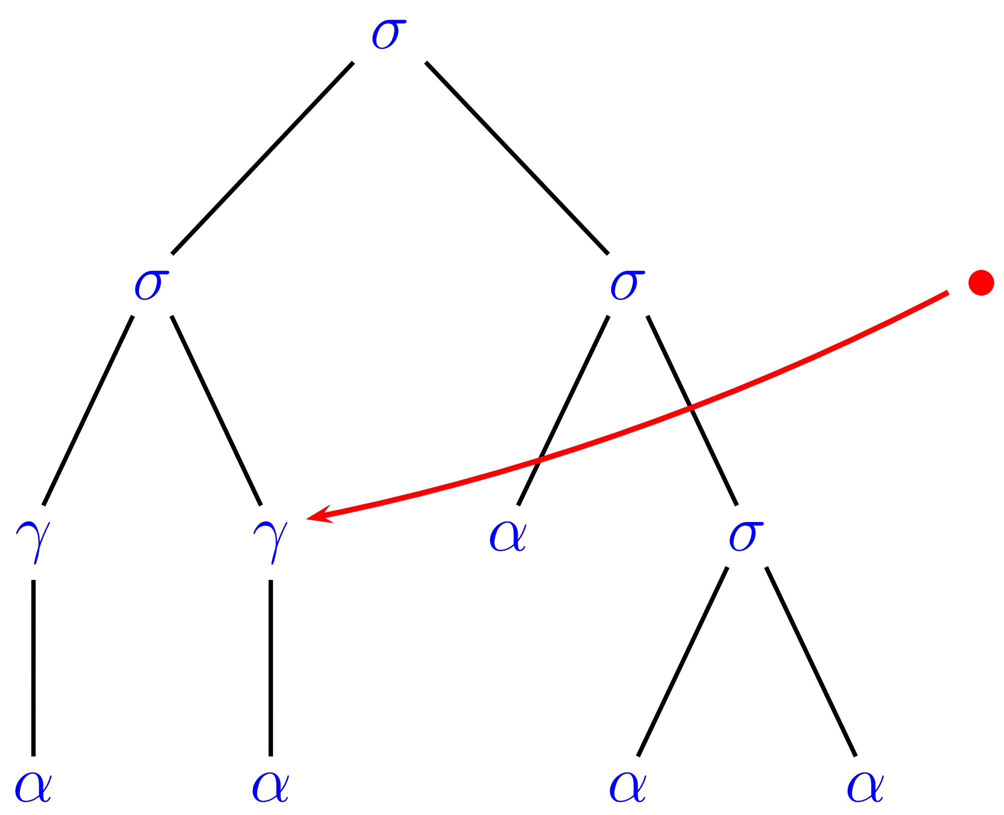
$test(u) = (\sigma, 1)$



$down_2$

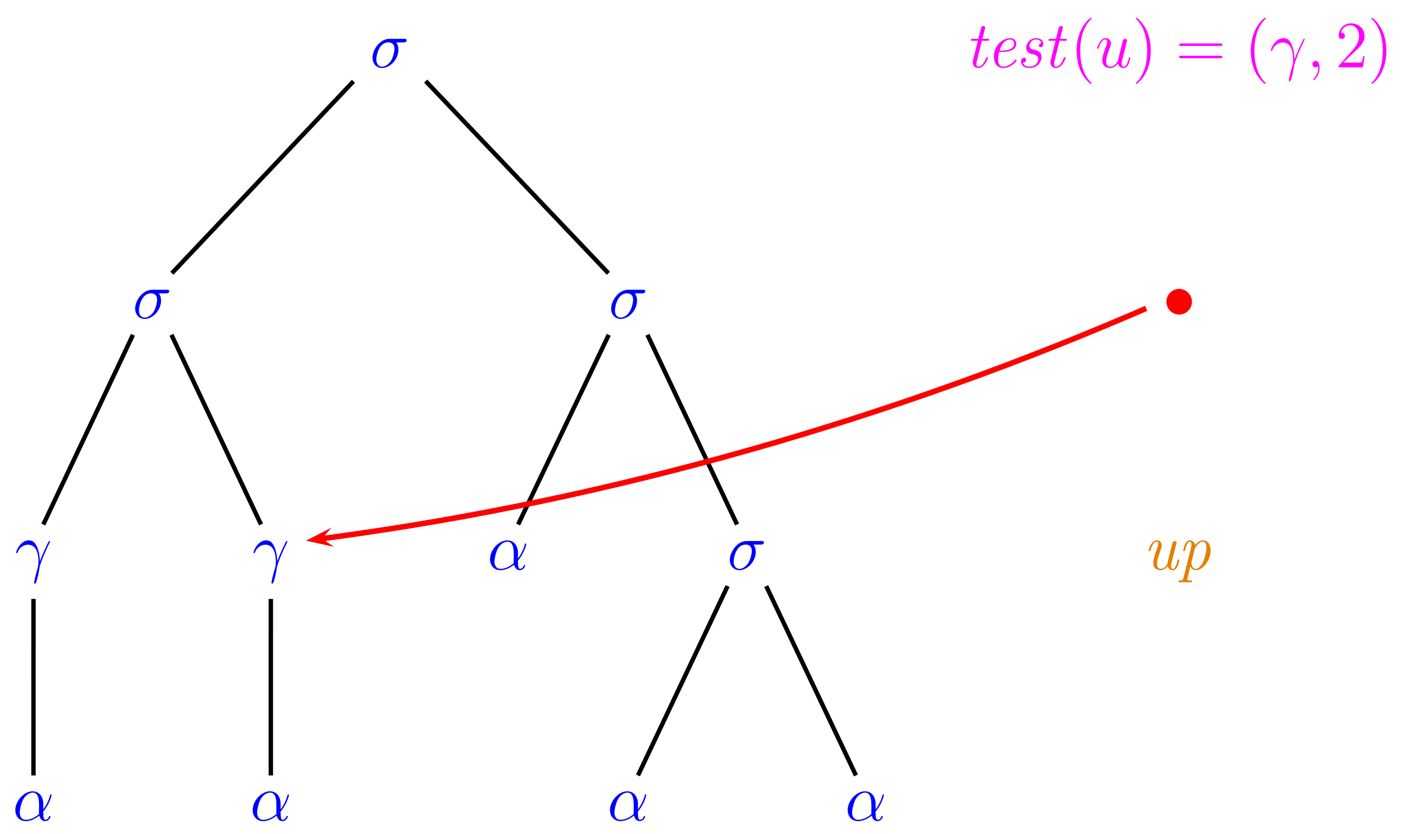
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



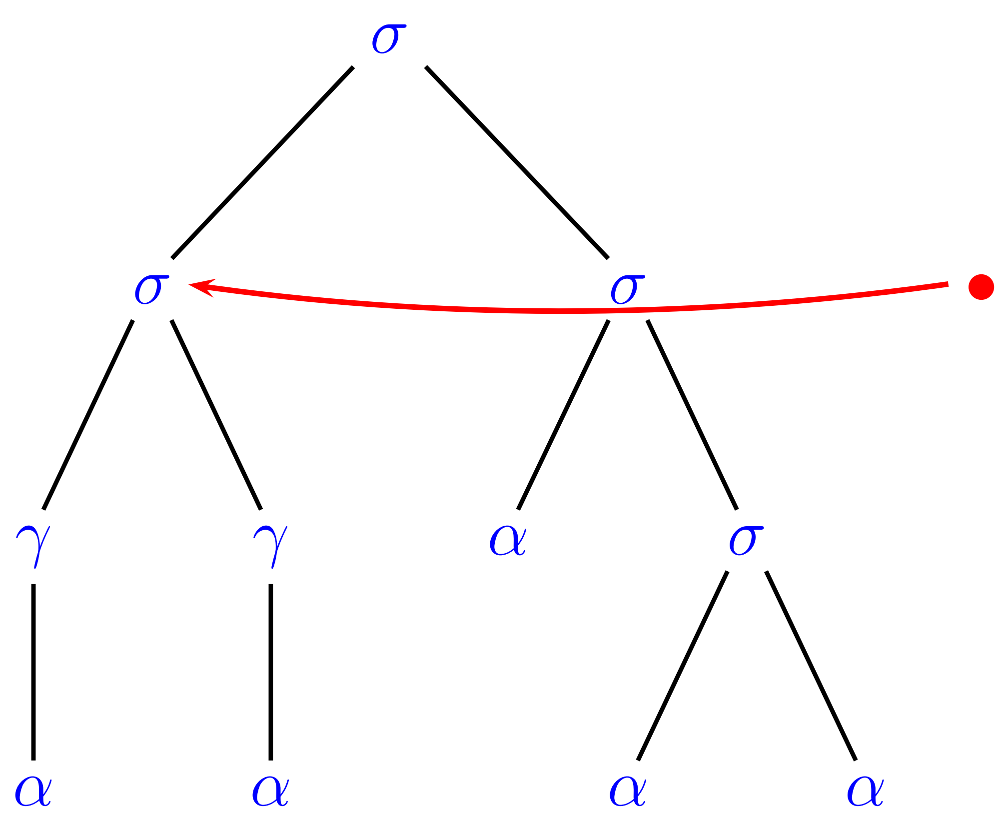
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



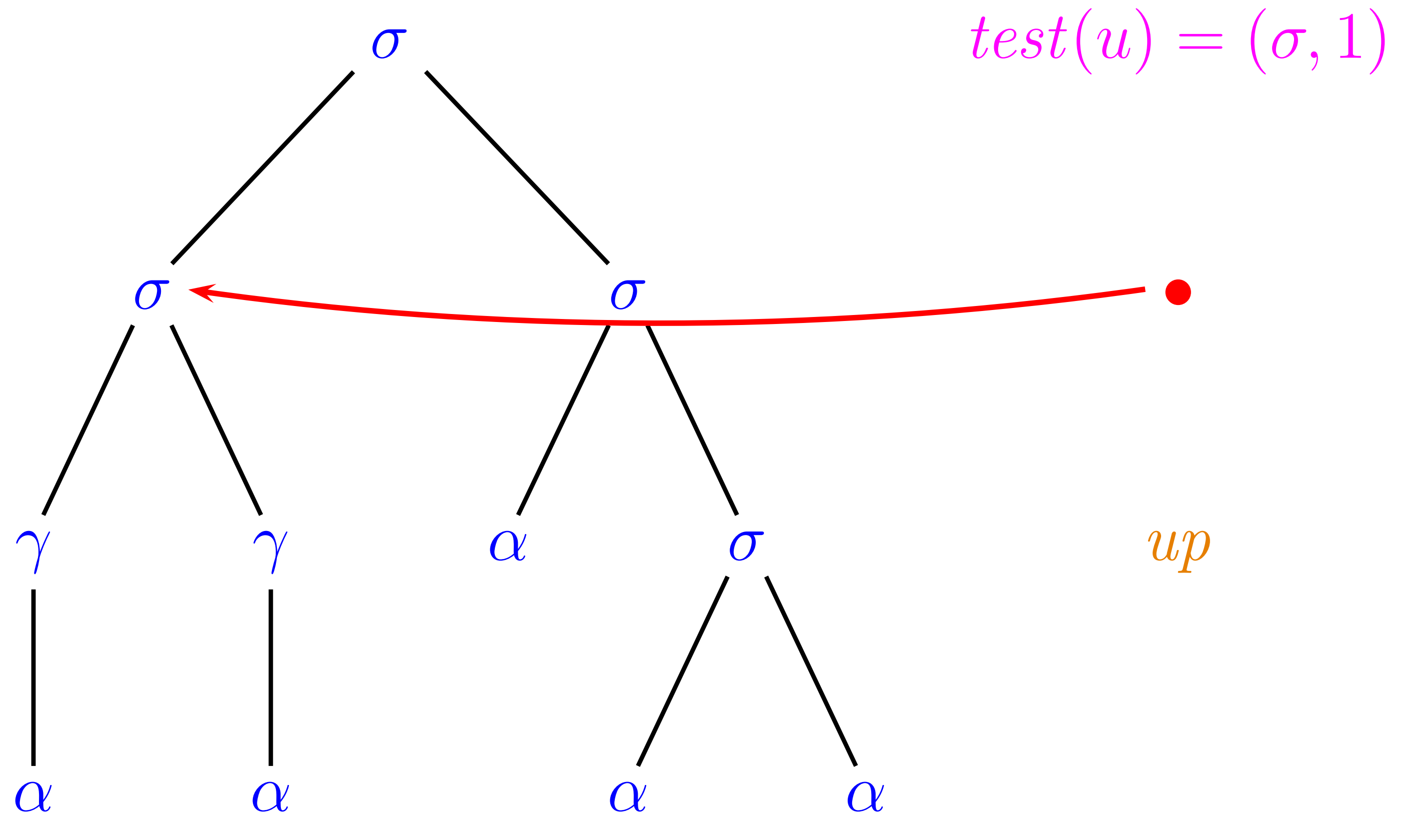
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



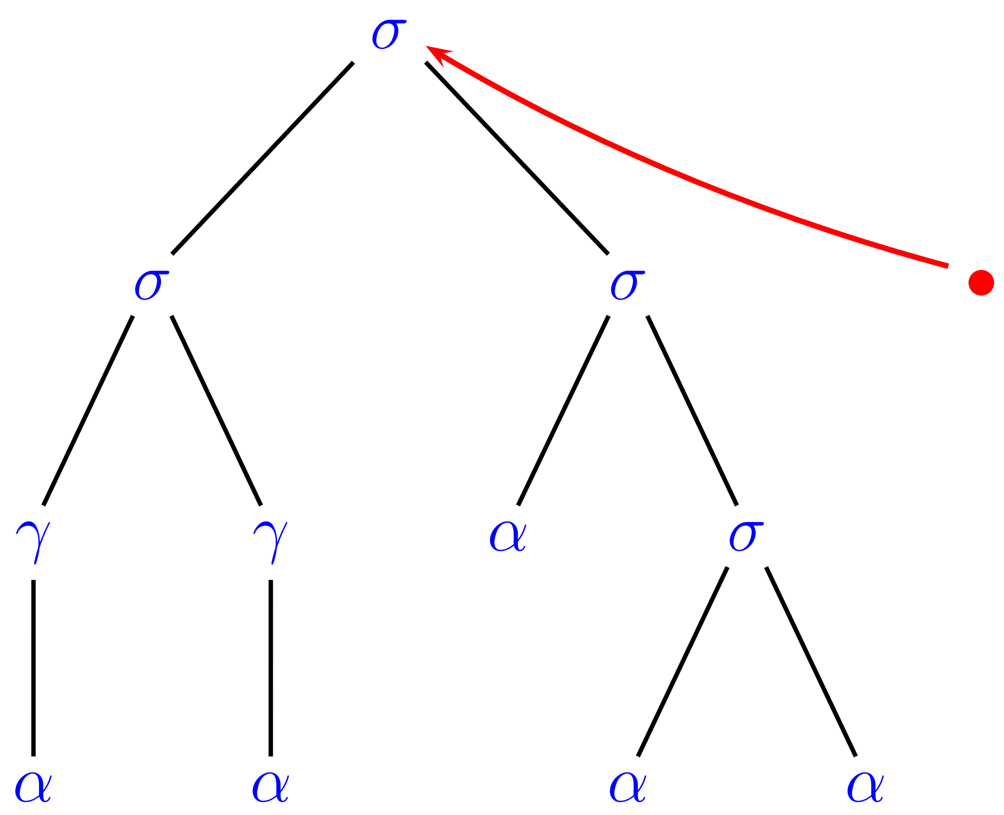
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



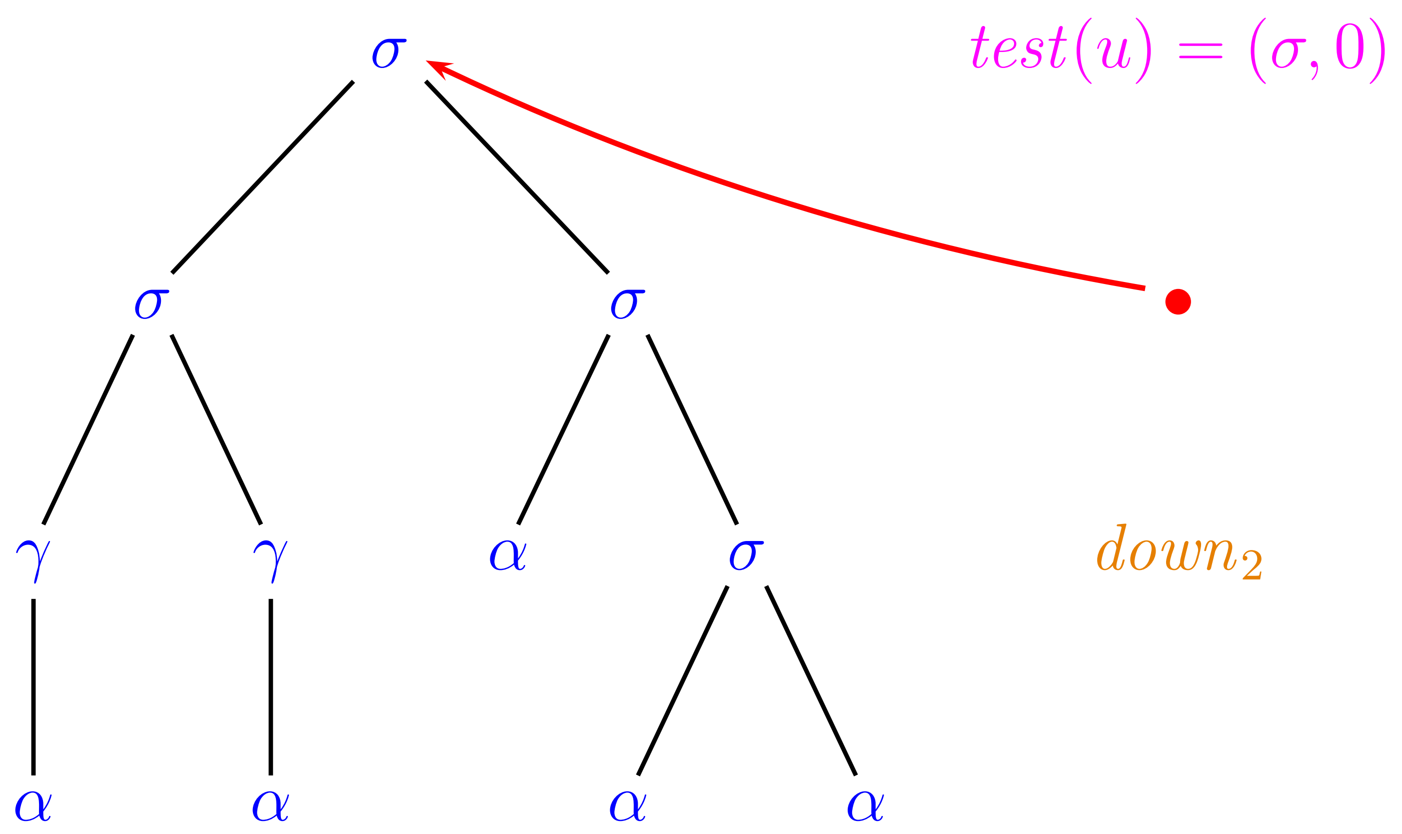
Fabejárás

utasítások: ${stay, up, down_1, \ldots, down_d}$



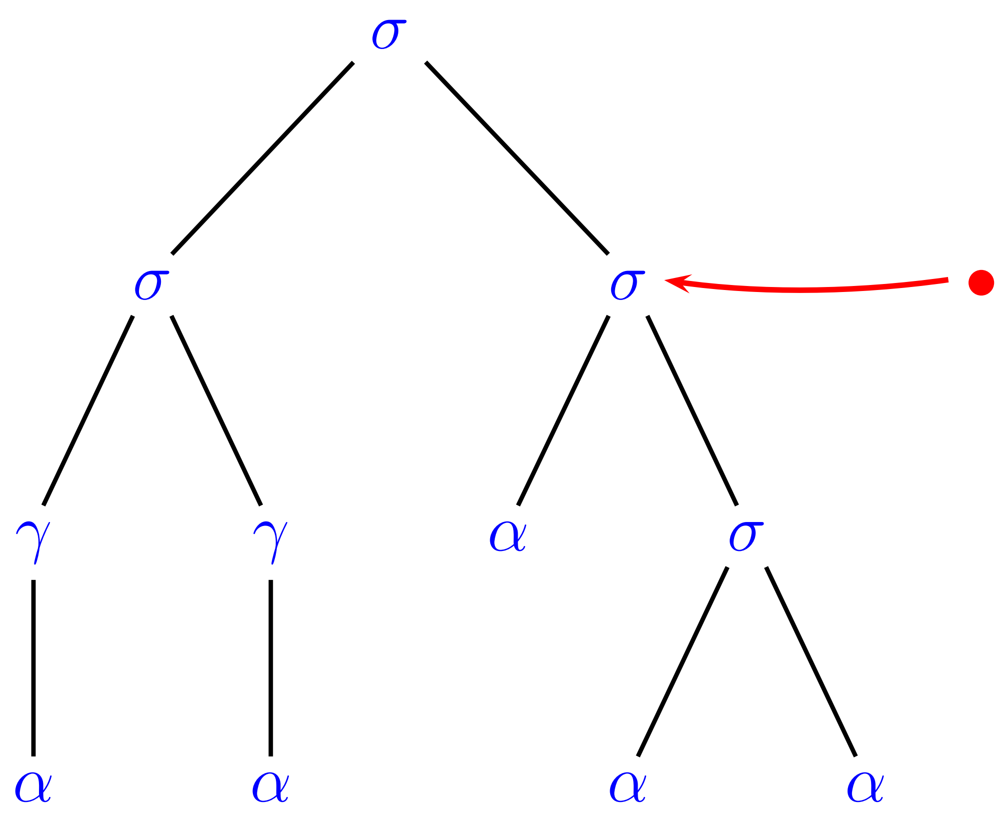
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



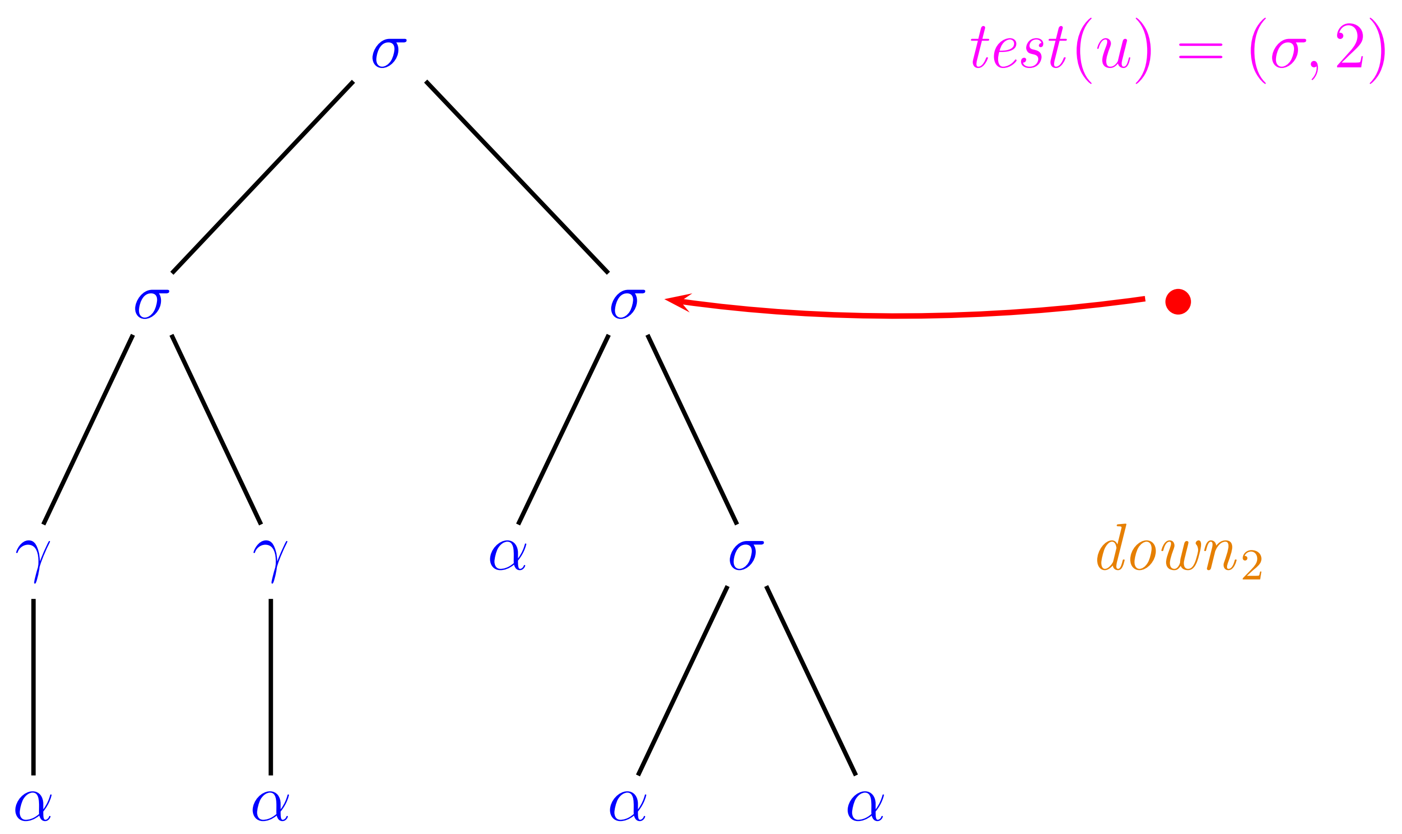
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



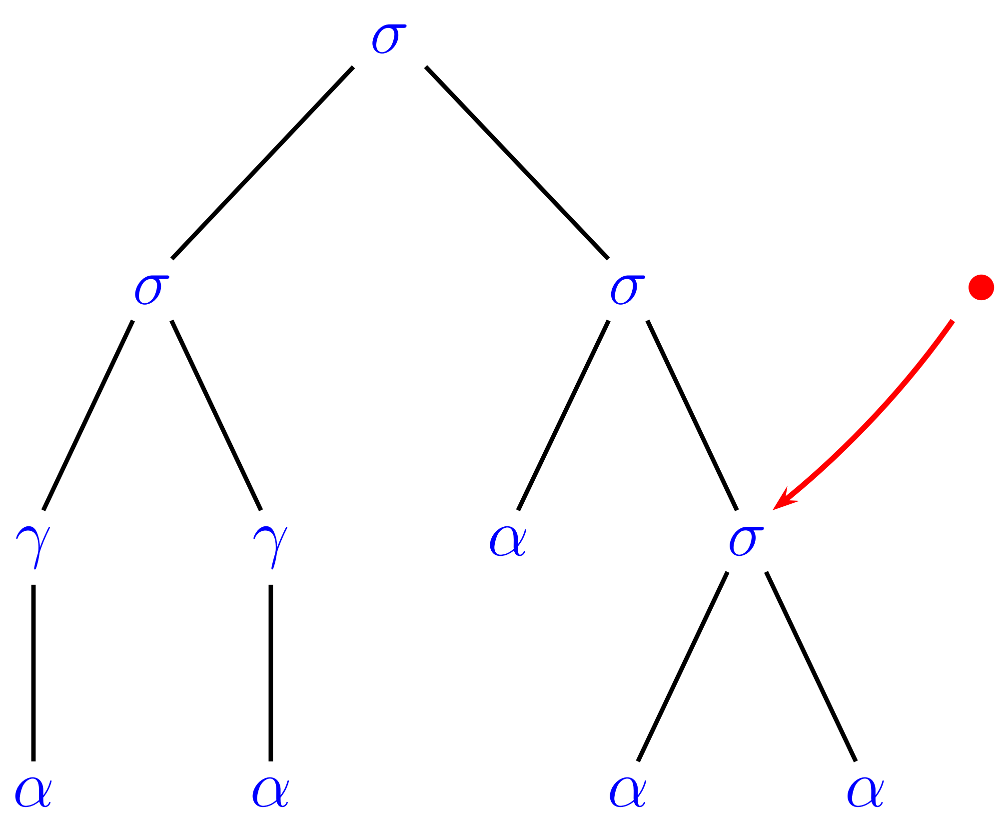
Fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d\}$



Fabejárás

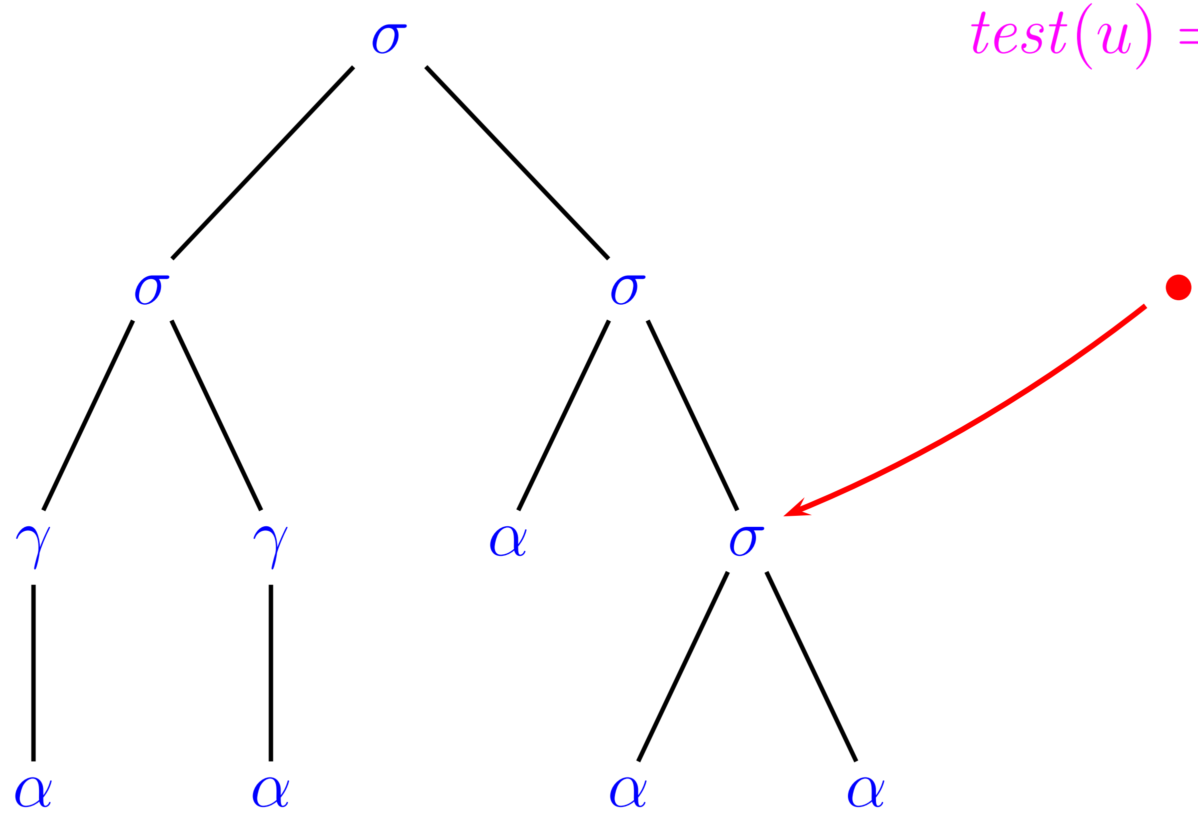
utasítások: $\{stay, up, down_1, \ldots, down_d\}$



Fabejárás

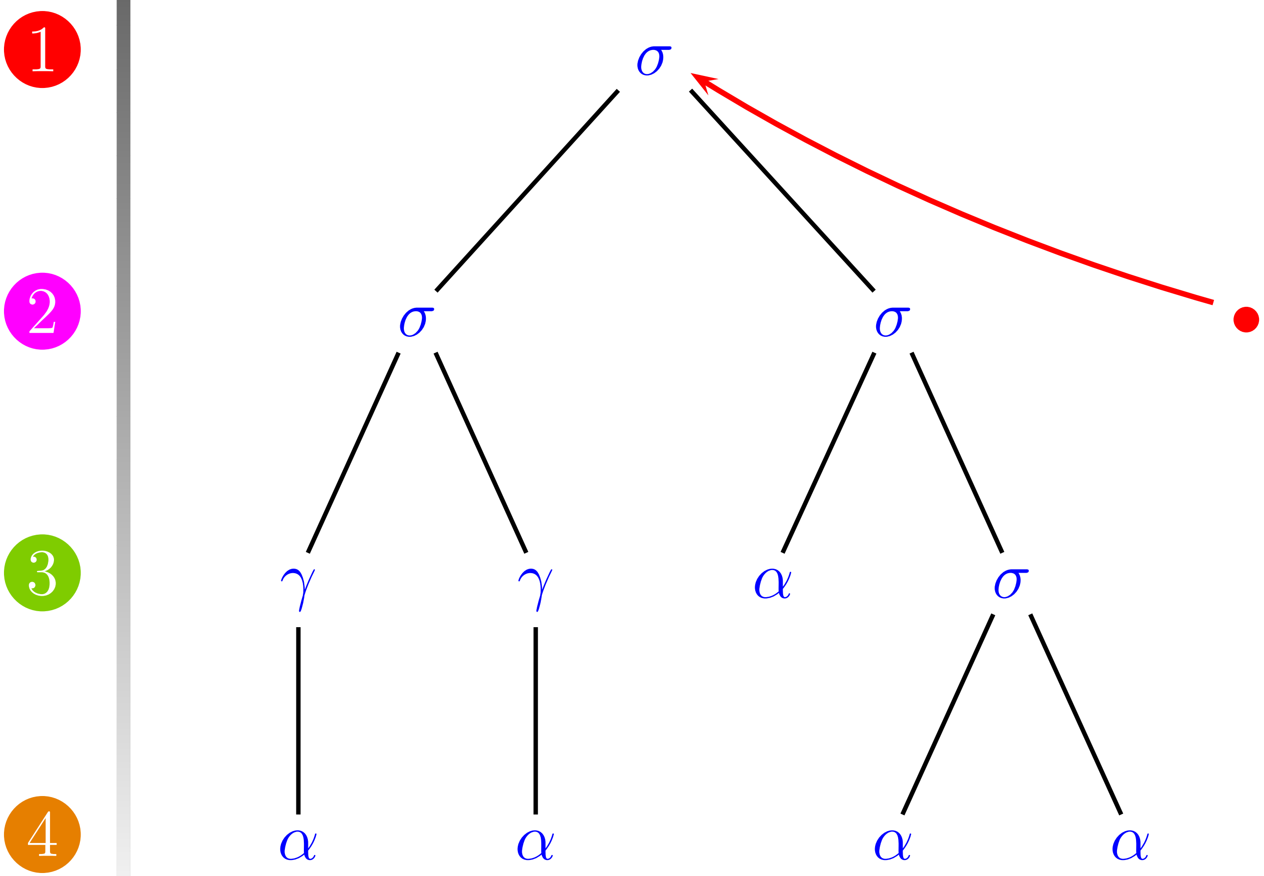
utasítások: $\{stay, up, down_1, \ldots, down_d\}$

$test(u) = (\sigma, 2)$



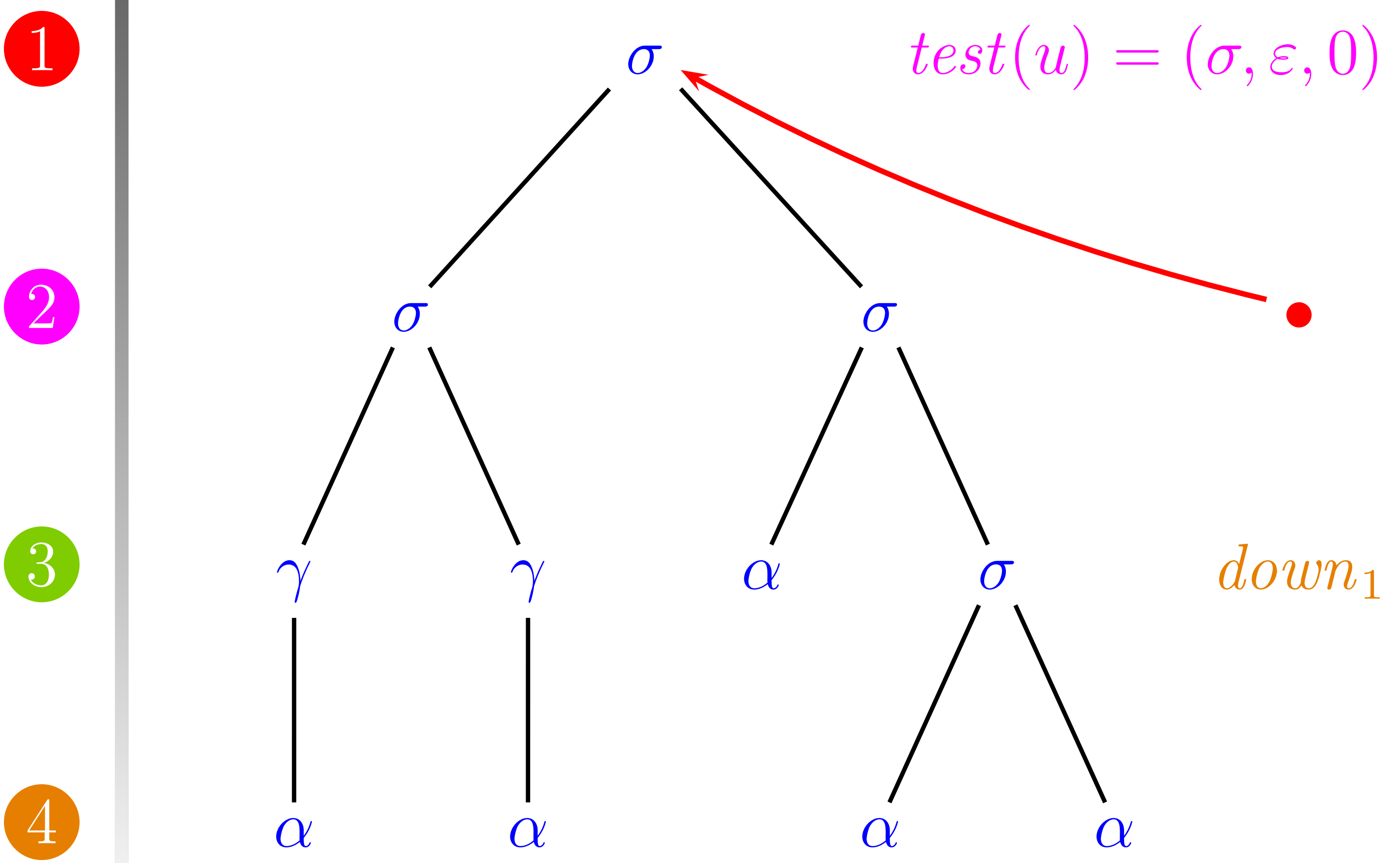
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



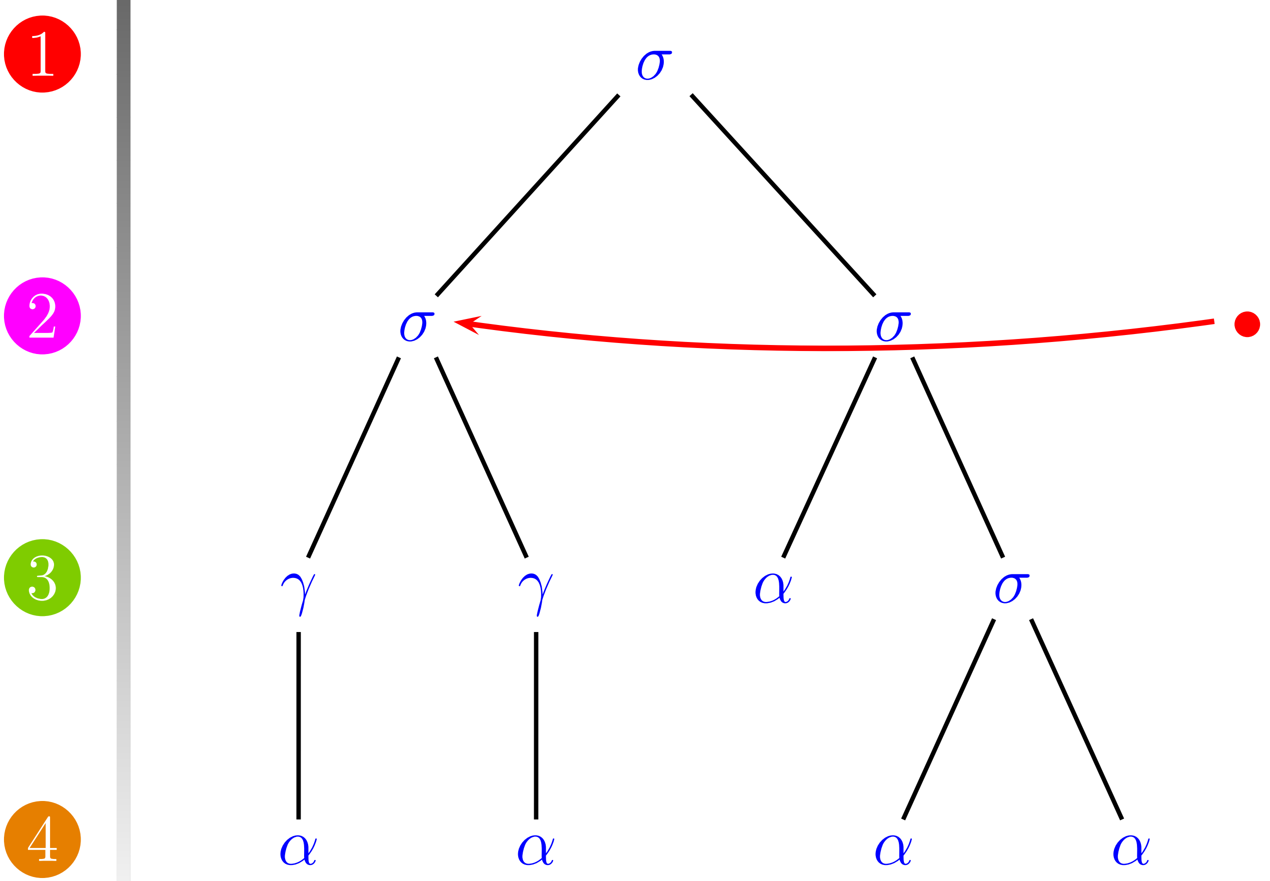
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$

1

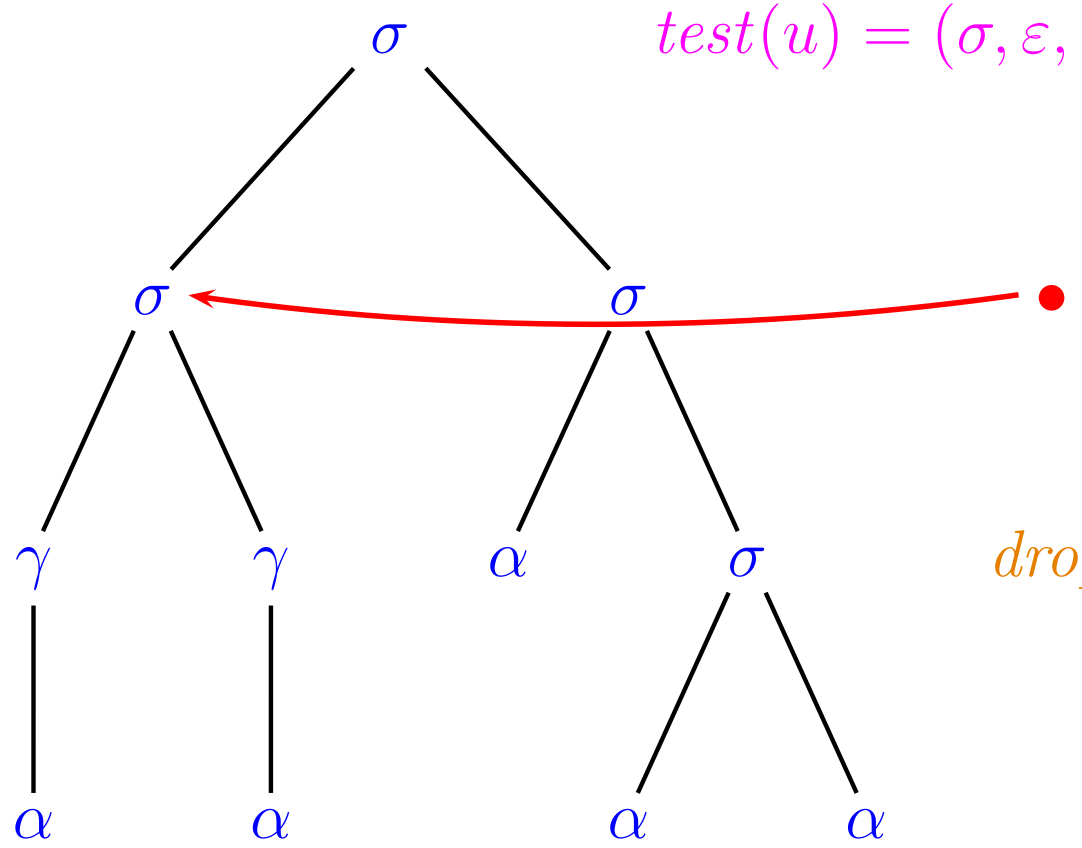
$test(u) = (\sigma, \varepsilon, 1)$

2

3

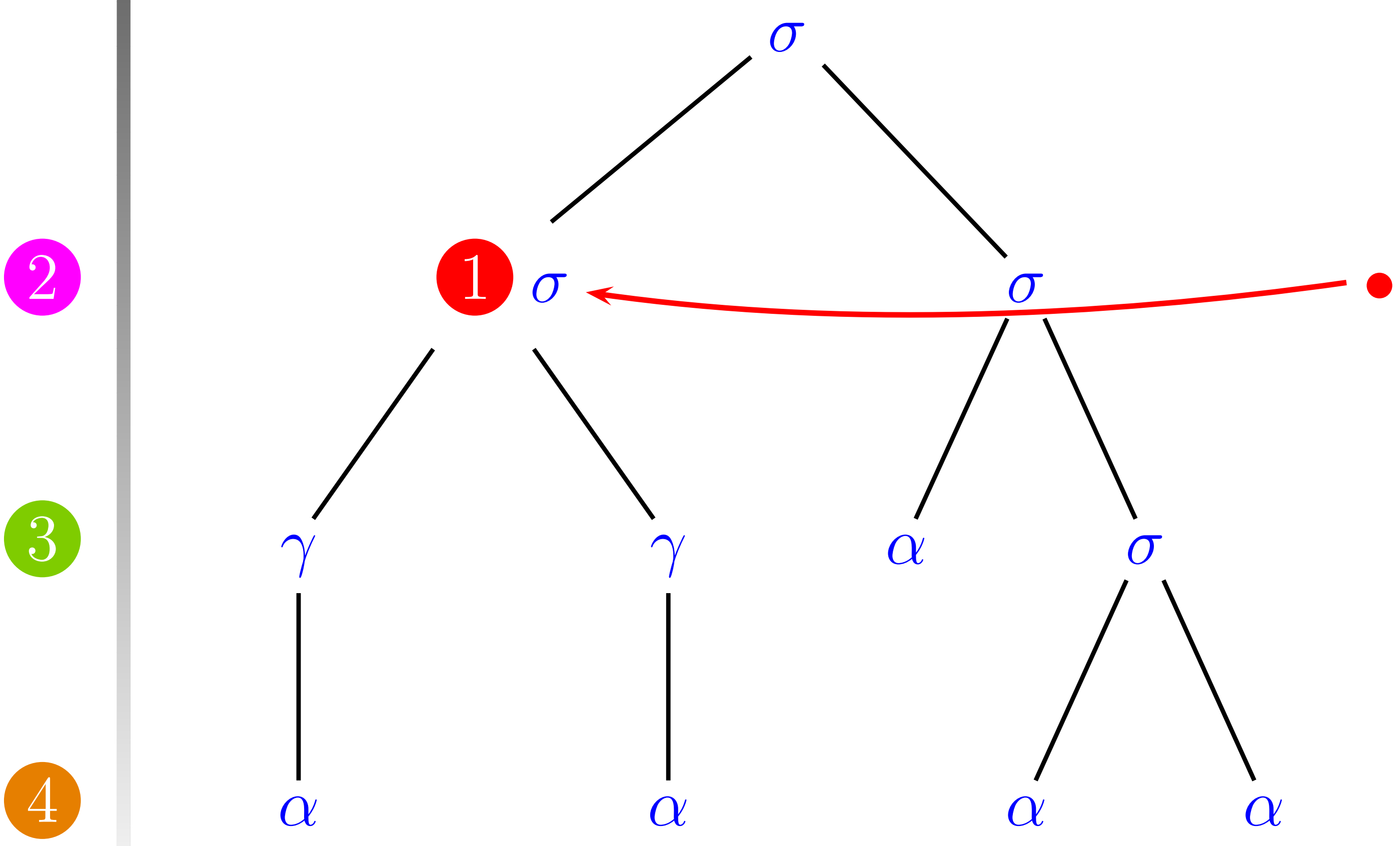
$drop$

4



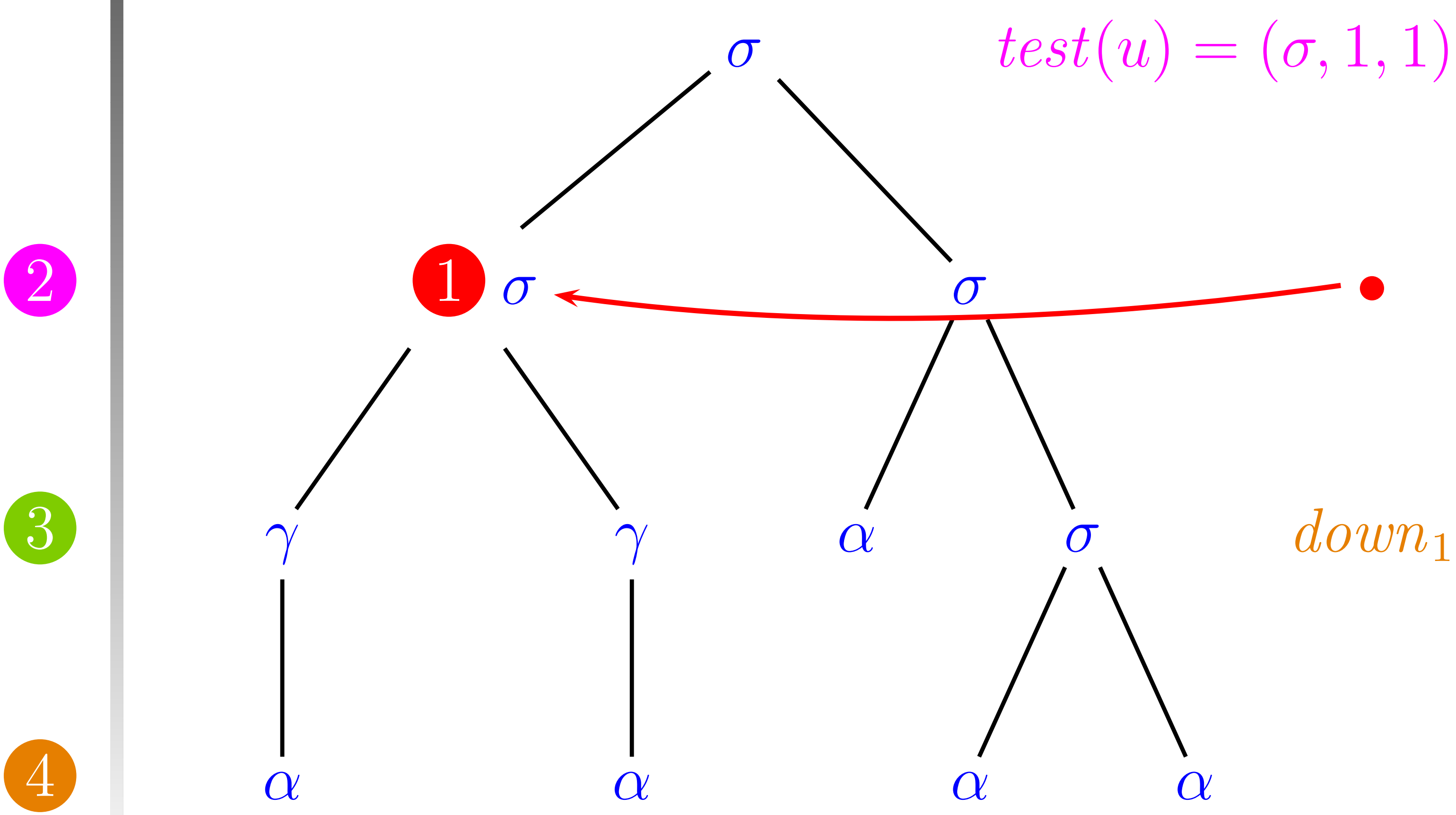
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



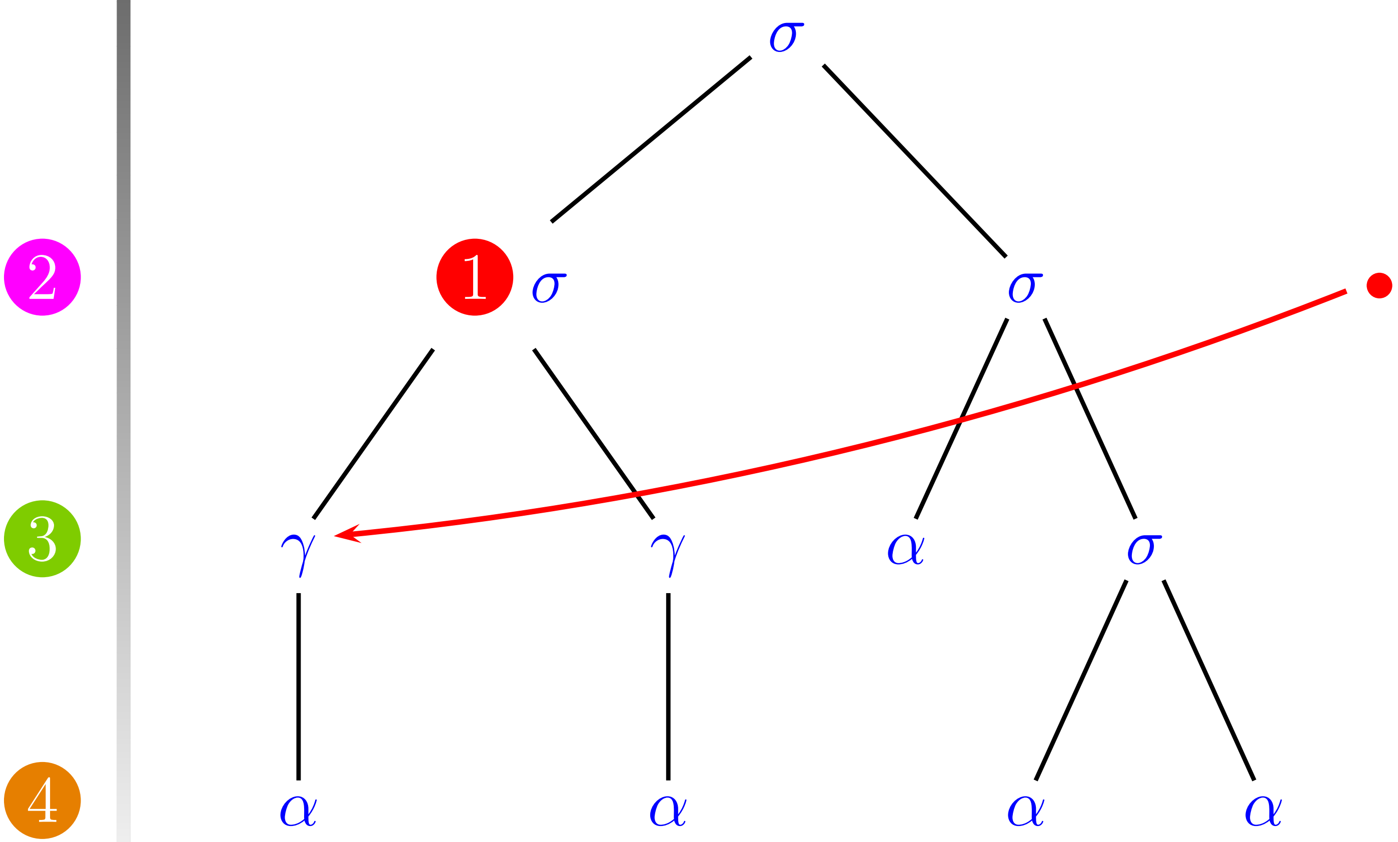
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



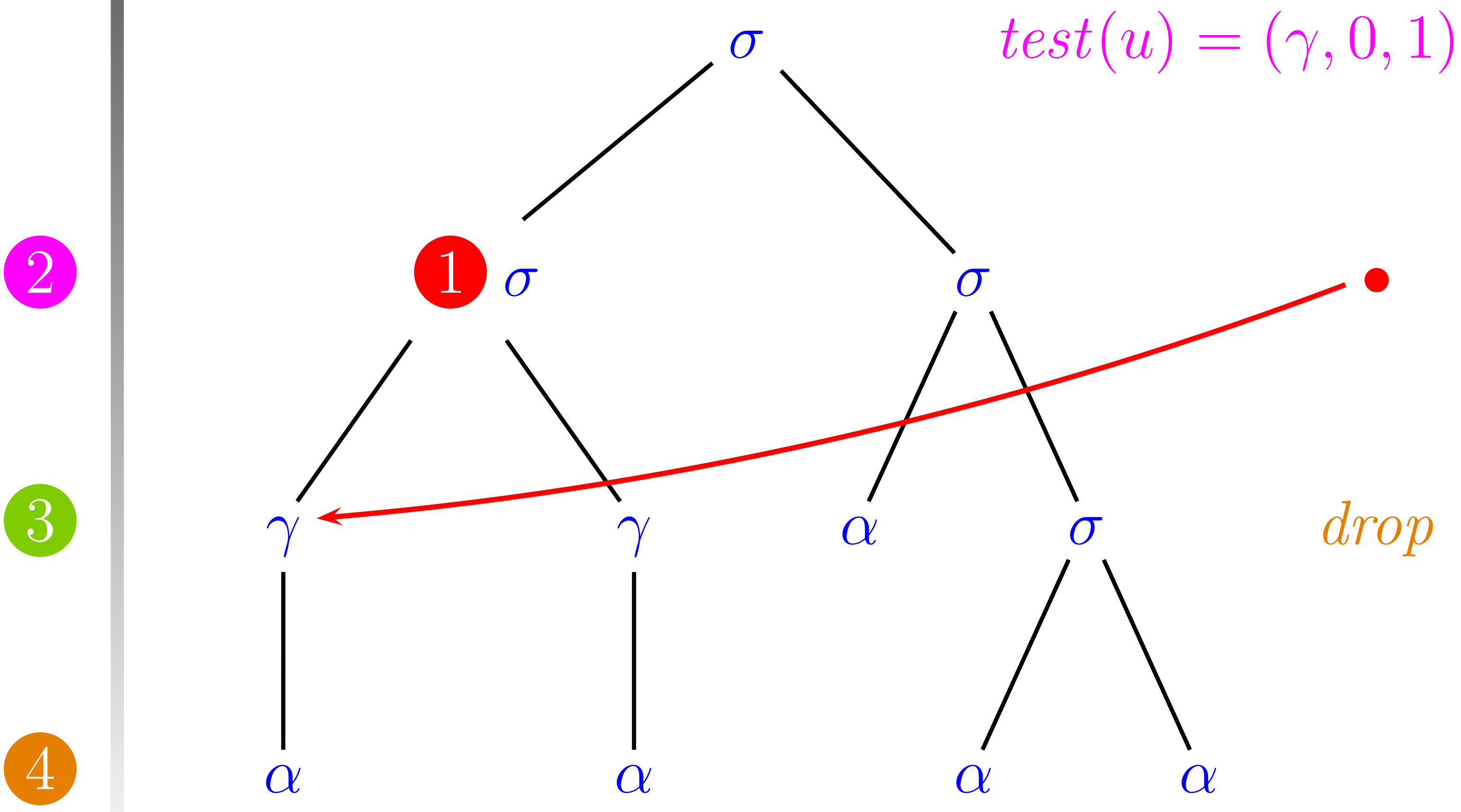
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



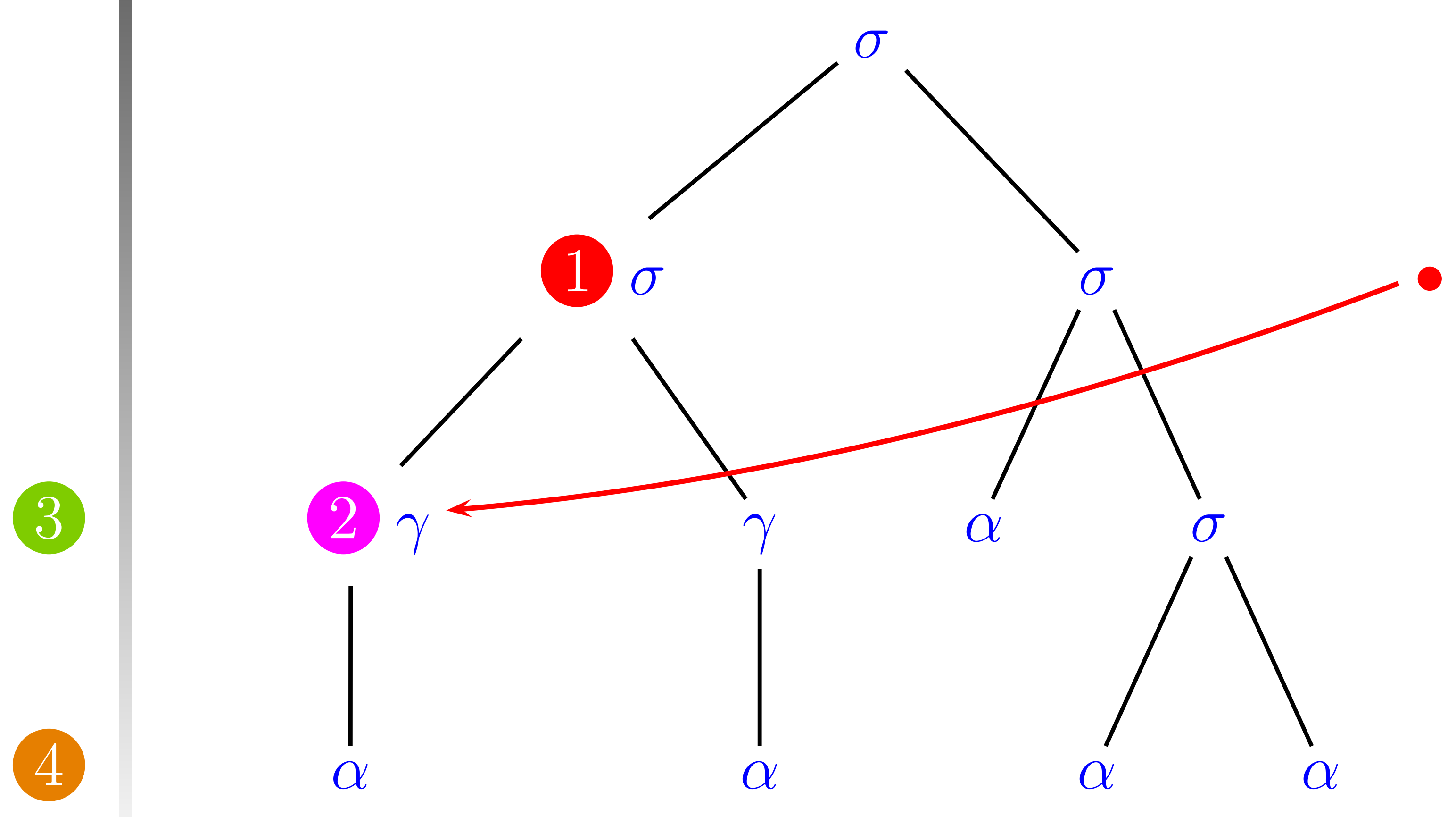
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



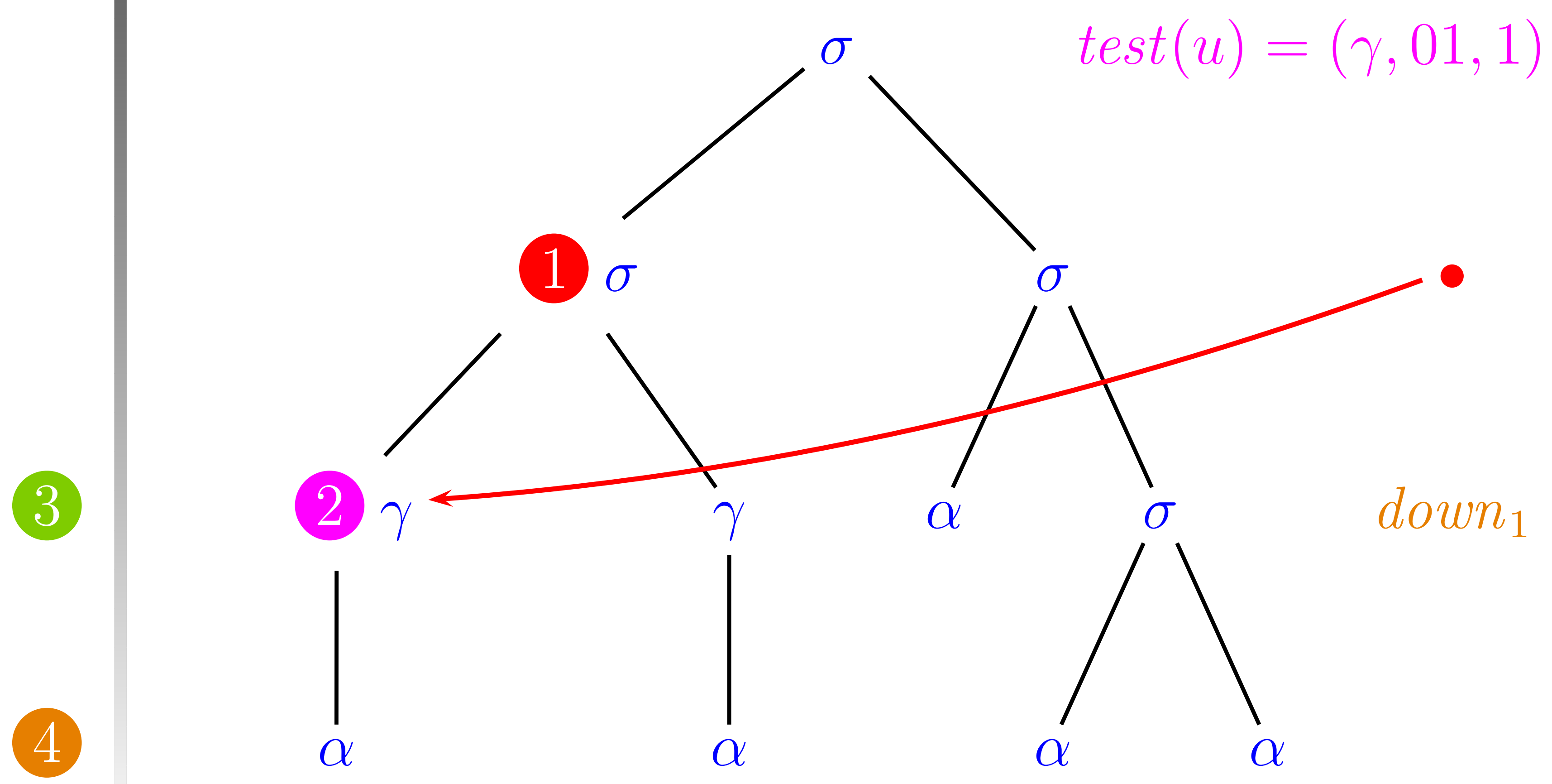
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



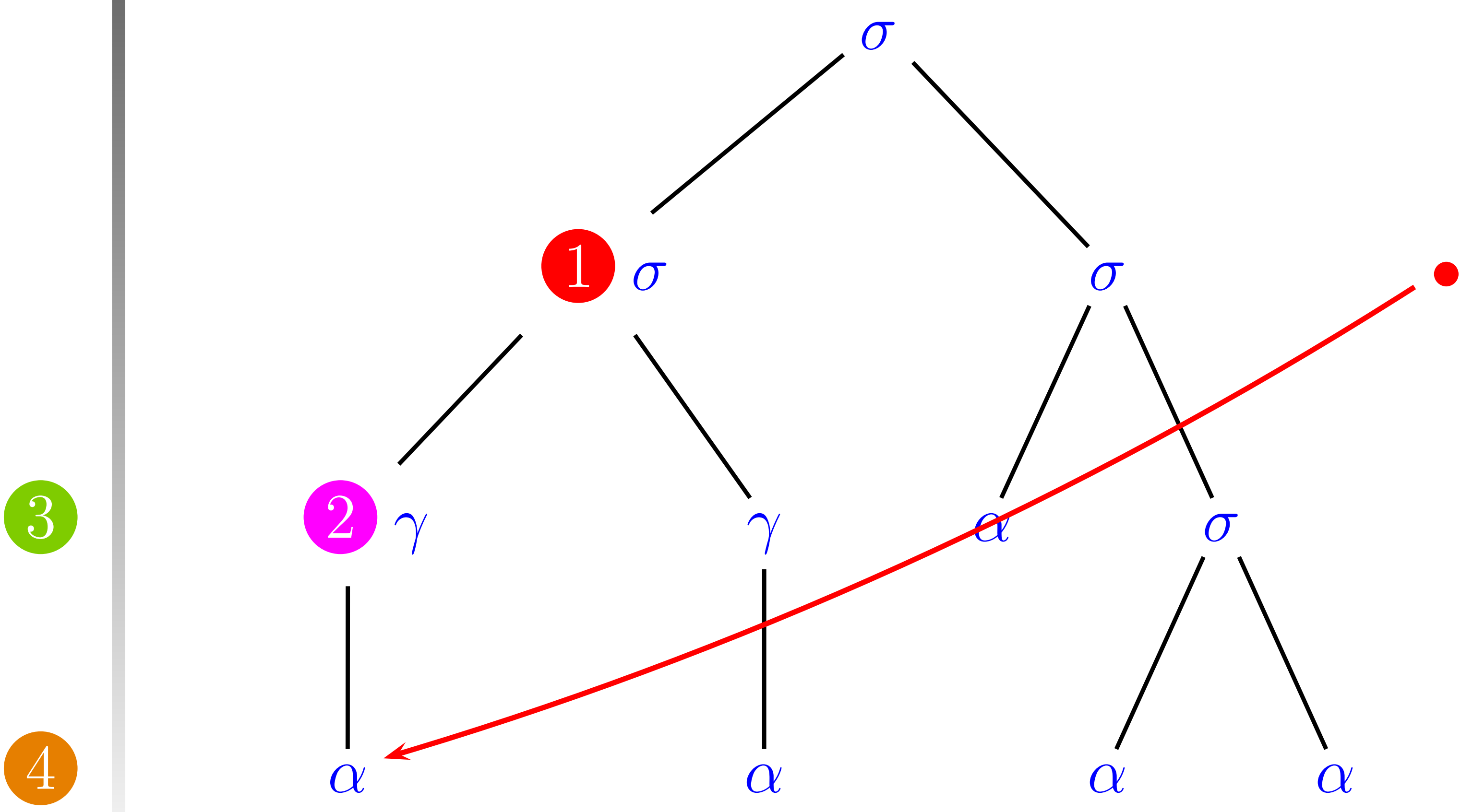
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



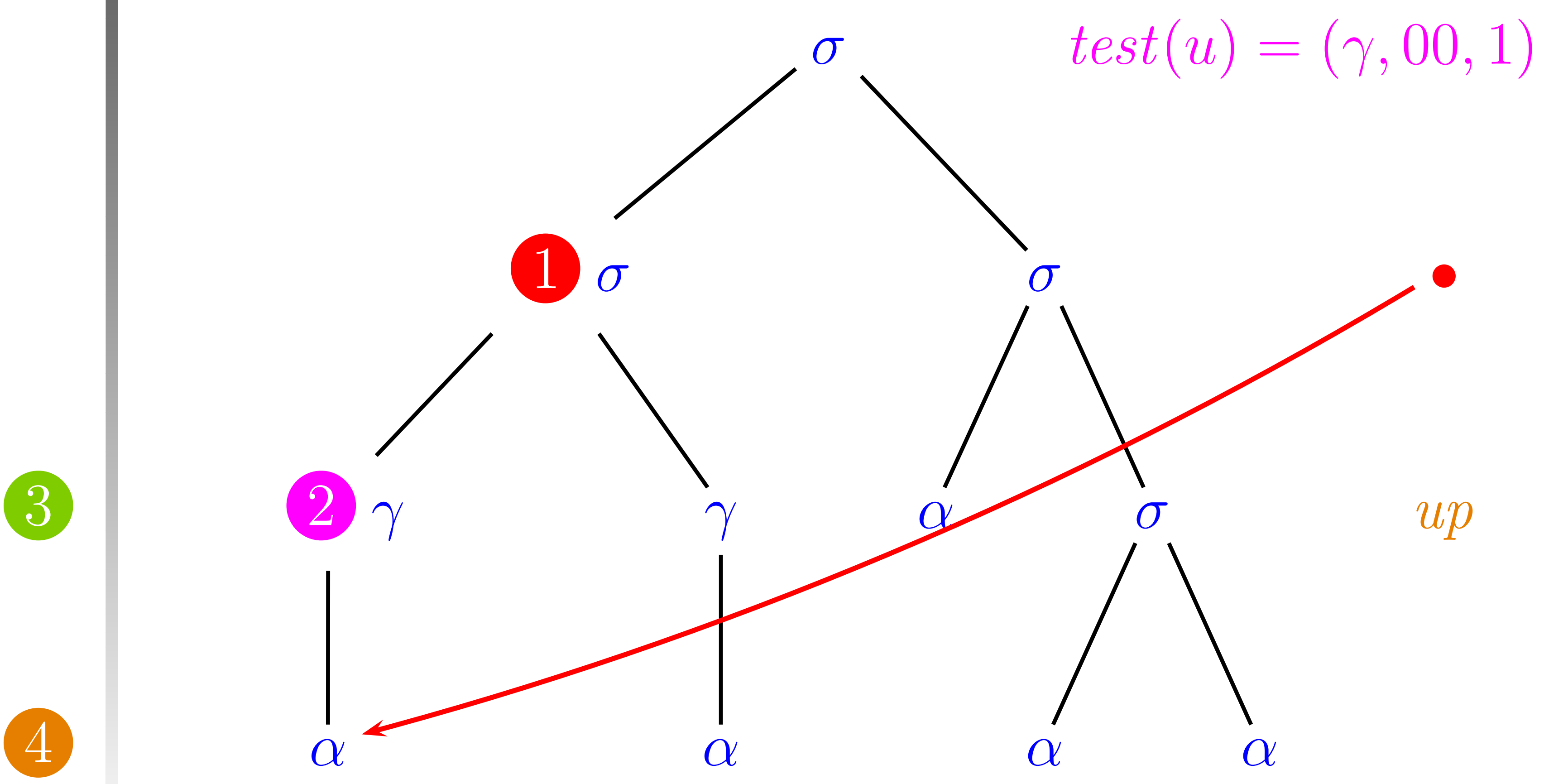
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



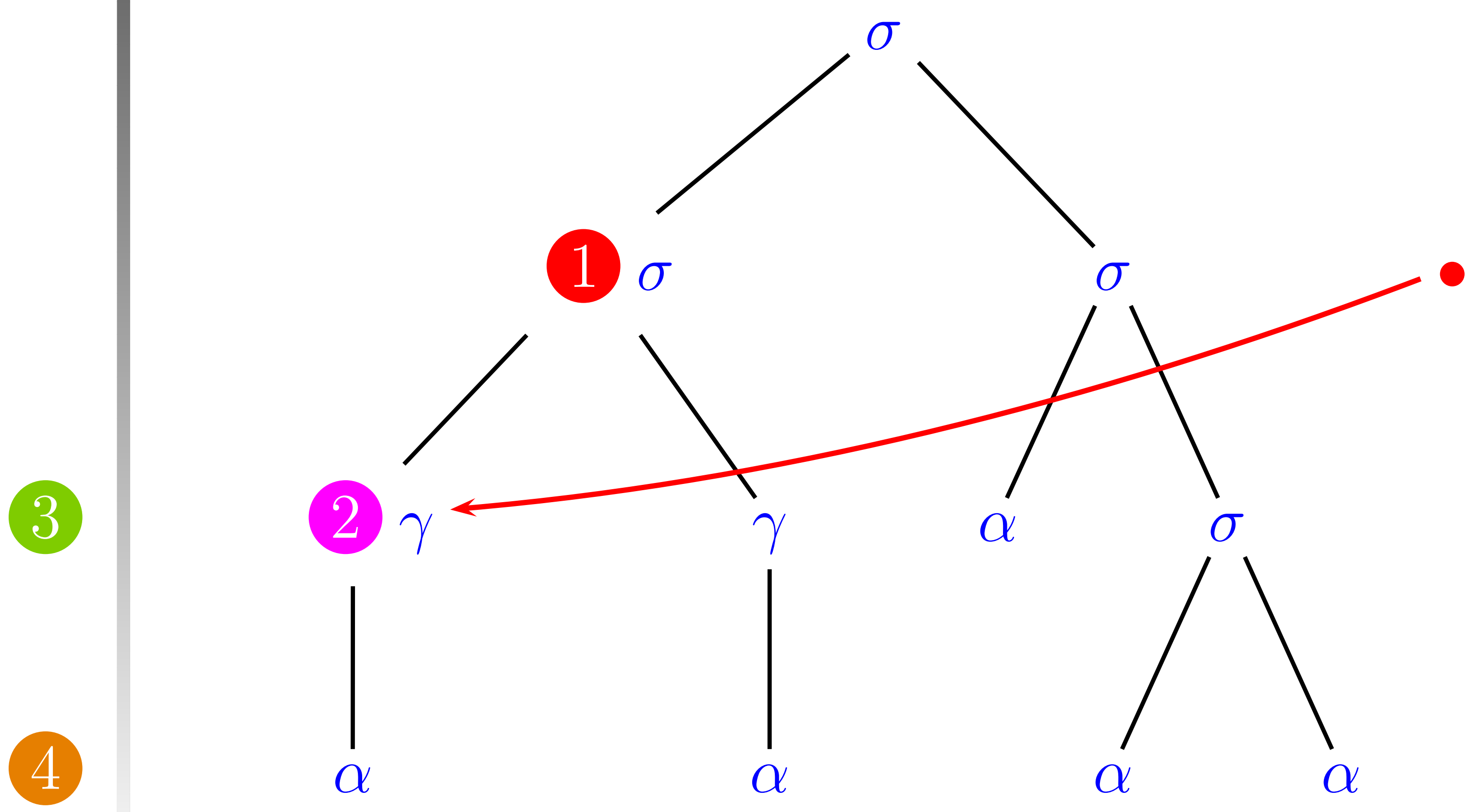
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



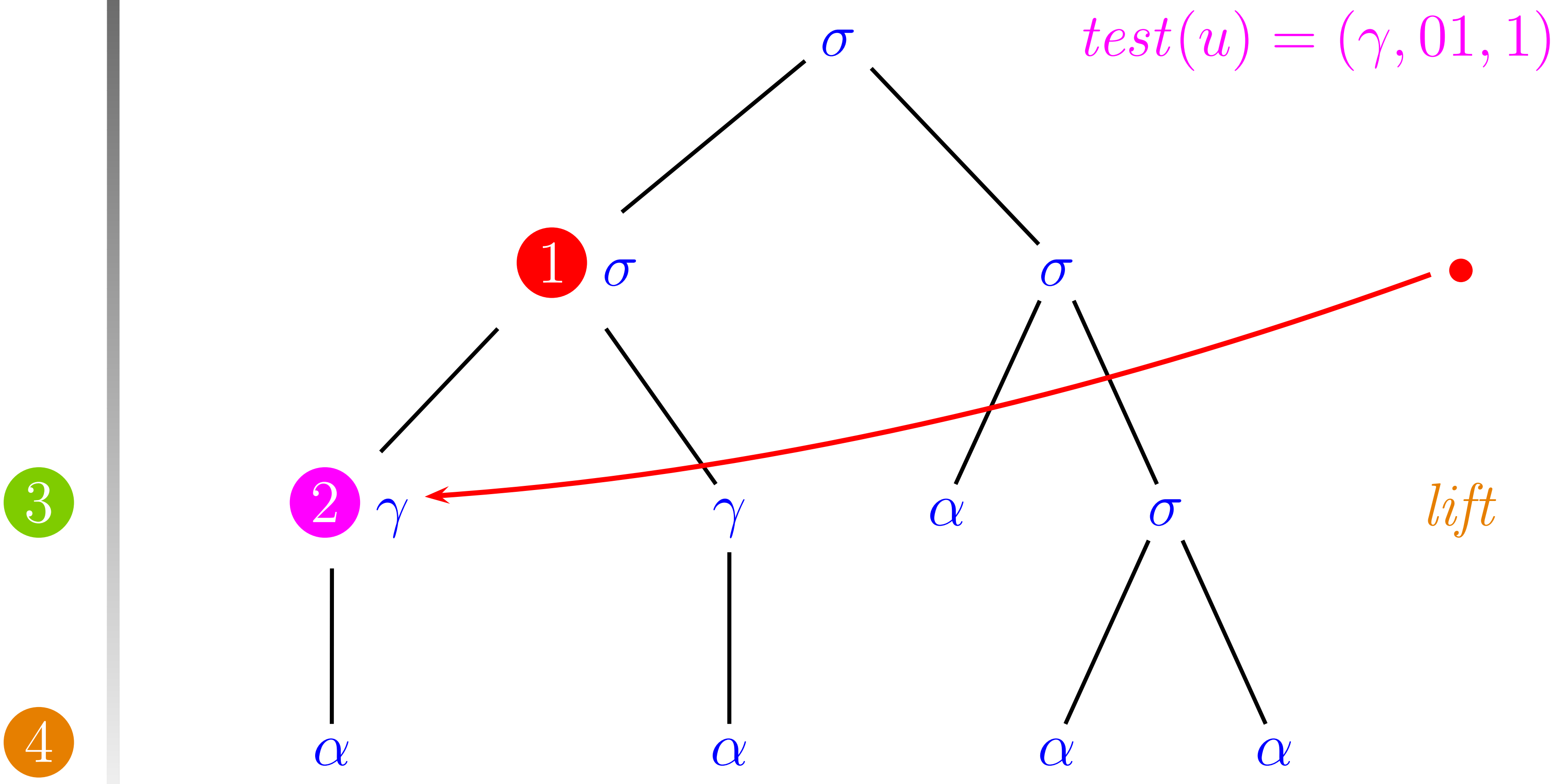
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



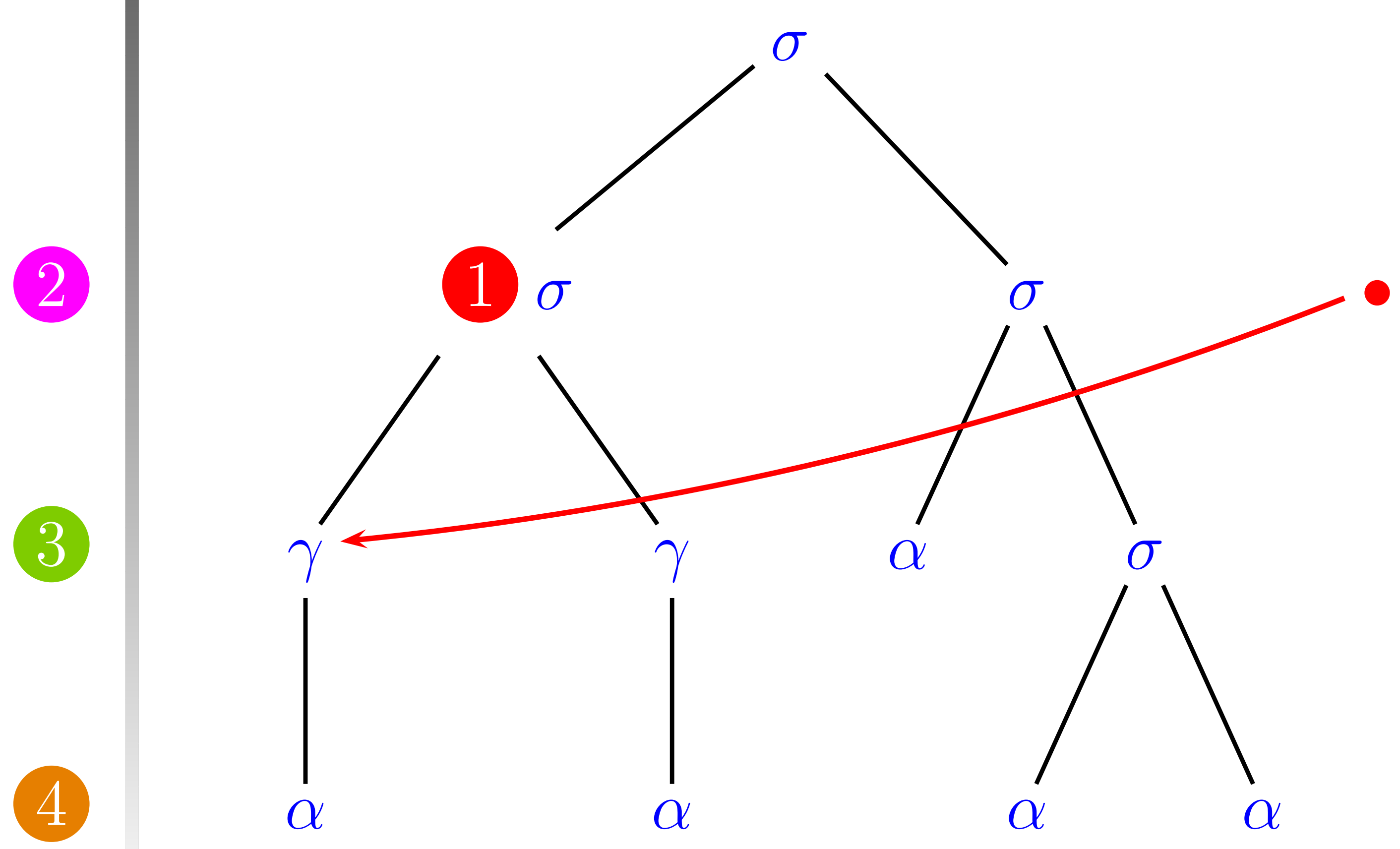
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



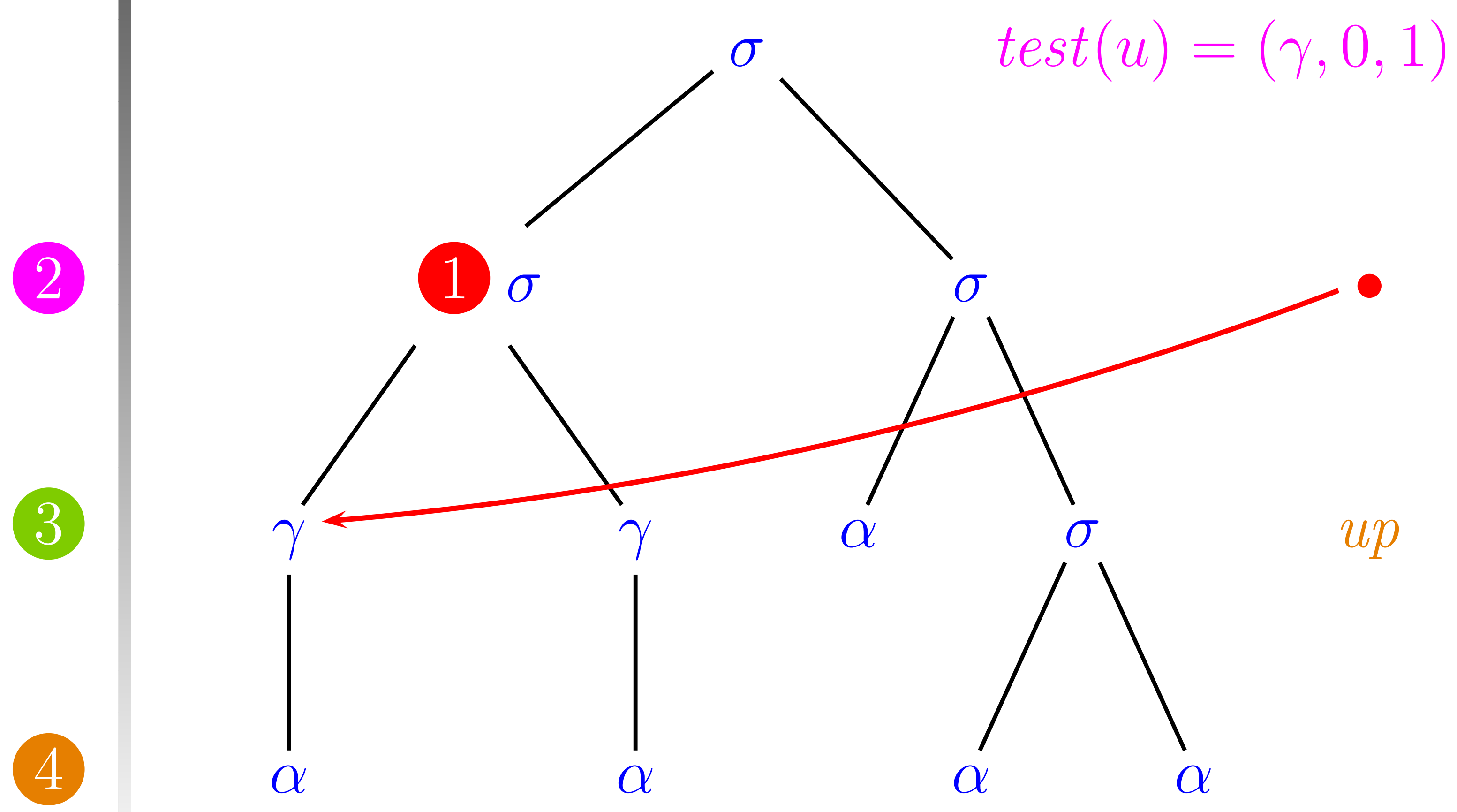
Kavics fabejárás

utasítások: ${stay, up, down_1, \ldots, down_d, drop, lift}$



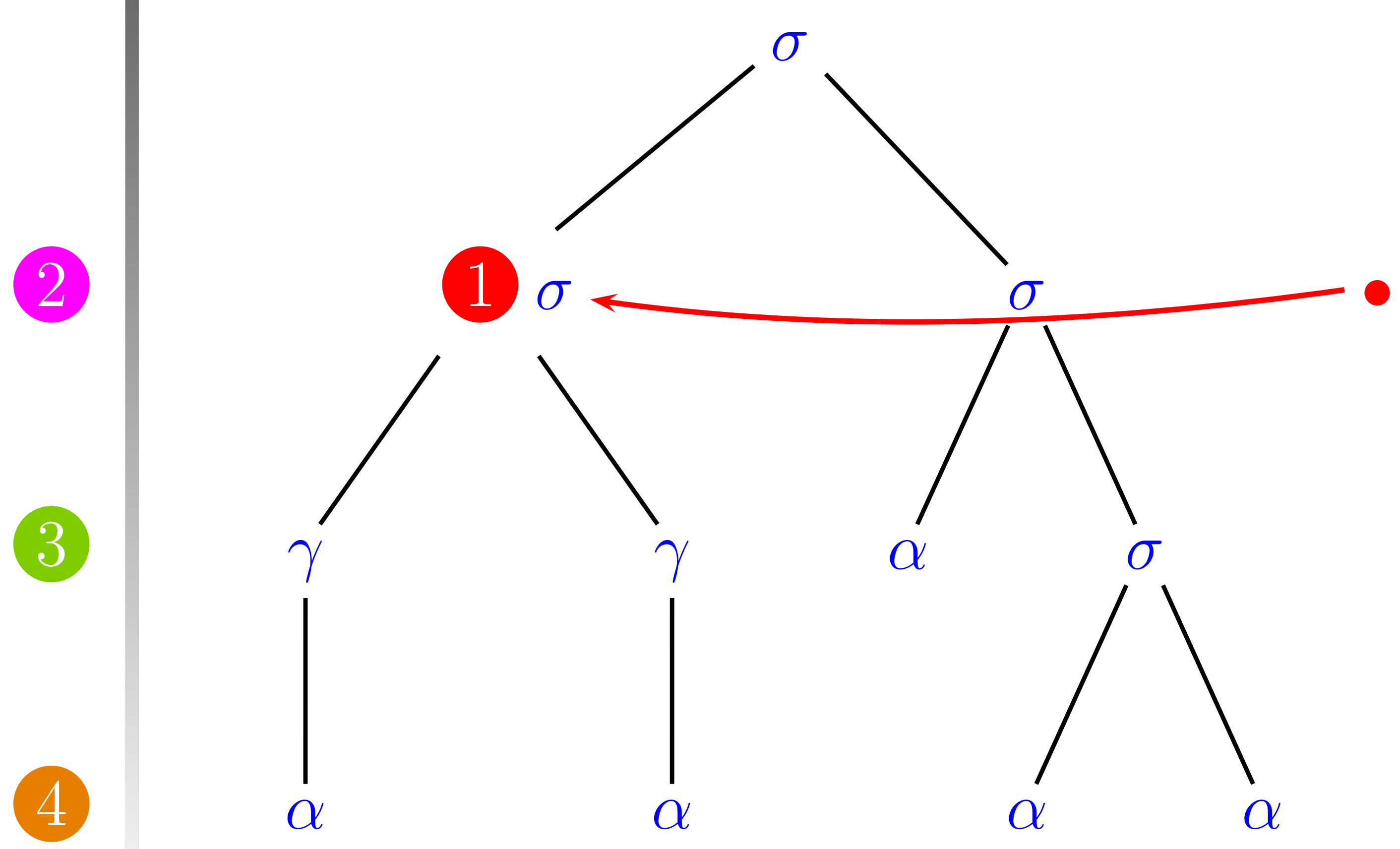
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



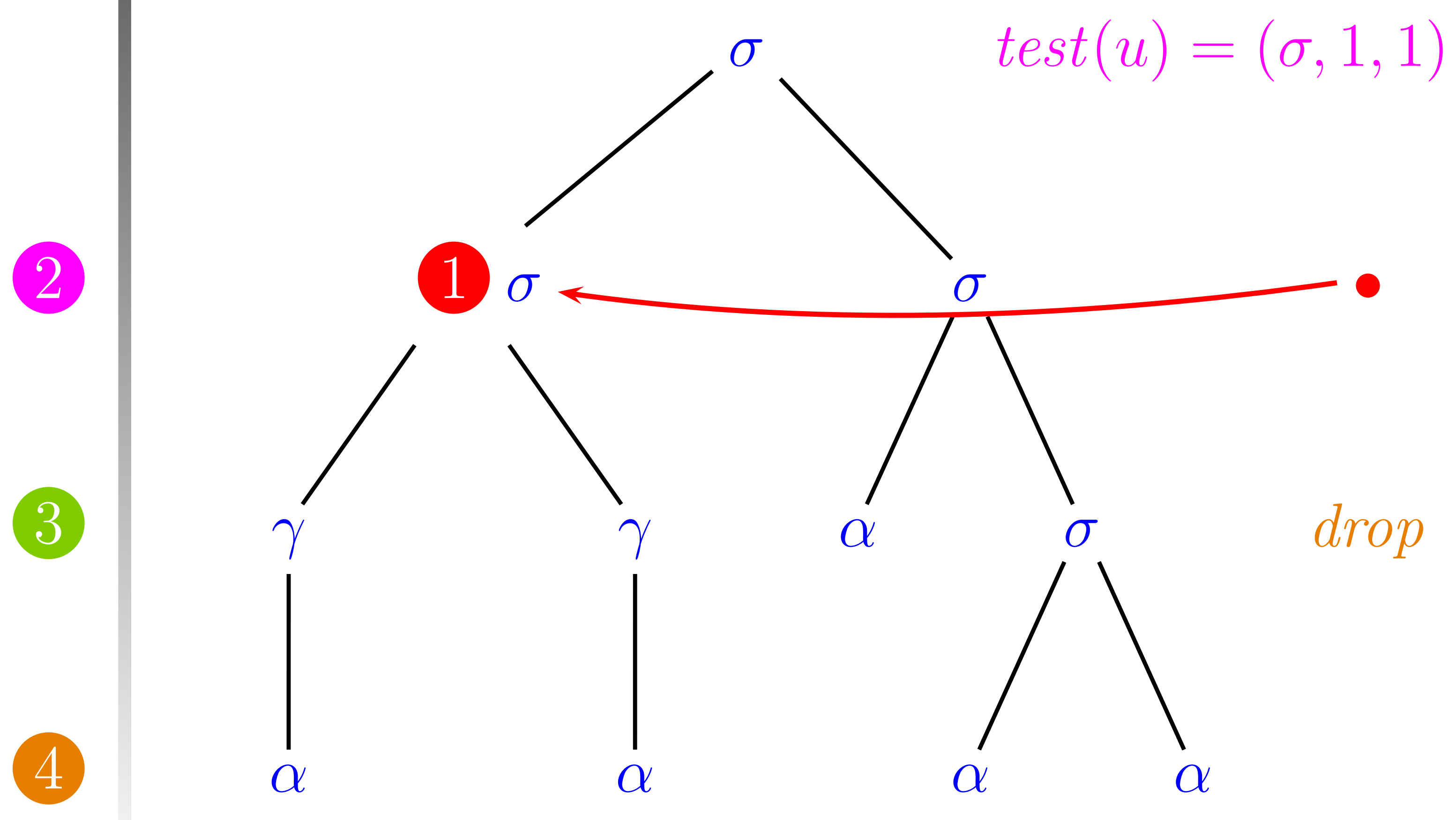
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



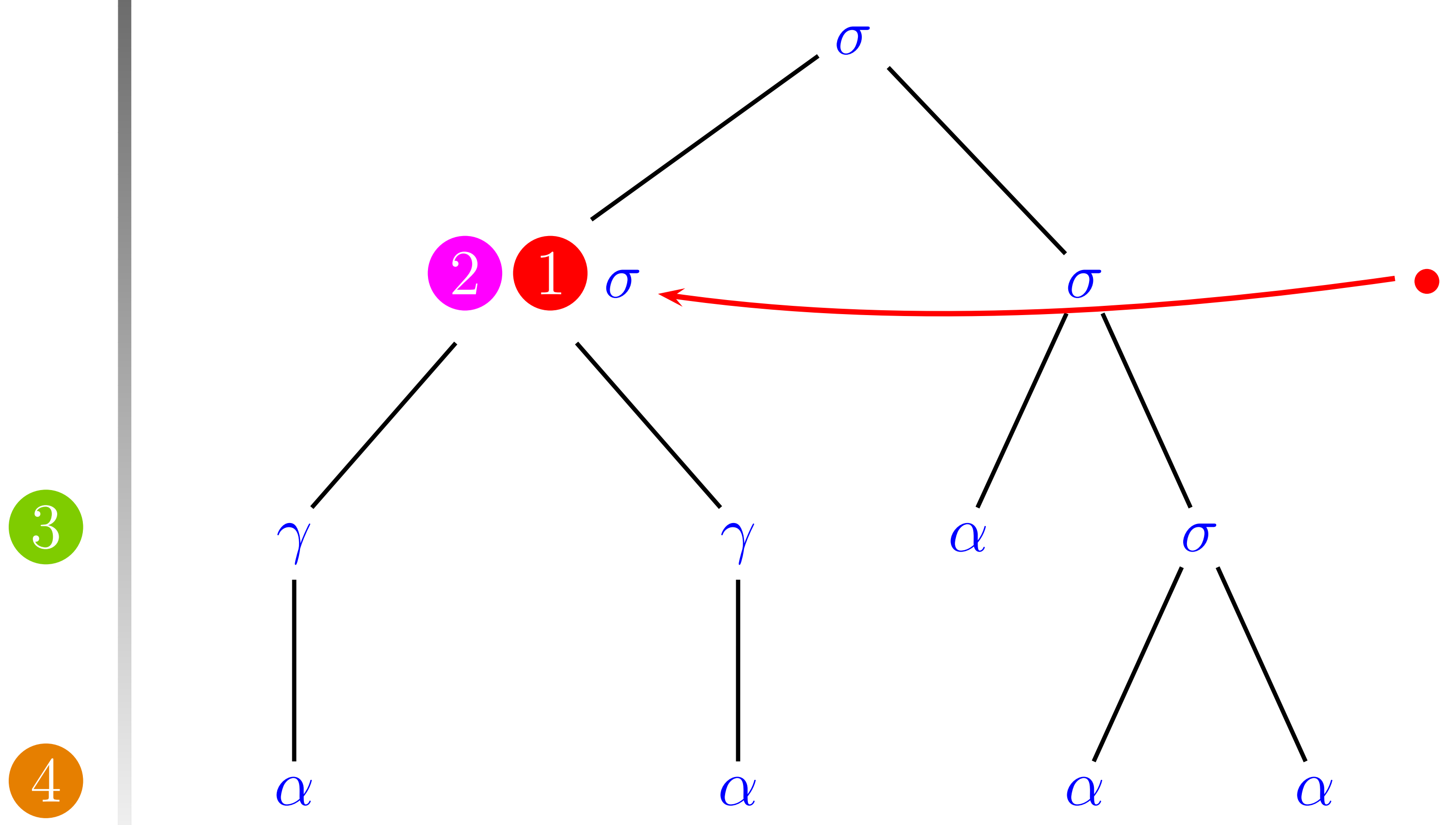
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



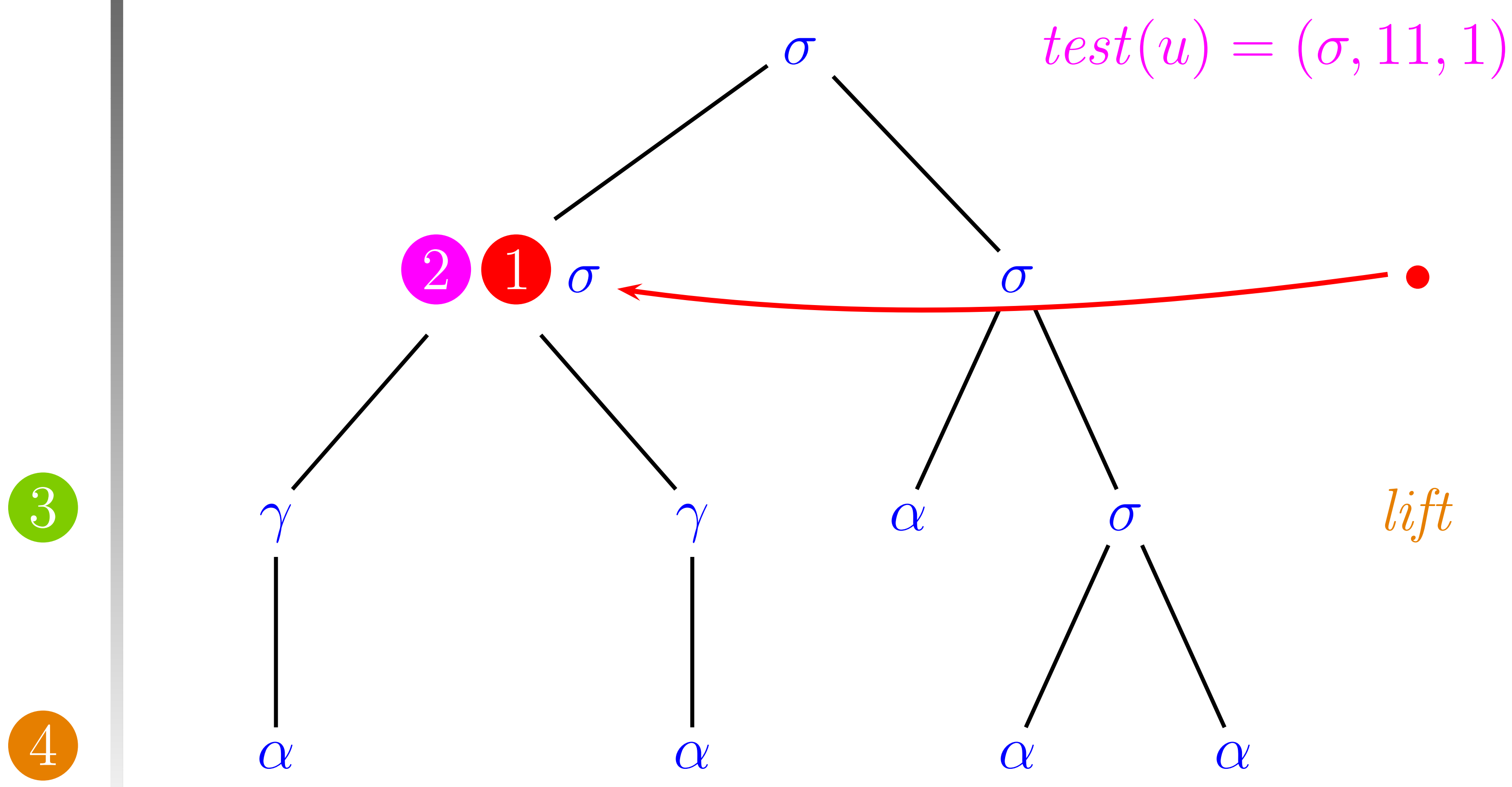
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



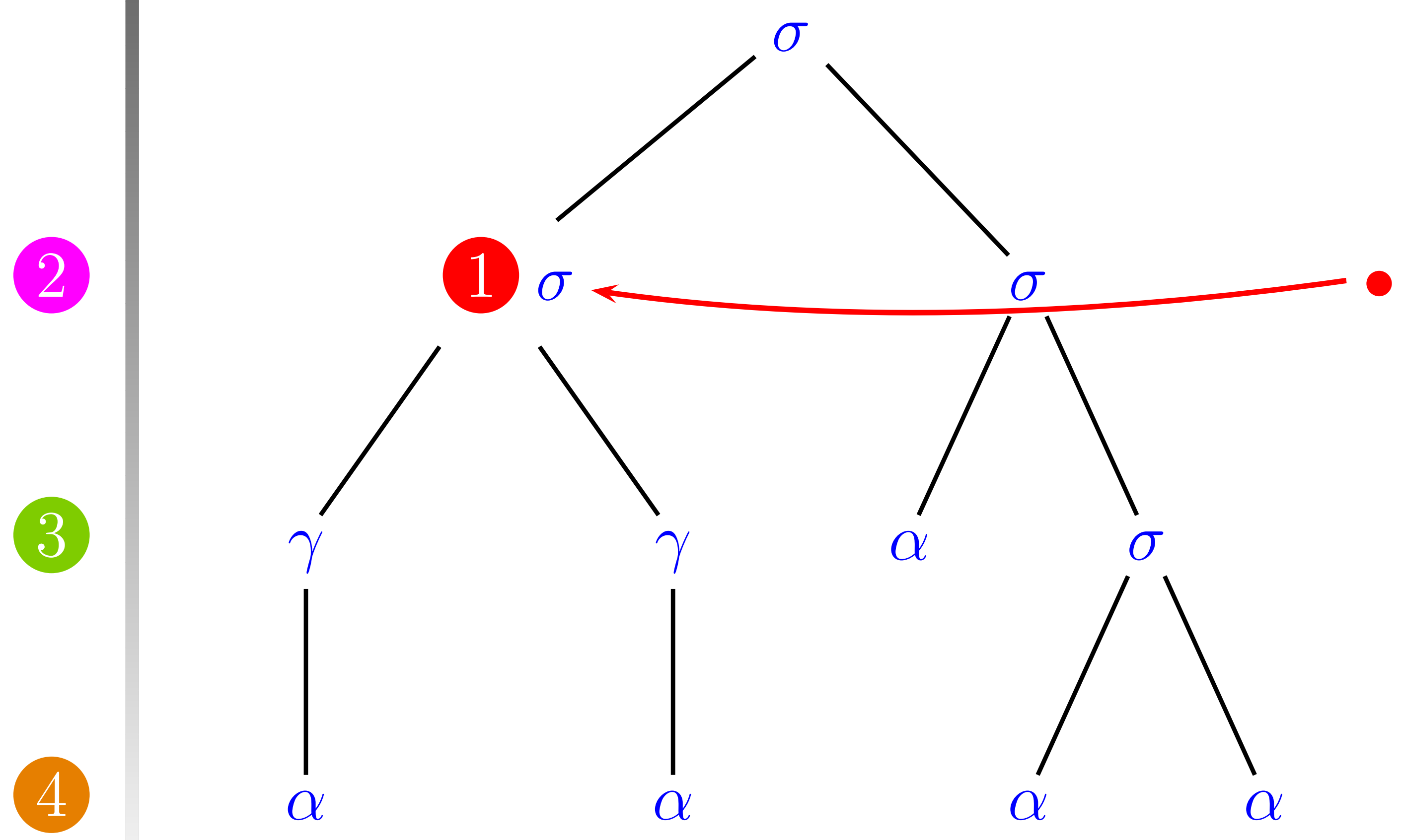
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



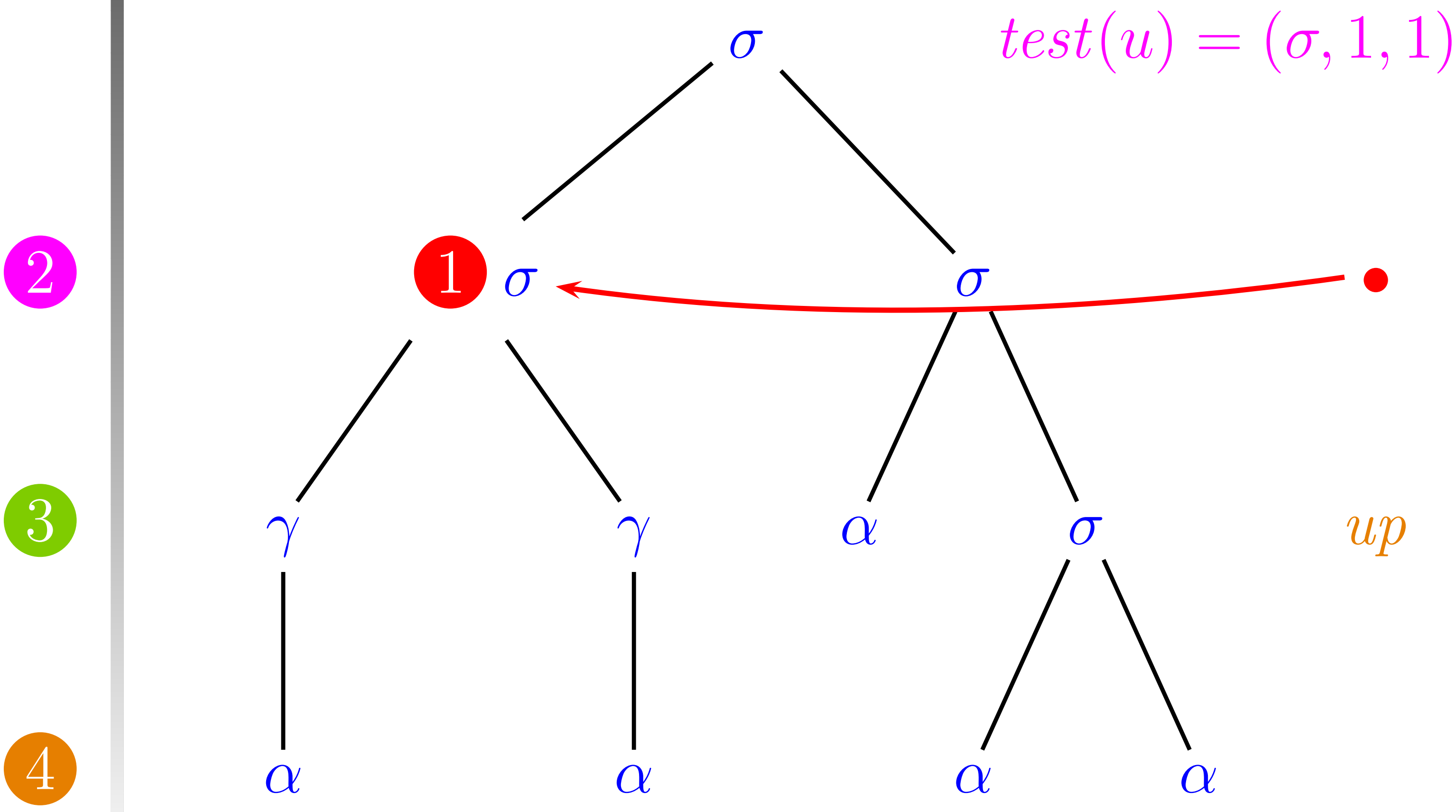
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



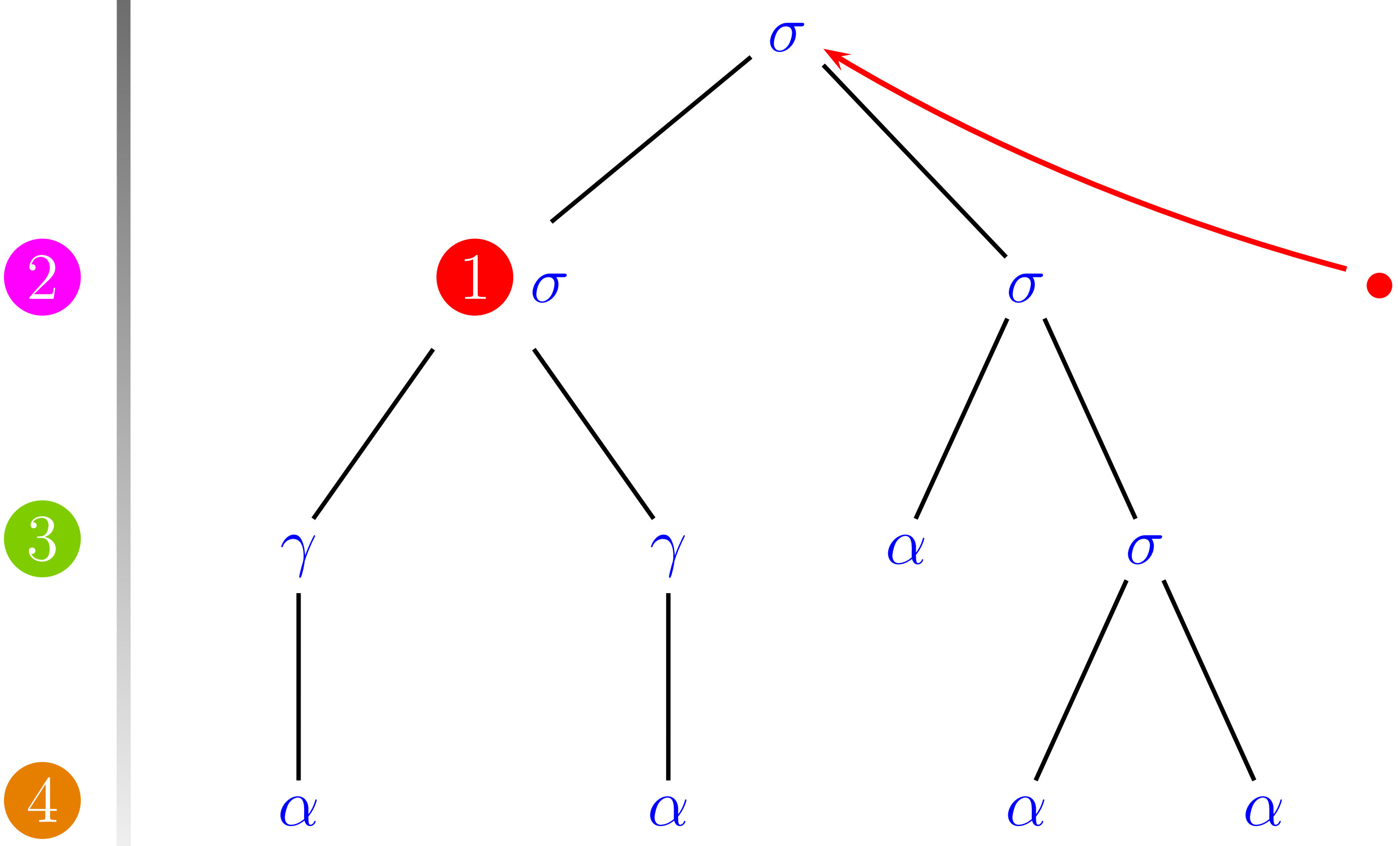
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



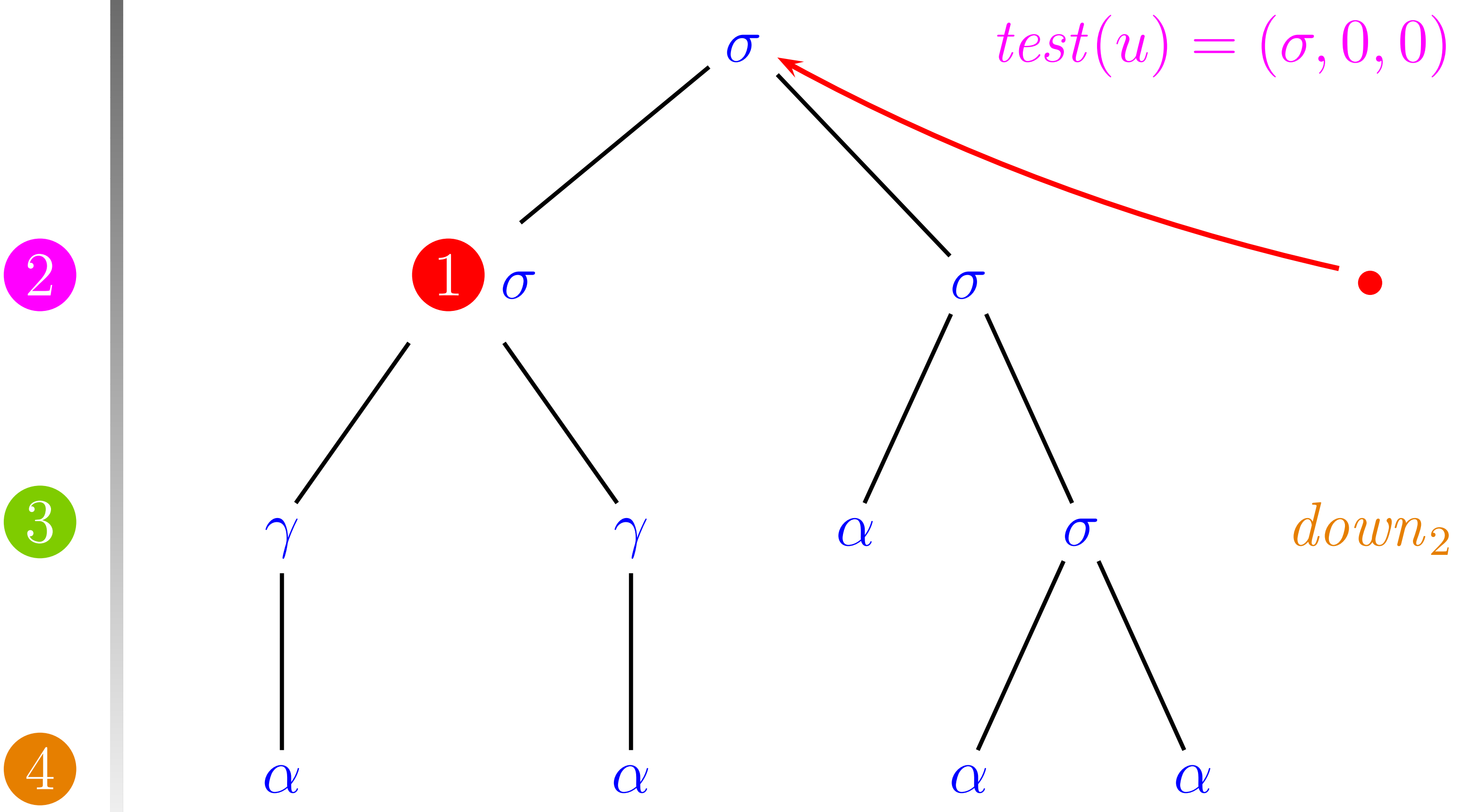
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



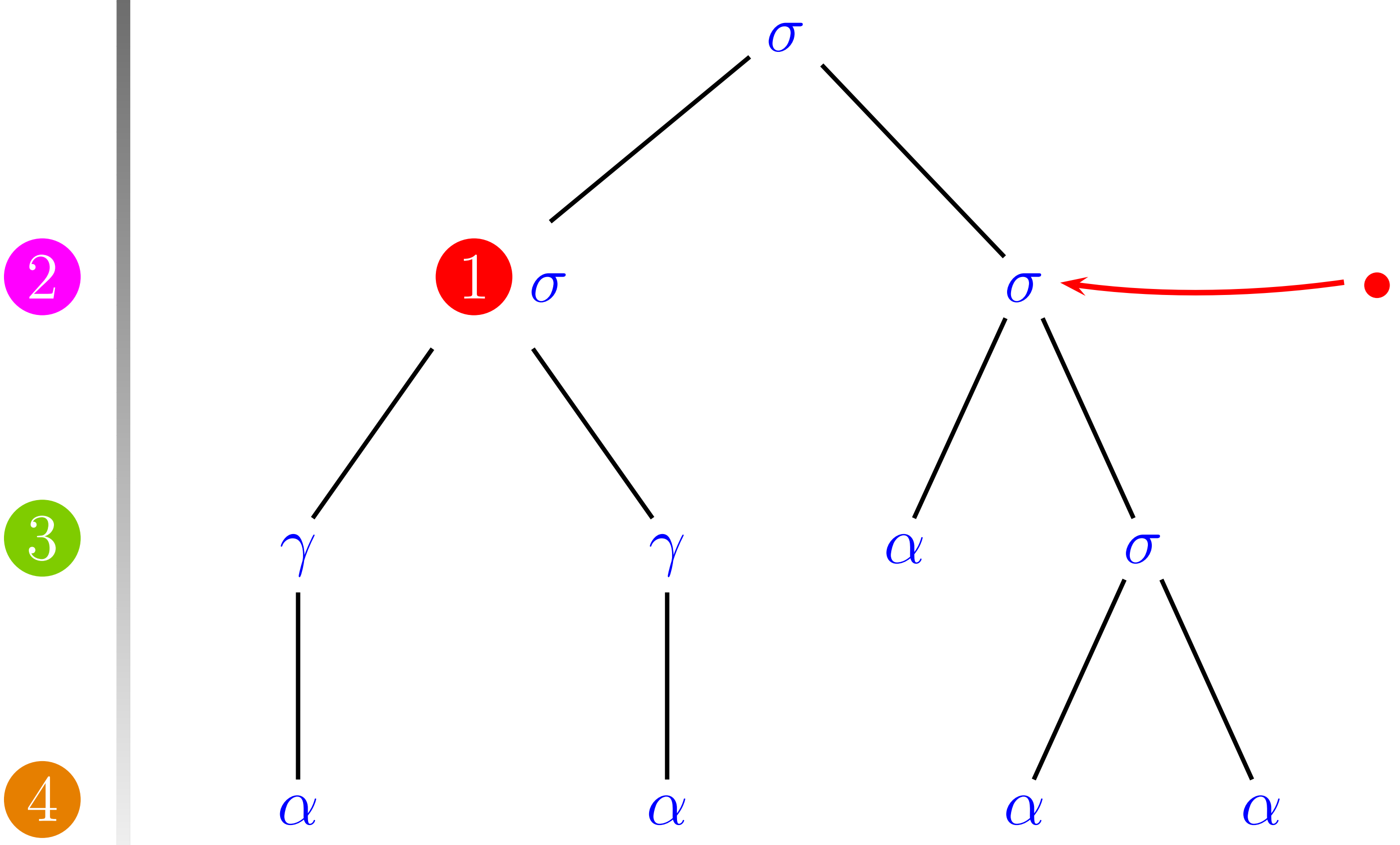
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



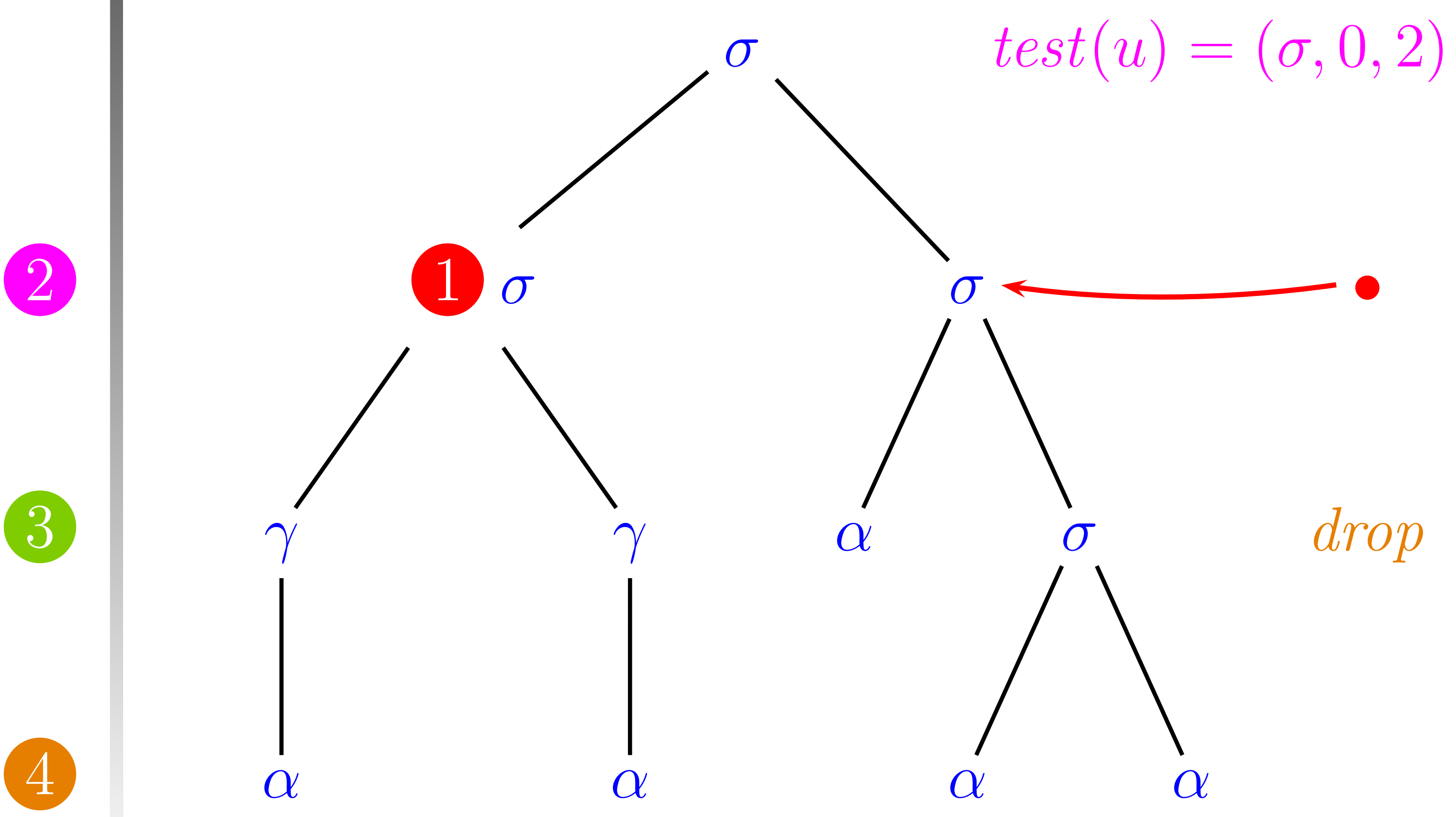
Kavics fabejárás

utasítások: ${stay, up, down_1, \ldots, down_d, drop, lift}$



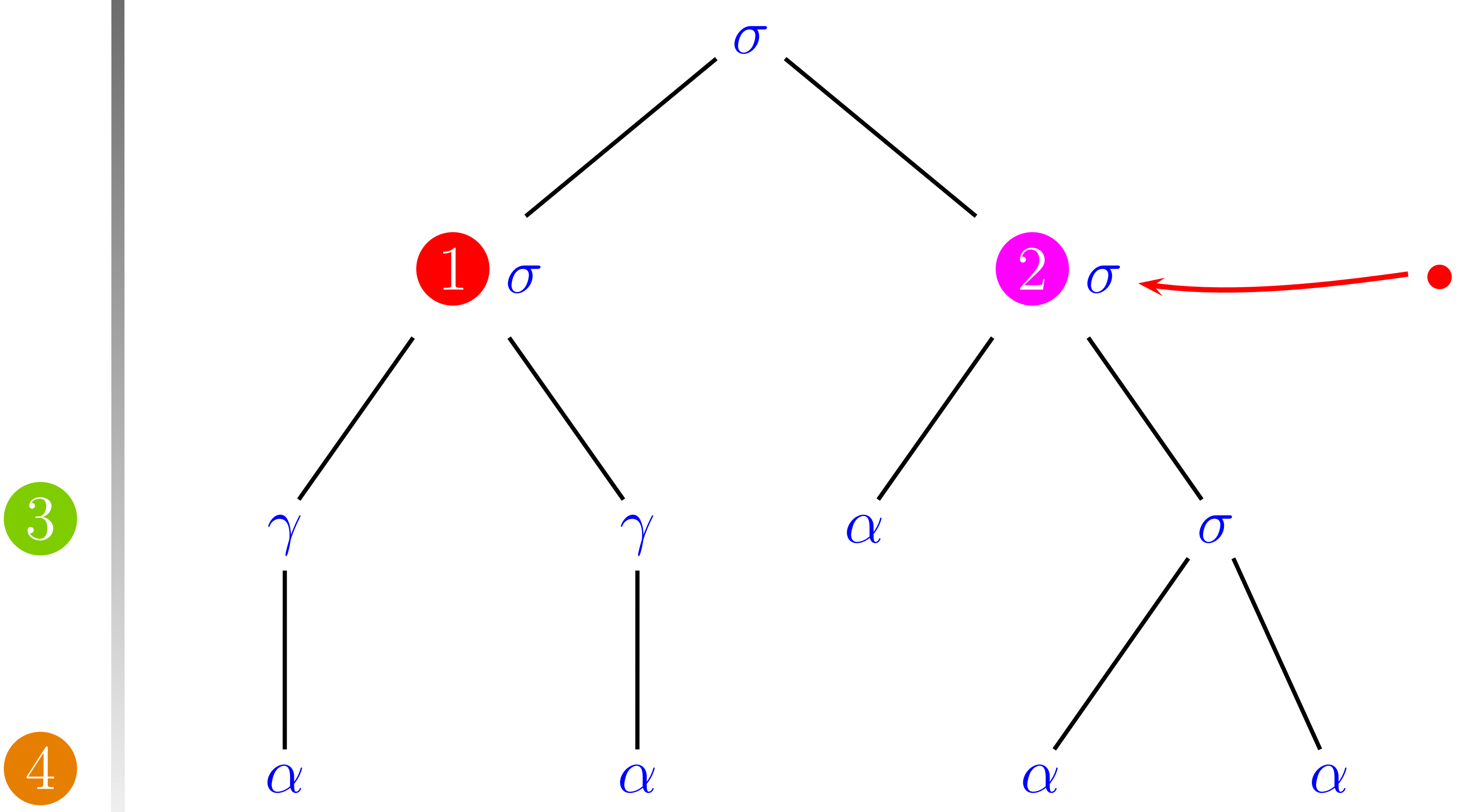
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



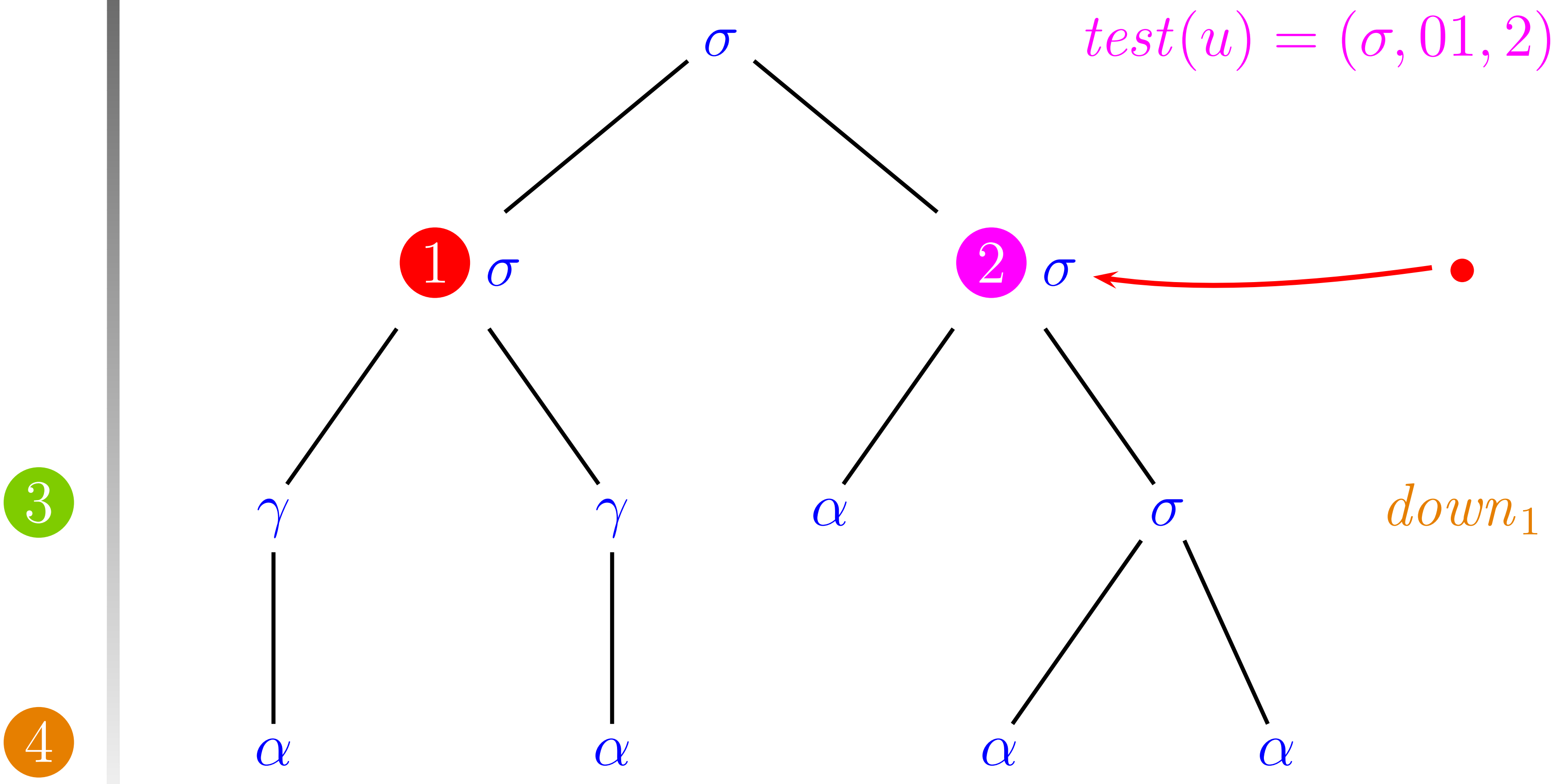
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



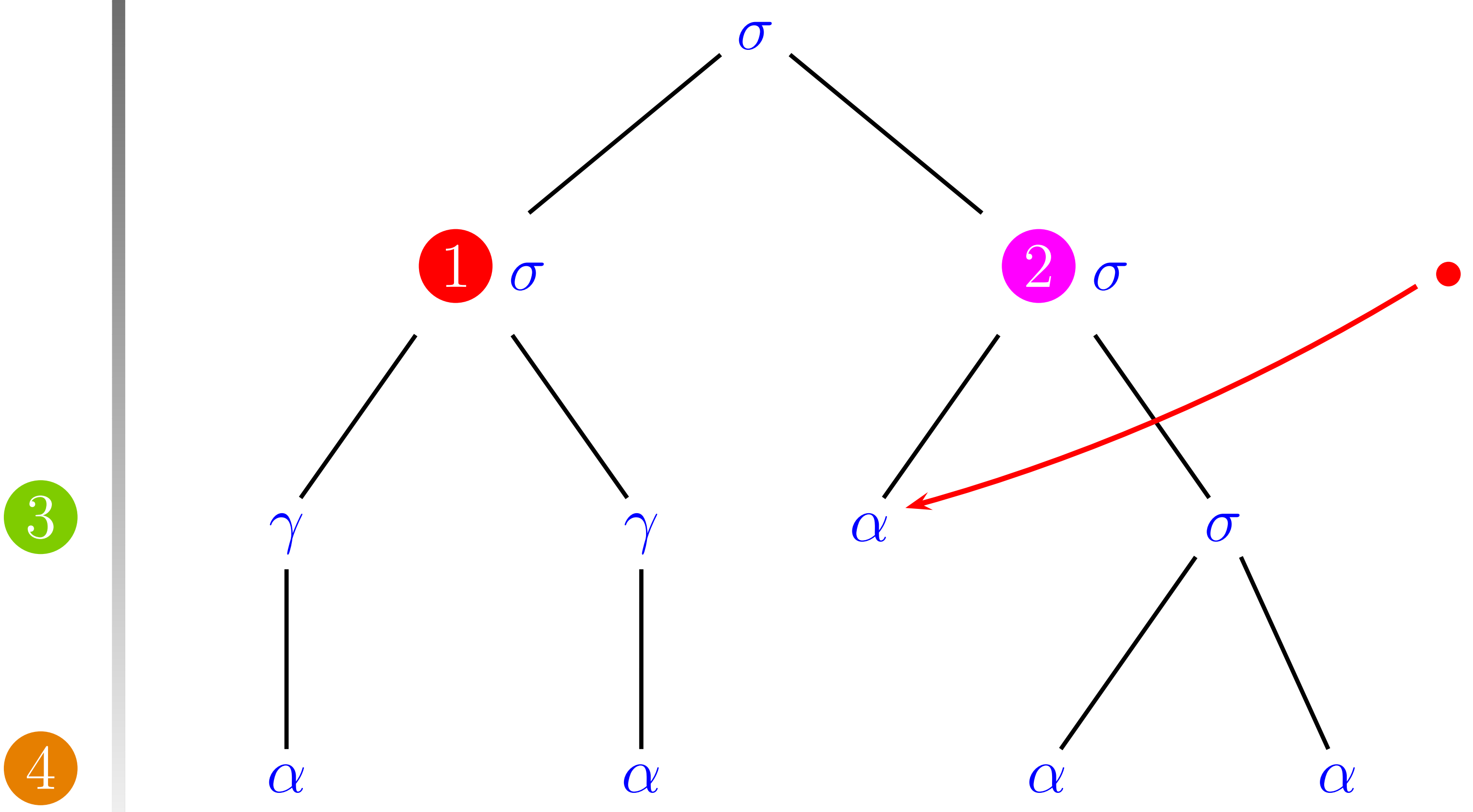
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



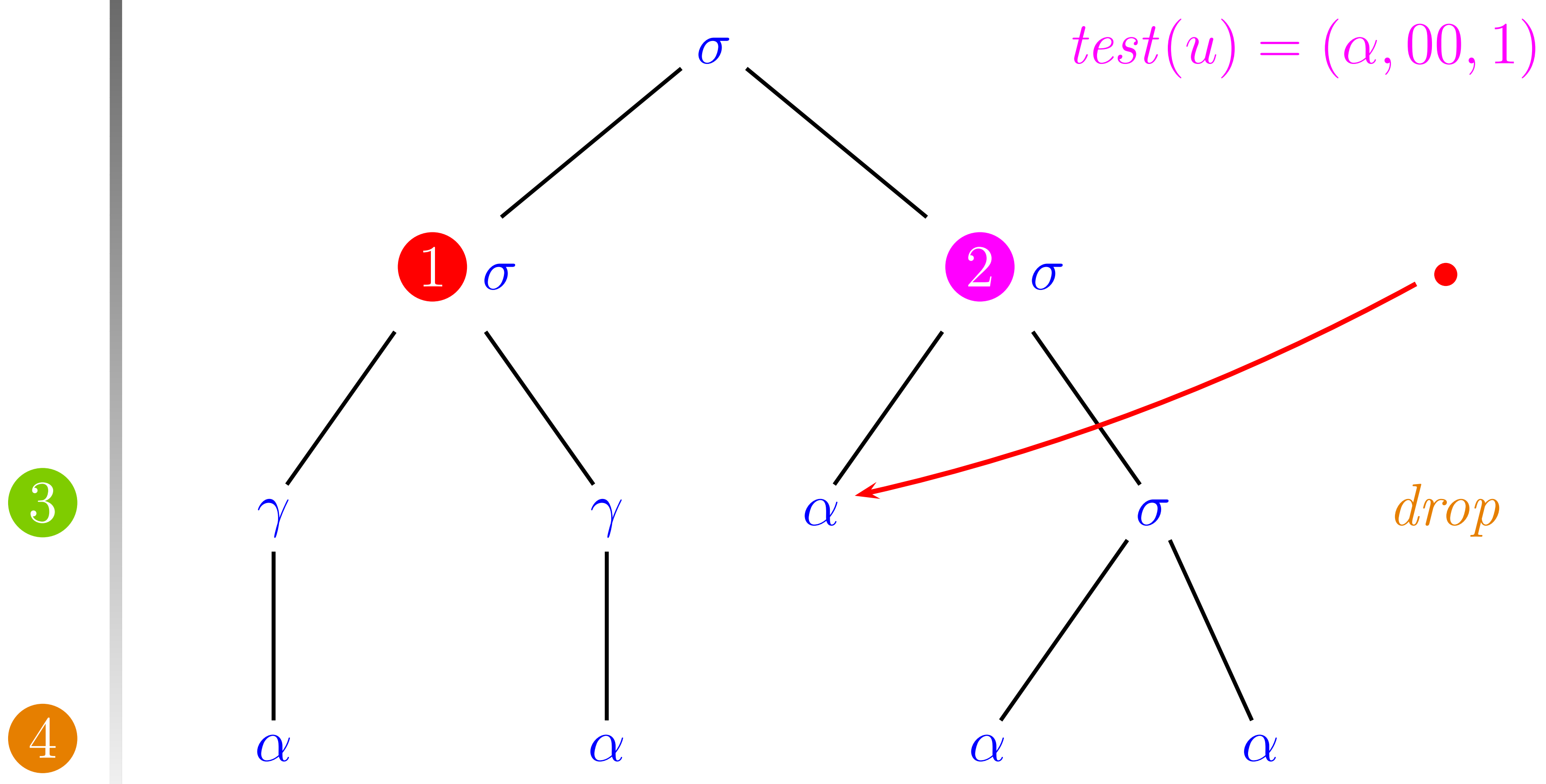
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



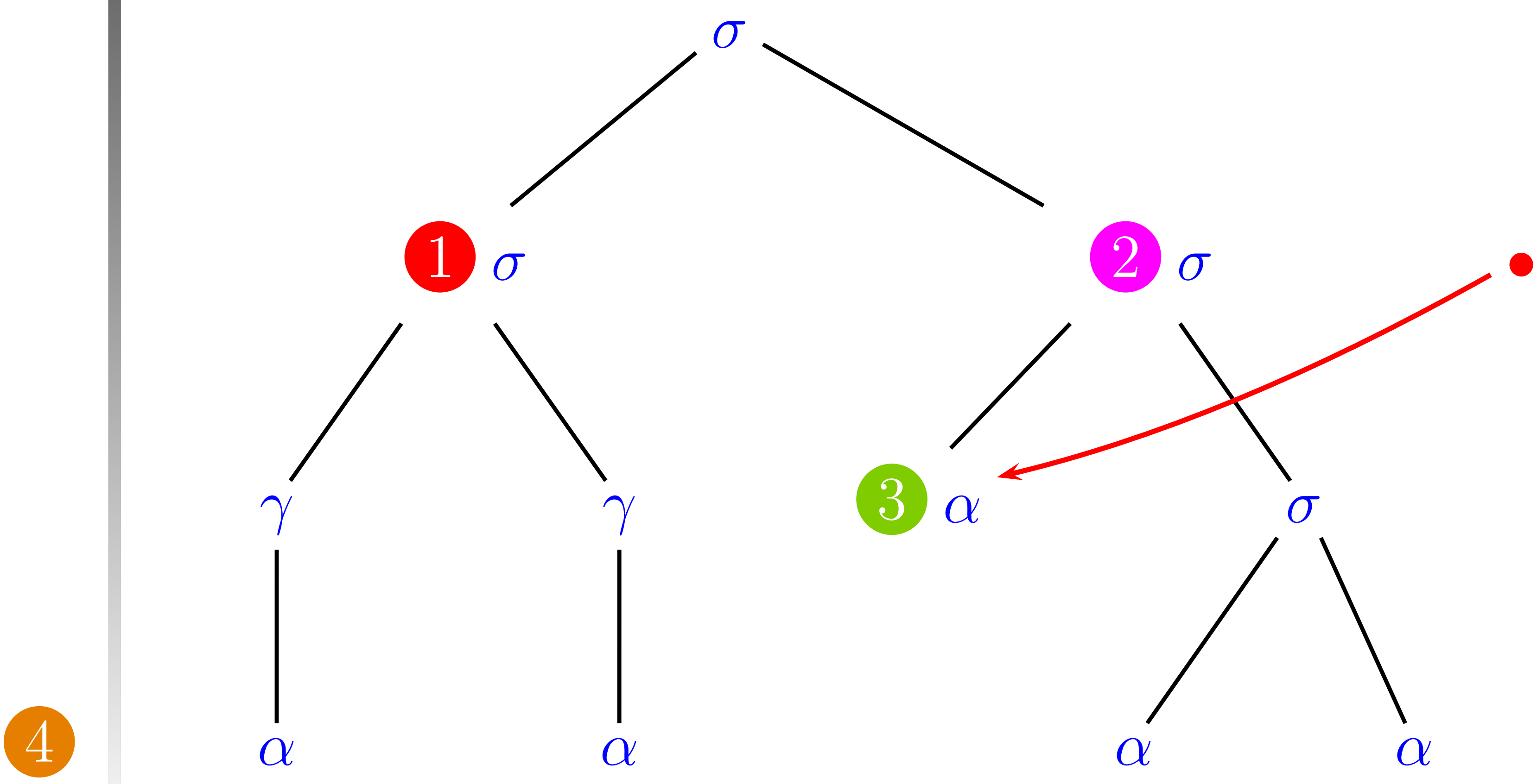
Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



Kavics fabejárás

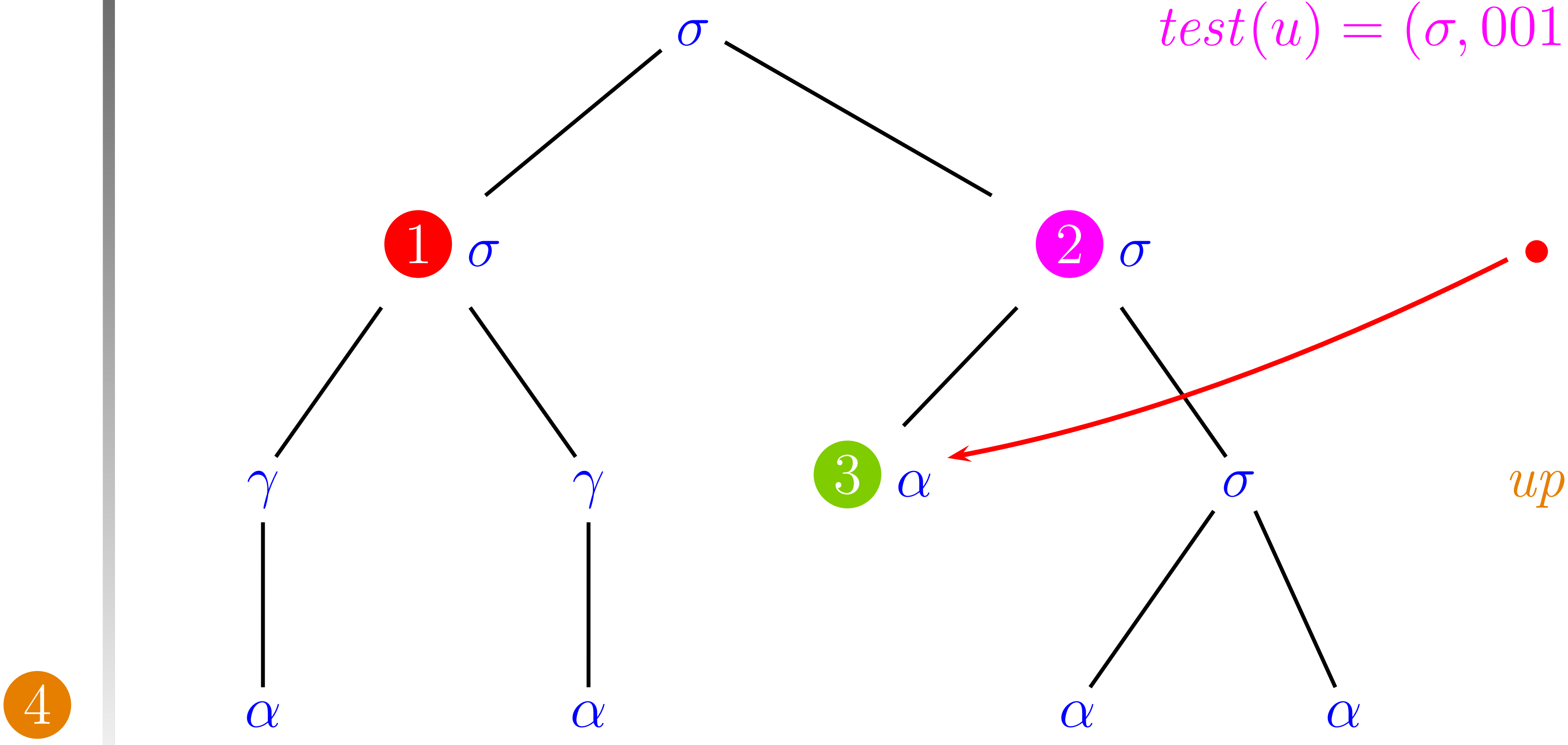
utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



Kavics fabejárás

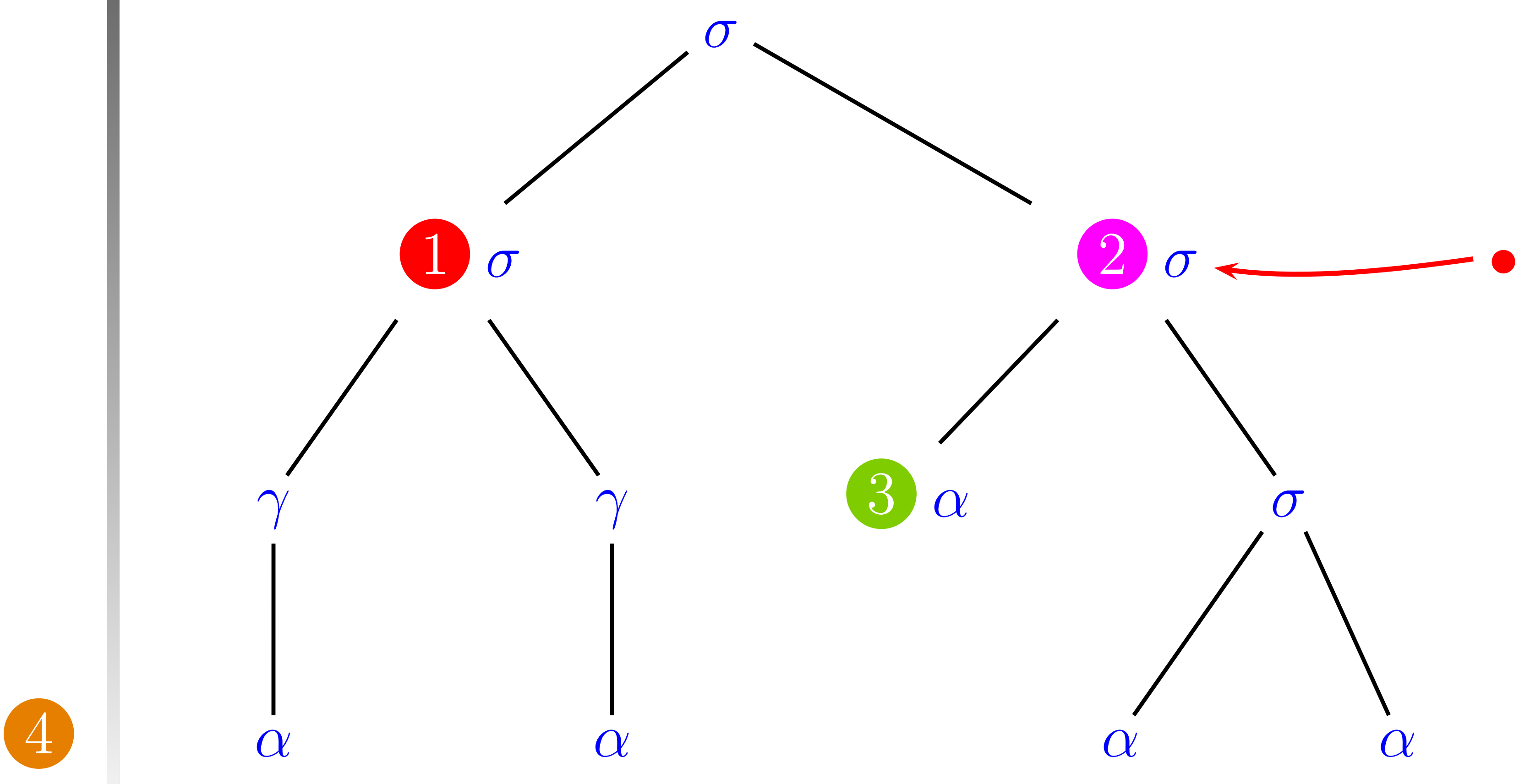
utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$

$test(u) = (\sigma, 001, 1)$



Kavics fabejárás

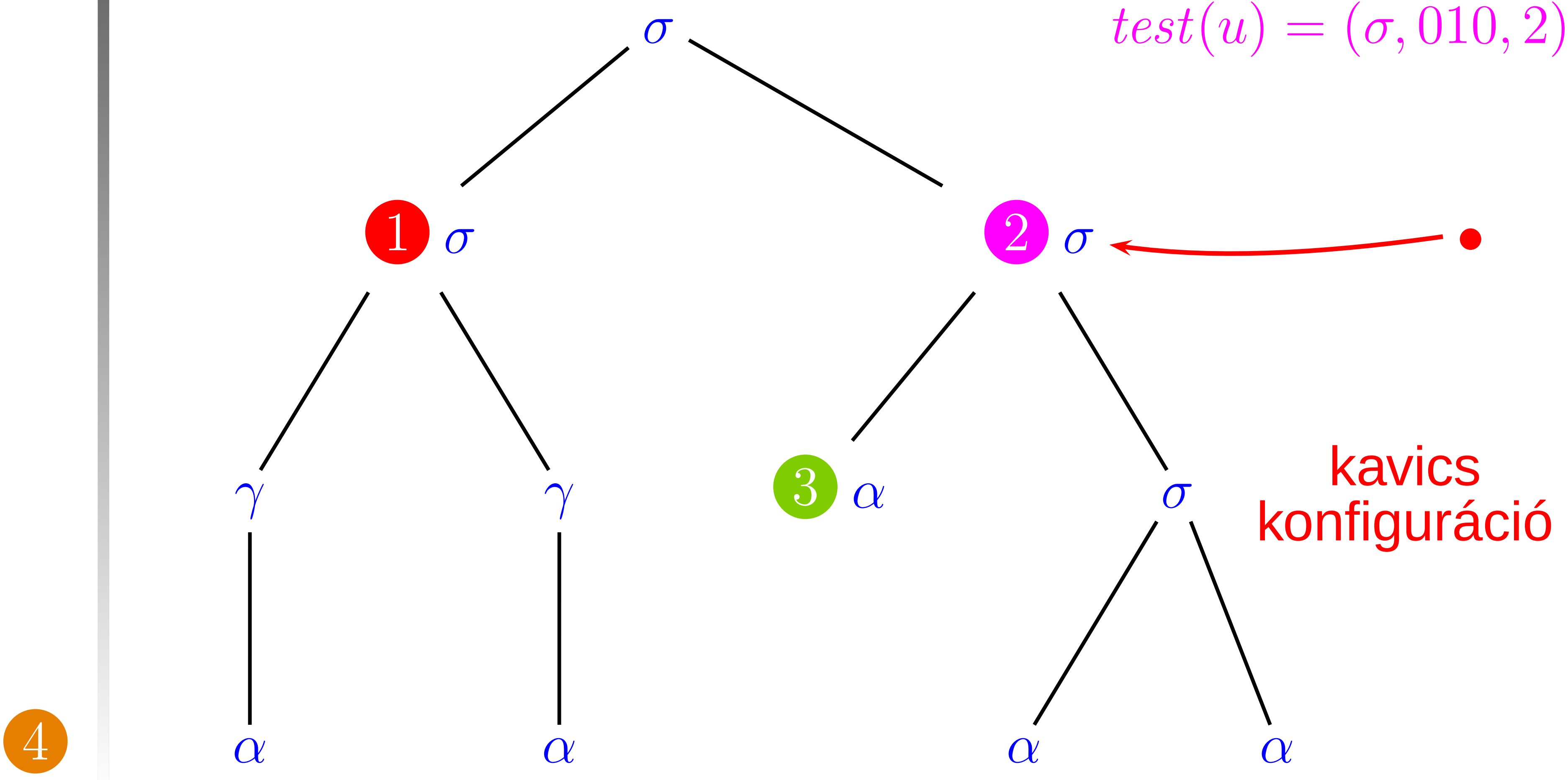
utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$



Kavics fabejárás

utasítások: $\{stay, up, down_1, \ldots, down_d, drop, lift\}$

$test(u) = (\sigma, 010, 2)$



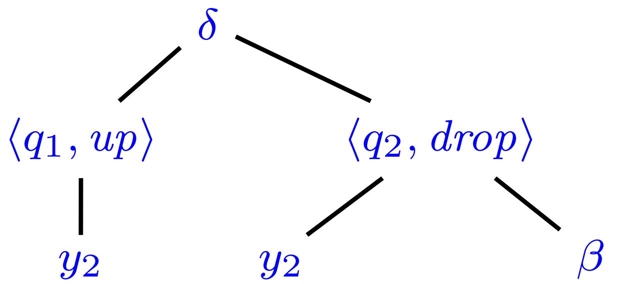
Kavics makró fatranszformátorok [EM03]

szintaxis: n-*kavics makró fatranszformátor* (n-pmtt) egy olyan $M = (Q, \Sigma, \Delta, q_0, R)$ rendszer, ahol

- Q egy rangolt ábécé, az *állapothalmaz*,
- Σ az *input ábécé*,
- Δ az *output ábécé*,
- $q_0 \in Q$ a *kezdőállapot* és
- R a *szabályhalmaz*.

Kavics makró fatranszformátorok [EM03]

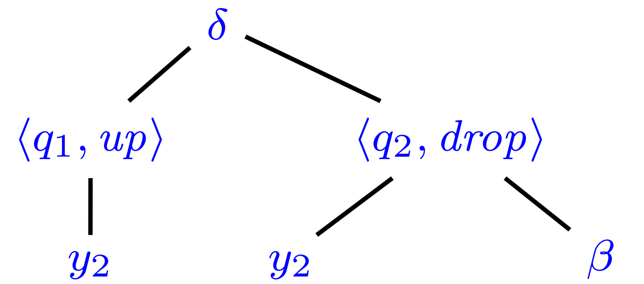
szabály: $\langle q, \sigma, 010, 2\rangle(y_1, y_2) \quad \rightarrow$



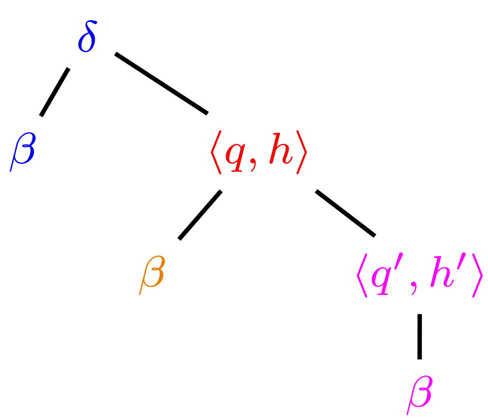
Kavics makró fatranszformátorok [EM03]

szabály:

$\langle q, \sigma, 010, 2\rangle(y_1, y_2) \;\rightarrow$

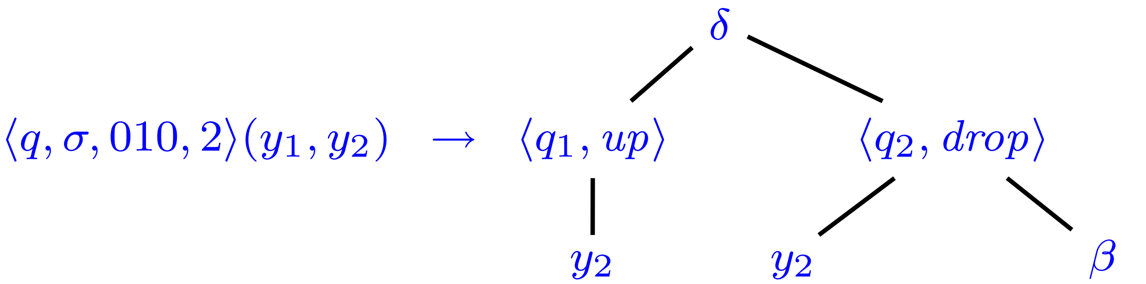


egy $s \in T_\Sigma$ input fa szerinti mondatforma:

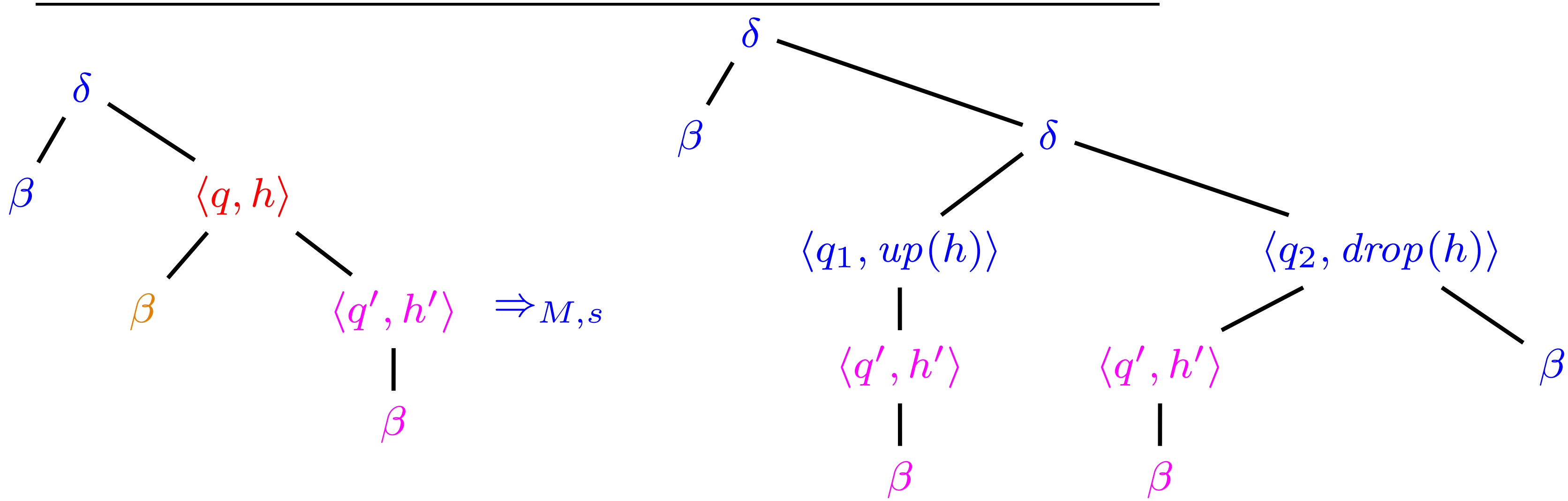


Kavics makró fatranszformátorok [EM03]

szabály:



egy $s \in T_\Sigma$ input fa szerinti mondatforma:



Kavics makró fatranszformátorok [EM03]

OI deriváció [EV85, ES77]: M csak legfelső konfigurációkat terjeszthet ki.

Kavics makró fatranszformátorok [EM03]

<u>szemantika:</u> Legyen M egy n-pmtt, $s \in T_\Sigma$ egy input fa.

- konfiguráció: $\langle q, (u, [u_1, \ldots, u_l])\rangle \in C_{M,s}$ (rangolt szimbólum),
$$rank(\langle q, (u, [u_1, \ldots, u_l])\rangle) = rank(q)$$
- mondatforma: $\xi \in T_{\Delta \cup C_{M,s}}$
- $\tau_M = \{(s,t) \in T_\Sigma \times T_\Delta \mid \langle q_0, (\varepsilon, [\,])\rangle \Rightarrow^*_{M,s} t\}$

<u>fatranszformációs osztályok:</u> n-$PMTT$, $PMTT$

Típus ellenőrzés probléma [MSV03]

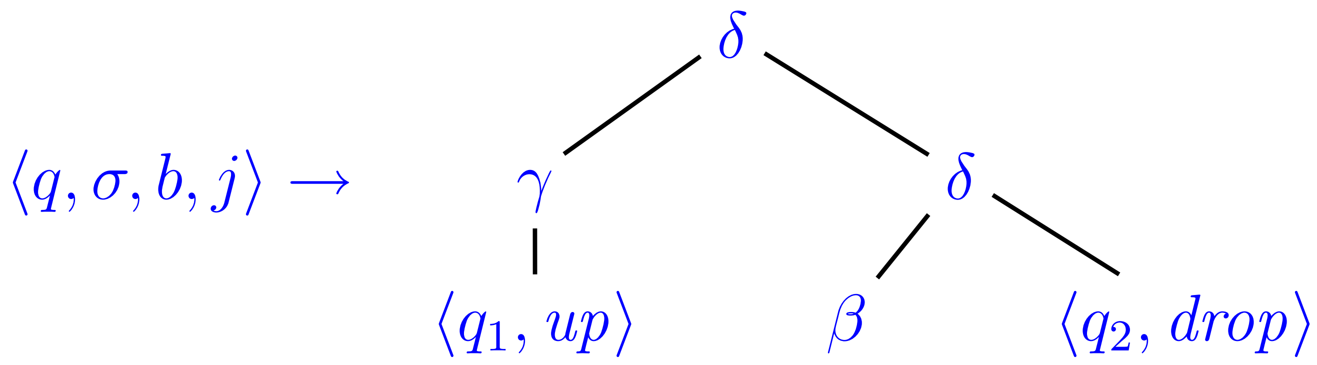
input: L_{in}, L_{out}, reguláris fanyelvek és τ, kavics makró fatranszformáció.

output: $\begin{cases} \text{"igen" ha } \tau(L_{in}) \subseteq L_{out} \\ \text{"nem" különben} \end{cases}$

nyitott kérdés. Eldönthető-e a típus ellenőrzés probléma?

Speciális pmtt-k

<u>*n-kavics fatranszformátor* (*n-ptt*)</u>: Az $M = (Q, \Sigma, q_0, R)$, n-pmtt egy *n-ptt*, ha minden Q-beli állapot rangja 0.



<u>*stay makró fatranszformátor* (*smtt*)</u>: Az $M = (Q, \Sigma, q_0, R)$, 0-pmtt egy *smtt*, ha nem használ up utasítást.

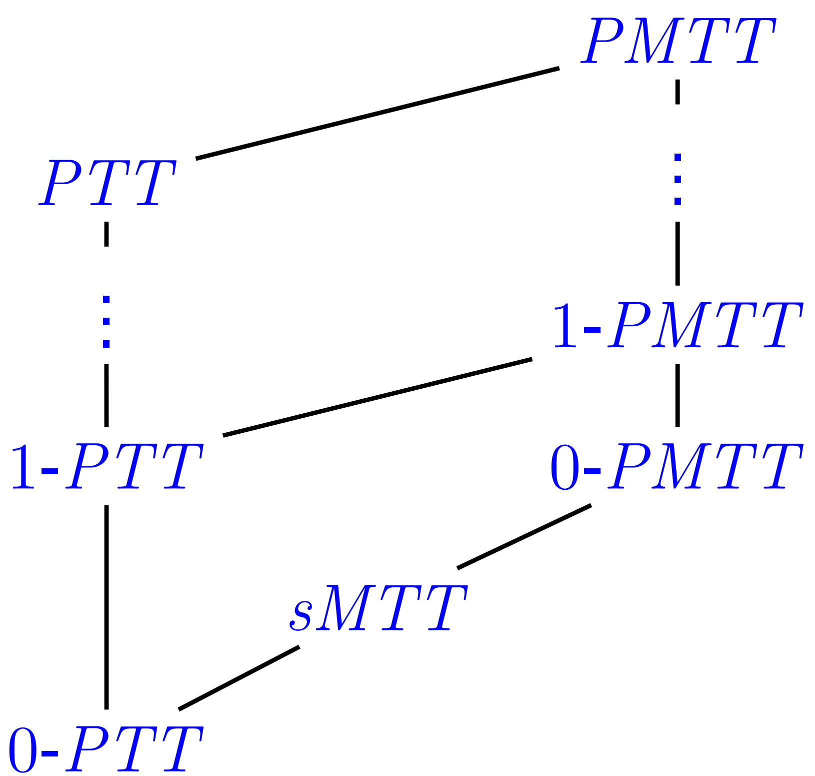
<u>fatranszformációs osztályok:</u> n-PTT, PTT, $sMTT$

Speciális pmtt-k

Lemma([EM03]). Minden $n \geq 0$-ra,

$$n\text{-}PTT \subseteq \underbrace{0\text{-}PTT \circ \ldots \circ 0\text{-}PTT}_{n \text{ db}} \circ 0\text{-}PTT = 0\text{-}PTT^{n+1}$$

Lemma([EM03]).

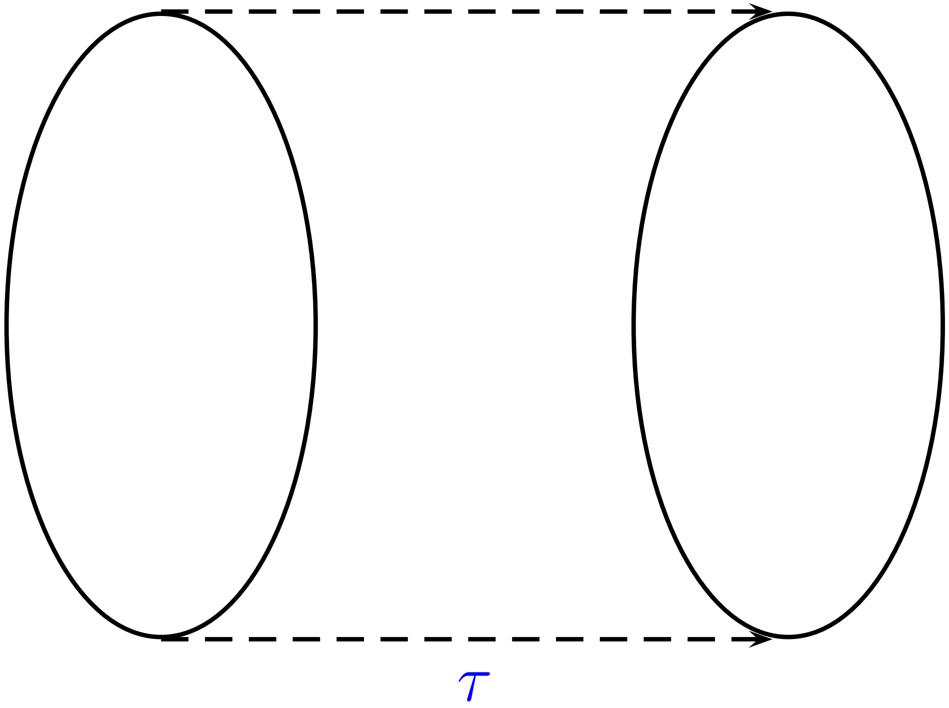


Speciális pmtt-k

Lemma. ([EM03]) Az smtt transzformációk (kompozícióinak) inverzei megőrzik a regularitást. (é.i: Minden $\tau \in sMTT$ fatranszformáció és L, reguláris fanyelvre $\tau^{-1}(L)$ reguláris.)

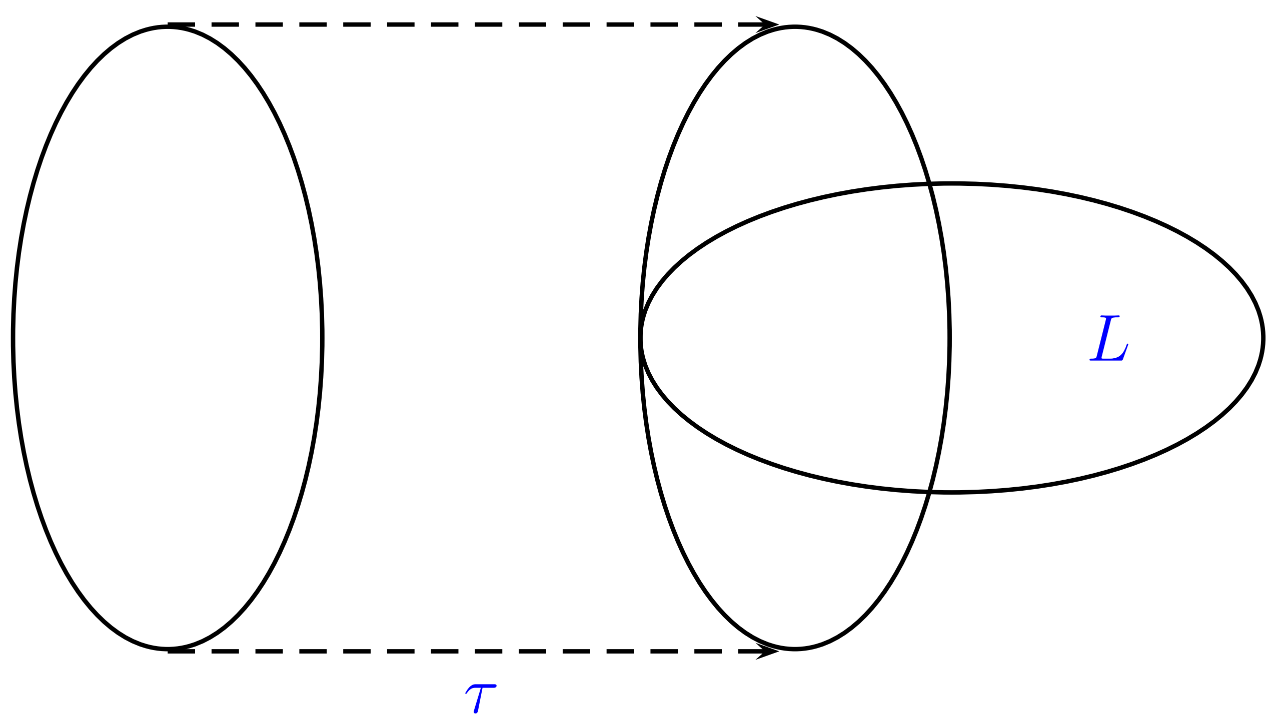
Speciális pmtt-k

Lemma. ([EM03]) Az smtt transzformációk (kompozícióinak) inverzei megőrzik a regularitást. (é.i: Minden $\tau \in sMTT$ fatranszformáció és L, reguláris fanyelvre $\tau^{-1}(L)$ reguláris.)



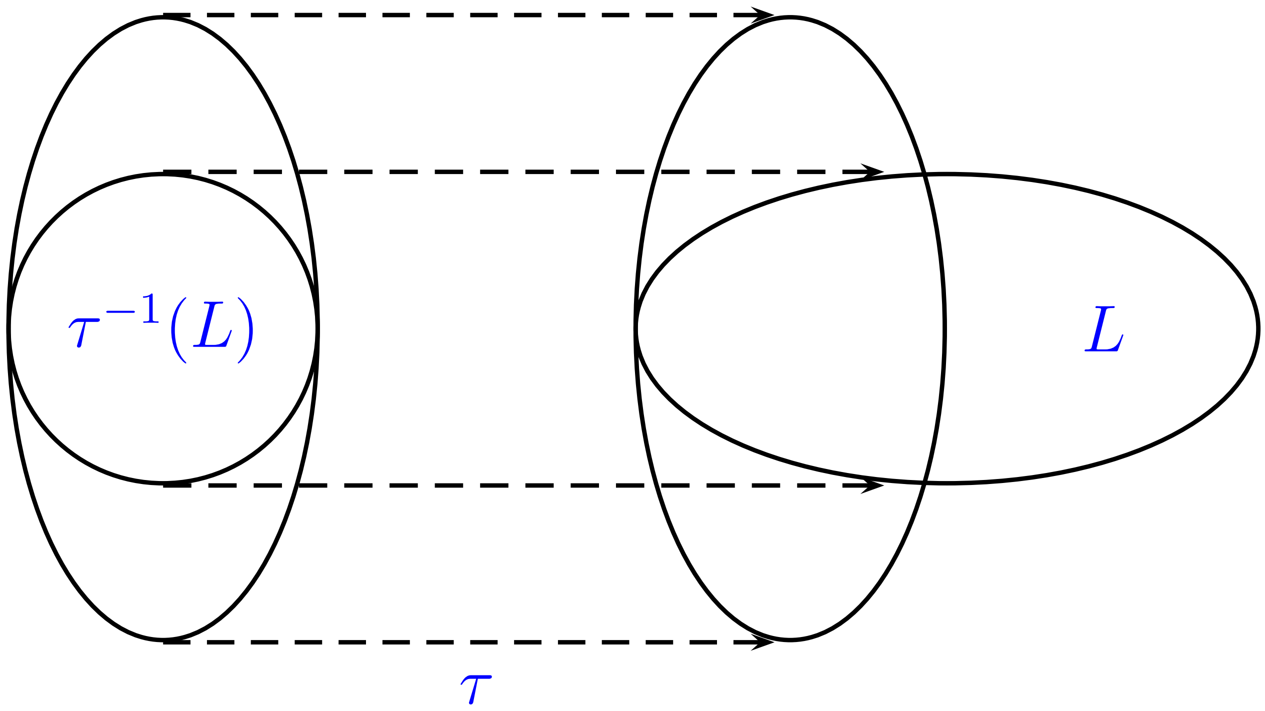
Speciális pmtt-k

Lemma. ([EM03]) Az smtt transzformációk (kompozícióinak) inverzei megőrzik a regularitást. (é.i: Minden $\tau \in sMTT$ fatranszformáció és L, reguláris fanyelvre $\tau^{-1}(L)$ reguláris.)



Speciális pmtt-k

Lemma. ([EM03]) Az smtt transzformációk (kompozícióinak) inverzei megőrzik a regularitást. (é.i: Minden $\tau \in sMTT$ fatranszformáció és L, reguláris fanyelvre $\tau^{-1}(L)$ reguláris.)



Speciális pmtt-k

Tétel. ([EM03]) Az smtt-k típus ellenőrzése eldönthető.

Következmény. A ptt-k típus ellenőrzése eldönthető.

Speciális pmtt-k

egyéb pmtt-k, melyekre a típus ellenőrzés eldönthető:

- determinisztikus pmtt-k.
  (azzal a kikötéssel, hogy $L_{out} \in dREG$)

- korlátozott parameter másolású pmtt-k.

References

[EM03] J. Engelfriet and S. Maneth. A Comparison of Pebble Tree Transducers with Macro Tree Transducers. *Acta Informatica*, 39:613–698, 2003.

[ES77] J. Engelfriet and E. M. Schmidt. IO and OI, Part I. *J. Comput. System Sci.*, 15(3):328–58, 1977.

[EV85] J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. System Sci.*, 31:71–146, 1985.

[GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[GS97] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 1–68. Springer-Verlag, 1997.

[MSV03] T. Milo, D. Suciu, and V. Vianu. Typechecking for XML transformers. *J. of Comput. Syst. Sci.*, 66:66–97, 2003.