

ADATSZERKEZETEK ÉS ALGORITMUSOK

Lengyel forma

Kifejezések

- Infix

- Amikor az operátor az operandusok között van
- $a + b$

- Postfix

- Amikor az operátor az operandusok mögött van
- $a b +$

- Prefix

- Amikor az operátor az operandusok előtt van
- $+a b$

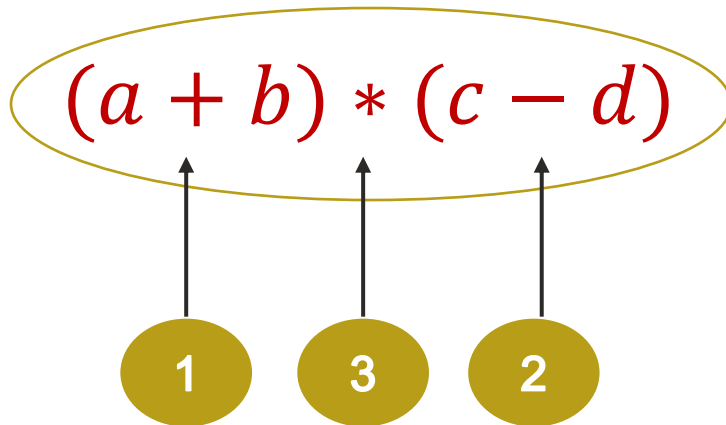
Lengyel forma

- J. Lukasewitz lengyel matematikus használta először
 - Prefix forma – lengyel forma
 - Postfix forma – fordított lengyel forma
- Átalakítás
 - Infix kifejezés
 - $(a + b) * (c - d)$
 - Postfix kifejezés
 - $a b + c d - *$
- Az Infix kifejezés átalakítható Postfix kifejezéssé
 - A kifejezések feldolgozásánál két fontos előnye van

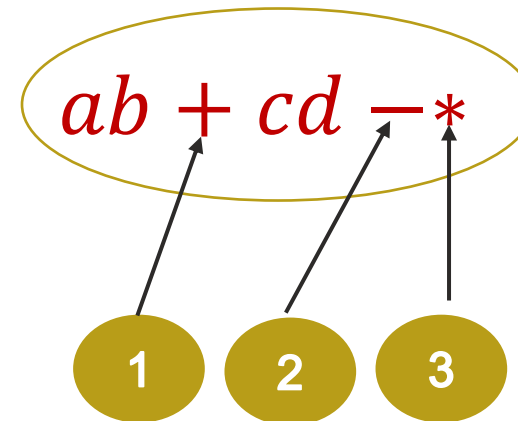
Fordított lengyel forma – előnyök

- A műveleti jelek olyan sorrendben követik egymást, amilyen sorrendben végre kell hajtani azokat.

Infix forma

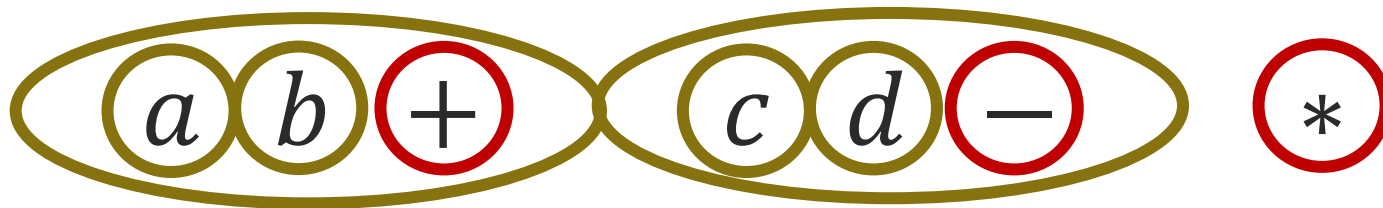


Lengyelforma



Fordított lengyel forma – előnyök

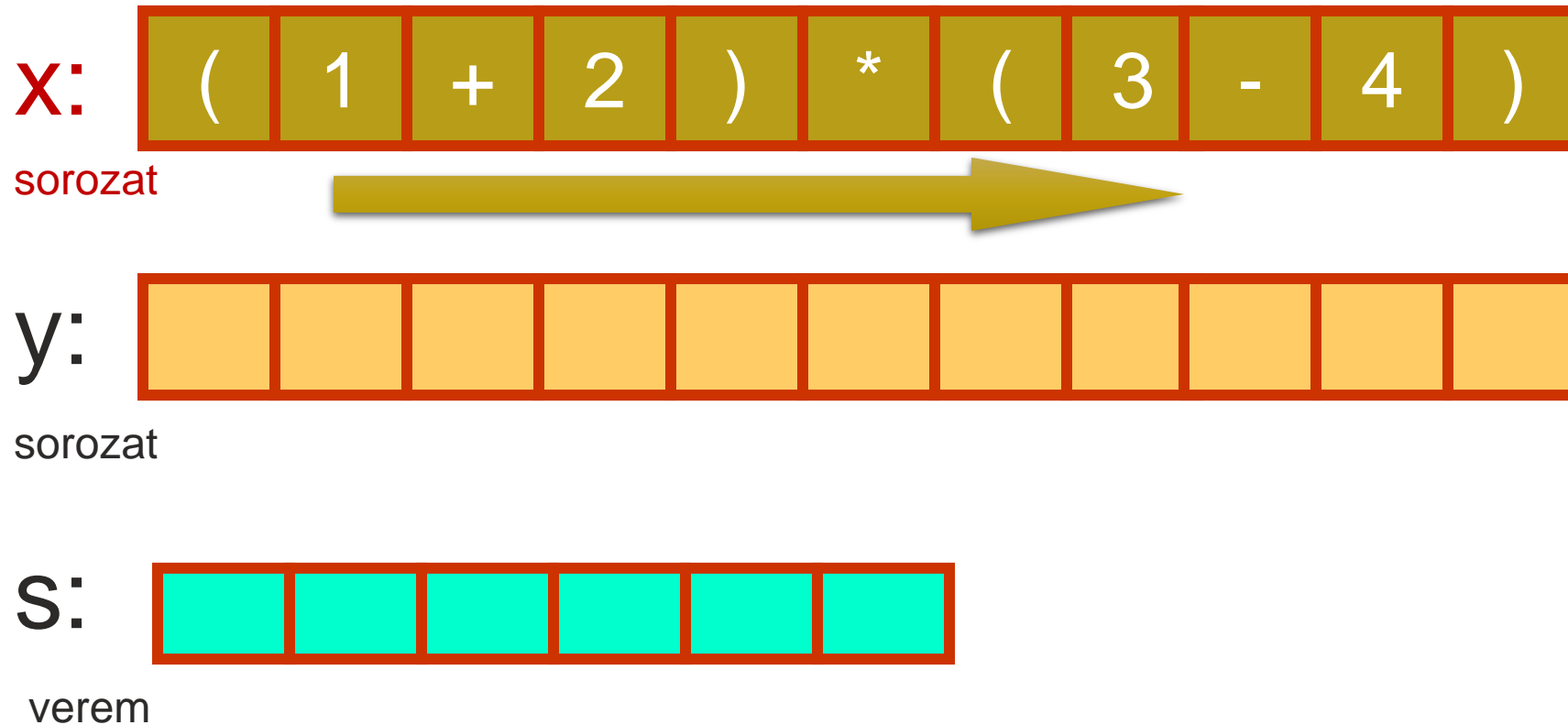
- A műveleti jel (operátor) közvetlenül az operandusai után áll
 - Infix forma: $(a + b) * (c - d)$
 - Postfix forma
 - Operátor
 - Operandus



További példák

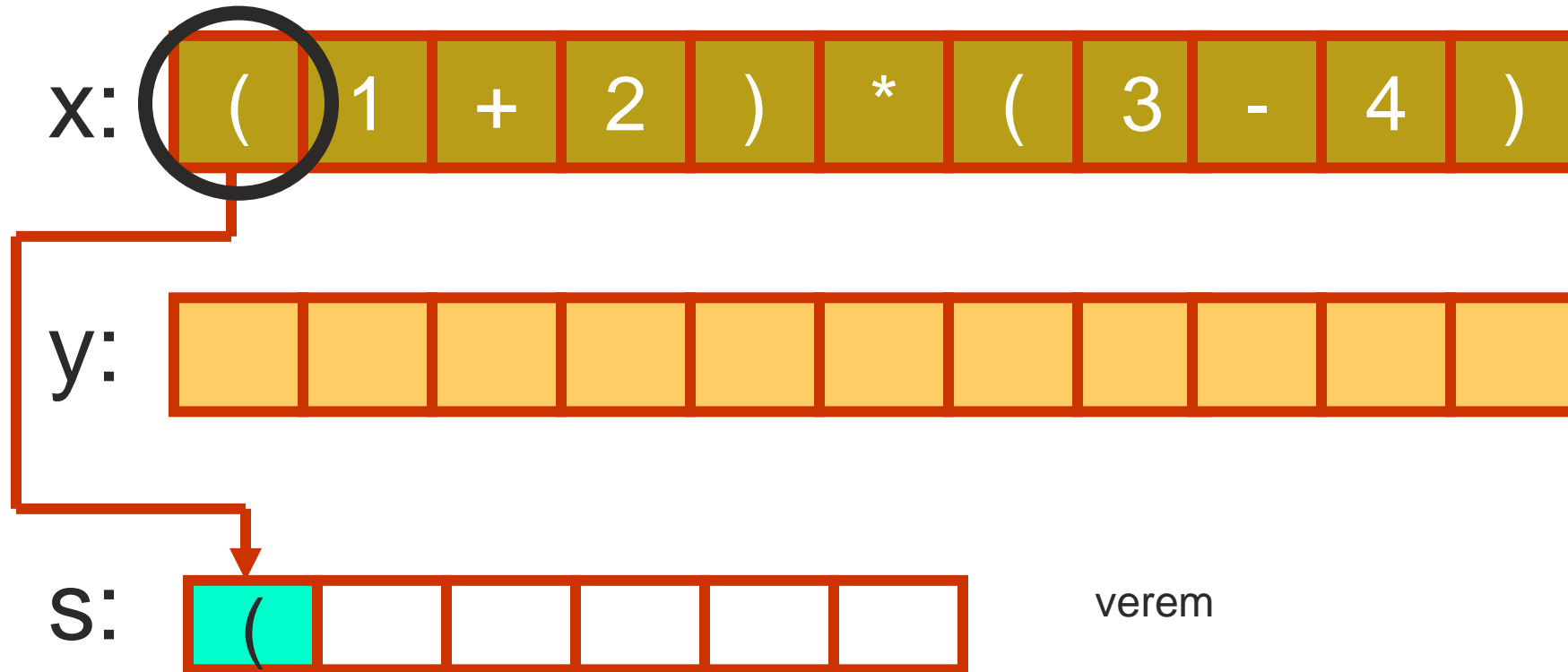
- Kifejezés lengyel formára hozása
- Szokásos forma
 - $(1 + 2) * (3 - 4)$
 - $(a + b * c) * (d * 3 - 4)$
- Fordított lengyel forma
 - $1\ 2\ +\ 3\ 4\ -\ *$
 - $abc\ *\ +d\ 3\ *\ 4\ -\ *$

Lengyel formára alakítás



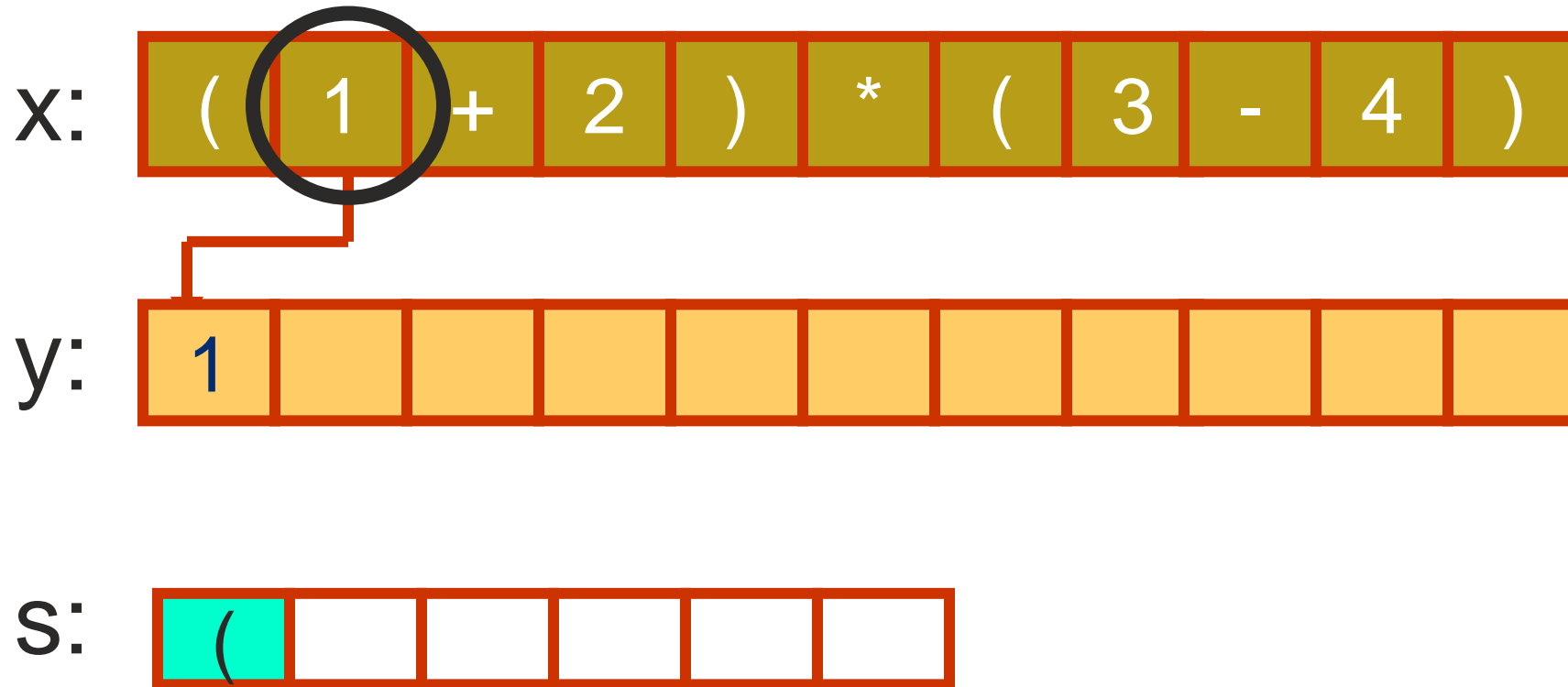
Az x sorozatot balról jobbra haladva dolgozzuk fel.
A sorozat végét ; jelzi.

Lengyel formára alakítás



A következő szimbólum **nyitózárójel**: **tegyük a verembe.**

Lengyel formára alakítás



A következő szimbólum operandus: írjuk ki.

Lengyel formára alakítás

x:

(1	+	2)	*	(3	-	4)
---	---	---	---	---	---	---	---	---	---	---

y:

1										
---	--	--	--	--	--	--	--	--	--	--

s:

(
---	--	--	--	--	--

A következő szimbólum **operátor**:

1. **legfeljebb a nyitózárójelig** kivesszük a veremből az operátornál nem kisebb prioritású operátorokat és **kiírjuk** azokat,

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

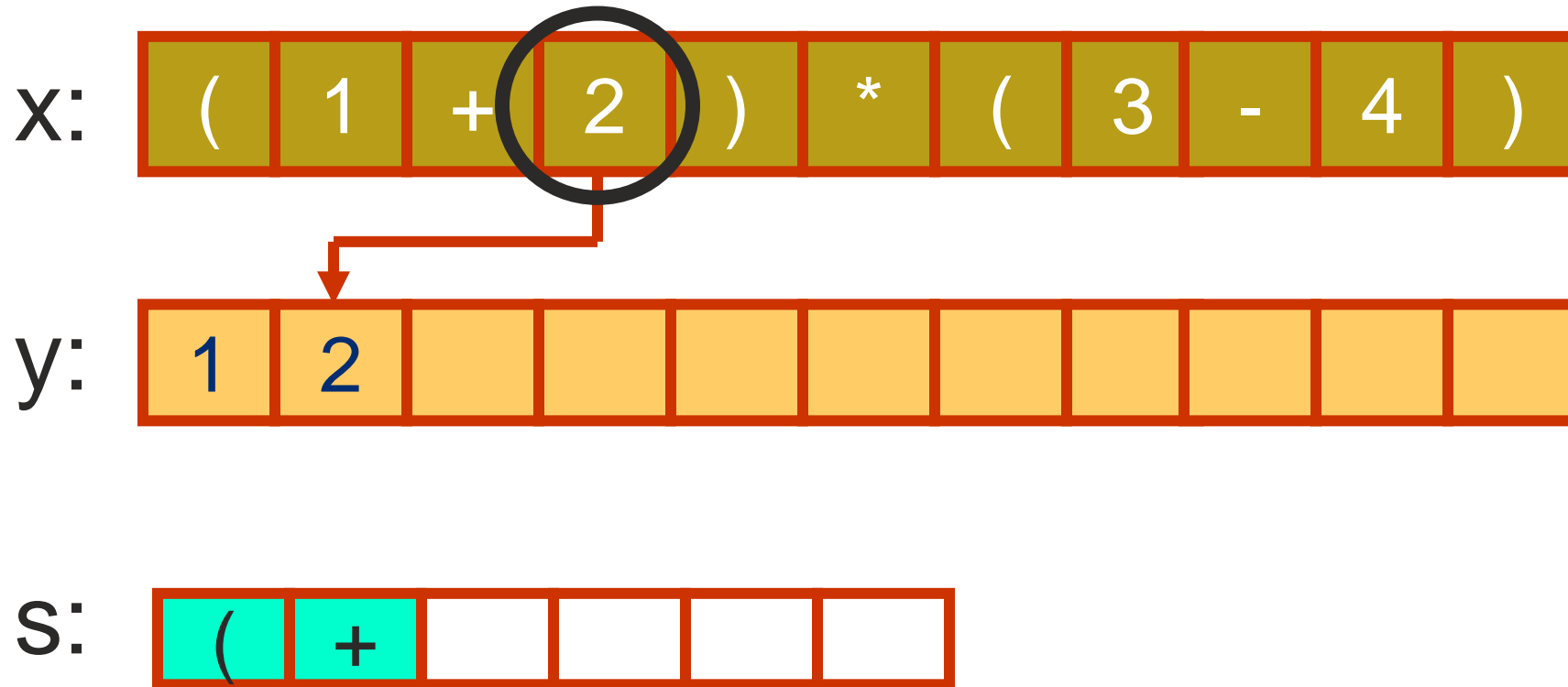
y: 1

S: (+

A következő szimbólum **operátor**:

1. legfeljebb a nyitózárójelig kivesszük a veremből az operátornál nem kisebb prioritású operátorokat és kiírjuk azokat,
2. Ezt az operátort betesszük a verembe.

Lengyel formára alakítás



A következő szimbólum operandus: írjuk ki.

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

y: 1 2 +

s: (+

A következő szimbólum csukózárójel:

1. Írjuk ki a verem tetején lévő elemeket egészen a nyitózárrójelig.

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

y: 1 2 +

s: (

A következő szimbólum **csukózárójel**:

1. Írjuk ki a verem tetején lévő elemeket egészen a nyitózárrójelig.
2. Vegyük ki a verem tetejéről a nyitózárrójelet.

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

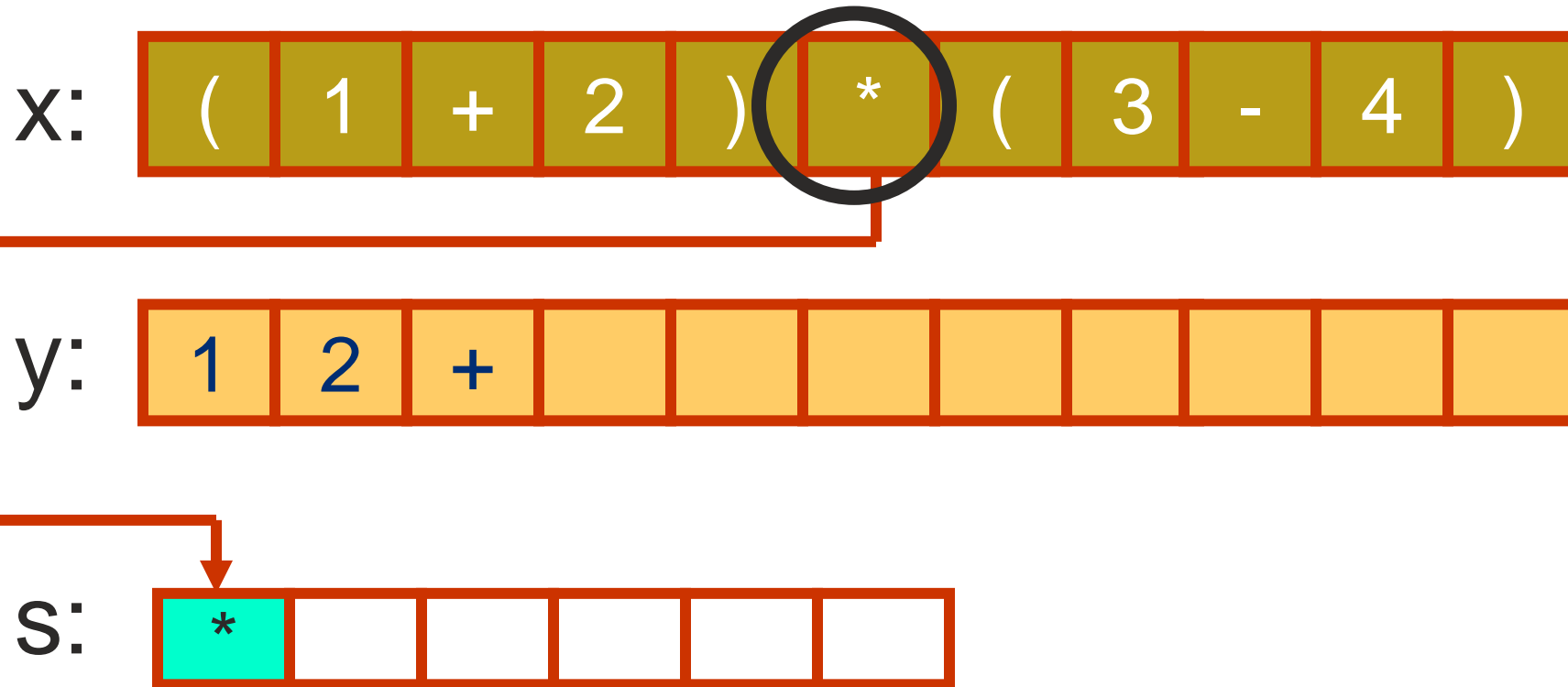
y: 1 2 +

s:

A következő szimbólum **operátor**:

1. **legfeljebb a nyitózárójelig** kivesszük a veremből az operátornál nem kisebb prioritású operátorokat és **kiírjuk** azokat,

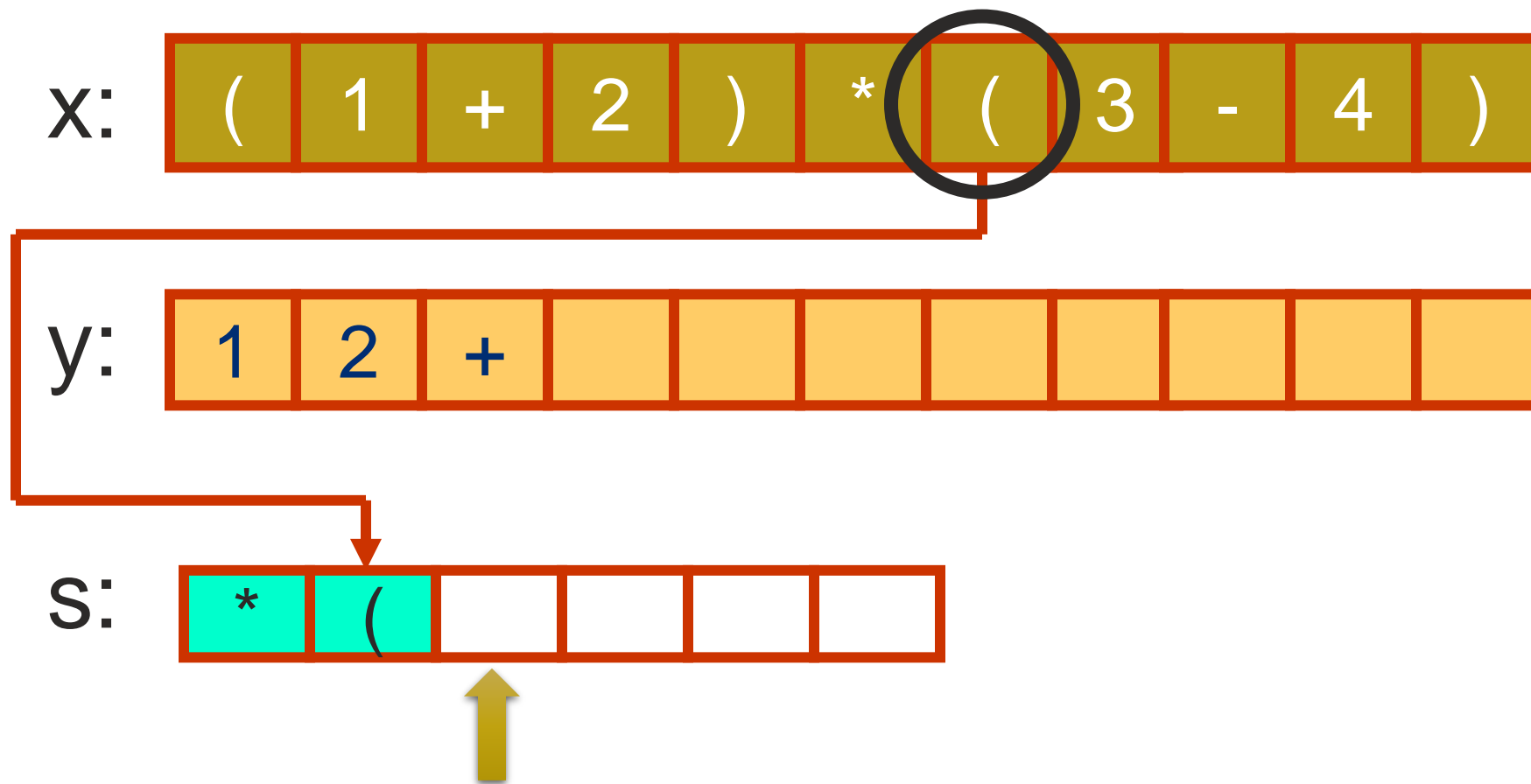
Lengyel formára alakítás



A következő szimbólum **operátor**:

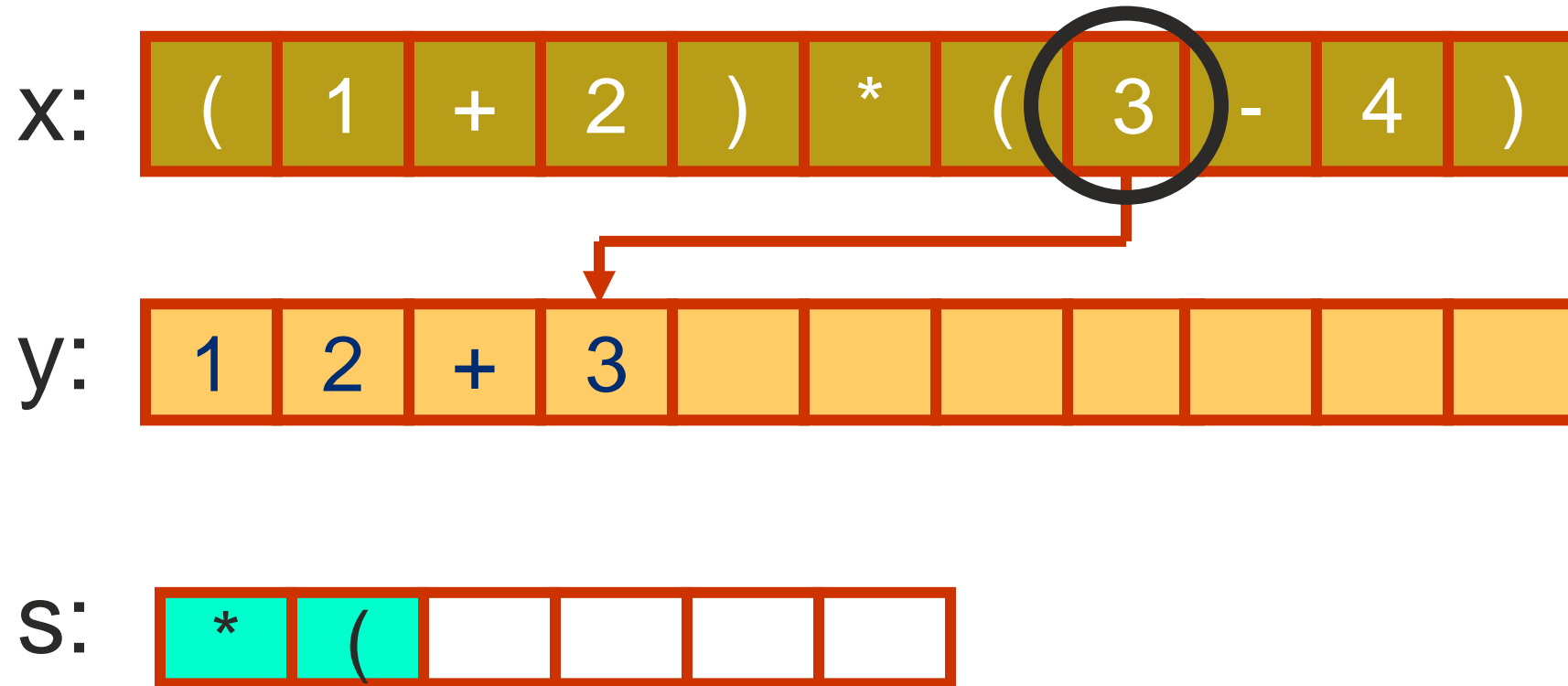
1. legfeljebb a nyitózárójelig kivesszük a veremből az operátornál nem kisebb prioritású operátorokat és kiírjuk azokat,
2. Ezt az operátort betesszük a verembe.

Lengyel formára alakítás



A következő szimbólum nyitózárójel: **tegyük a verembe.**

Lengyel formára alakítás



A következő szimbólum operandus: írjuk ki.

Lengyel formára alakítás

x:

(1	+	2)	*	(3	-	4)
---	---	---	---	---	---	---	---	---	---	---

y:

1	2	+	3							
---	---	---	---	--	--	--	--	--	--	--

S:

*	(
---	---	--	--	--	--

A következő szimbólum **operátor**:

1. **legfeljebb a nyitózárrójelig kivesszük a veremből az** operátornál nem kisebb prioritású operátorokat és **kiírjuk** azokat,

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

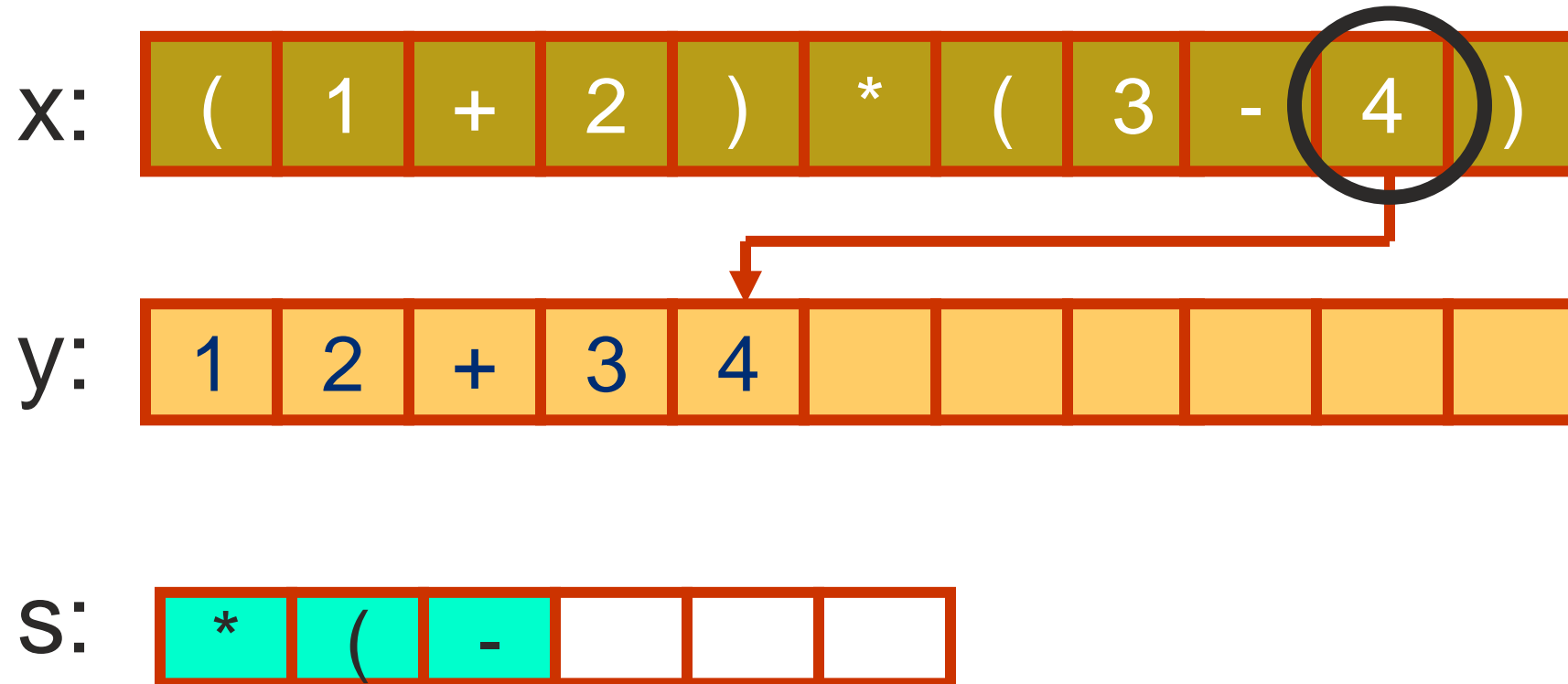
y: 1 2 + 3

S: * (-

A következő szimbólum **operátor**:

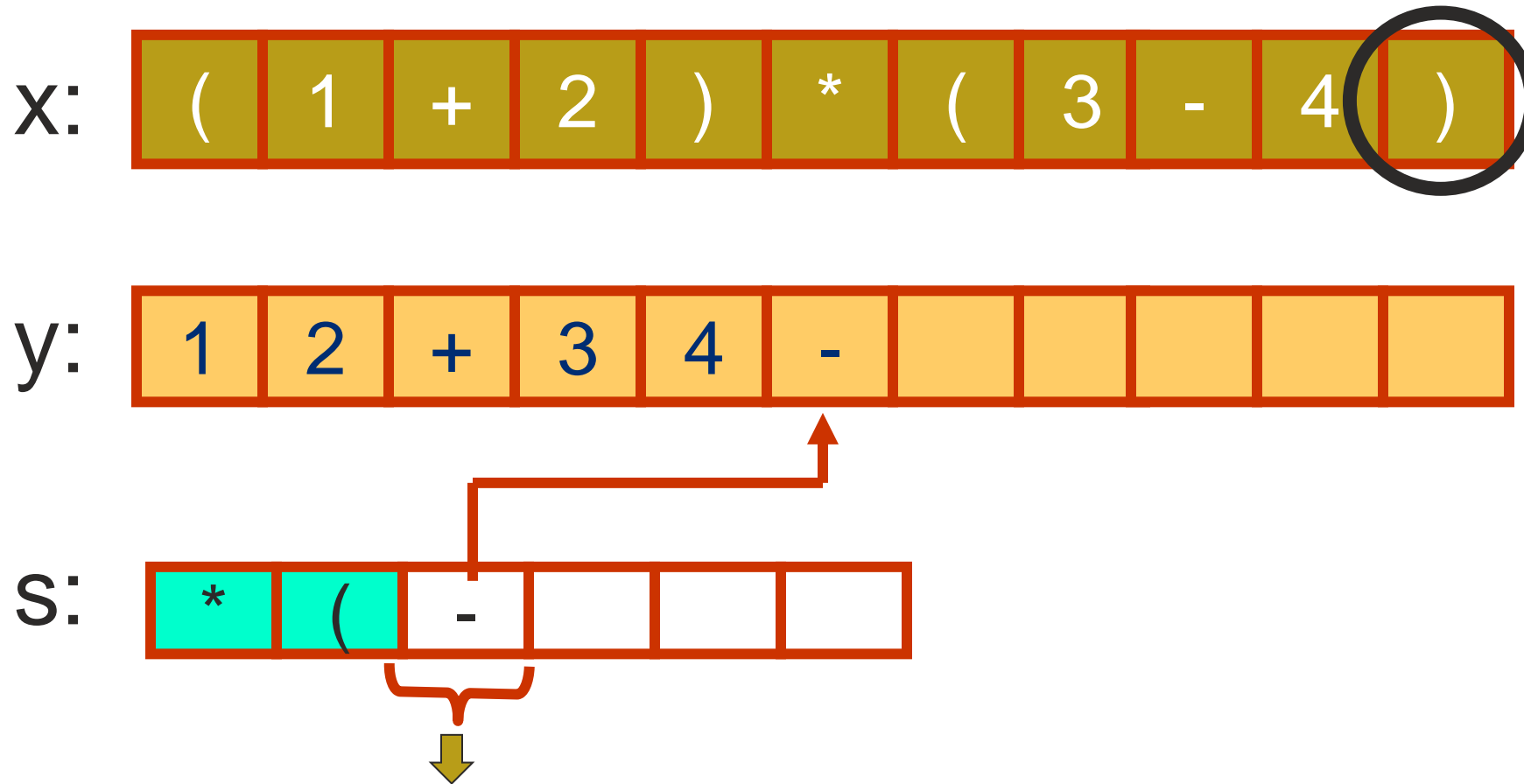
1. legfeljebb a nyitózárrójelig kivesszük a veremből az operátornál nem kisebb prioritású operátorokat és kiírjuk azokat,
2. Ezt az operátort betesszük a verembe.

Lengyel formára alakítás



A következő szimbólum operandus: írjuk ki.

Lengyel formára alakítás



A következő szimbólum **csukózárójel**:

1. Írjuk ki a verem tetején lévő elemeket egészen a nyitózárójelig.

Lengyel formára alakítás

x: (1 + 2) * (3 - 4)

y: 1 2 + 3 4 -

S: * (

A következő szimbólum **csukózárójel**:

1. Irjuk ki a verem tetején lévő elemeket egészen a nyitózárrójelig.
2. Vegyük ki a verem tetejéről a nyitózárrójelt.

Lengyel formára alakítás

x:

(1	+	2)	*	(3	-	4)	
---	---	---	---	---	---	---	---	---	---	---	--

y:

1	2	+	3	4	-	*				
---	---	---	---	---	---	---	--	--	--	--

s:

*					
---	--	--	--	--	--

Elértük a kifejezés végét:
Írjuk ki a veremben lévő összes elemet.

Lengyel formára alakítás

- Tegyük fel, hogy az x „token”-ekből álló sor tartalmazza a szintaktikailag helyes kifejezést,
- Egy token lehet:
 - Operandus, vagy
 - Operátor (bináris), vagy
 - „(” vagy „)”
 - az y sorba hozzuk létre a postfix formájú kifejezést,
- közben felhasználva az s vermet, mely operátorokat és nyitózárójelet tartalmazhat

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor	
		s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
			y.in(s.pop)	y.in(s.pop)
			s.pop	s.push(e)
		¬s.isempty		
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty			
$\neg x.isempty$			
e ← x.out			
I / e=operandus?		N	
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
$\neg s.isempty$			
y.in(s.pop)			

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
I / y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

Lengyel formára alakítás

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

Lengyel forma kiértékelése

- $1\ 2\ +\ 3\ 4\ -\ *$
 - Kiértékelés után
 - -3

Lengyel forma kiértékelése

y:

1	2	+	3	4	-	*				
---	---	---	---	---	---	---	--	--	--	--

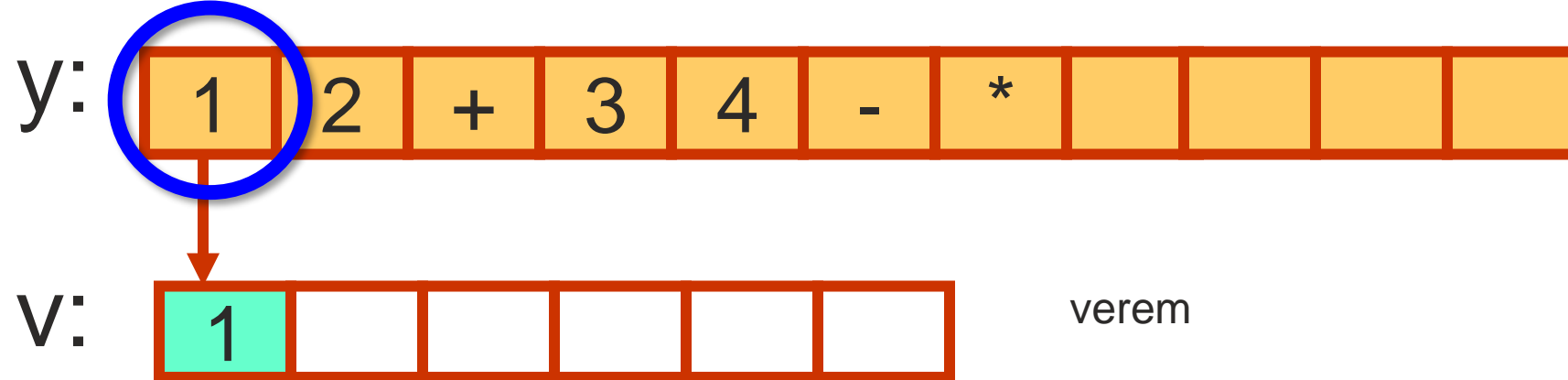
sorozat

V:

--	--	--	--	--	--

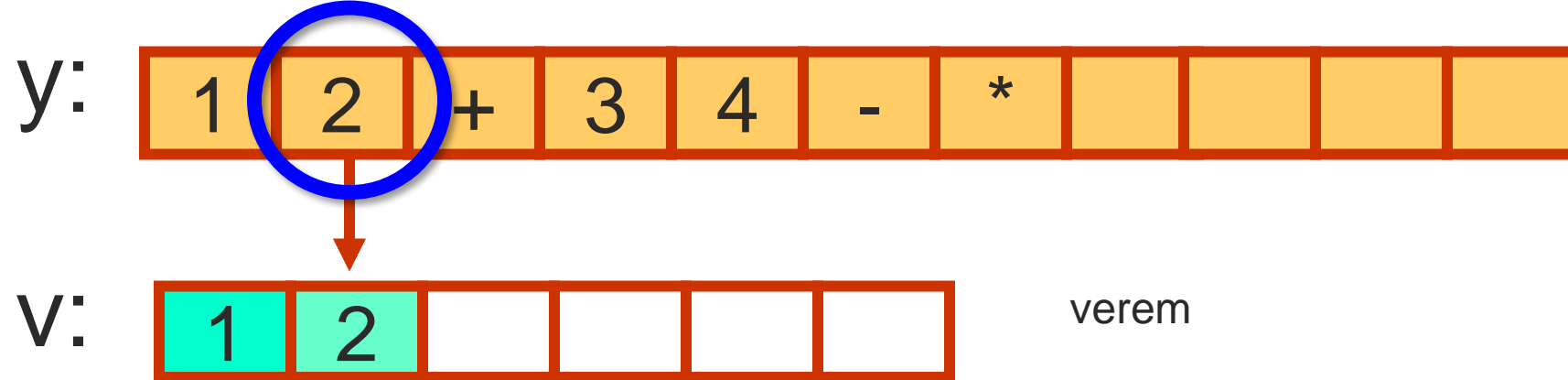
verem

Lengyel forma kiértékelése



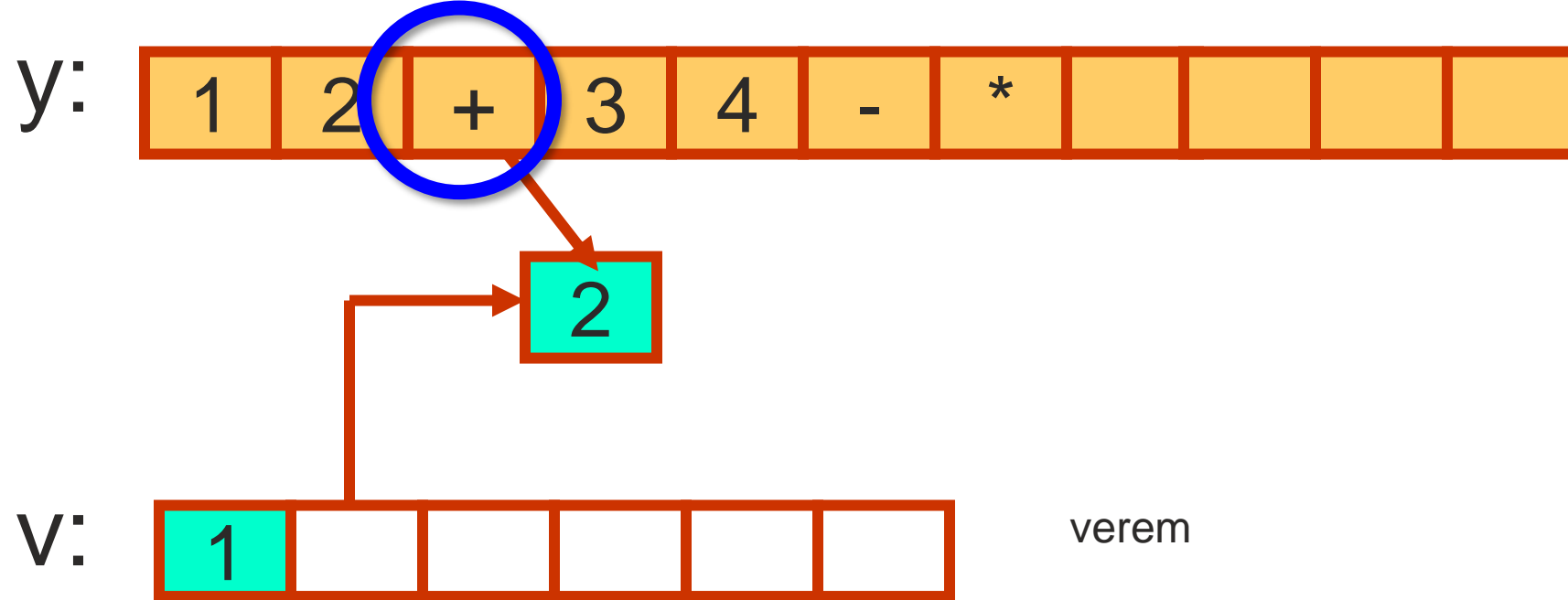
A következő szimbólum operandus:
Tegyük a verembe.

Lengyel forma kiértékelése



A következő szimbólum operandus:
Tegyük a verembe.

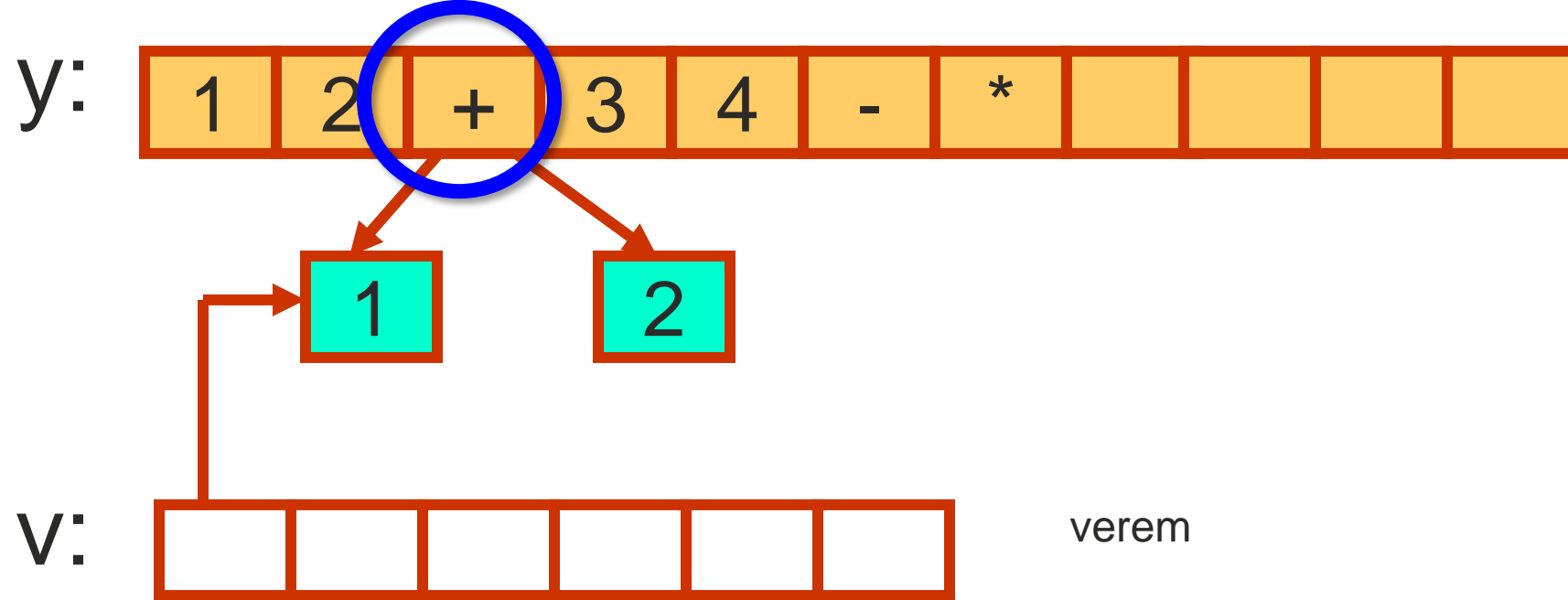
Lengyel forma kiértékelése



A következő szimbólum operátor:

1. Vegyük ki a veremből a második operandust.

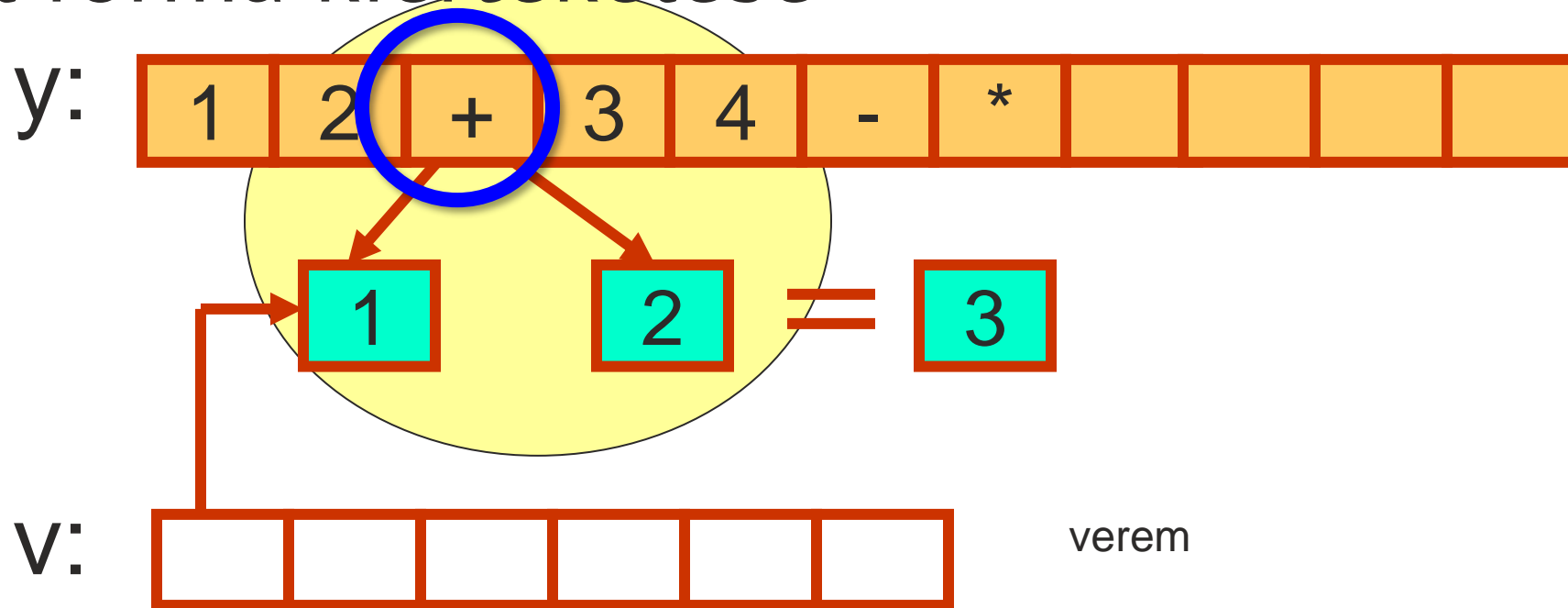
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. **Vegyük ki a veremből az első operandust.**

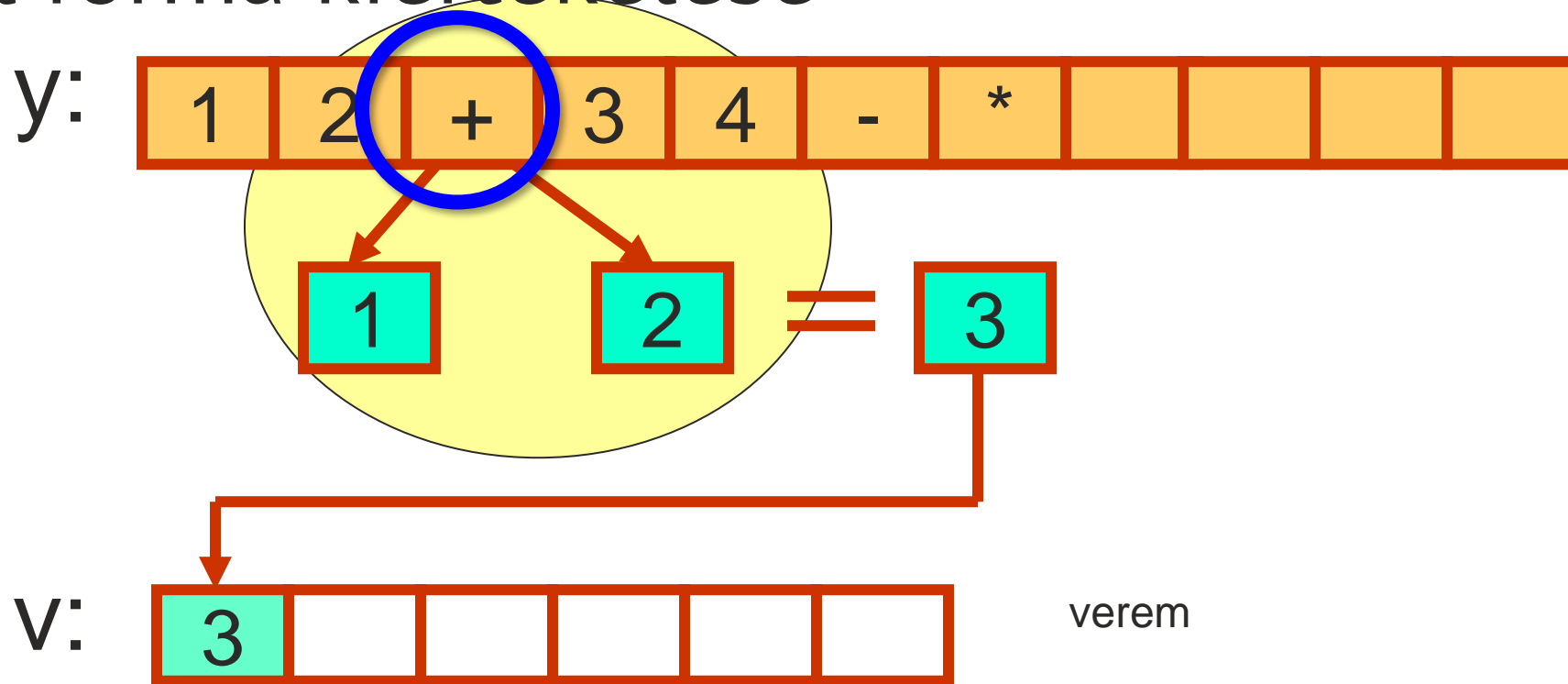
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.

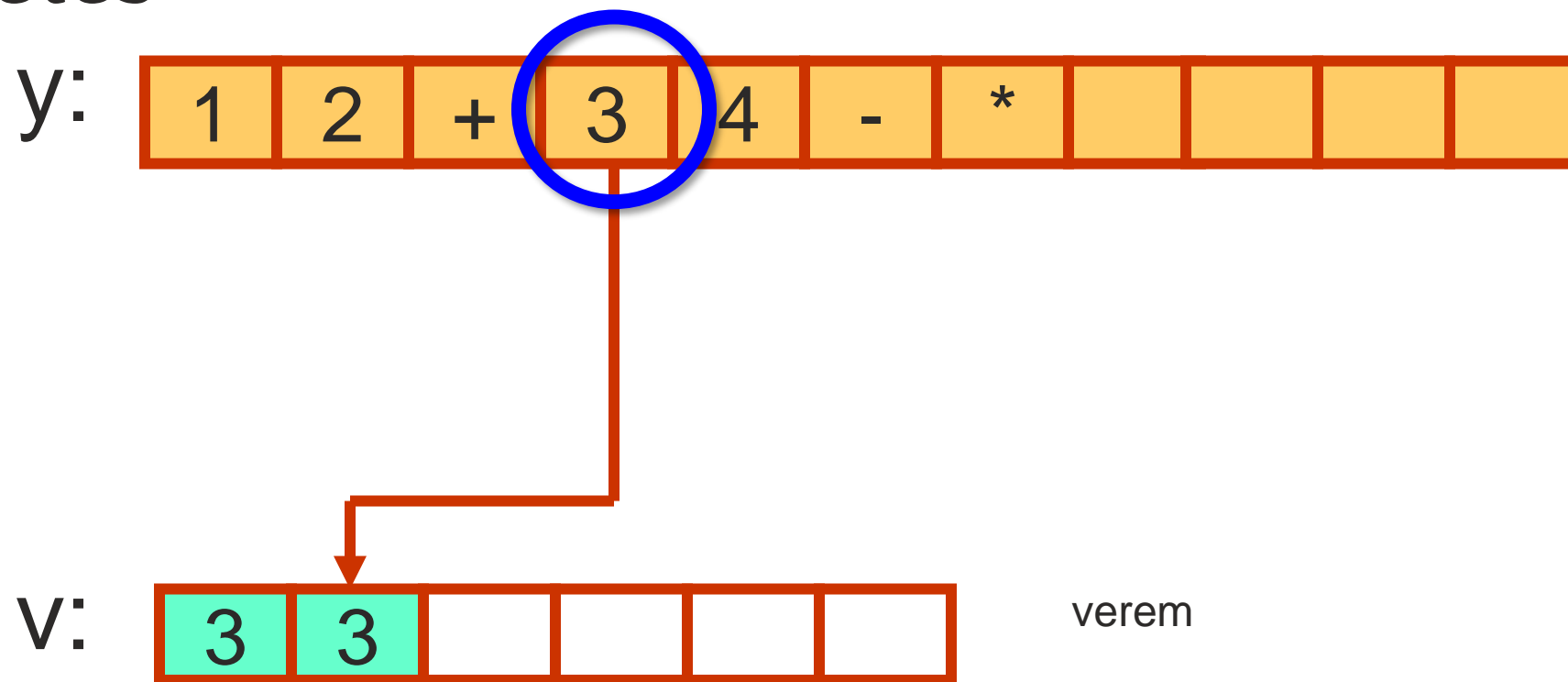
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

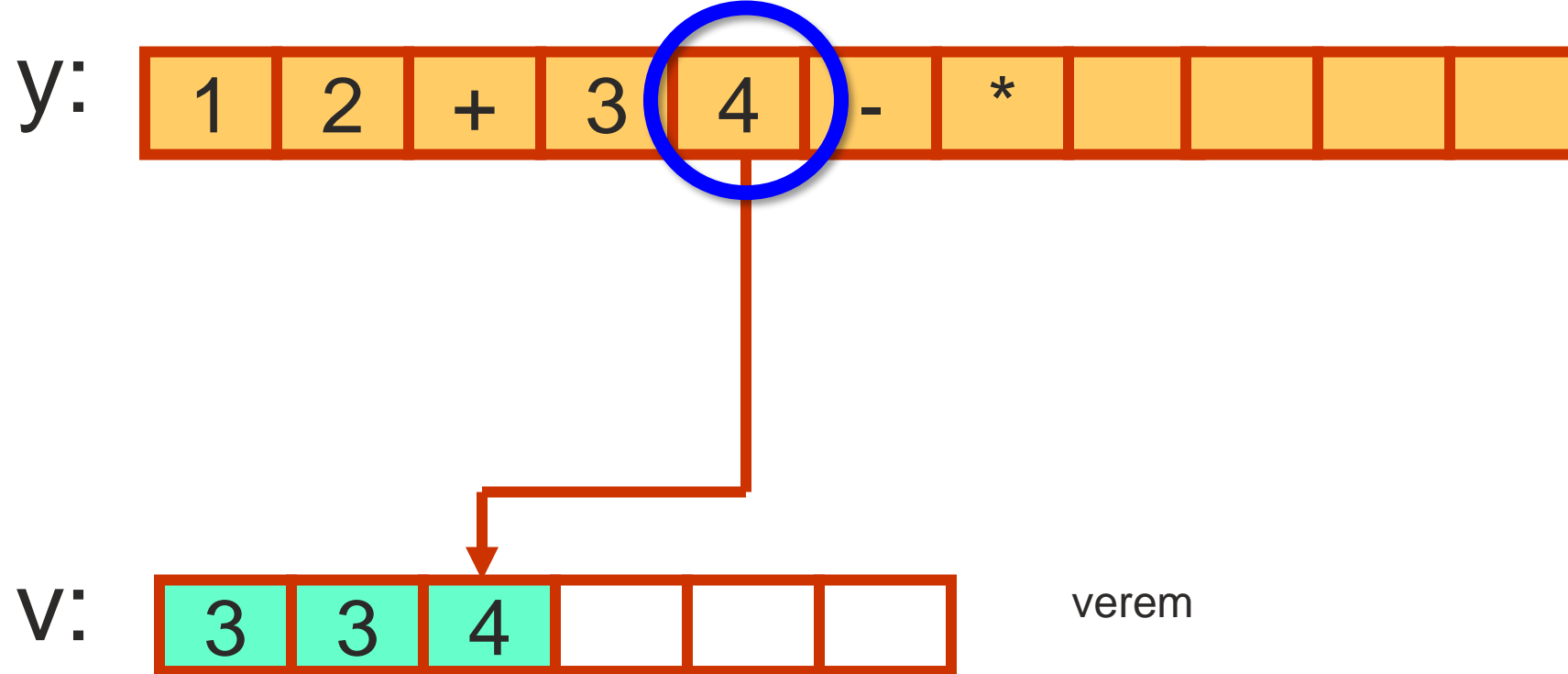
1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.
4. Az így kapott eredményt tegyük a verembe.

Kiértékelés



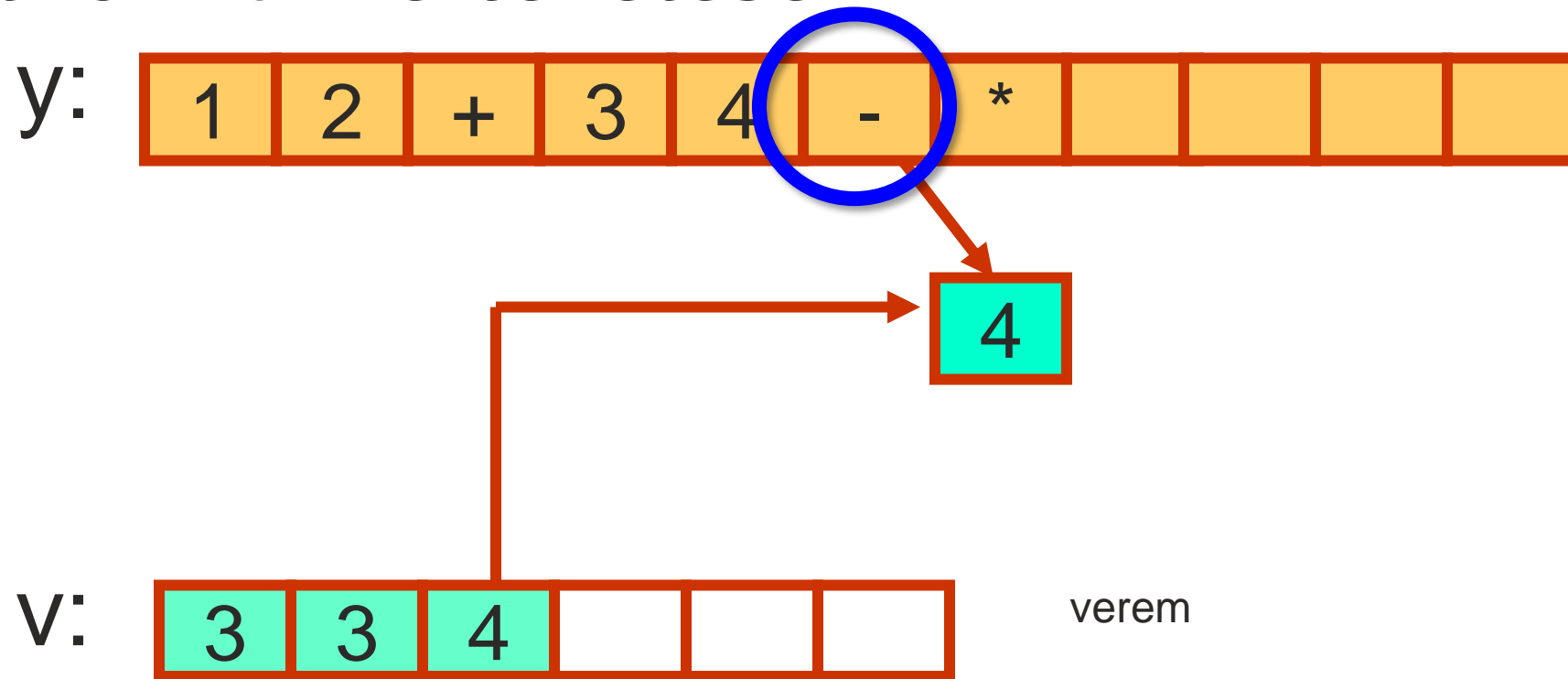
A következő szimbólum operandus:
Tegyük a verembe.

Lengyel forma kiértékelése



A következő szimbólum operandus:
Tegyük a verembe.

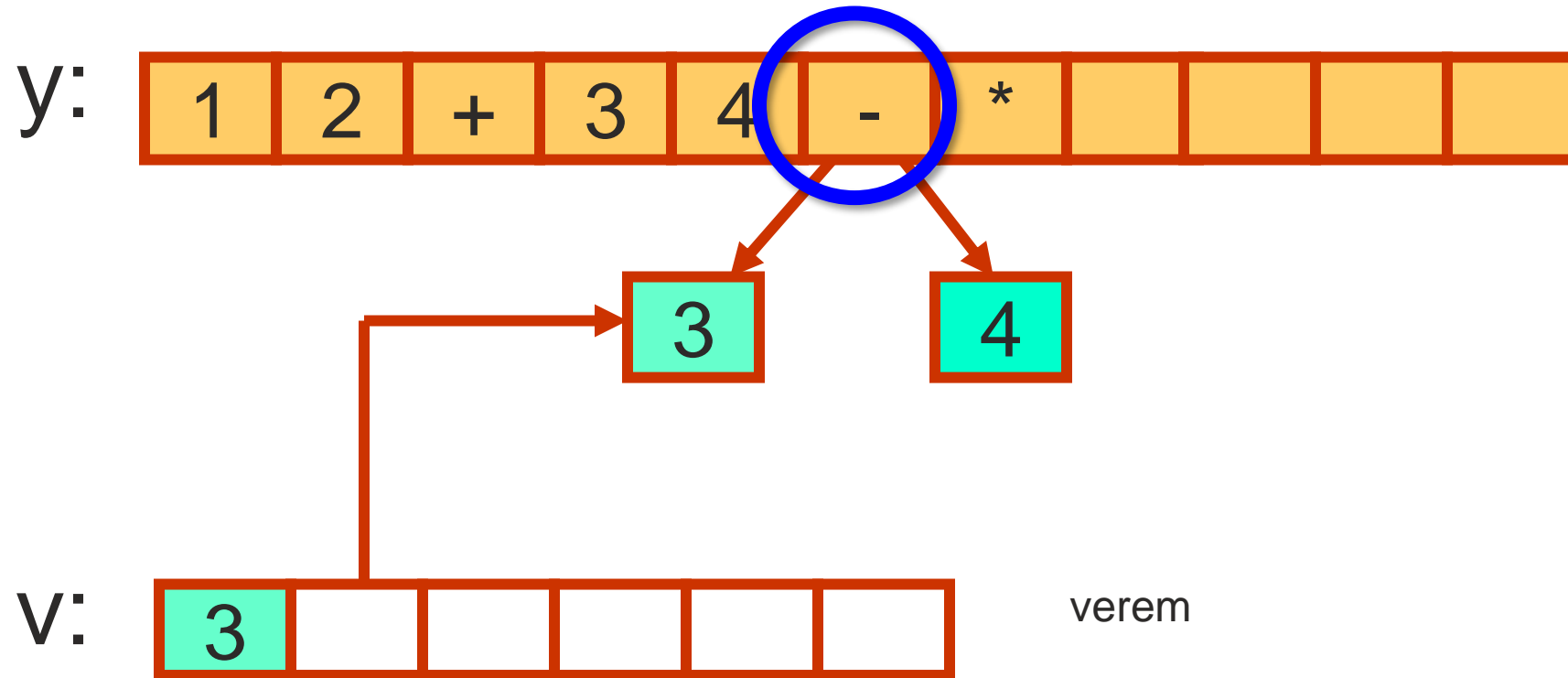
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust.

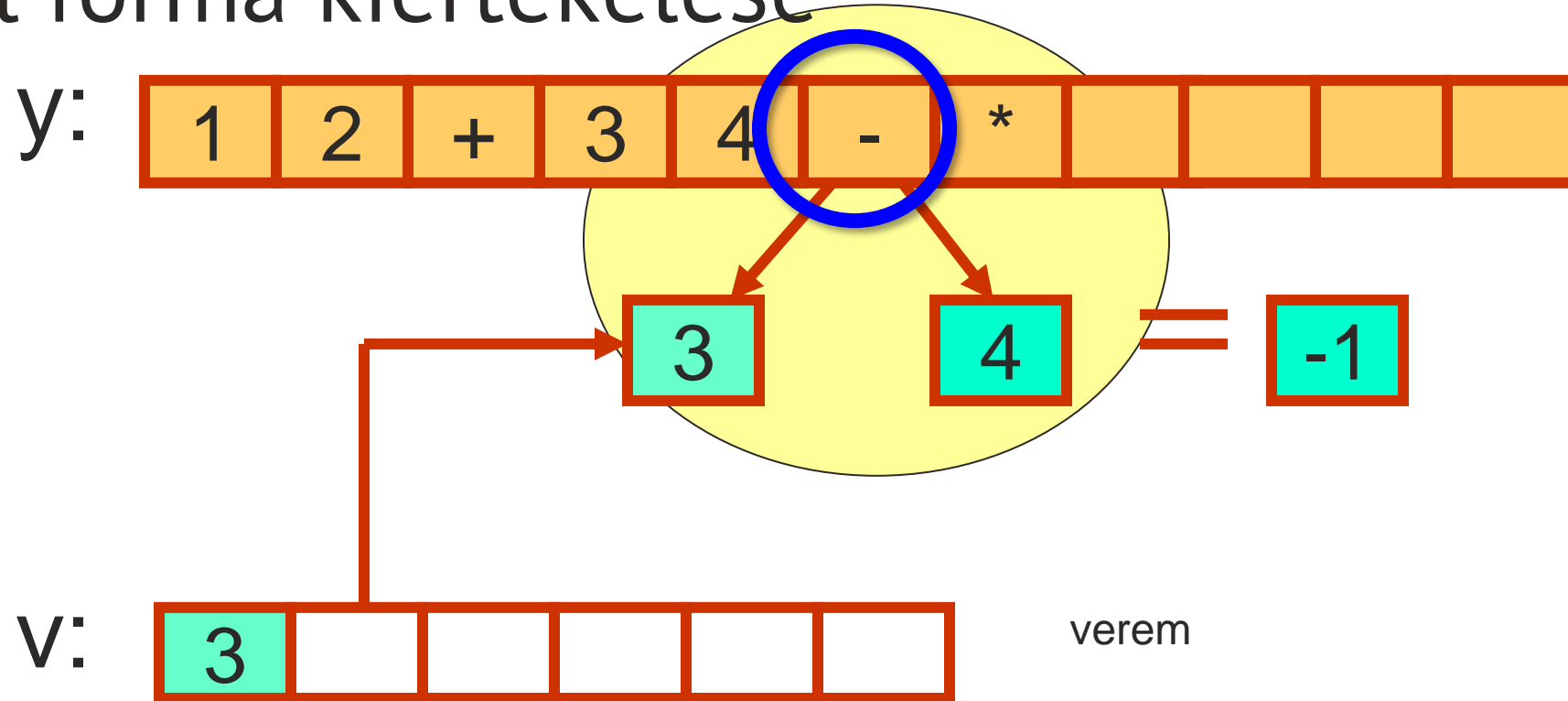
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.

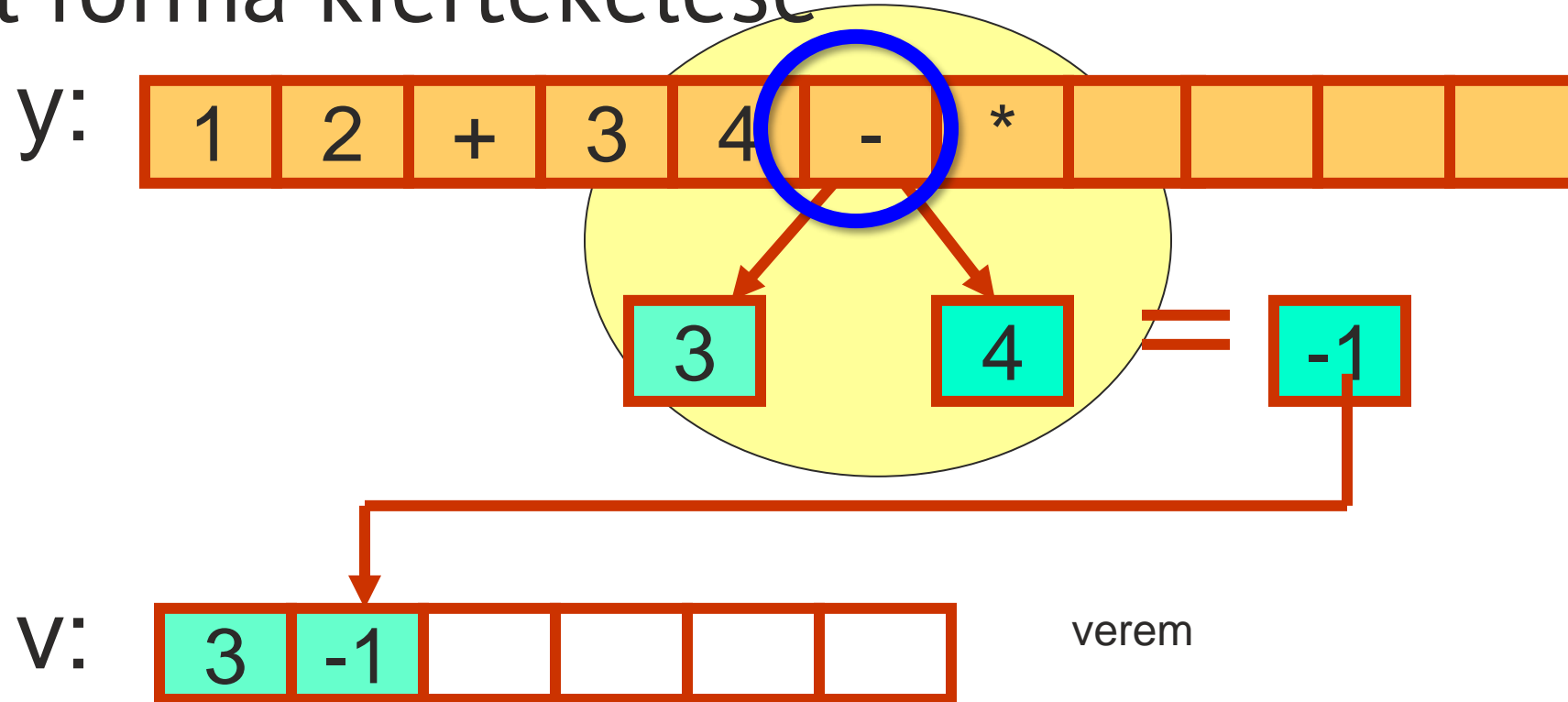
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.

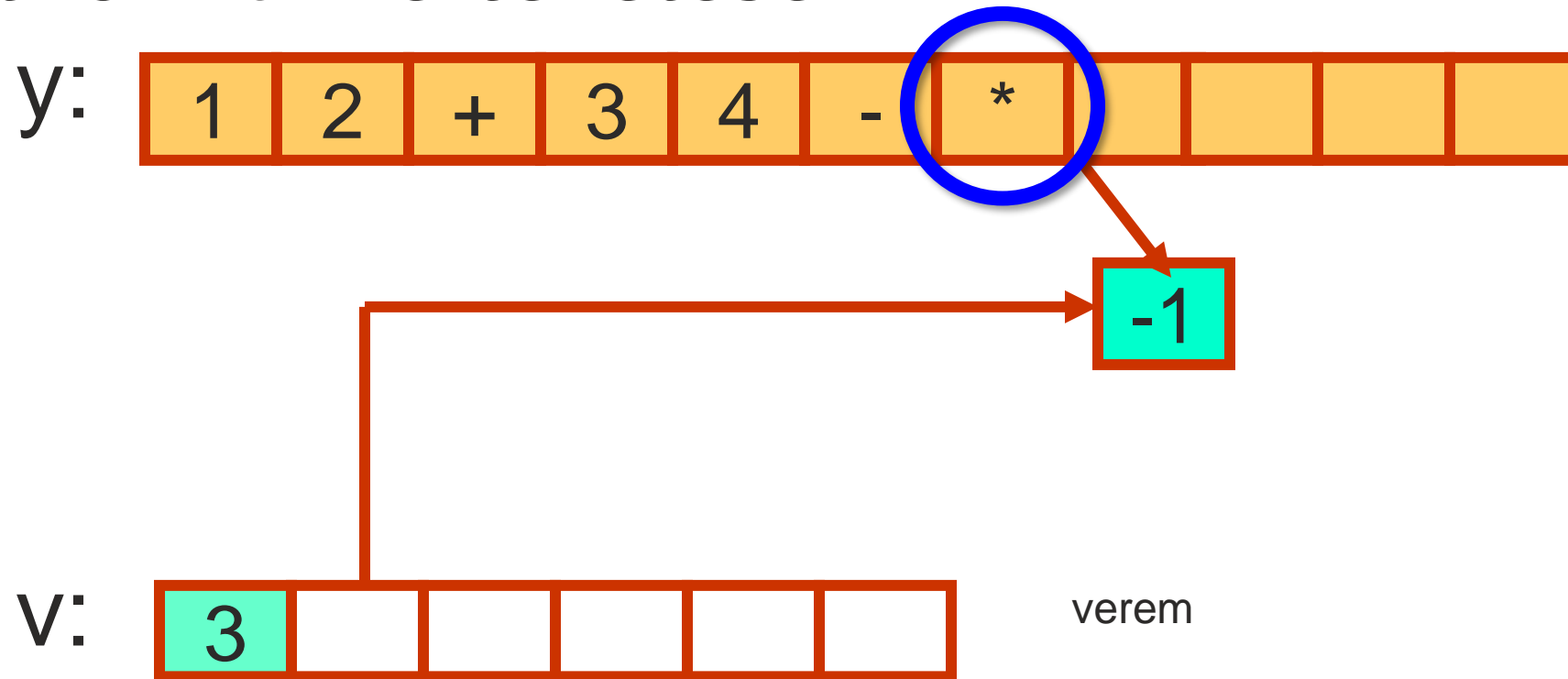
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.
4. Az így kapott eredményt tegyük a verembe.

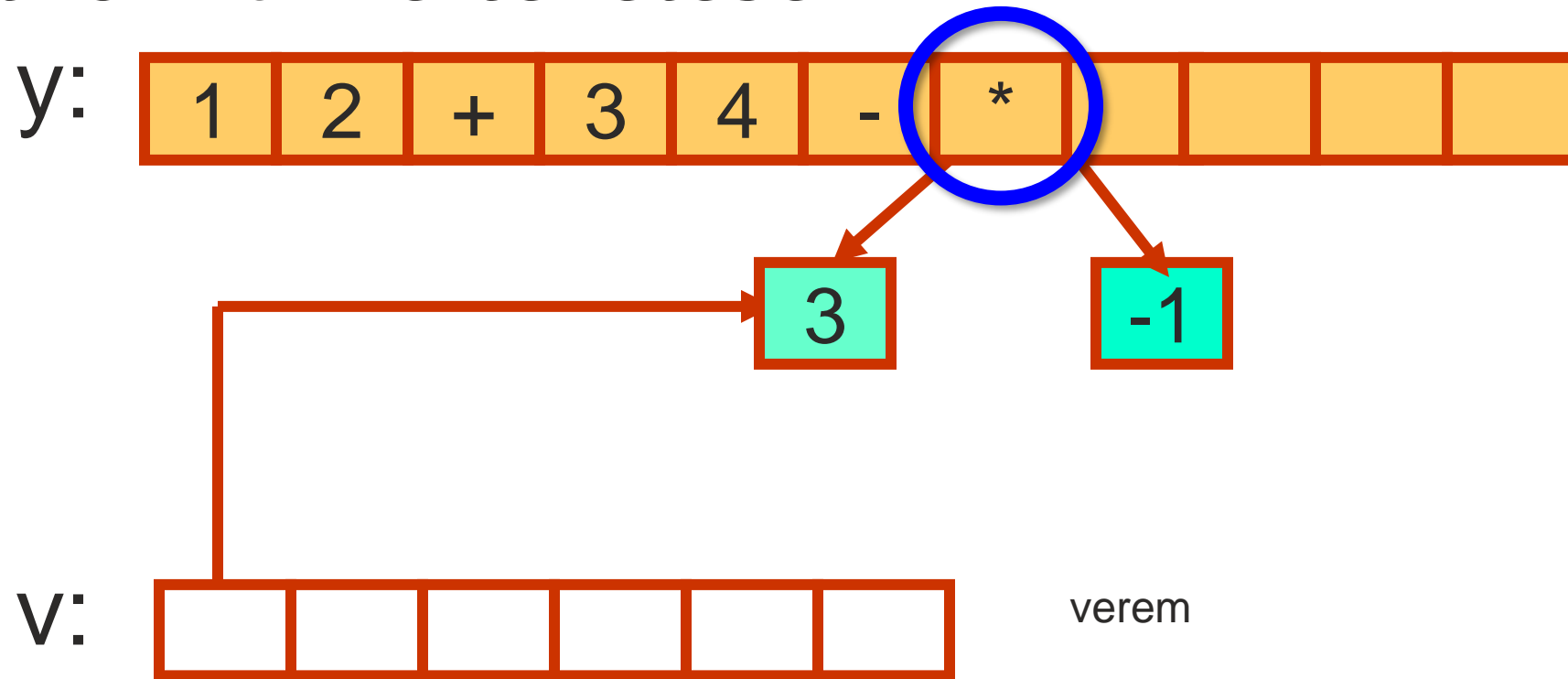
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust.

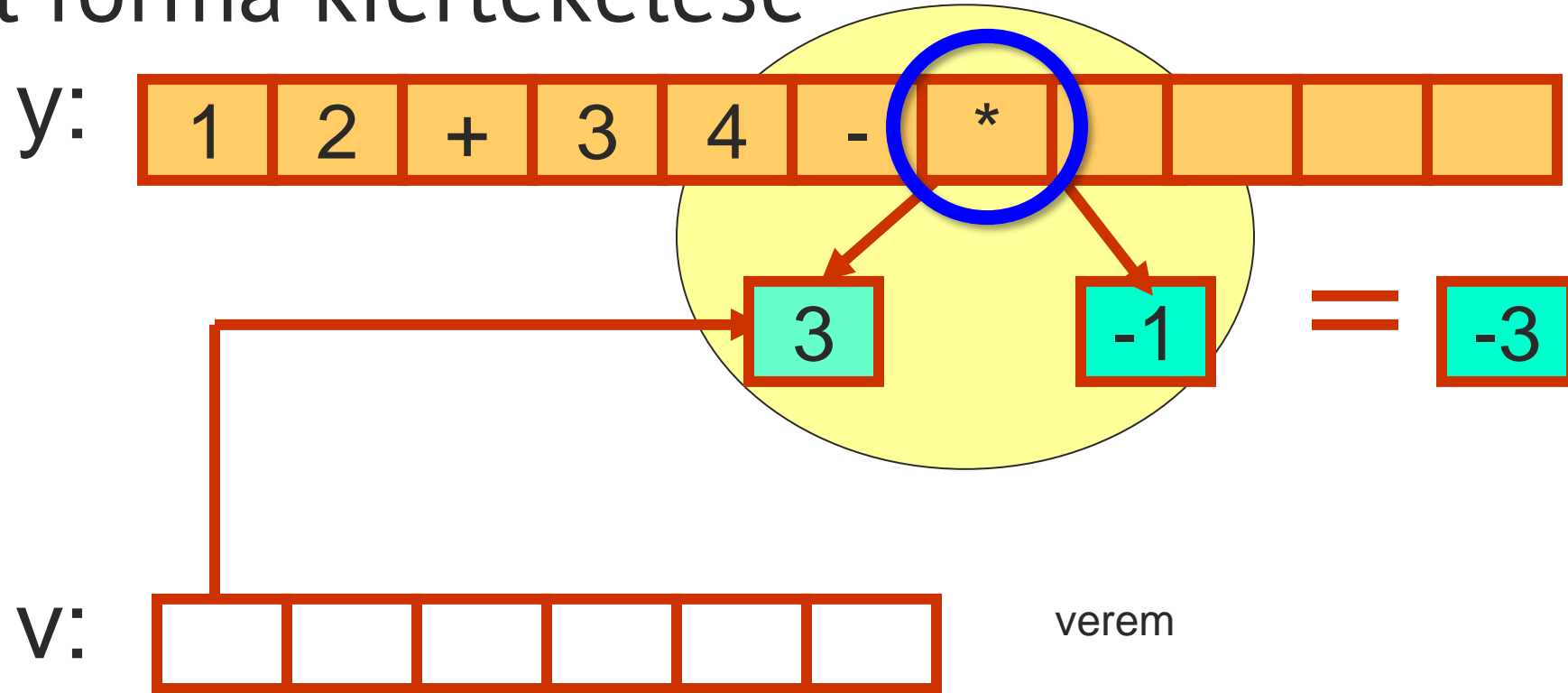
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.

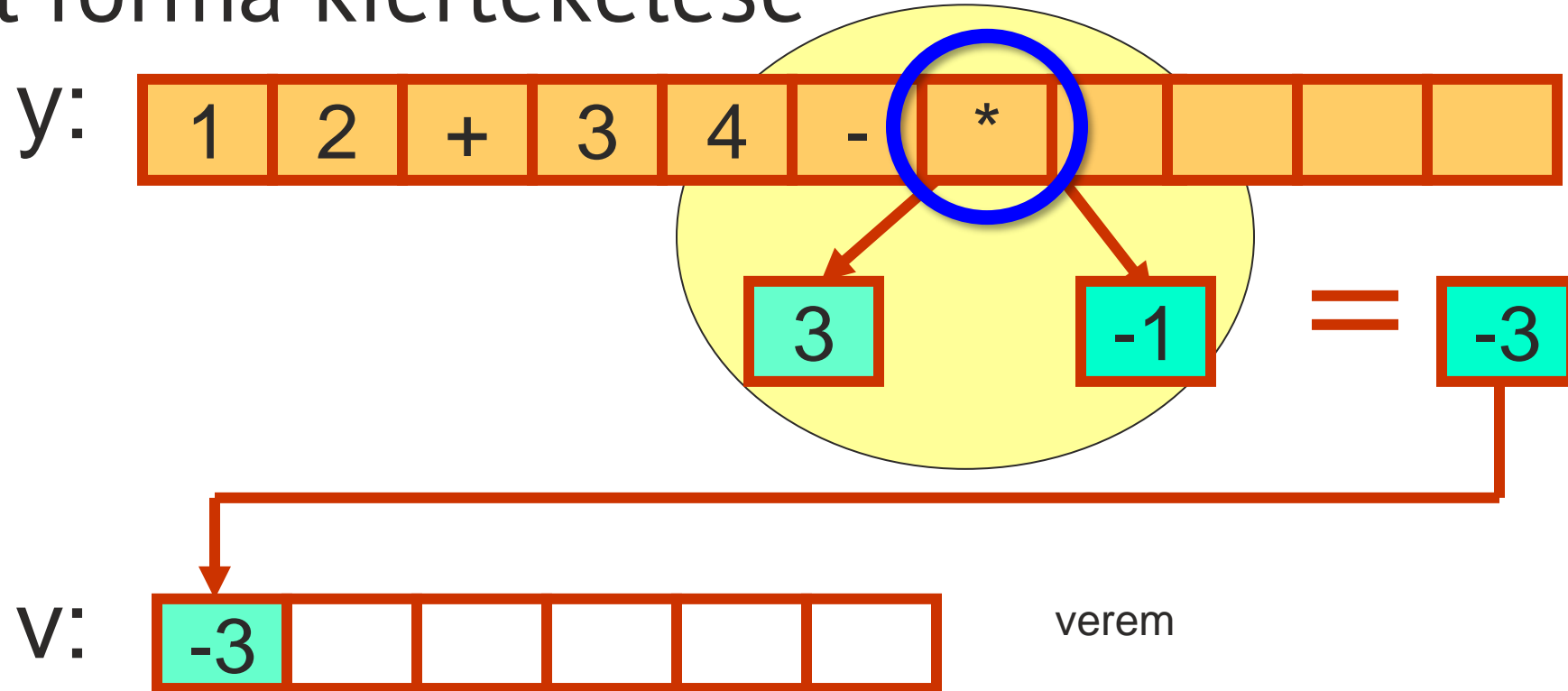
Lengyel forma kiértékelése



A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.

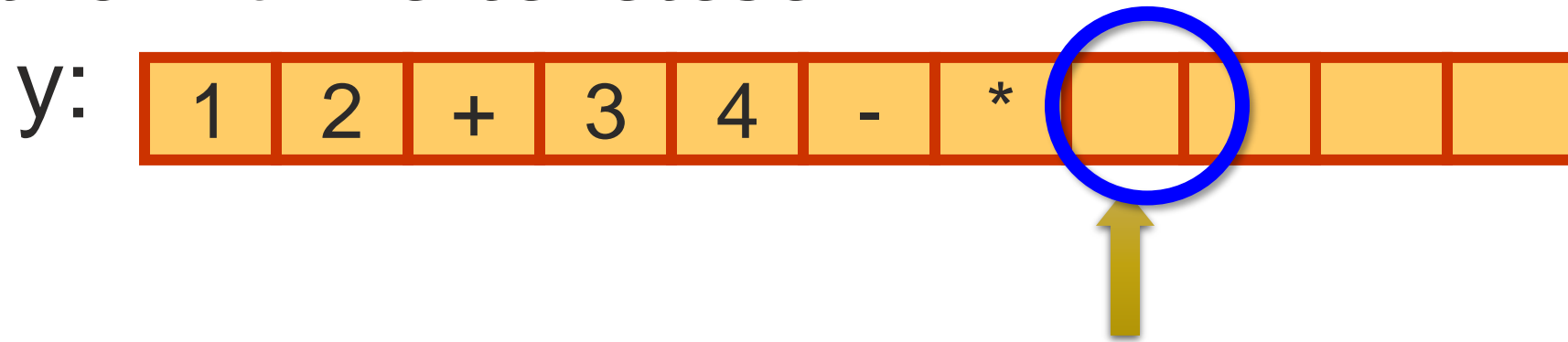
Lengyel forma kiértékelése



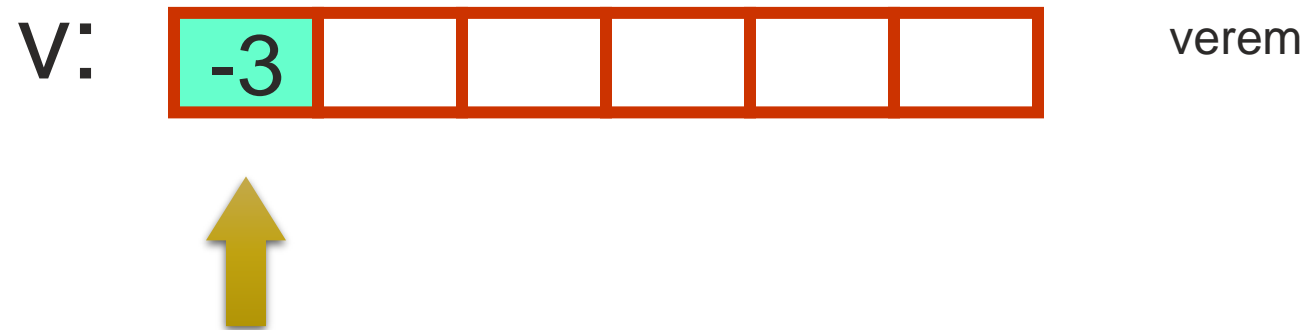
A következő szimbólum **operátor**:

1. Vegyük ki a veremből a második operandust
2. Vegyük ki a veremből az első operandust.
3. Számítsuk ki az adott műveleti jellel a kifejezés értékét.
4. Az így kapott eredményt tegyük a verembe.

Lengyel forma kiértékelése



Elértük a kifejezés végét



Az eredmény a verem tetején van.

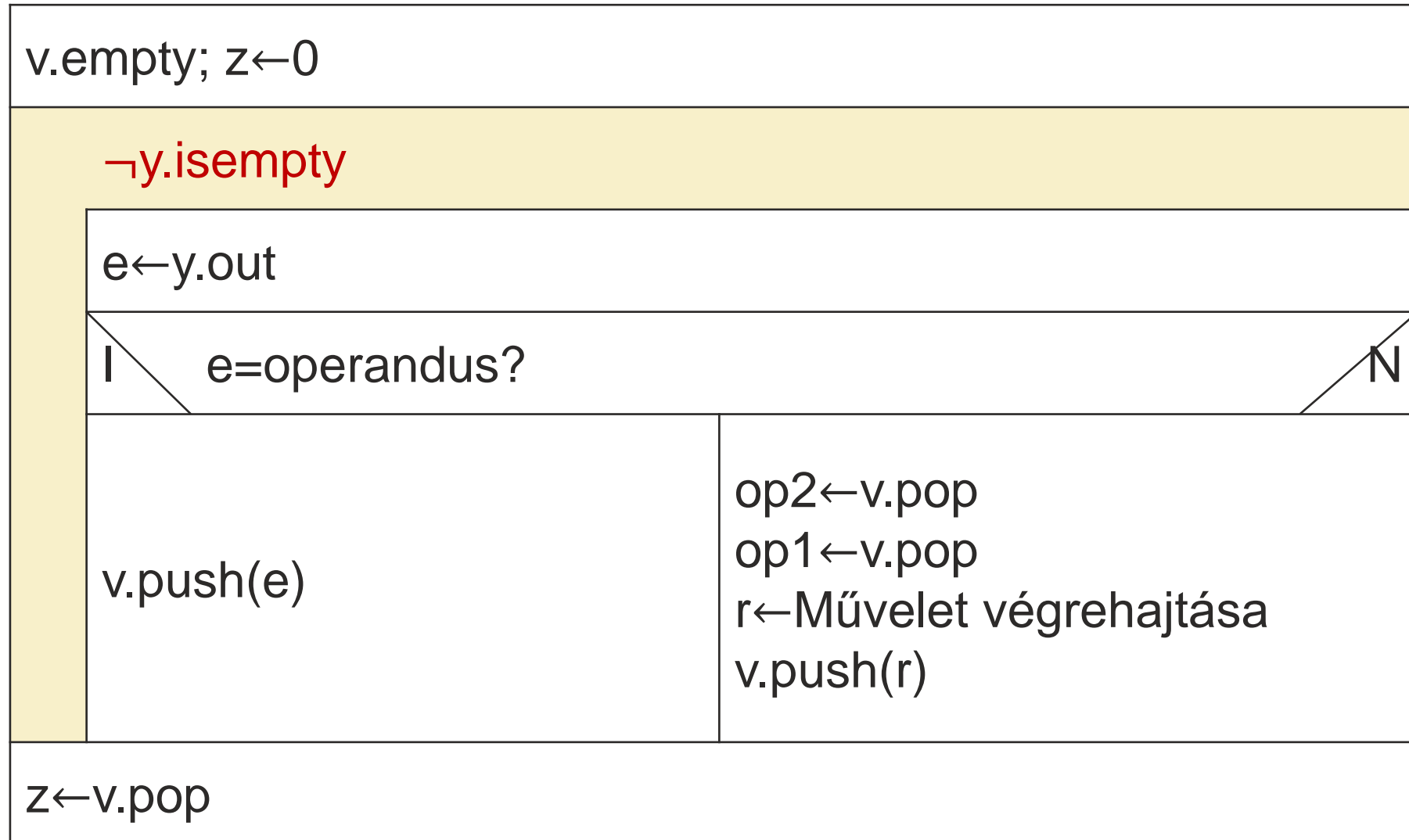
Lengyel forma kiértékelése

- Tegyük fel, hogy az y sor tartalmazza a postfix alakban lévő kifejezést,
- Értékeljük ki, felhasználva a v vermet, mely operandusokat tartalmazhat
- Az eredményt tároljuk a z változóban

Lengyel forma kiértékelése

v.empty; z←0	
¬y.isempty	
e←y.out	
I	N
e=operandus?	
v.push(e)	op2←v.pop op1←v.pop r←Művelet végrehajtása v.push(r)
z←v.pop	

Lengyel forma kiértékelése



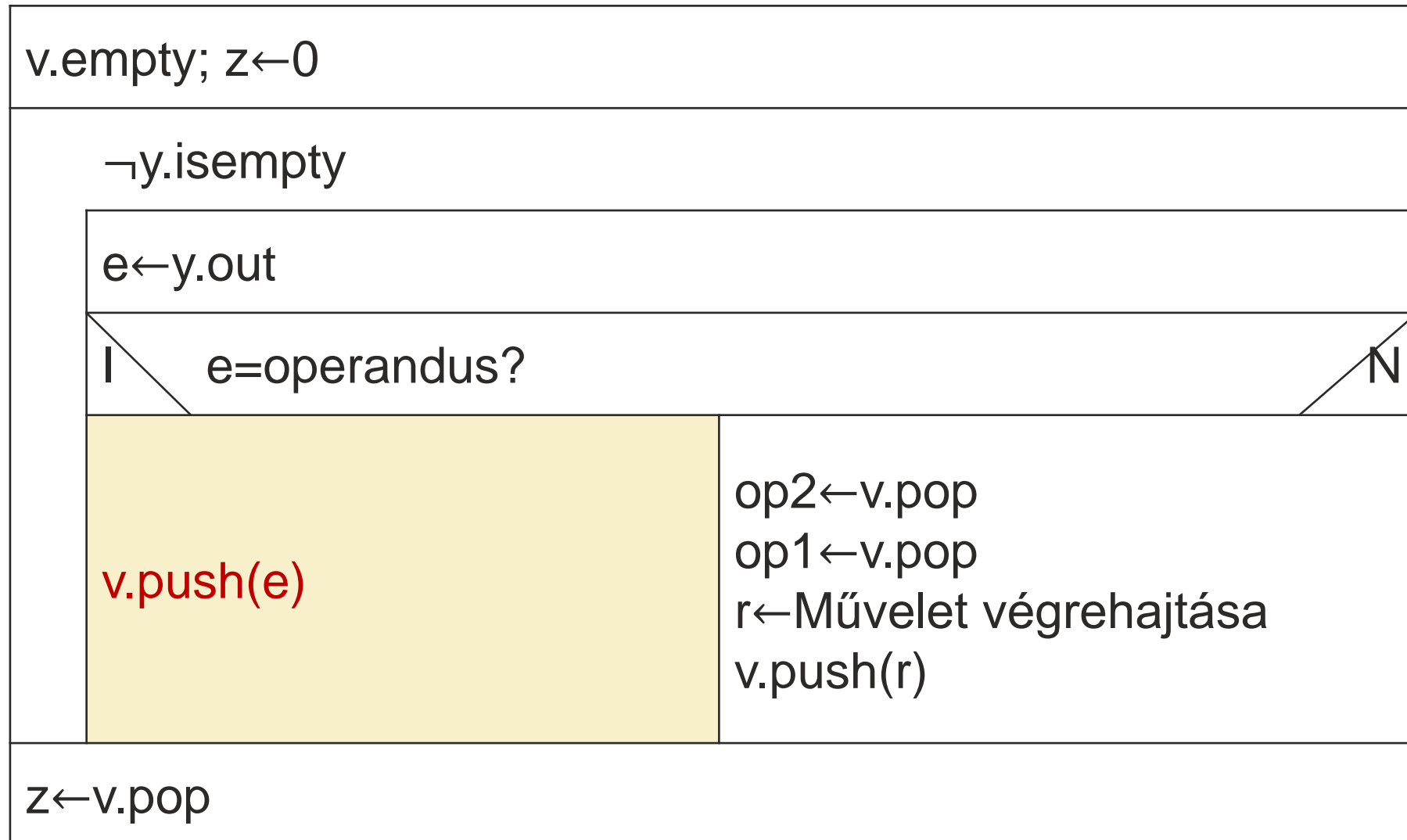
Lengyel forma kiértékelése

v.empty; z←0	
¬y.isempty	
e←y.out	
I	N
e=operandus?	
v.push(e)	op2←v.pop op1←v.pop r←Művelet végrehajtása v.push(r)
z←v.pop	

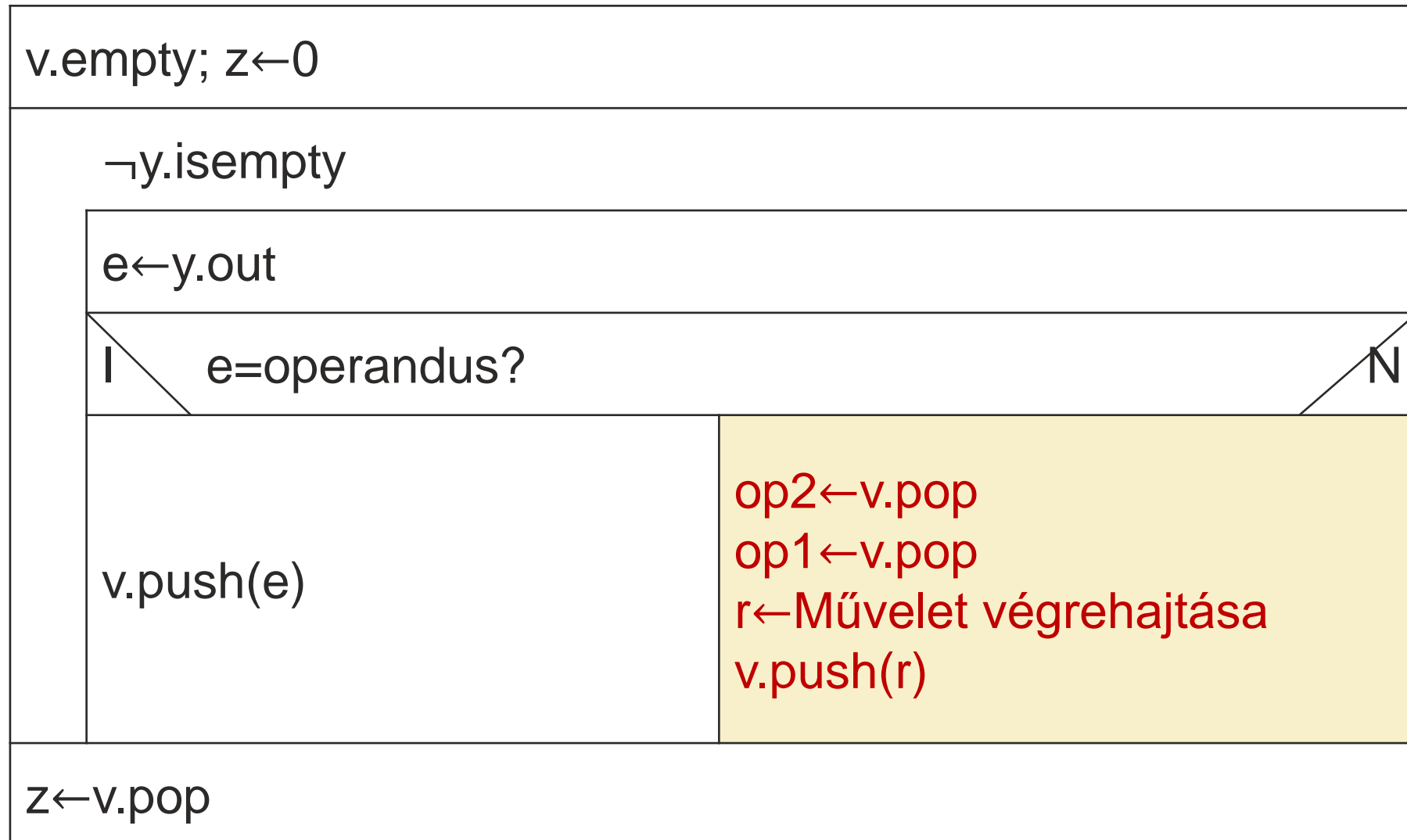
Lengyel forma kiértékelése

v.empty; z←0	
¬y.isempty	
e←y.out	
I / e=operandus? / N	
v.push(e)	op2←v.pop op1←v.pop r←Művelet végrehajtása v.push(r)
z←v.pop	

Lengyel forma kiértékelése



Lengyel forma kiértékelése



Lengyel forma kiértékelése

v.empty; z←0	
¬y.isempty	
e←y.out	
I	N
e=operandus?	
v.push(e)	op2←v.pop op1←v.pop r←Művelet végrehajtása v.push(r)
z←v.pop	

Lengyel forma kiértékelése

v.empty; z←0	
¬y.isempty	
e←y.out	
I	N
e=operandus?	
v.push(e)	op2←v.pop op1←v.pop r←Művelet végrehajtása v.push(r)
z←v.pop	

Tömbök

Következő téma

De előtte még egy animáció a lengyel formához

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Input:
x – sor

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--

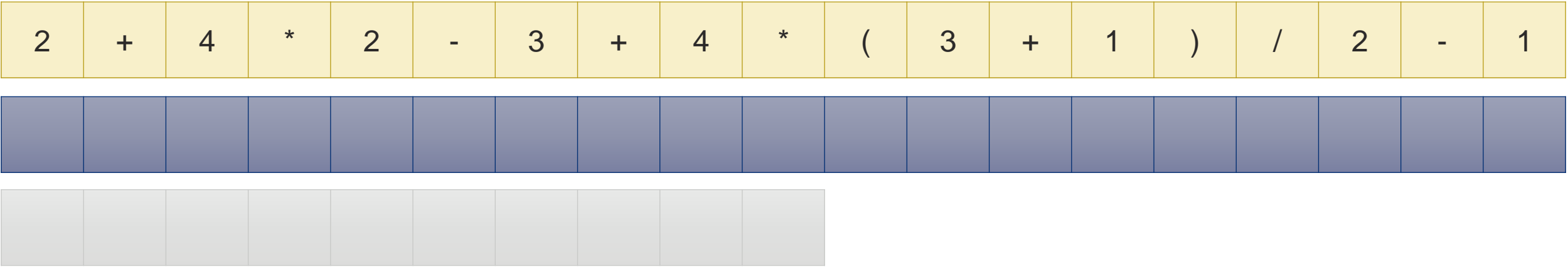
Input:

x – sor

Feldolgozás során használt tárolók:

y – sor (lengyelforma lesz benne)

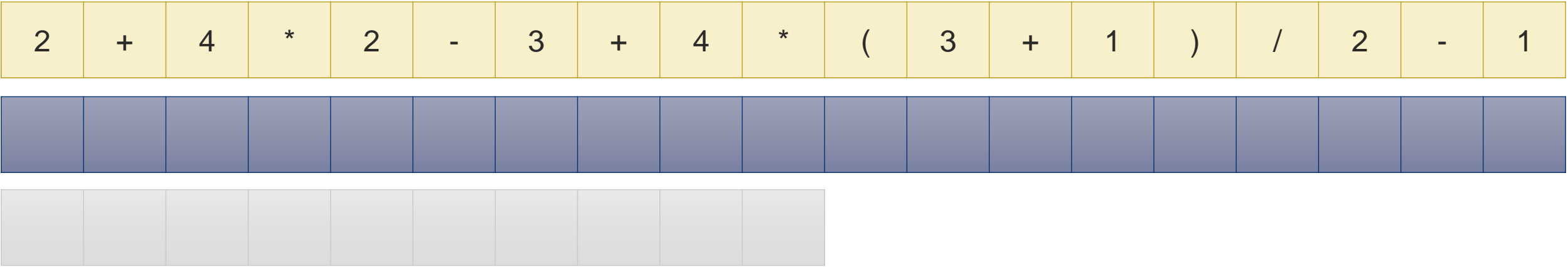
s – verem (operátorokat tárol)



Input:
x – sor

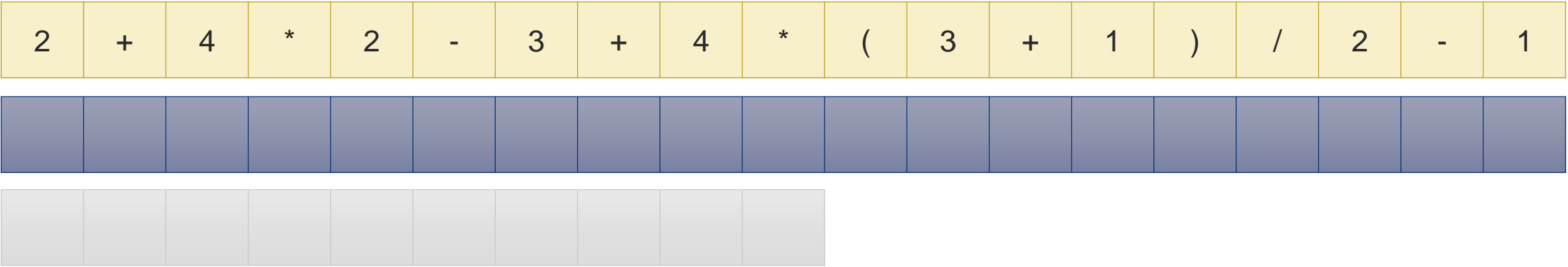
Feldolgozás során használt tárolók:
y – sor (lengyelforma lesz benne)
s – verem (operátorokat tárol)

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

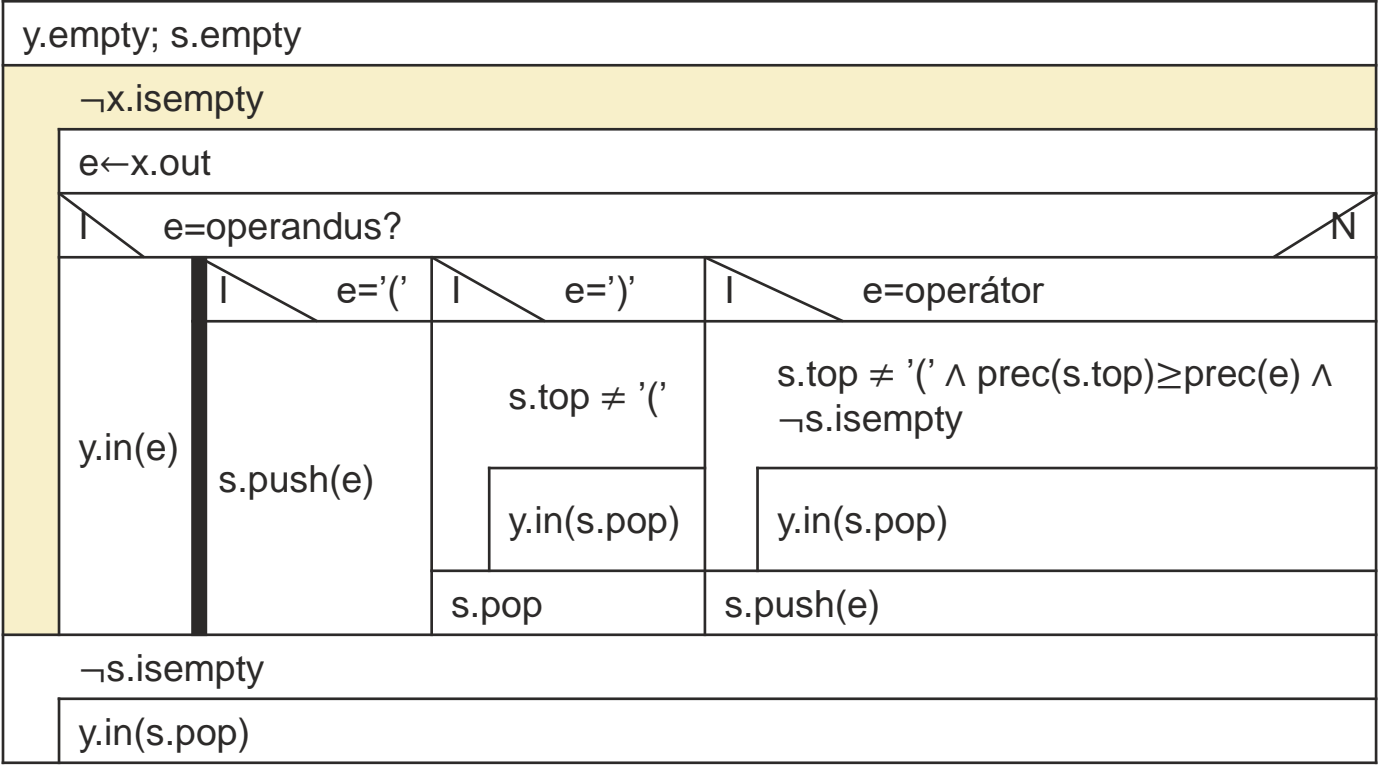


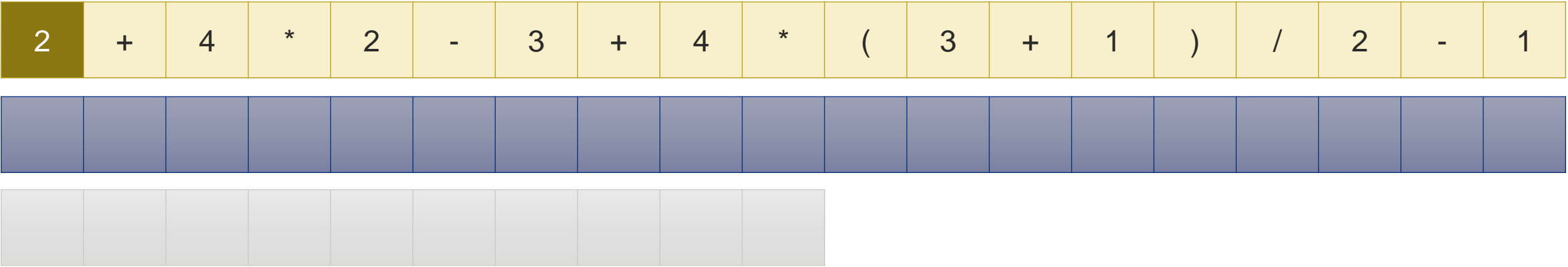
A használt tárolók létrehozása,
ürítése.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I e=operandus? N			
y.in(e)	I e='(' N	I e=')'	I e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			



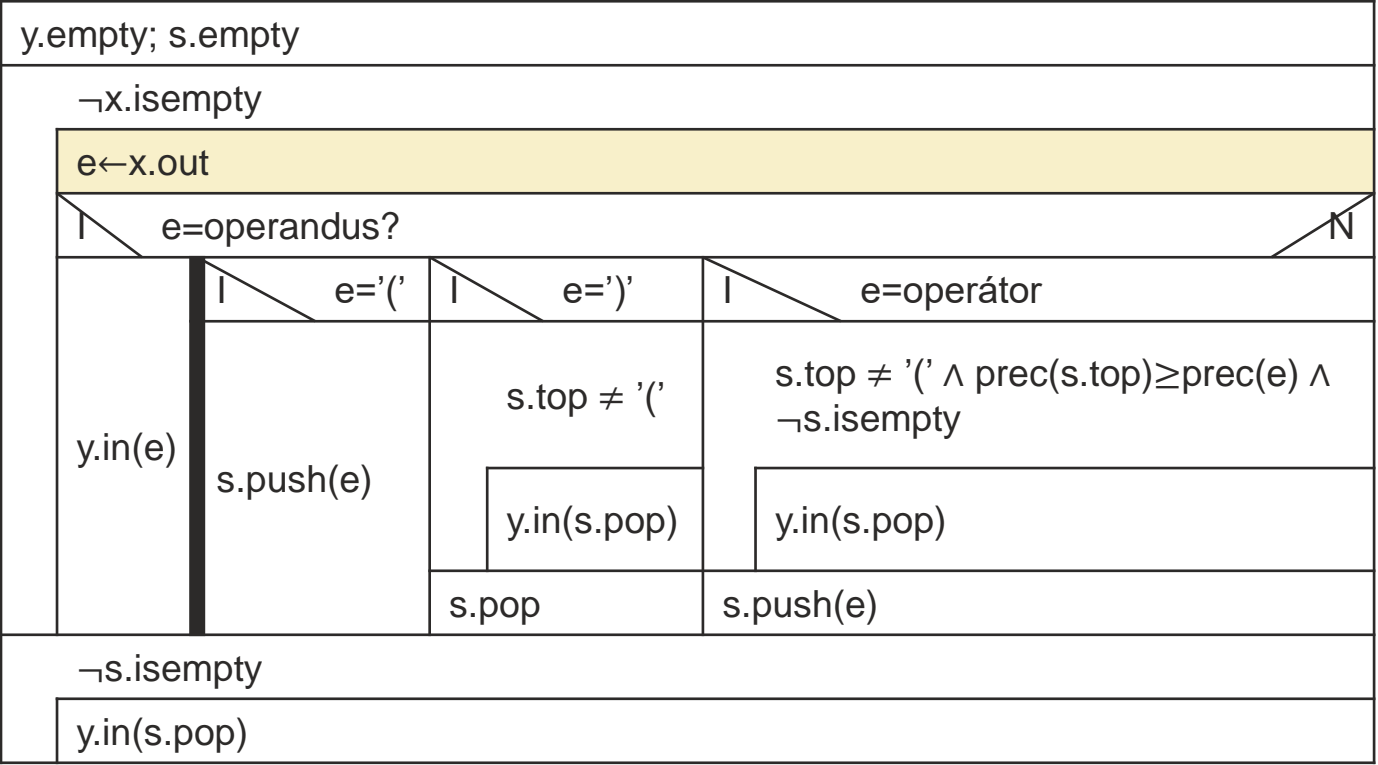
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

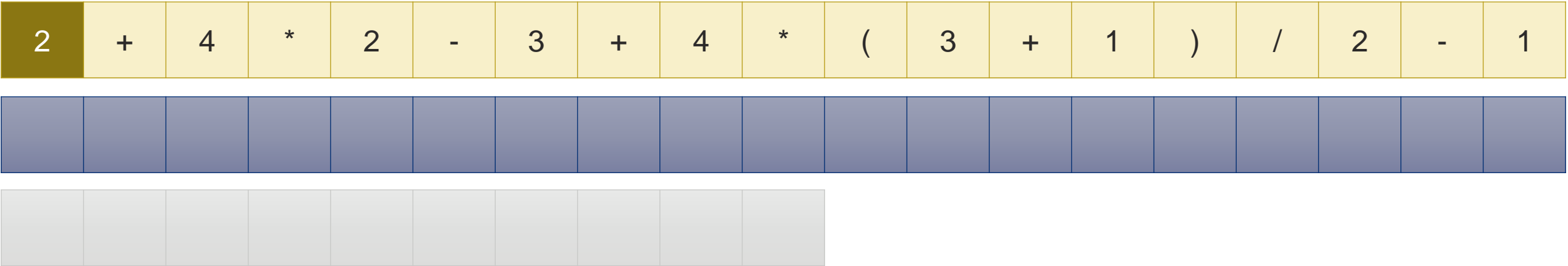




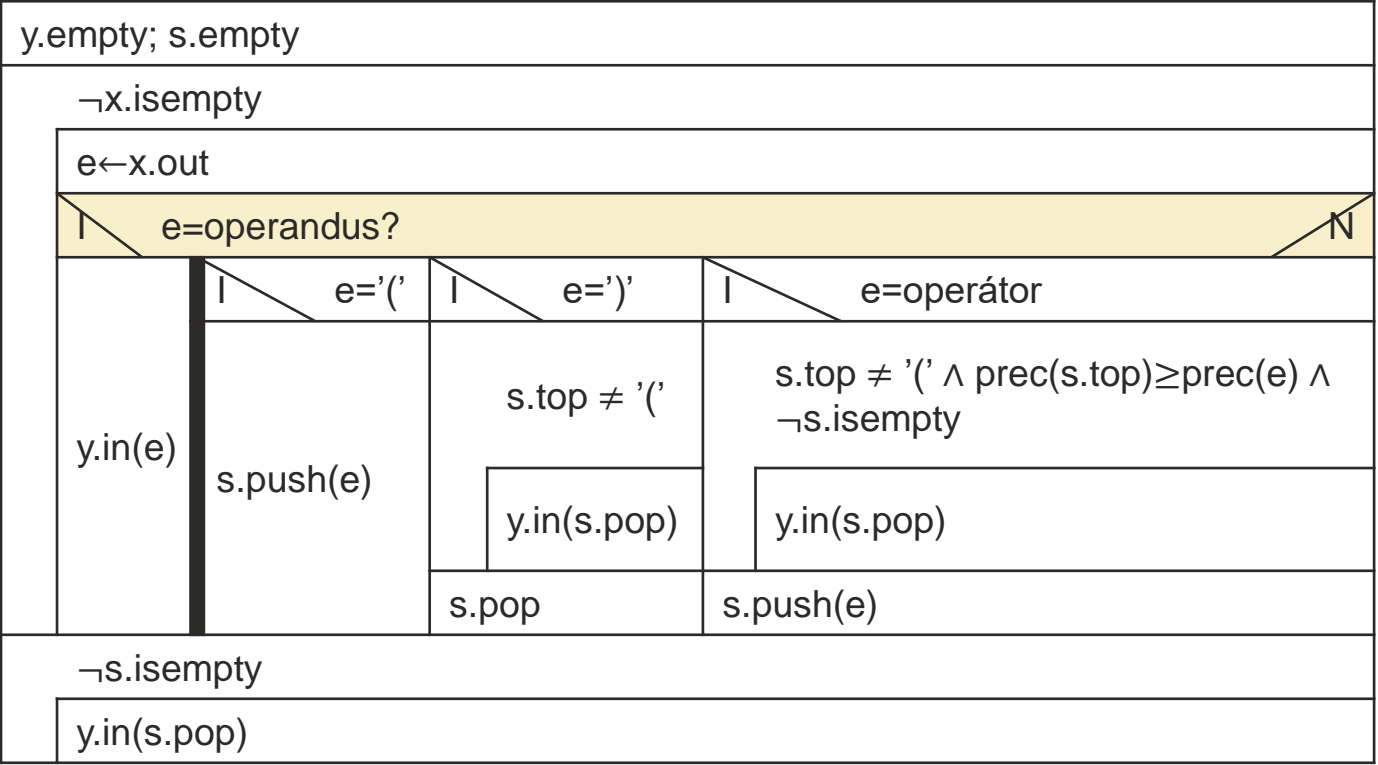
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

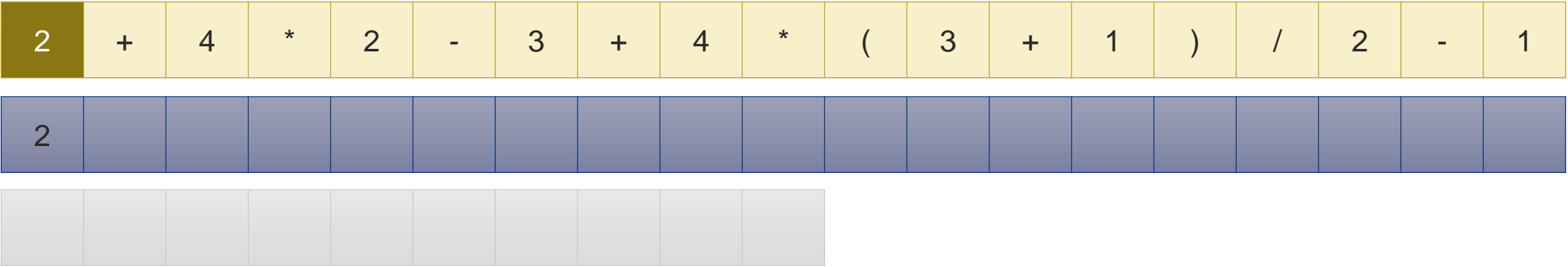
Az x sor következő elemét kivesszük a sorból.





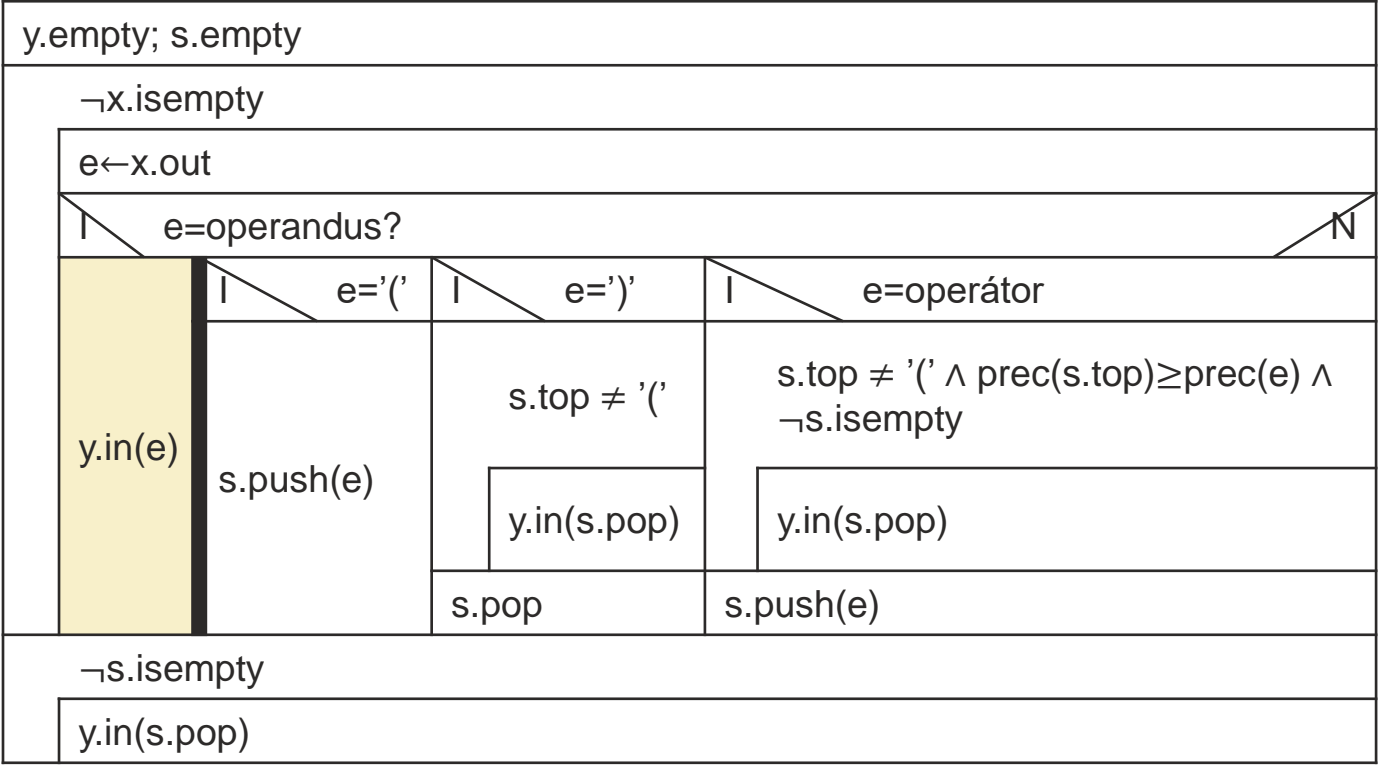
A kivett elem operandus?

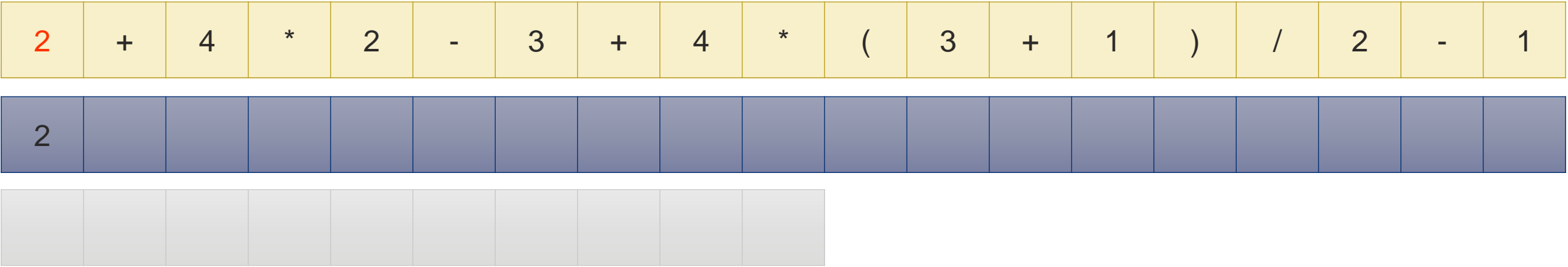




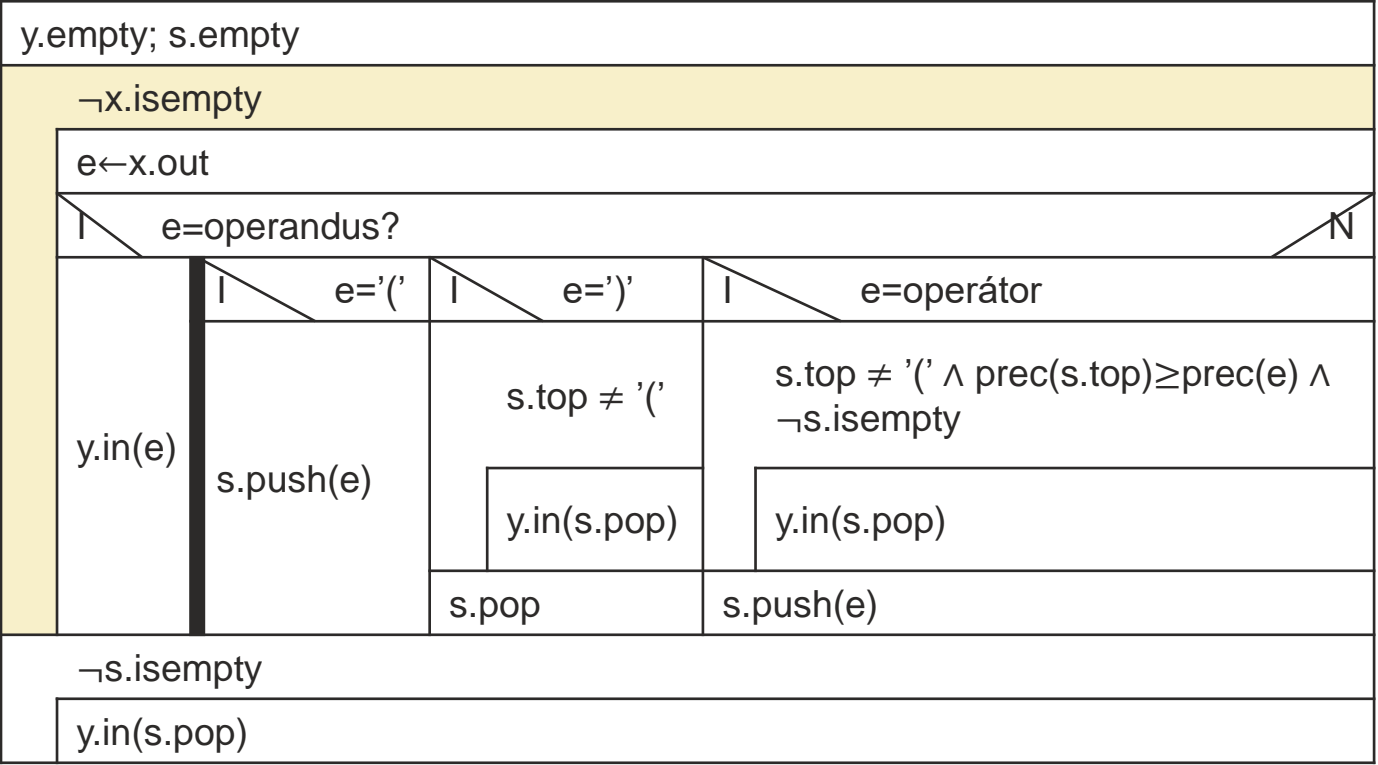
A kivett elem operandus?

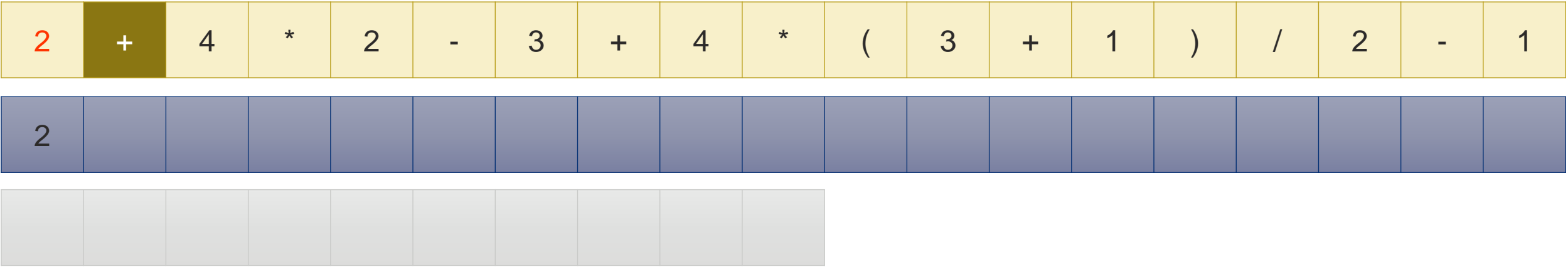
Igen → így azt az y sorba tesszük.





Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

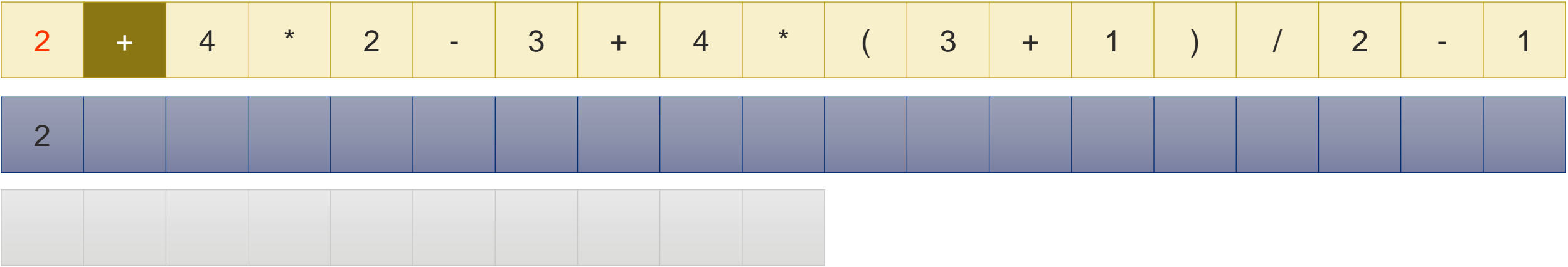




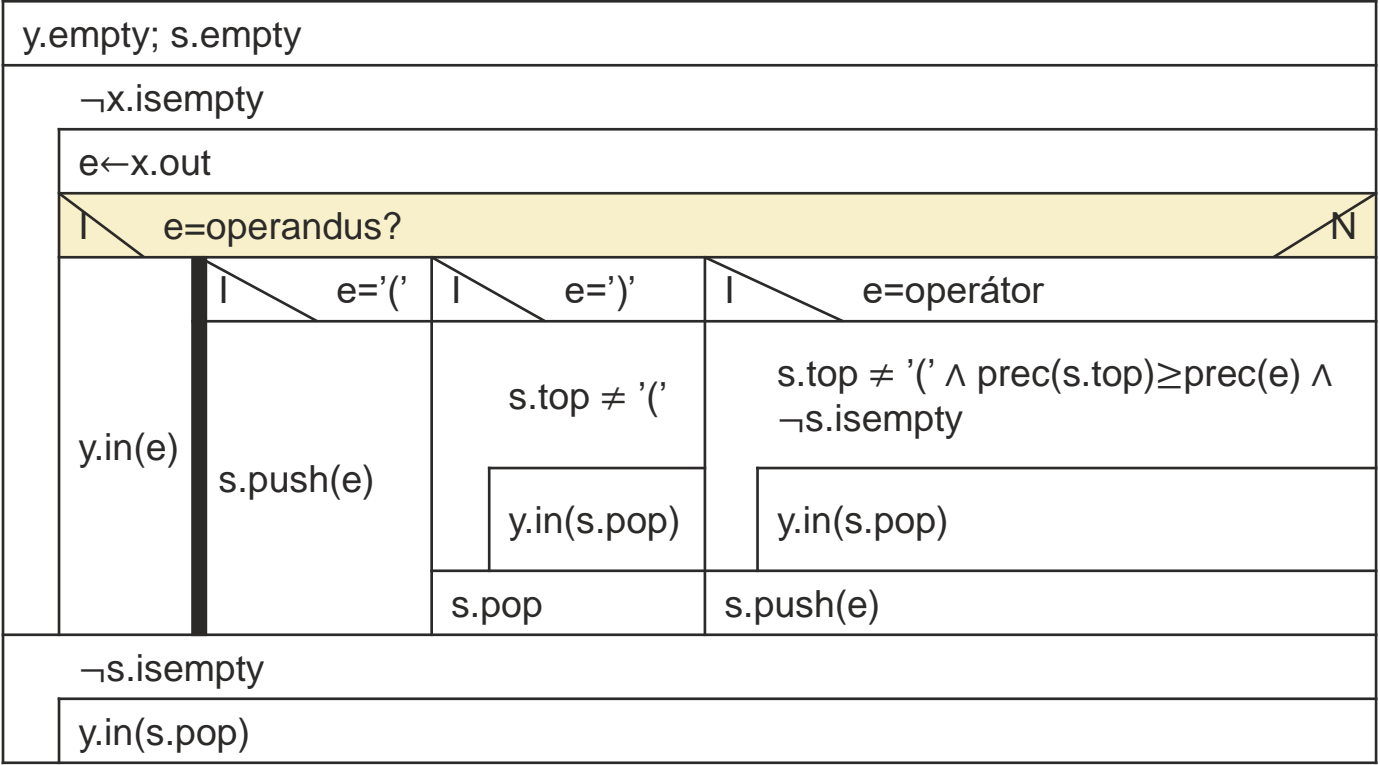
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

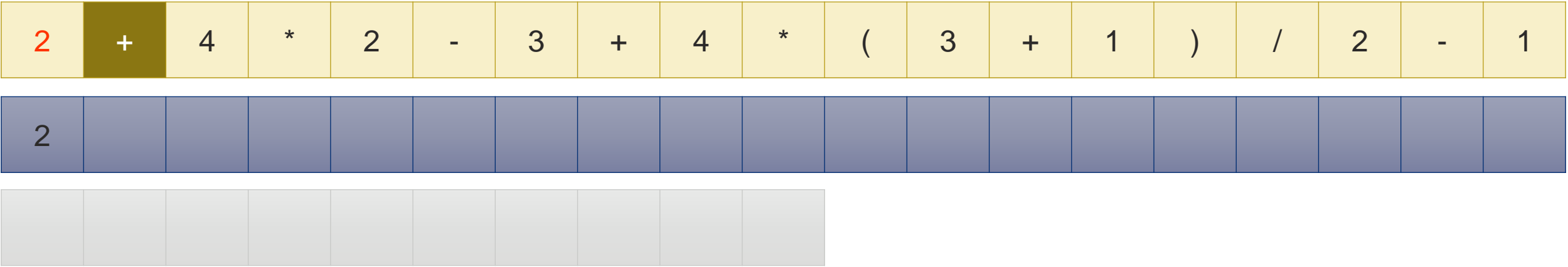
Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I \ e=operandus? / N				
y.in(e)	I \ e='(' /	I \ e=')' /	I \ e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				



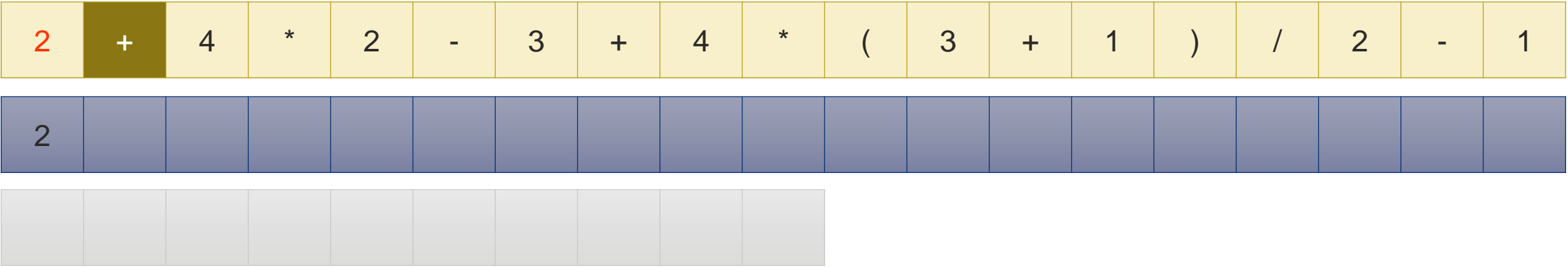
A kivett elem operandus?





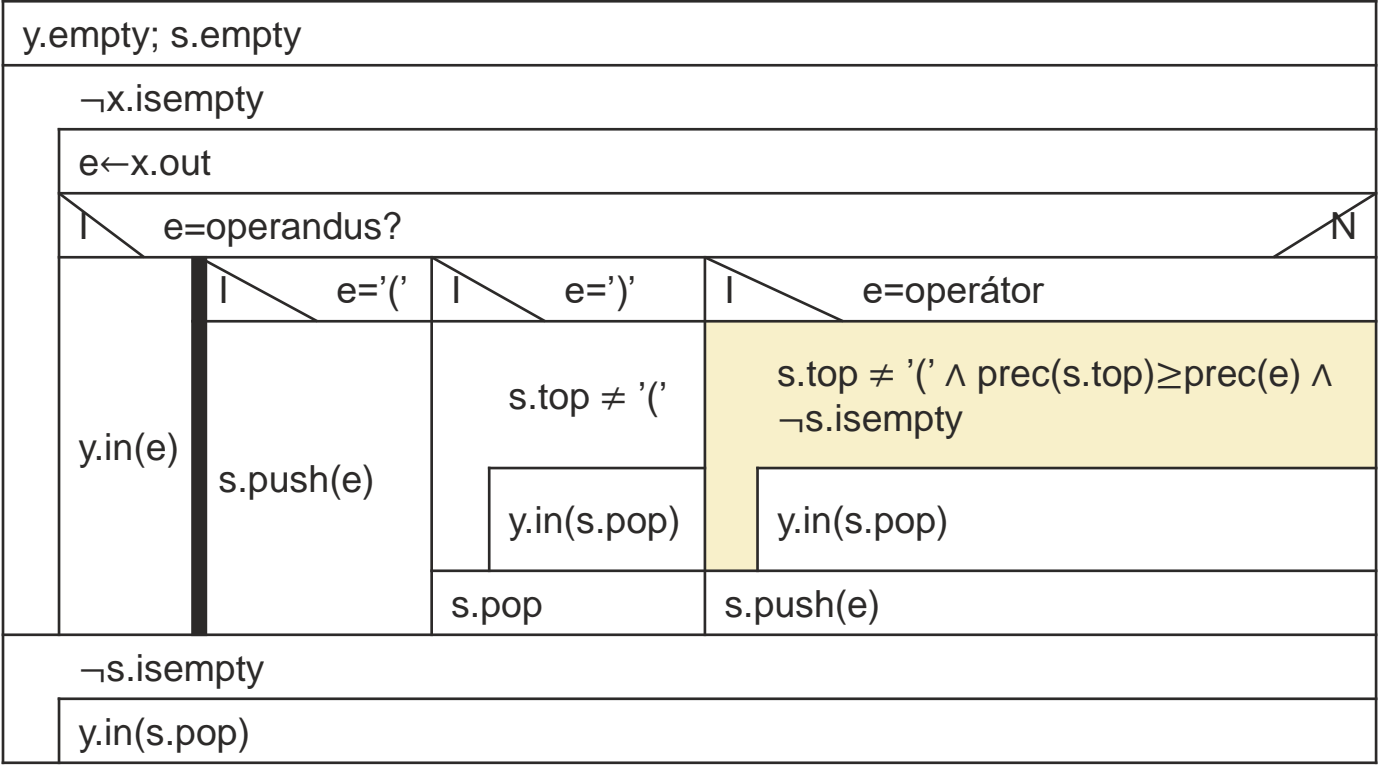
A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?

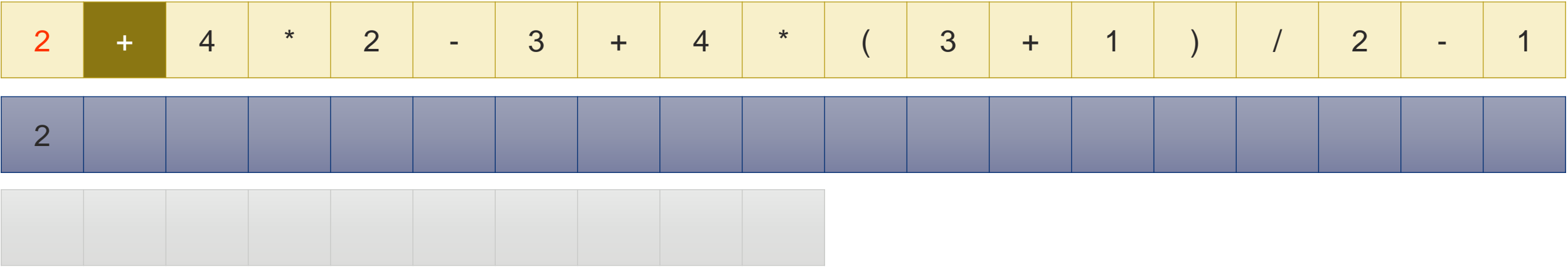
y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			



Nem → A kivett elem operátor, vagy valamely zárójel?

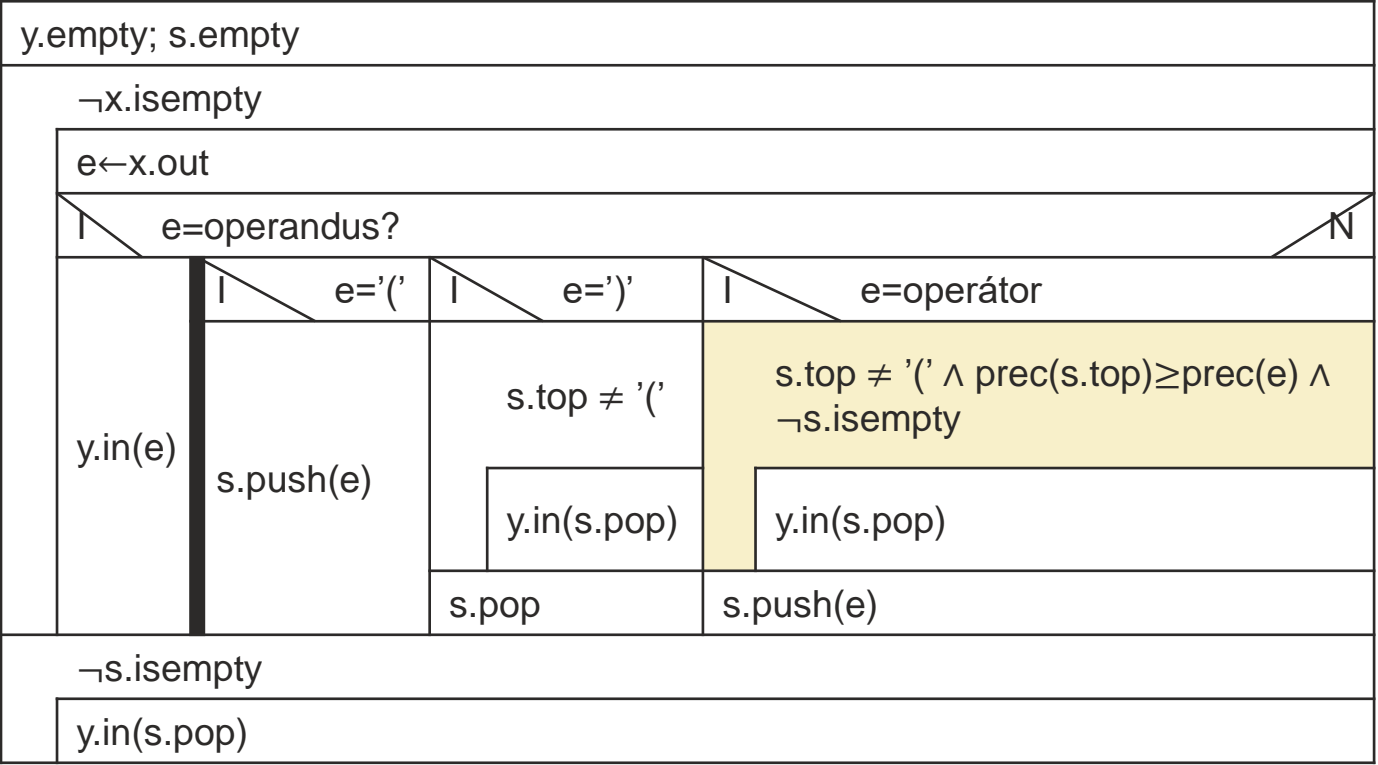
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

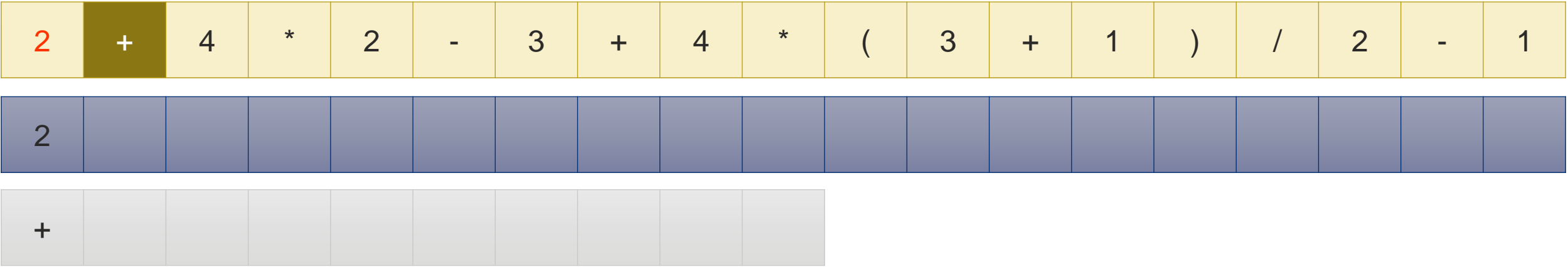




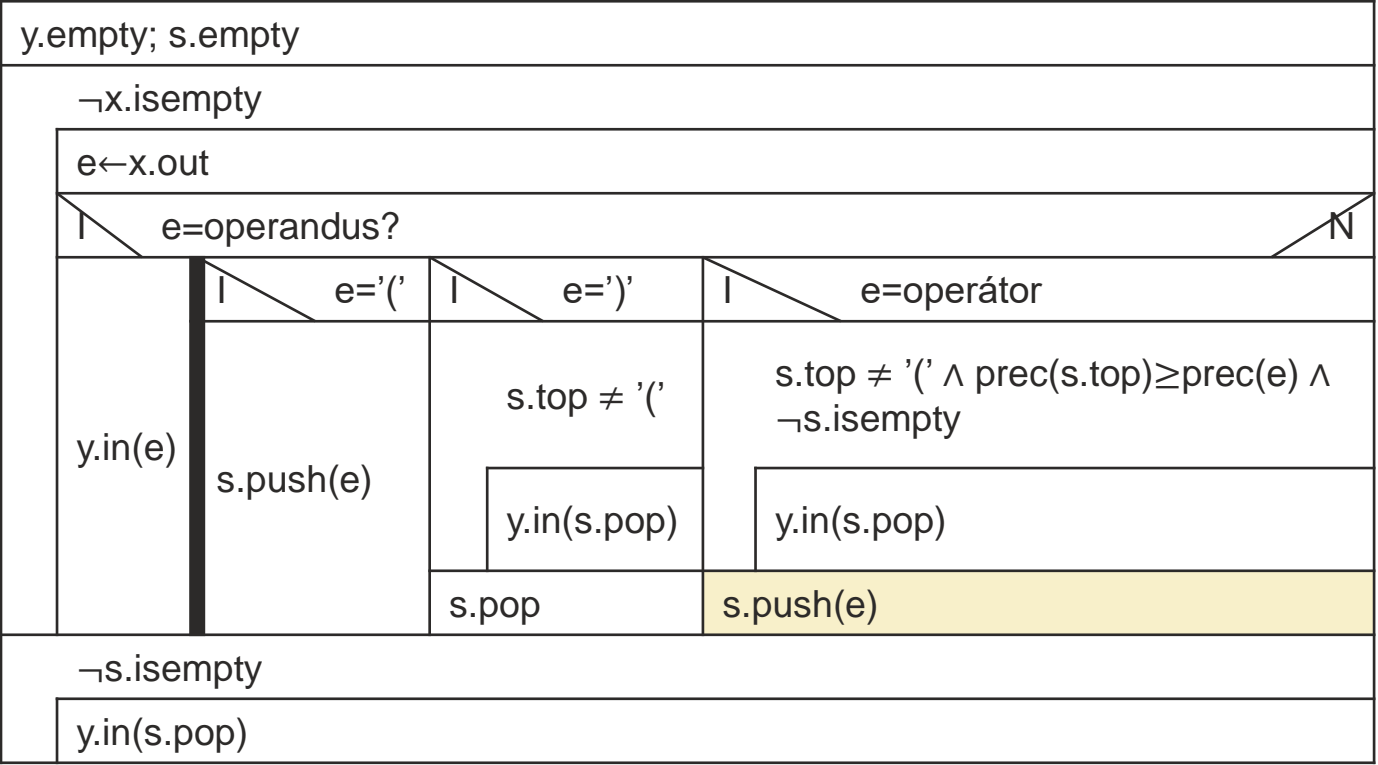
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

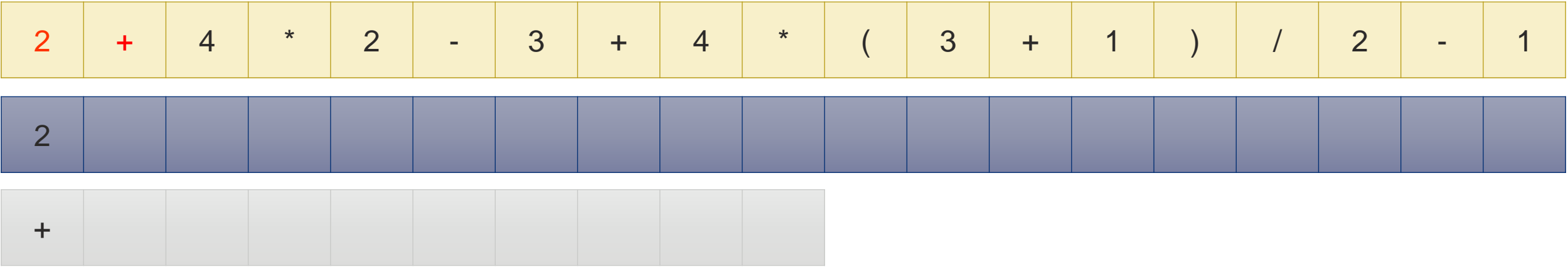
Üres, ezért a feltétel nem teljesül, a ciklusmag nem kerül végrehajtásra.



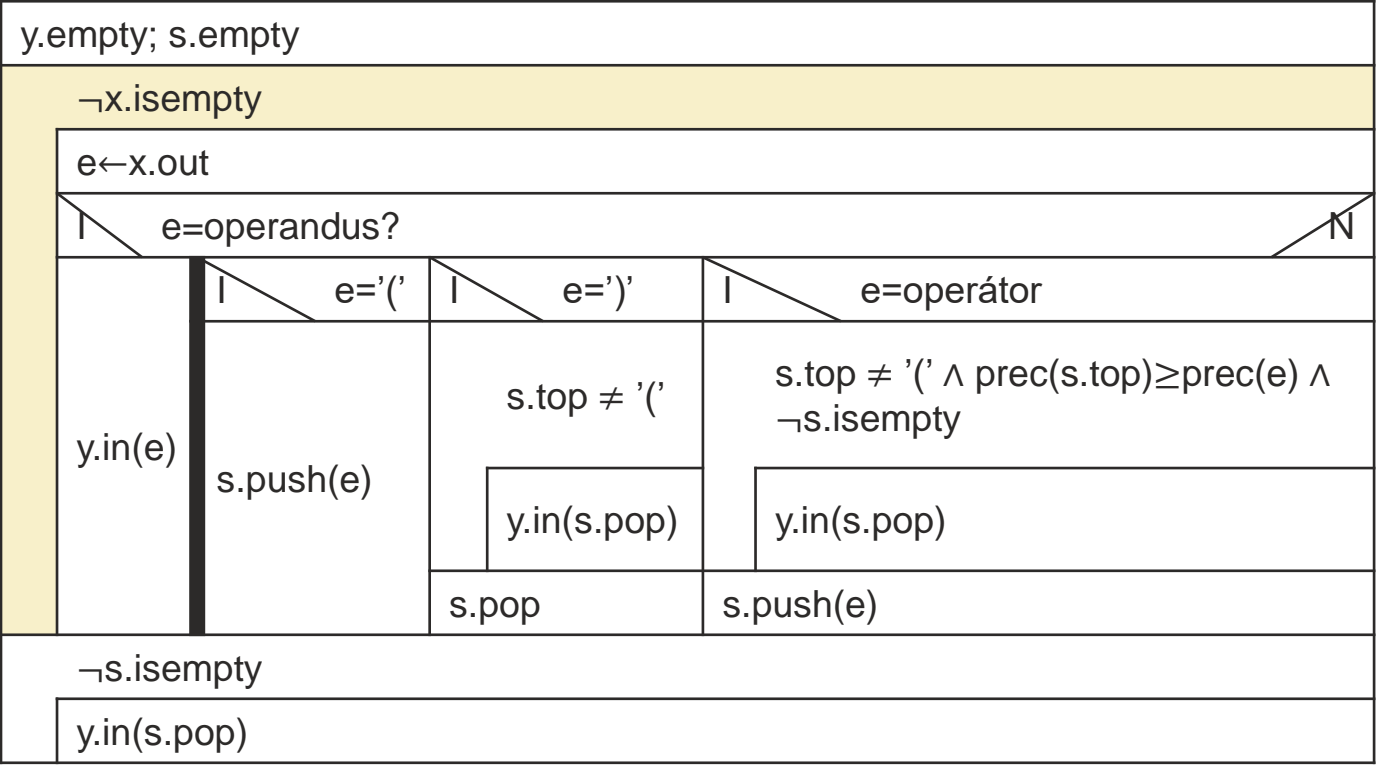


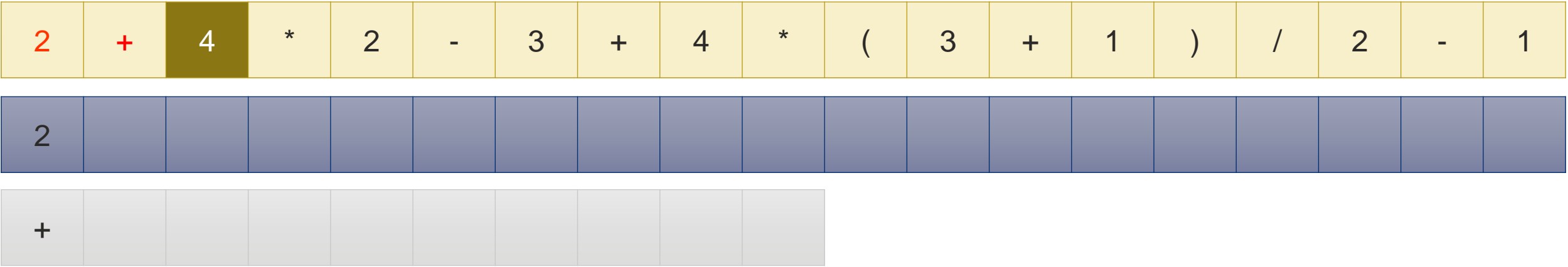
Üres, ezért a feltétel nem teljesül, a ciklusmag nem kerül végrehajtásra.
Végül az aktuális operátort a verem tetejére tesszük.





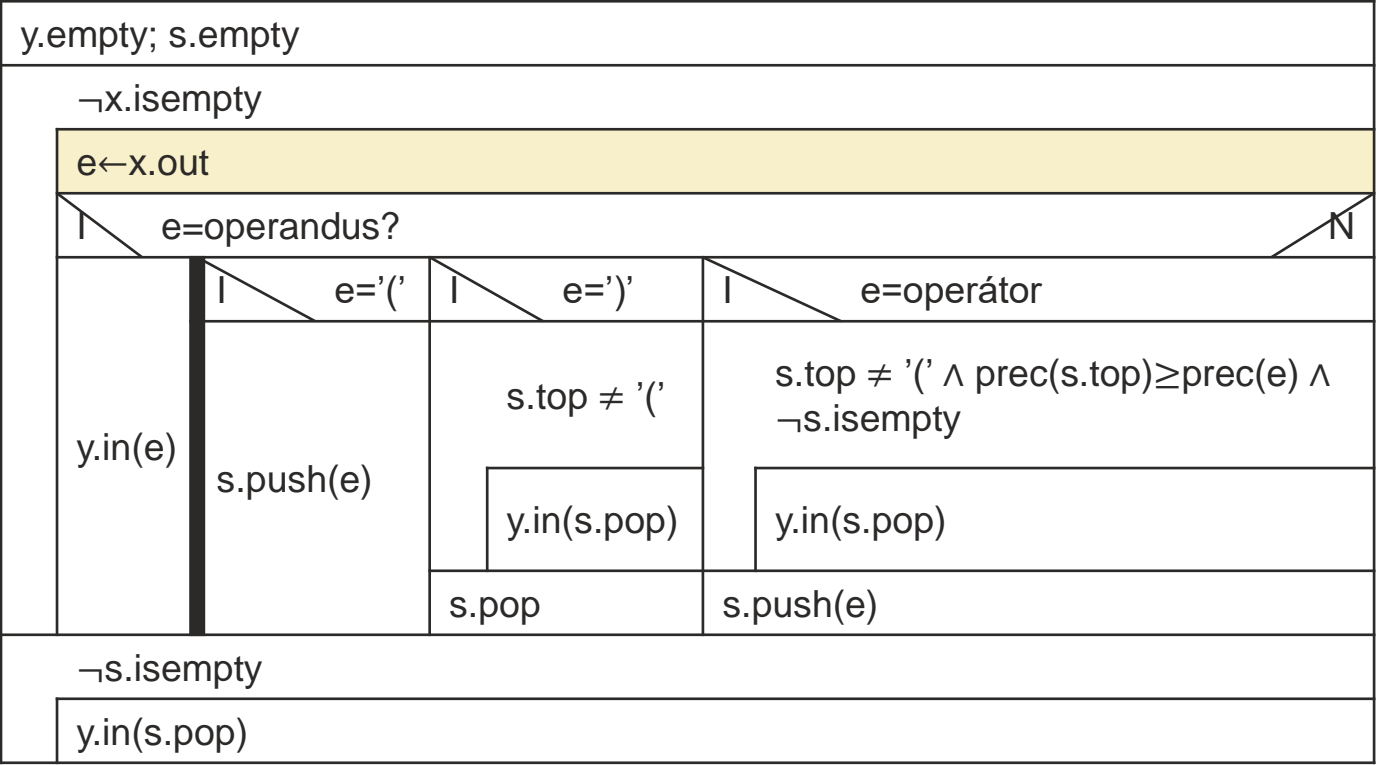
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.





Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.



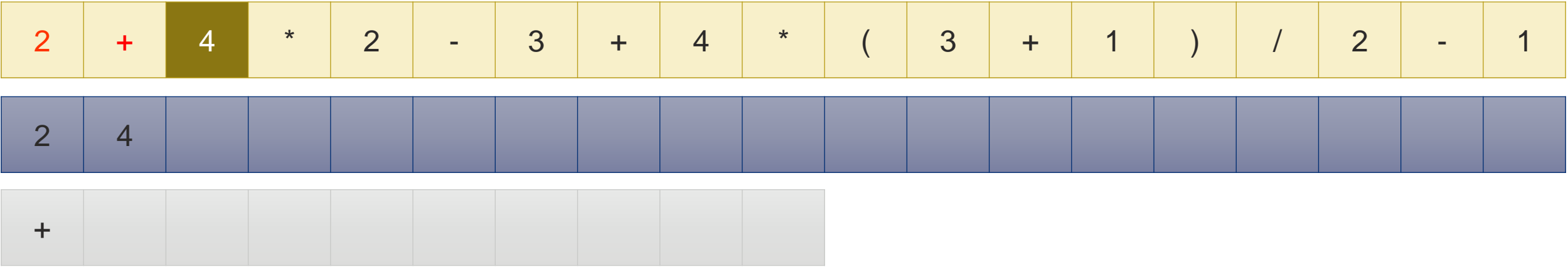
2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2																		
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

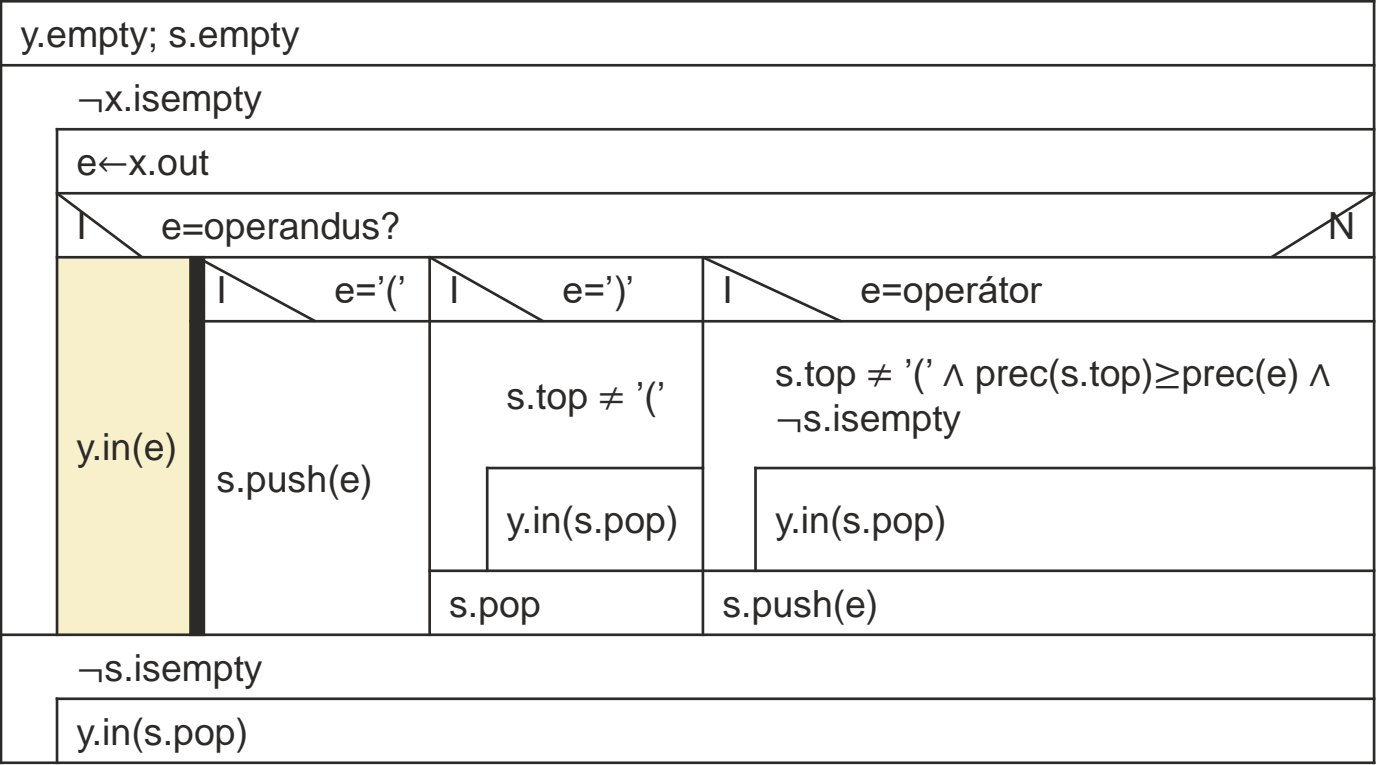
+																		
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

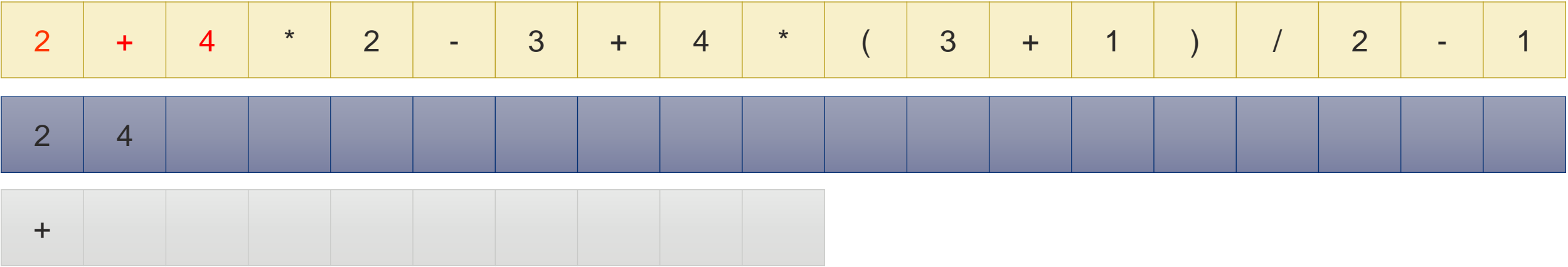
A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

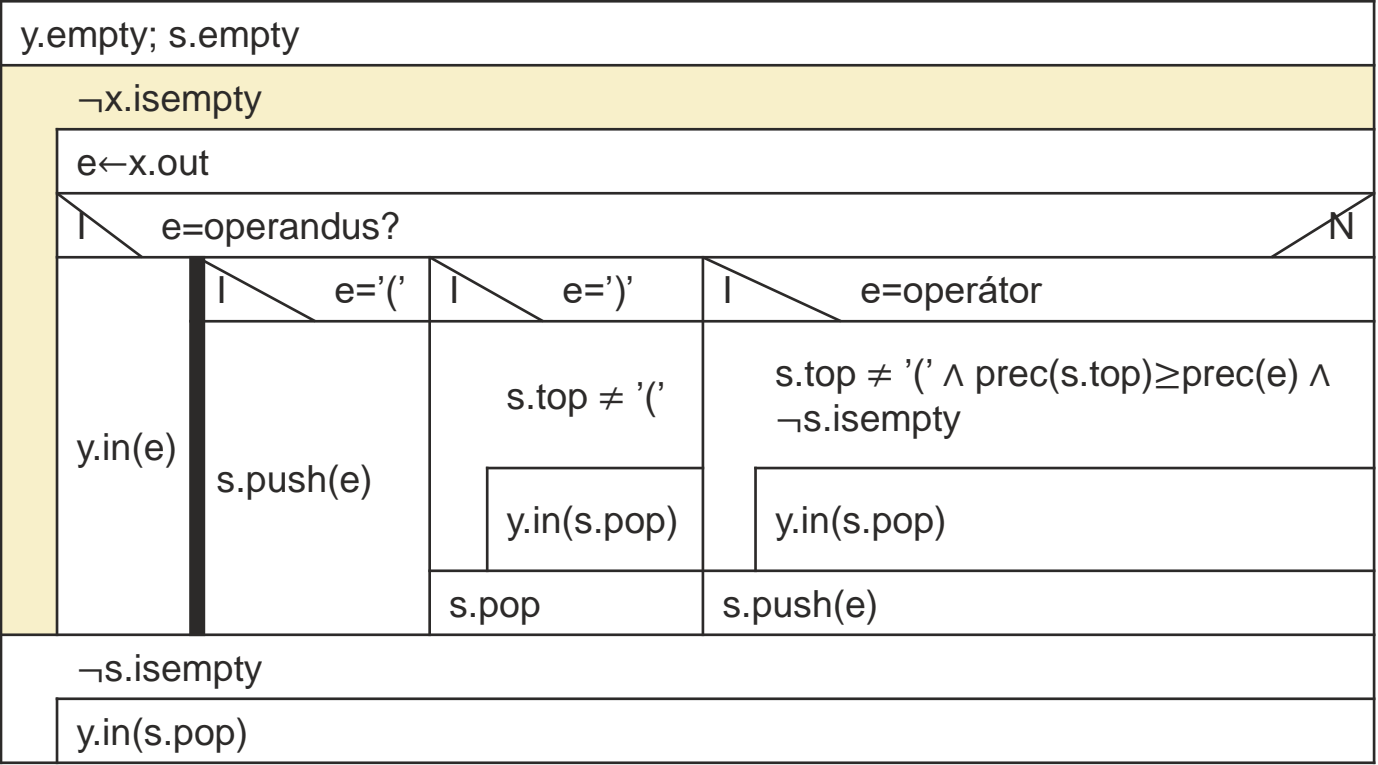


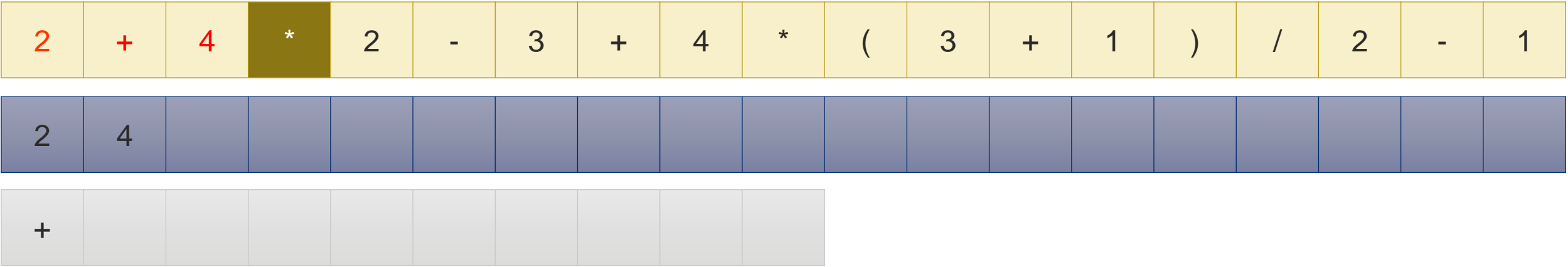
A kivett elem operandus?
Igen → így azt az y sorba tesszük.





Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

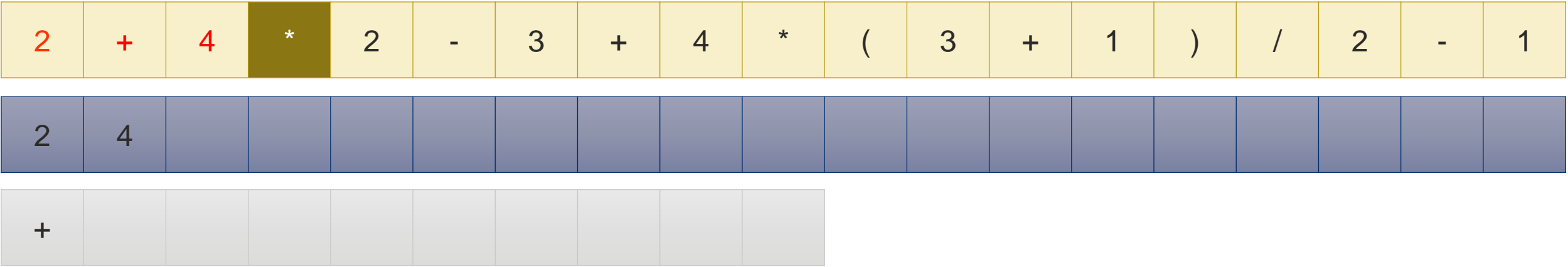




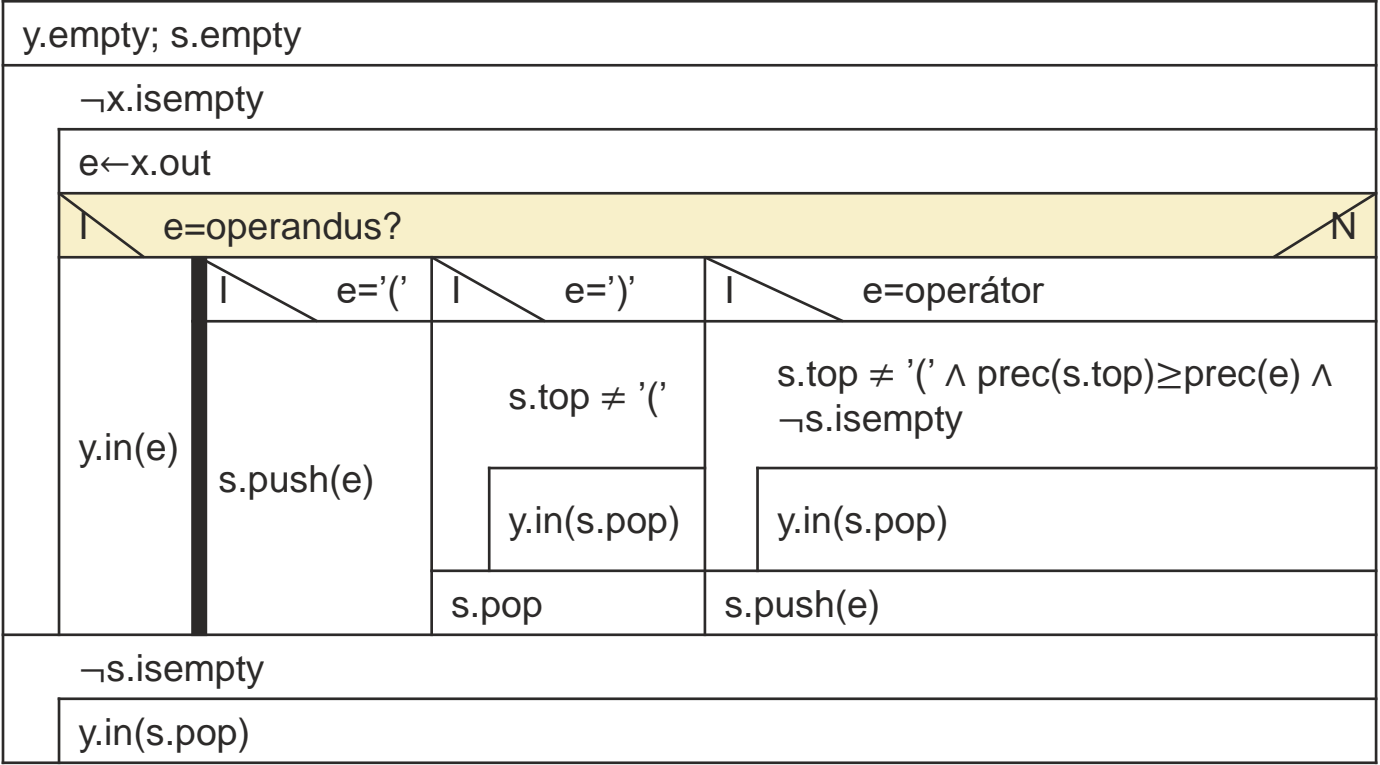
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

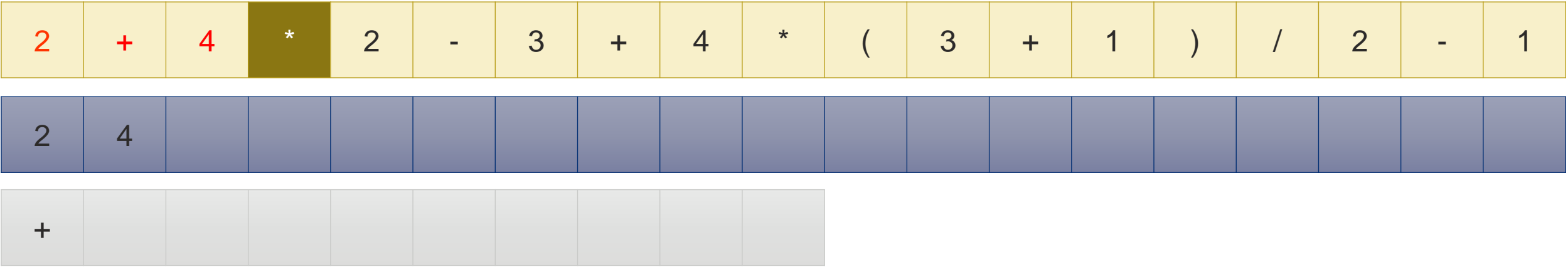
Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

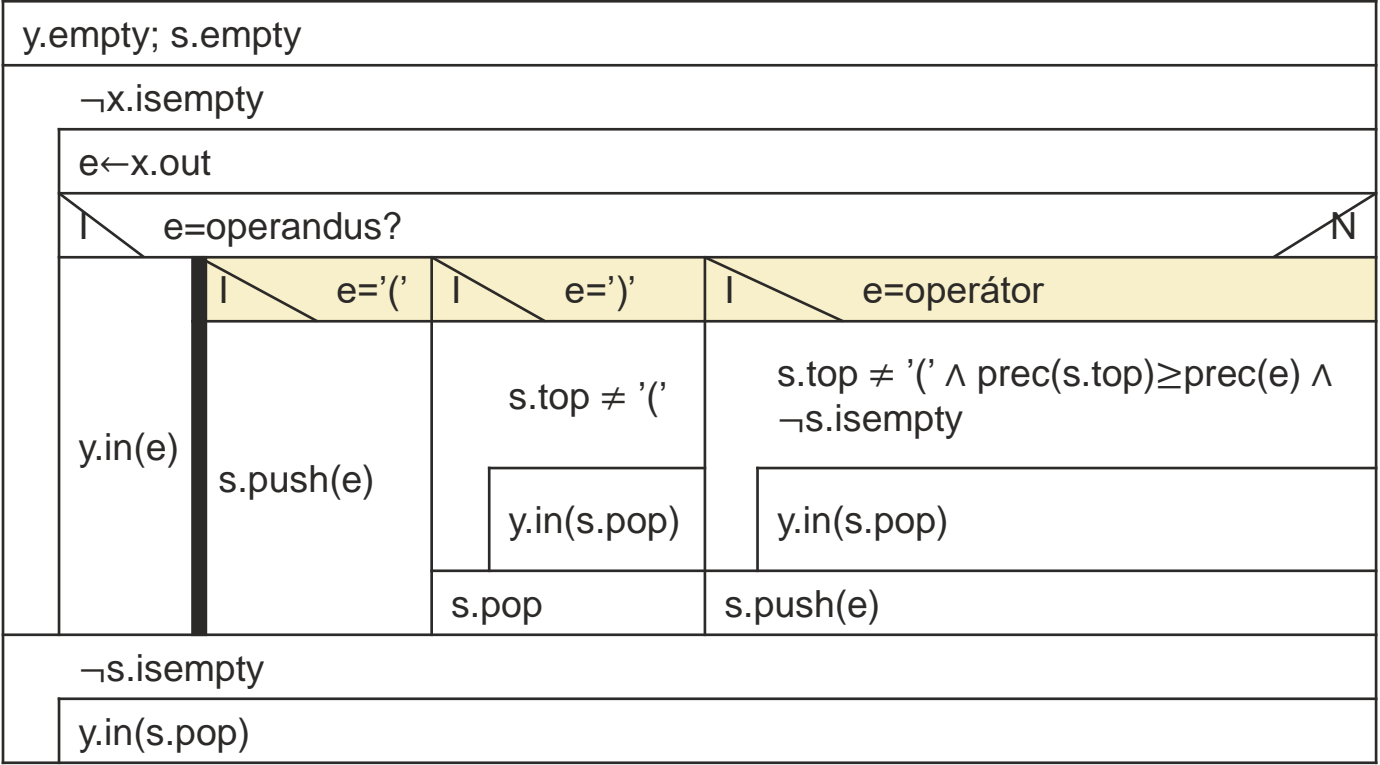


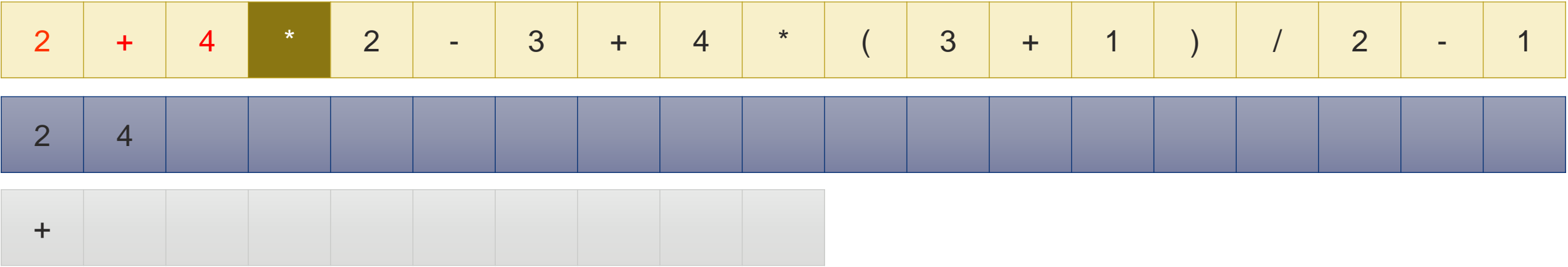
A kivett elem operandus?





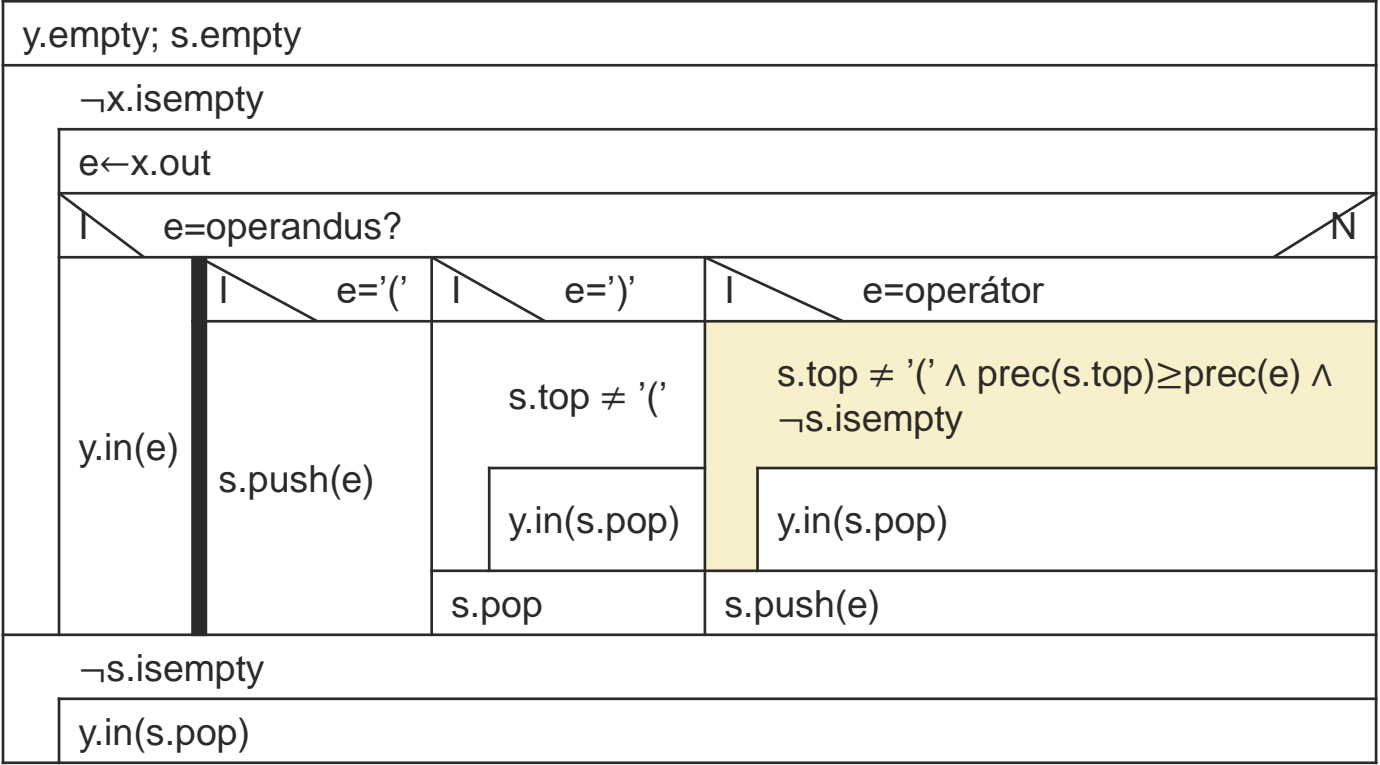
A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?

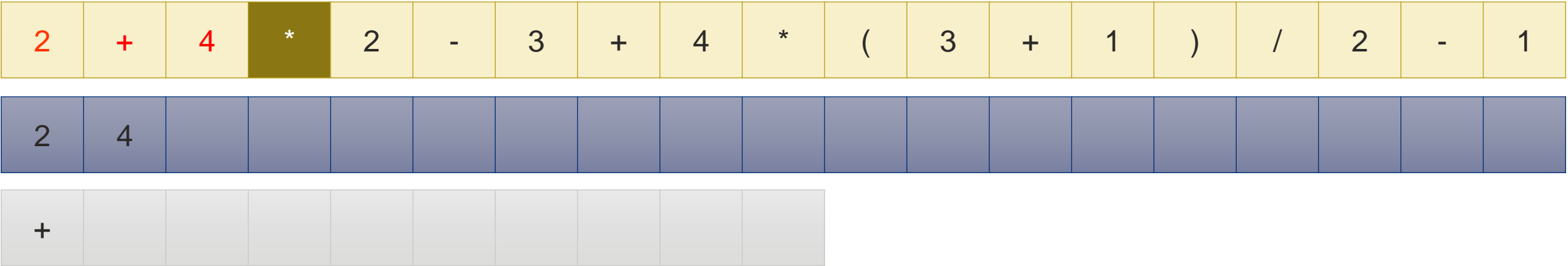




Nem → A kivett elem operátor, vagy valamely zárójel?

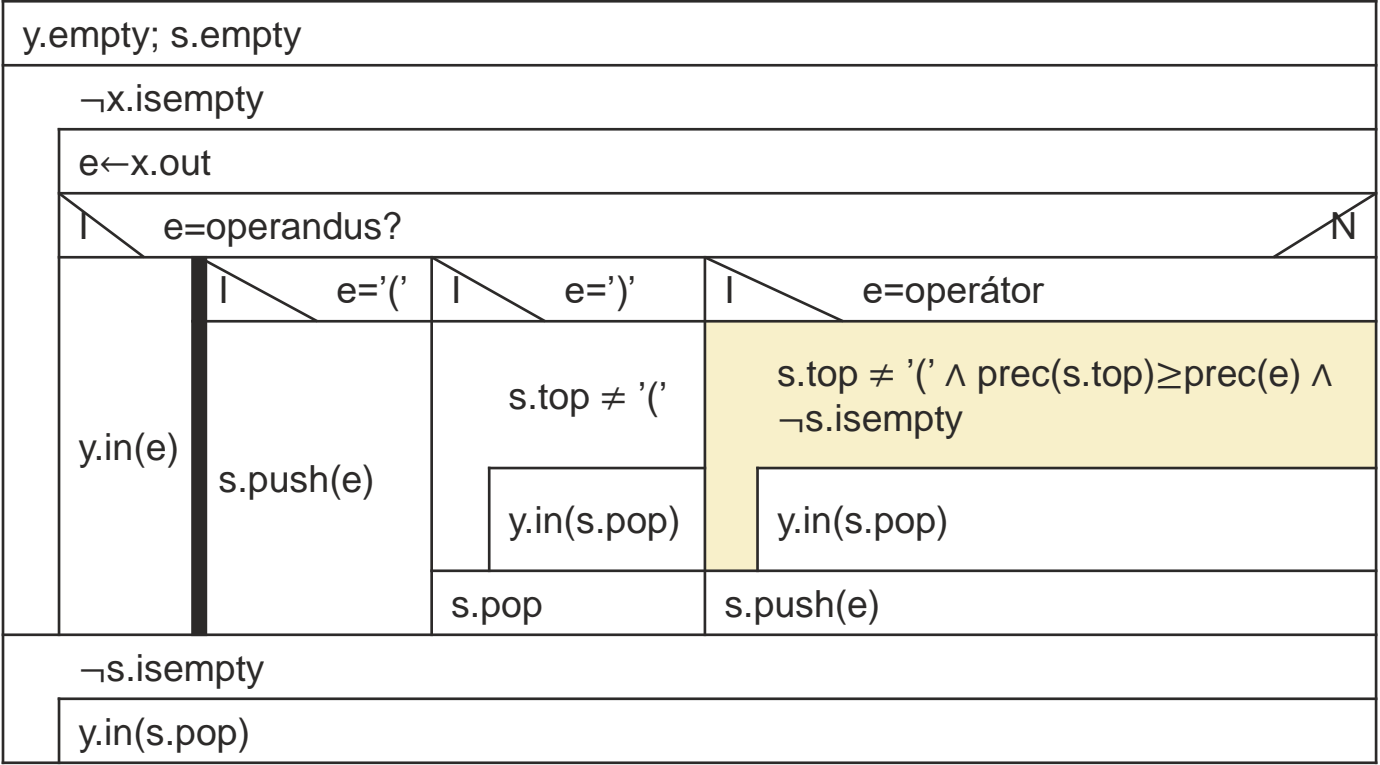
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

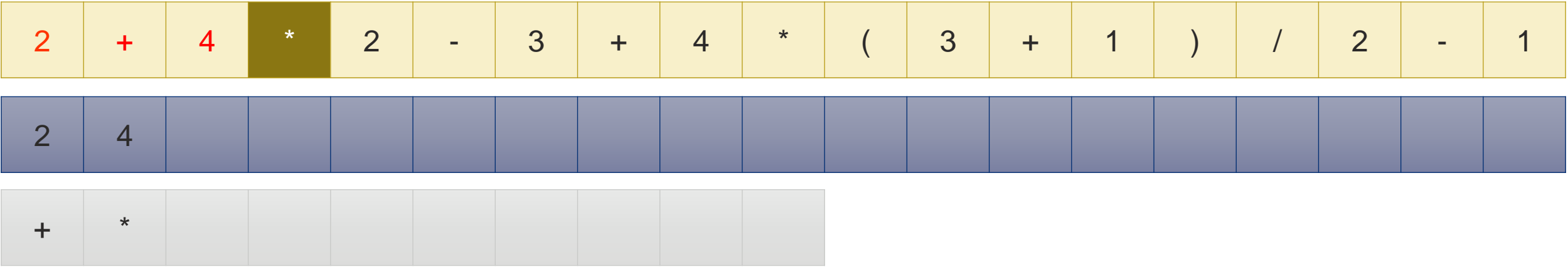




Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

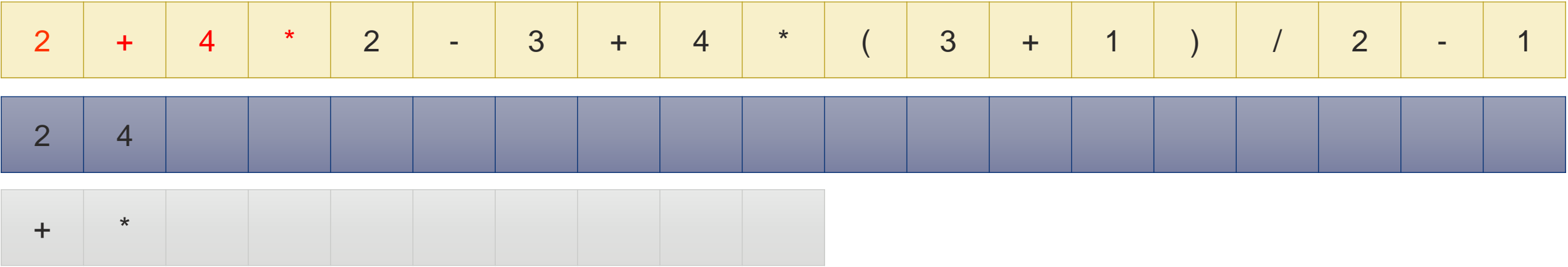
A verem tetején levő operátor precedenciája alacsonyabb.



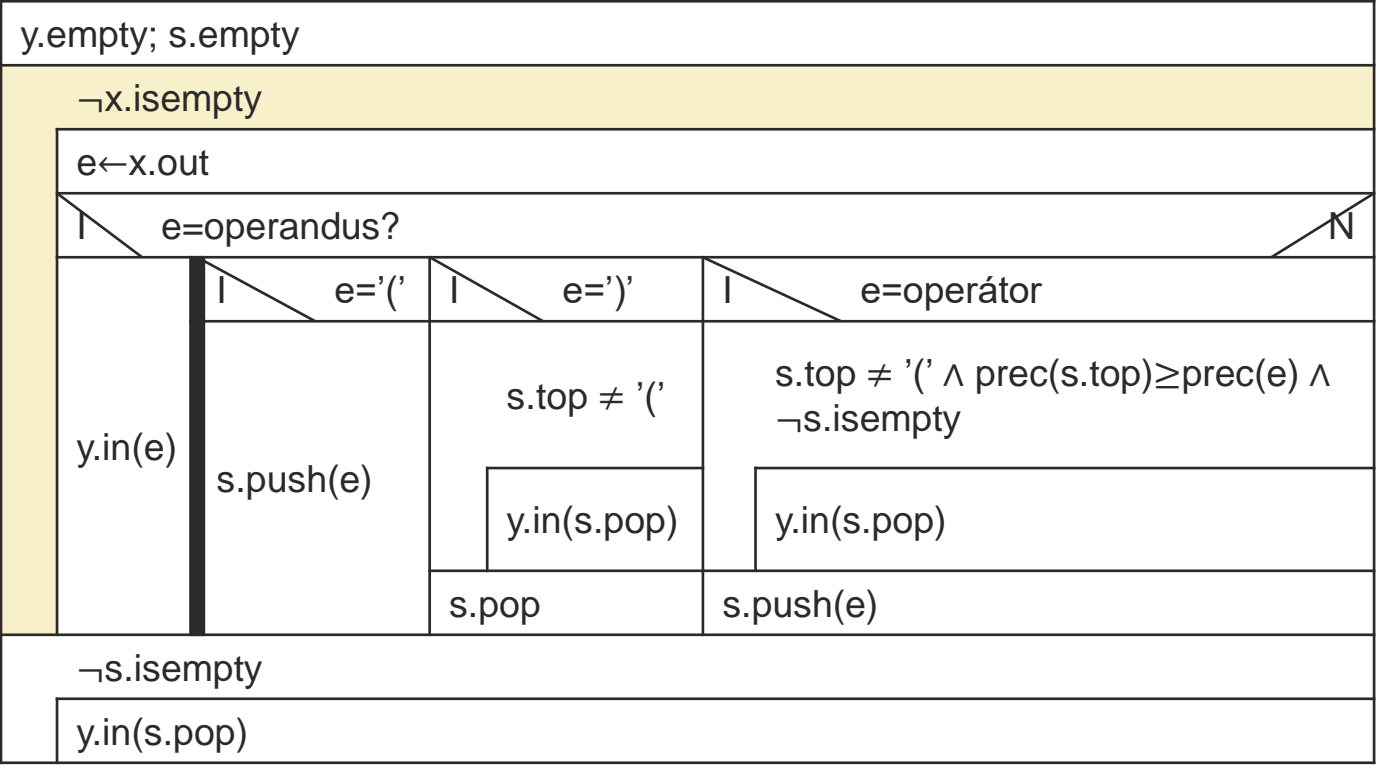


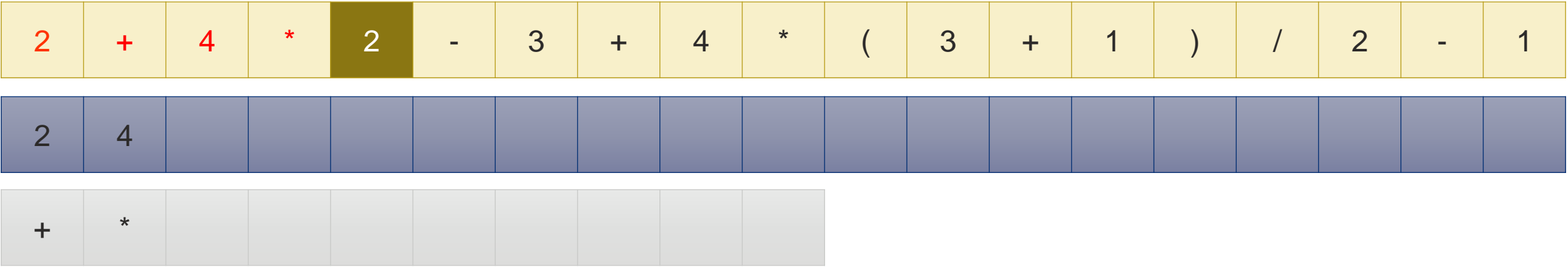
A verem tetején levő operátor precedenciája alacsonyabb.
Végül az aktuális operátort a verem tetejére tesszük.

y.empty; s.empty			
¬x.isempty			
e←x.out			
e=operandus? N			
y.in(e)	e='('	e=')'	e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			



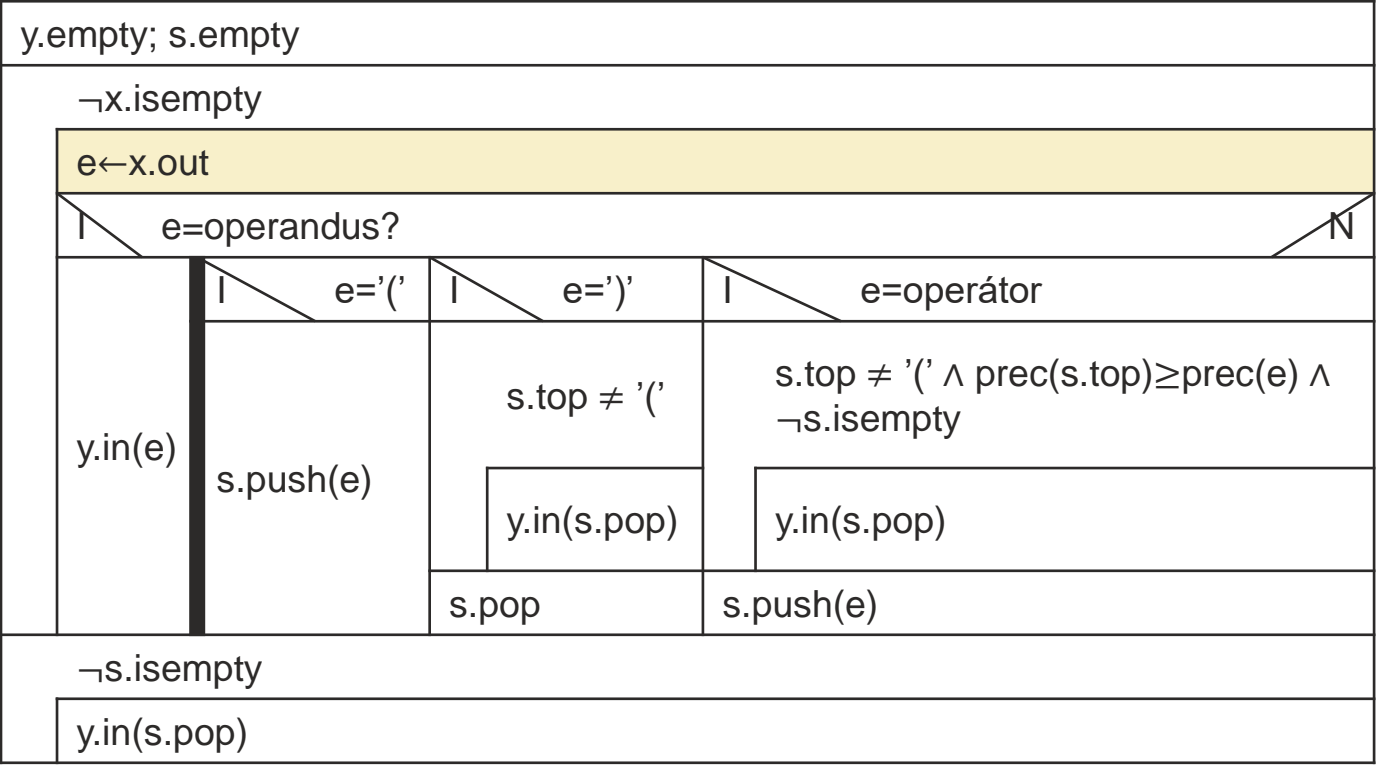
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

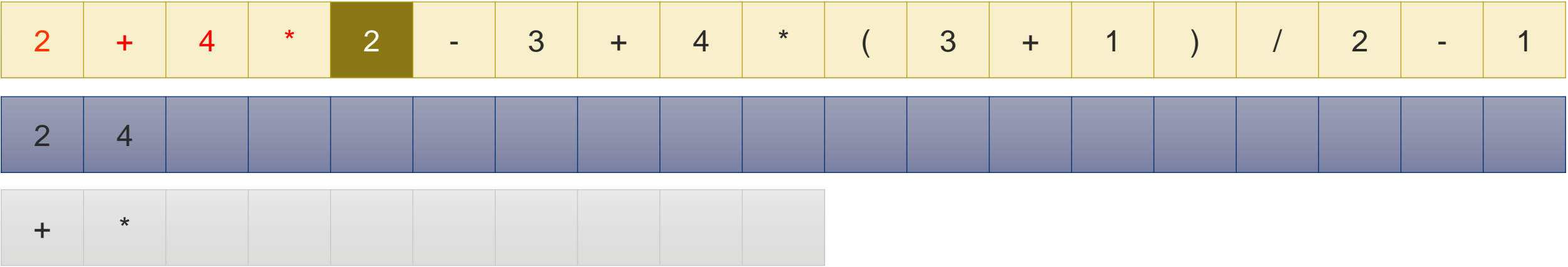




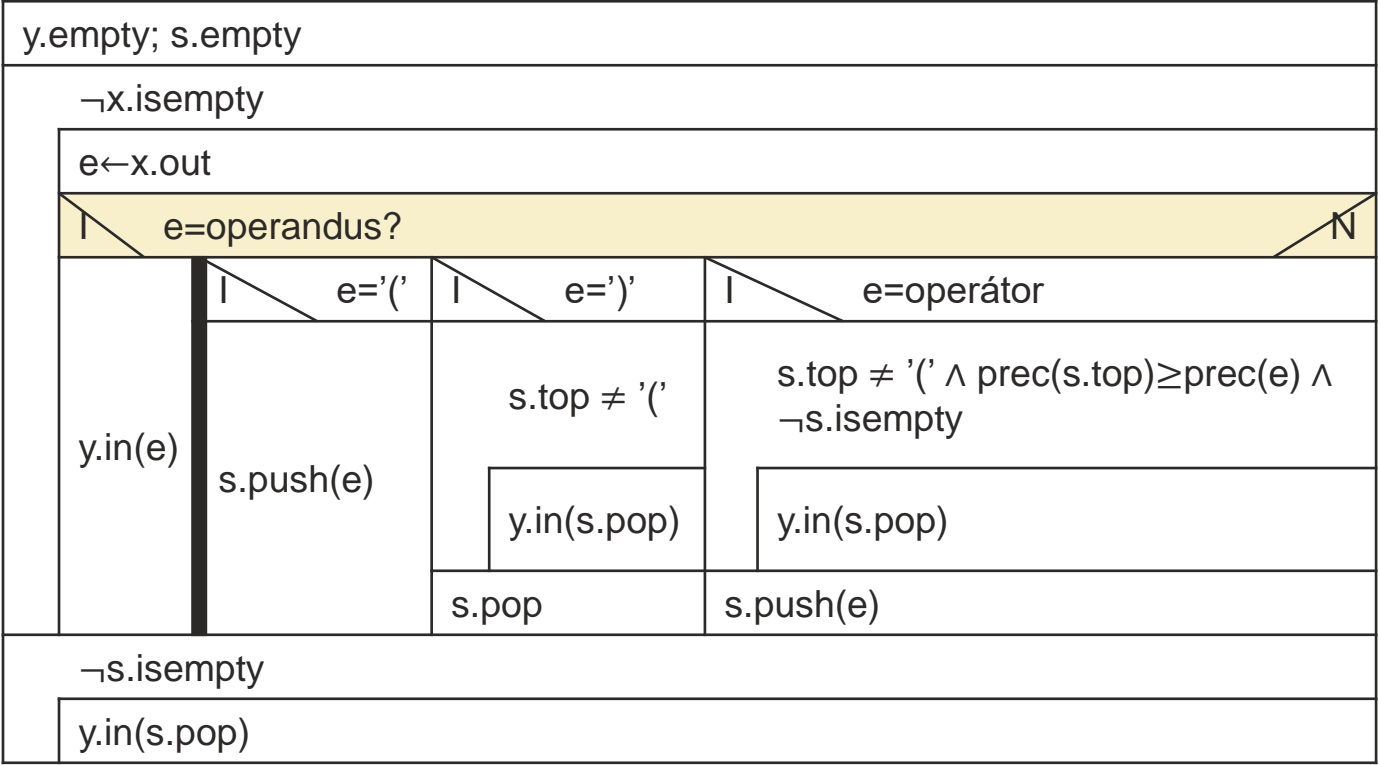
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

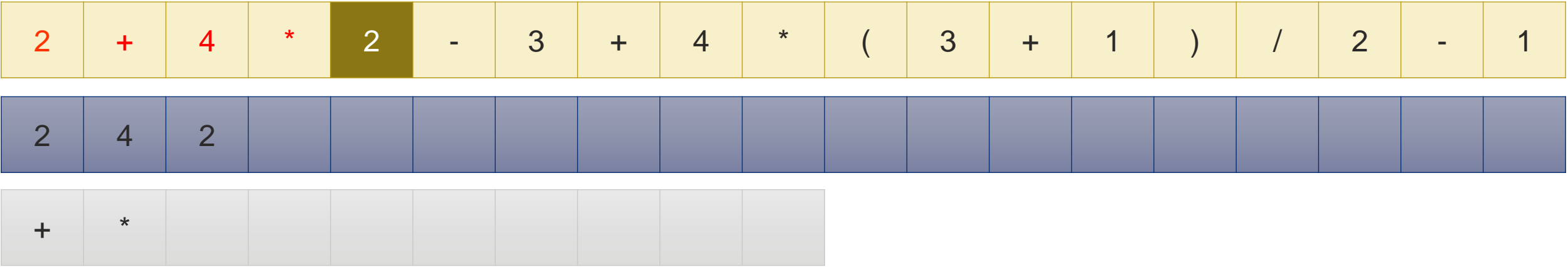
Az x sor következő elemét kivesszük a sorból.



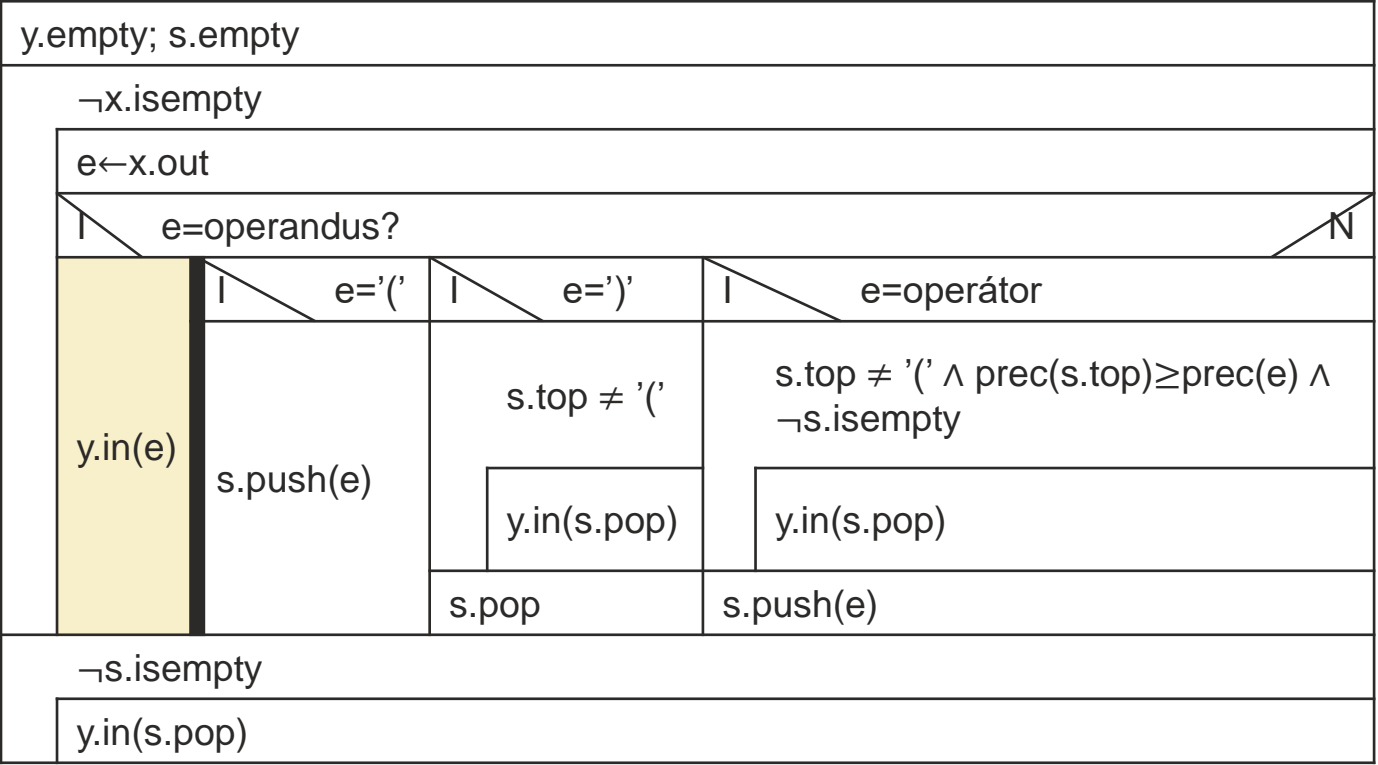


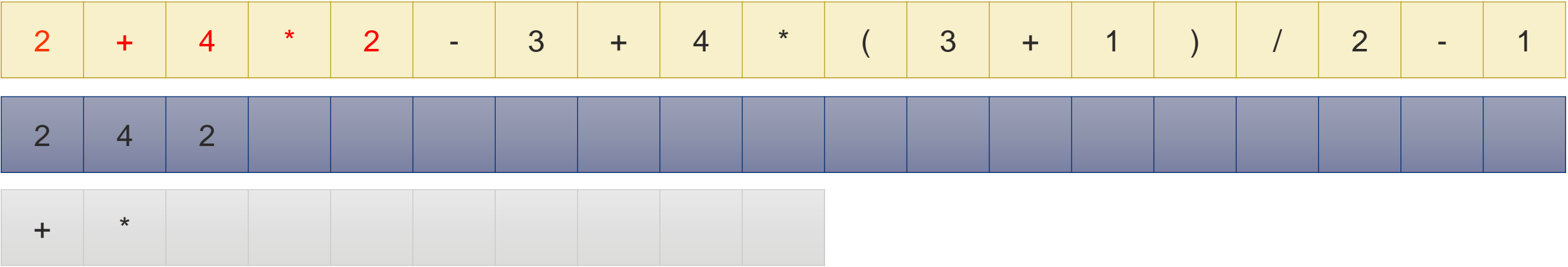
A kivett elem operandus?



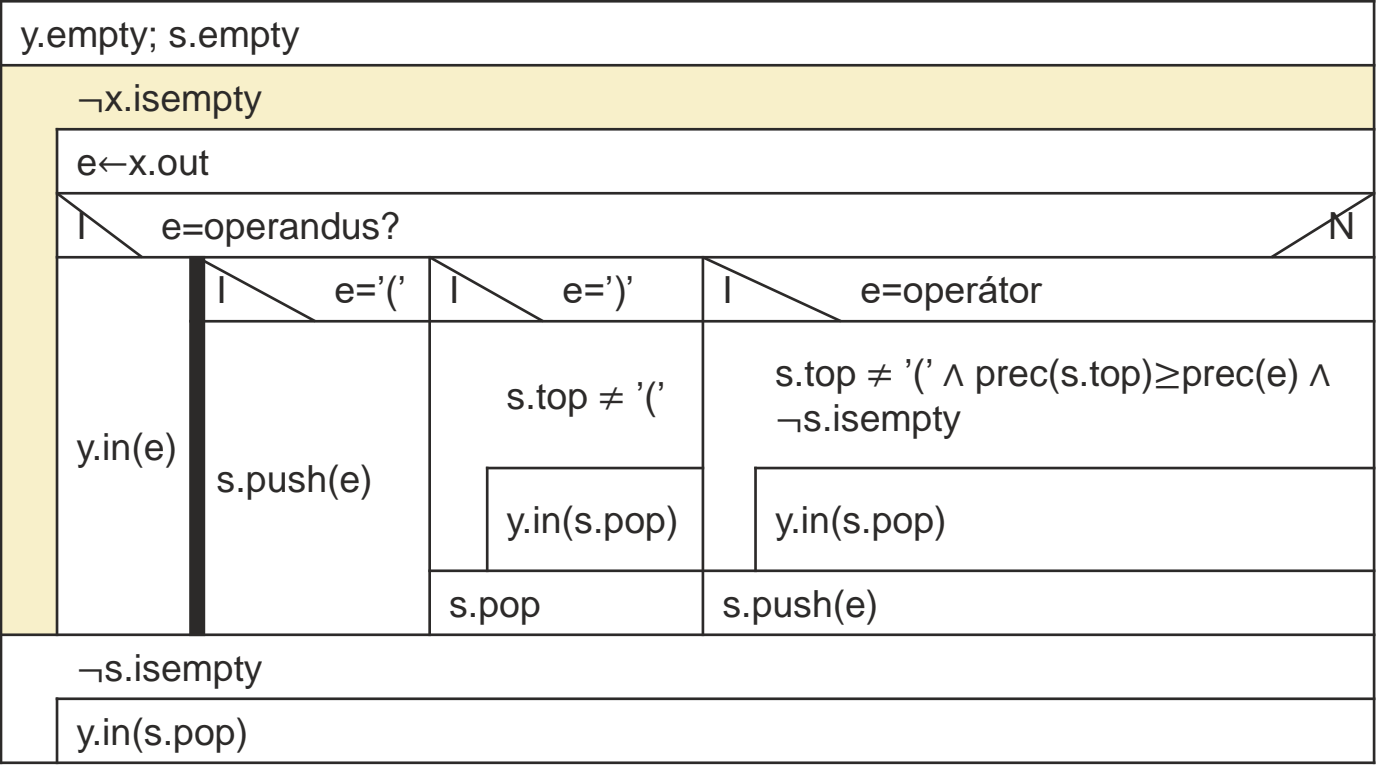


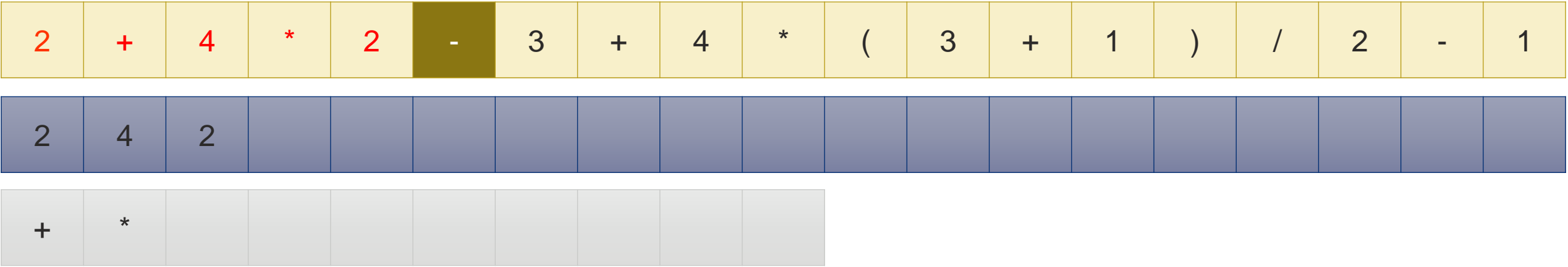
A kivett elem operandus?
Igen → így azt az y sorba tesszük.





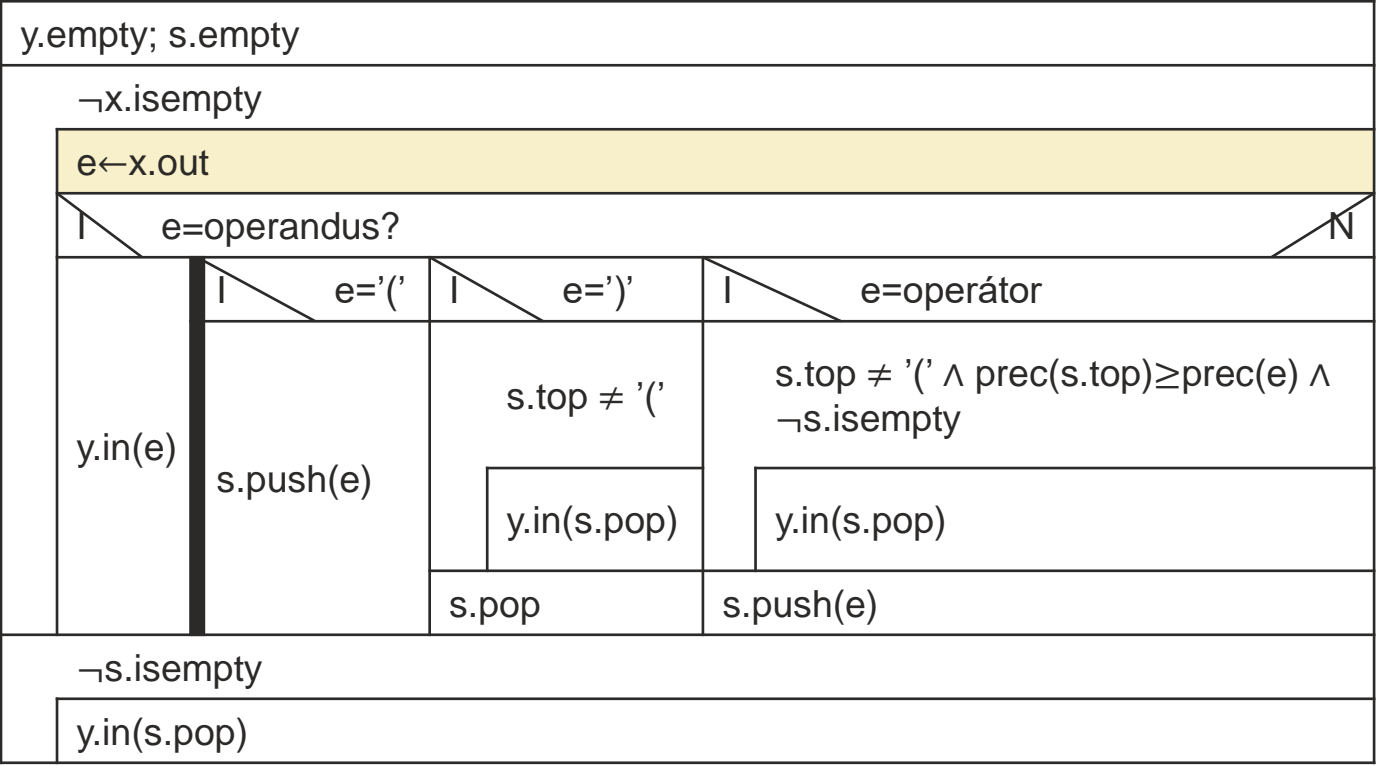
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

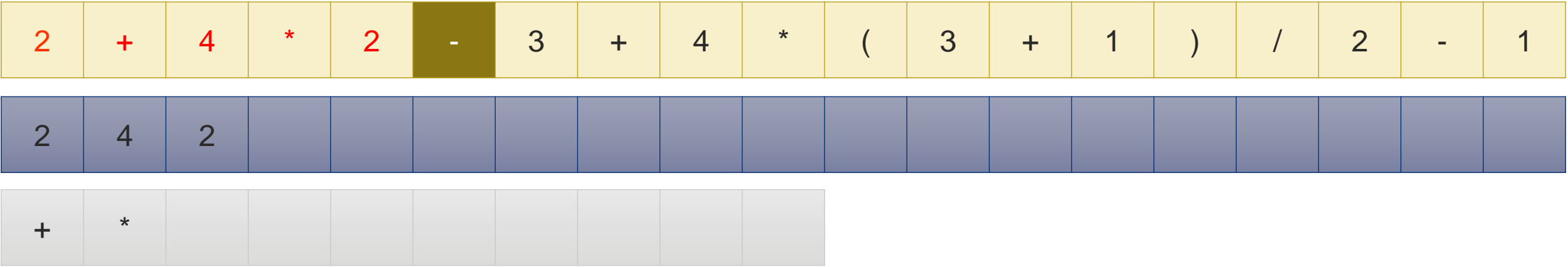




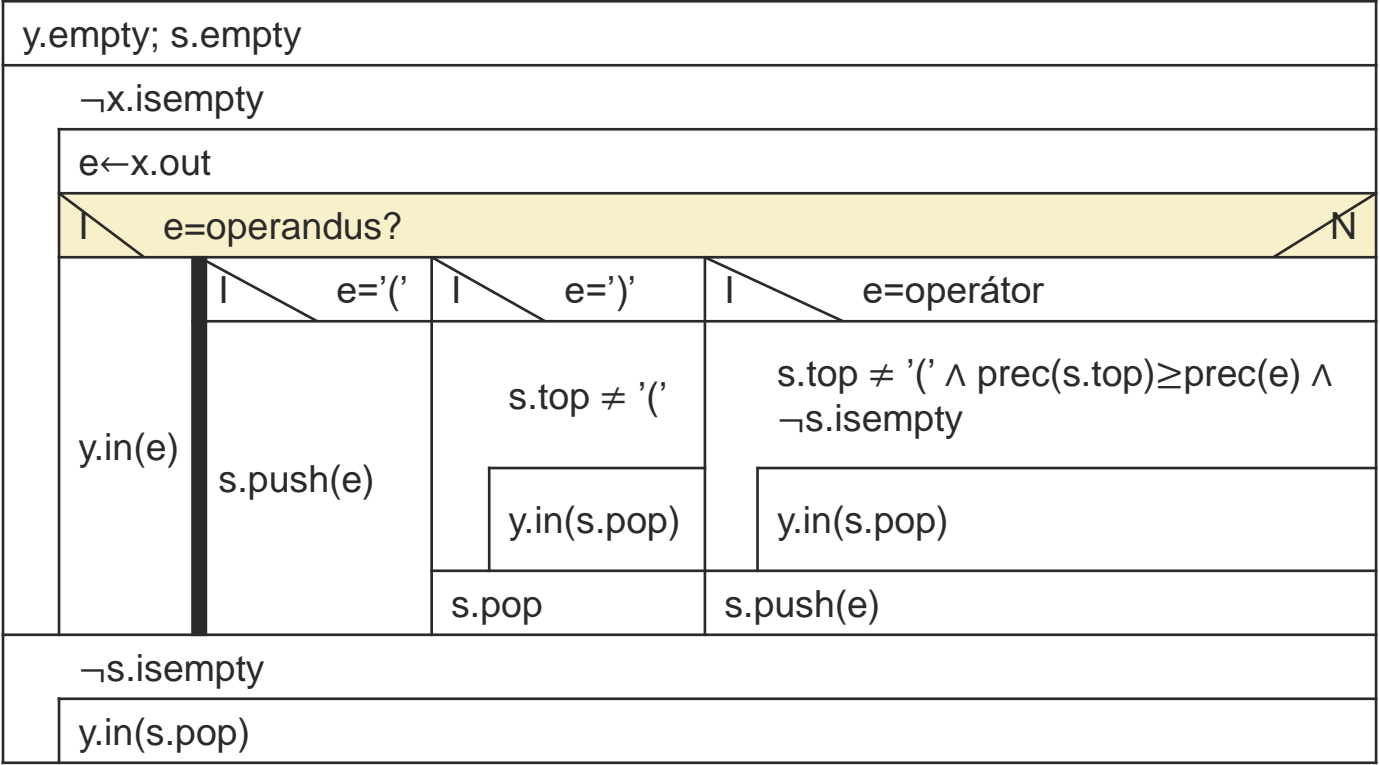
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

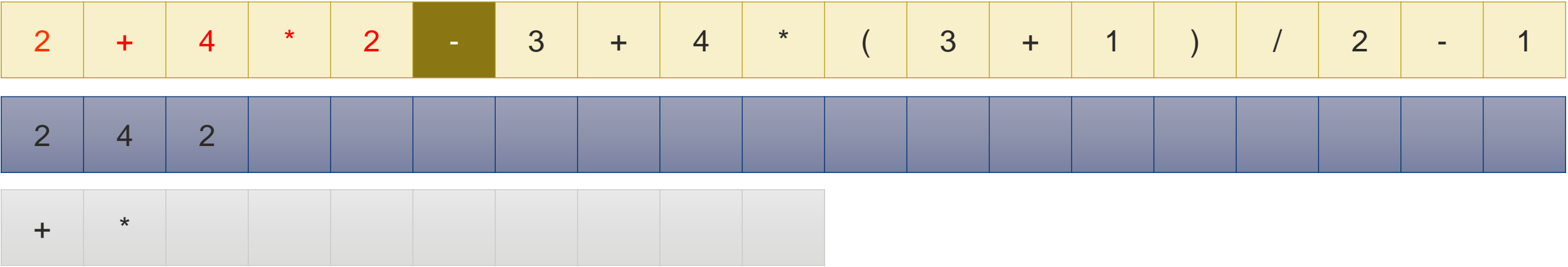
Az x sor következő elemét kivesszük a sorból.



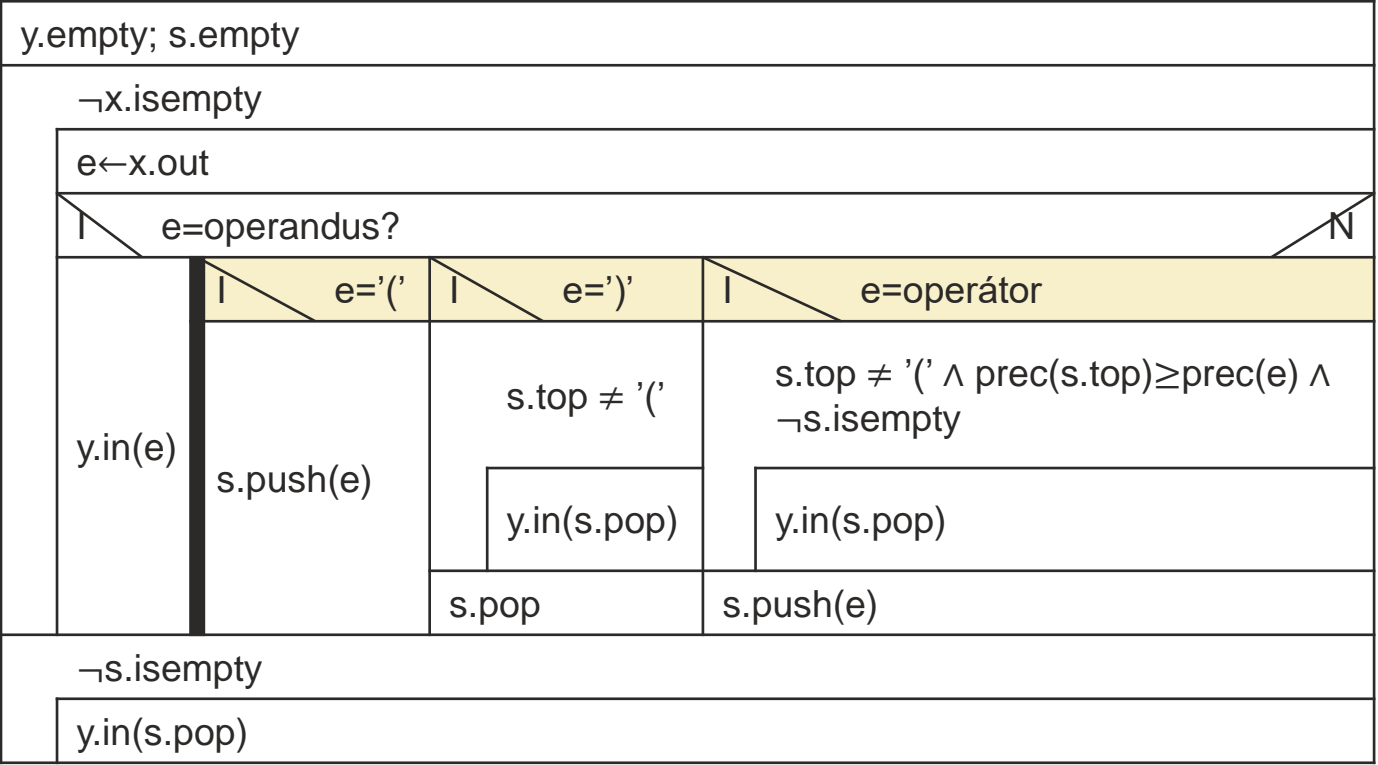


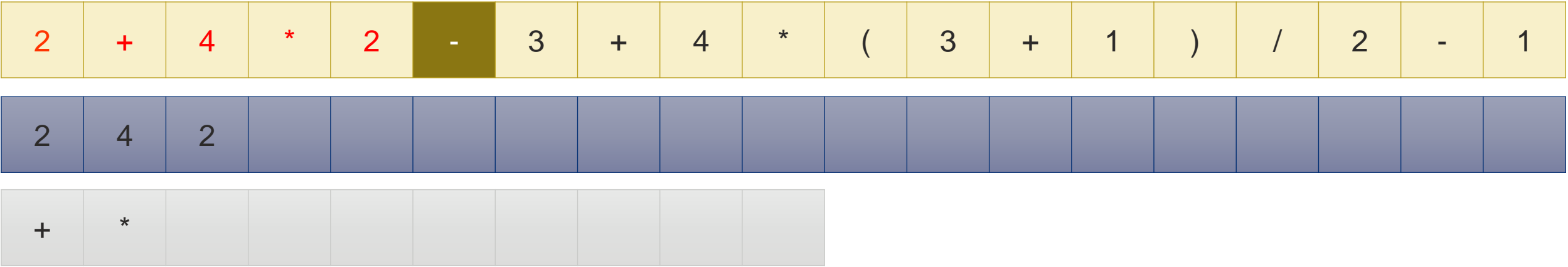
A kivett elem operandus?





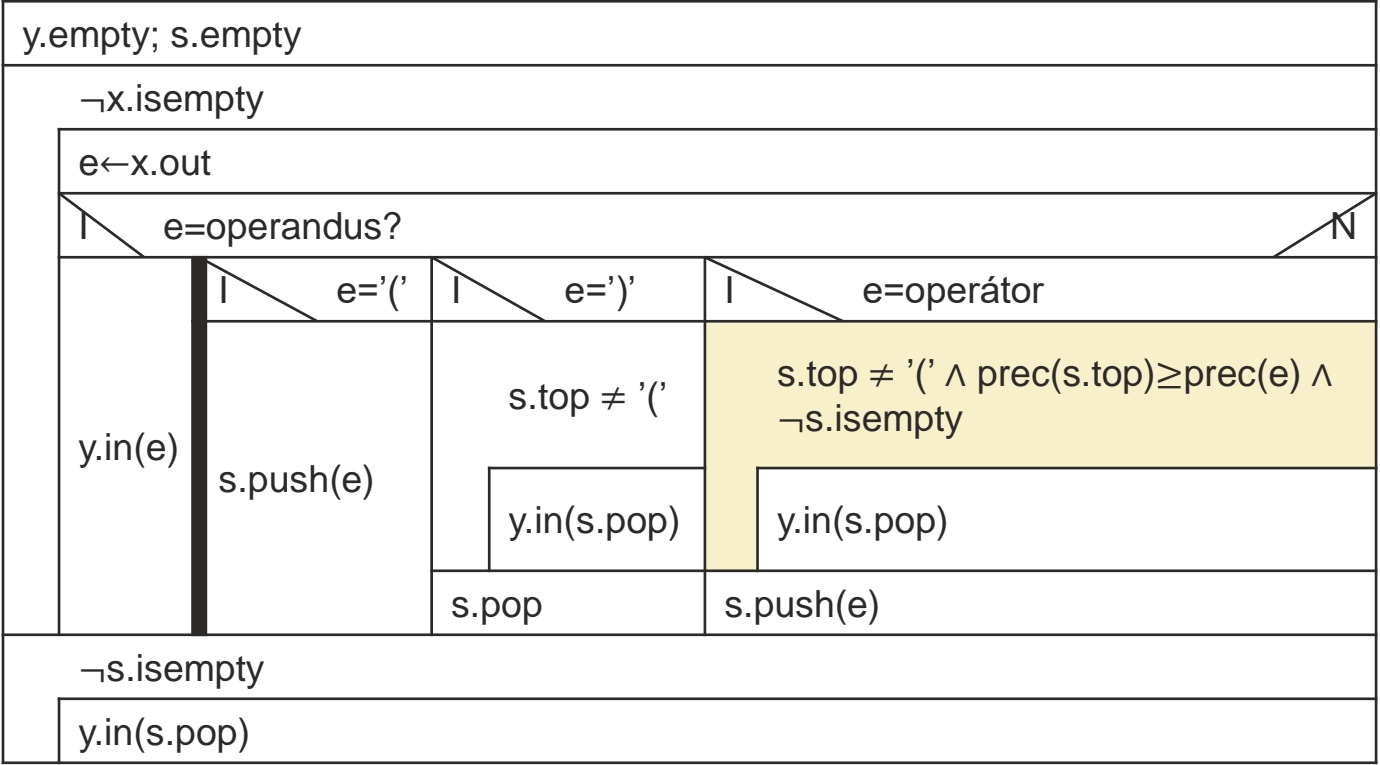
A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?

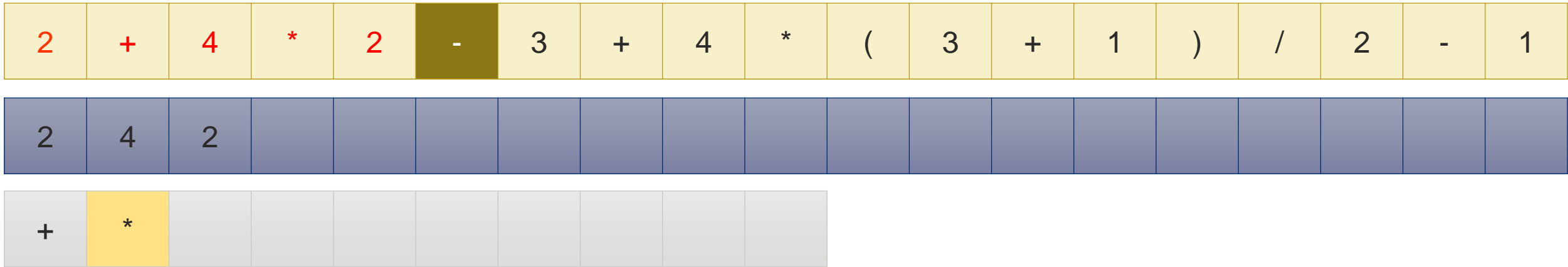




Nem → A kivett elem operátor, vagy valamely zárójel?

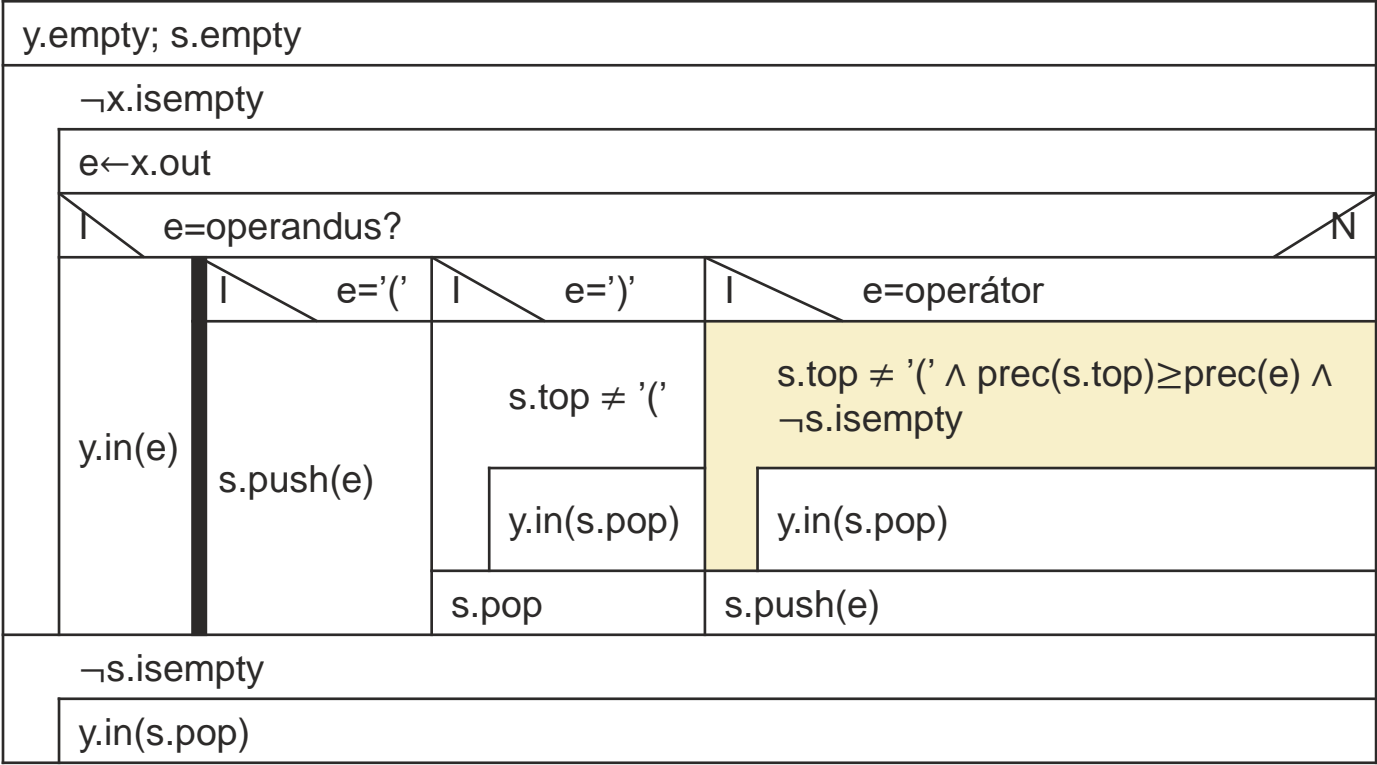
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

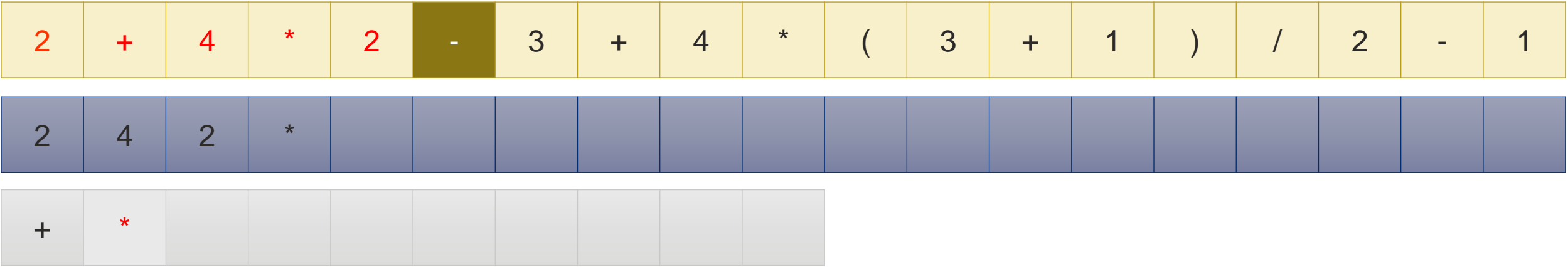




Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

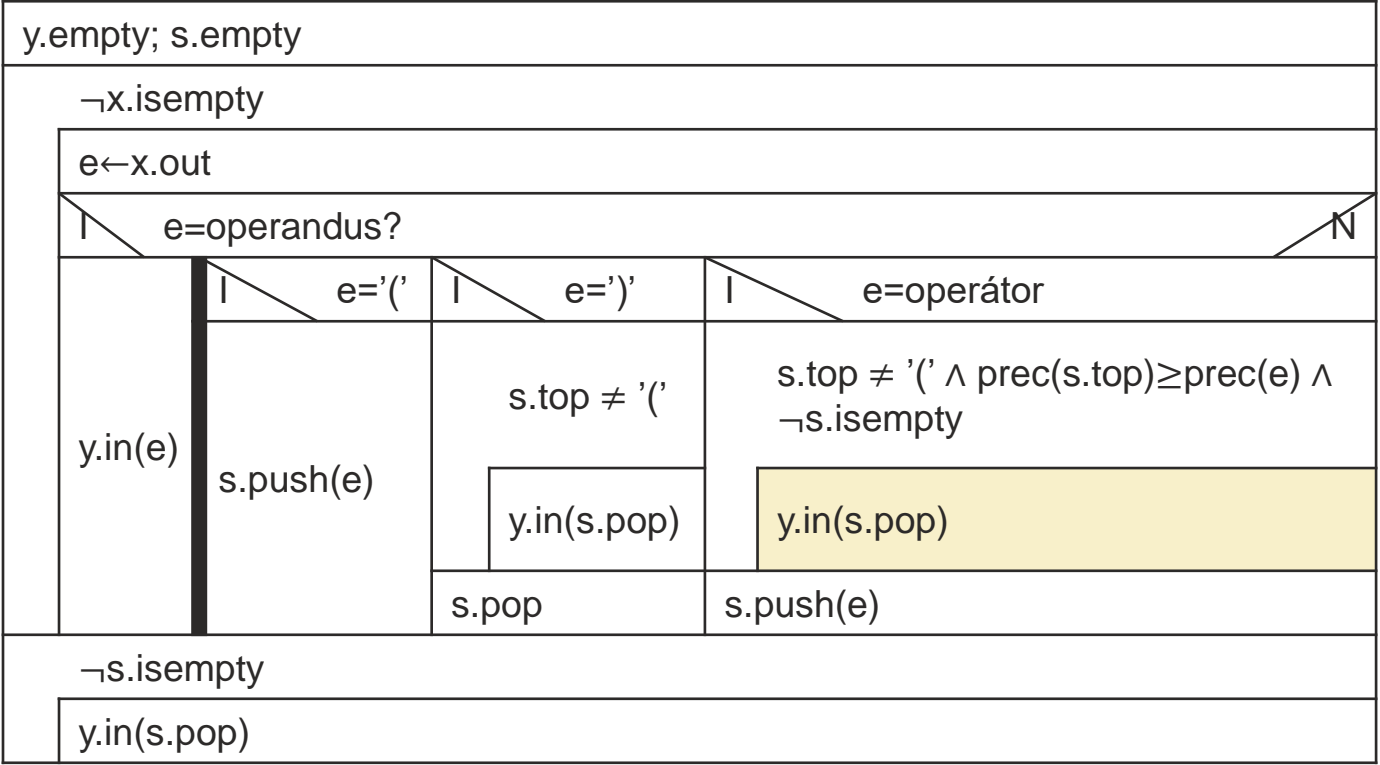
A verem tetején levő művelet (*) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.

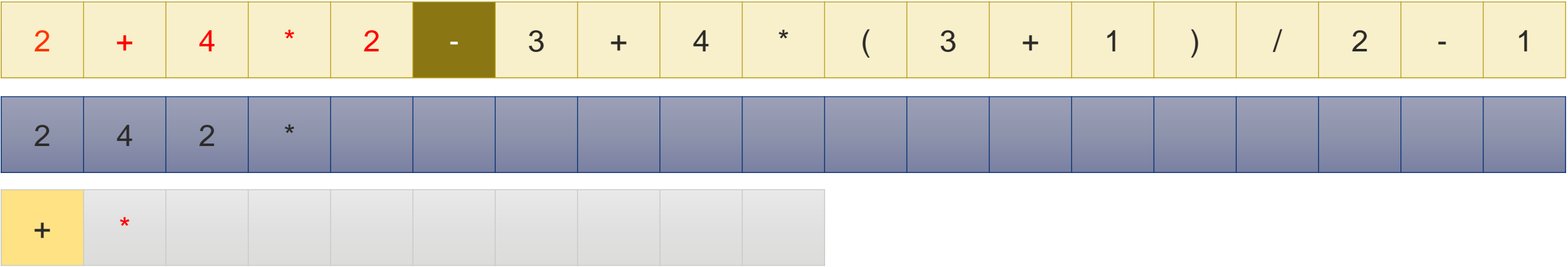




A verem tetején levő művelet (*) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.

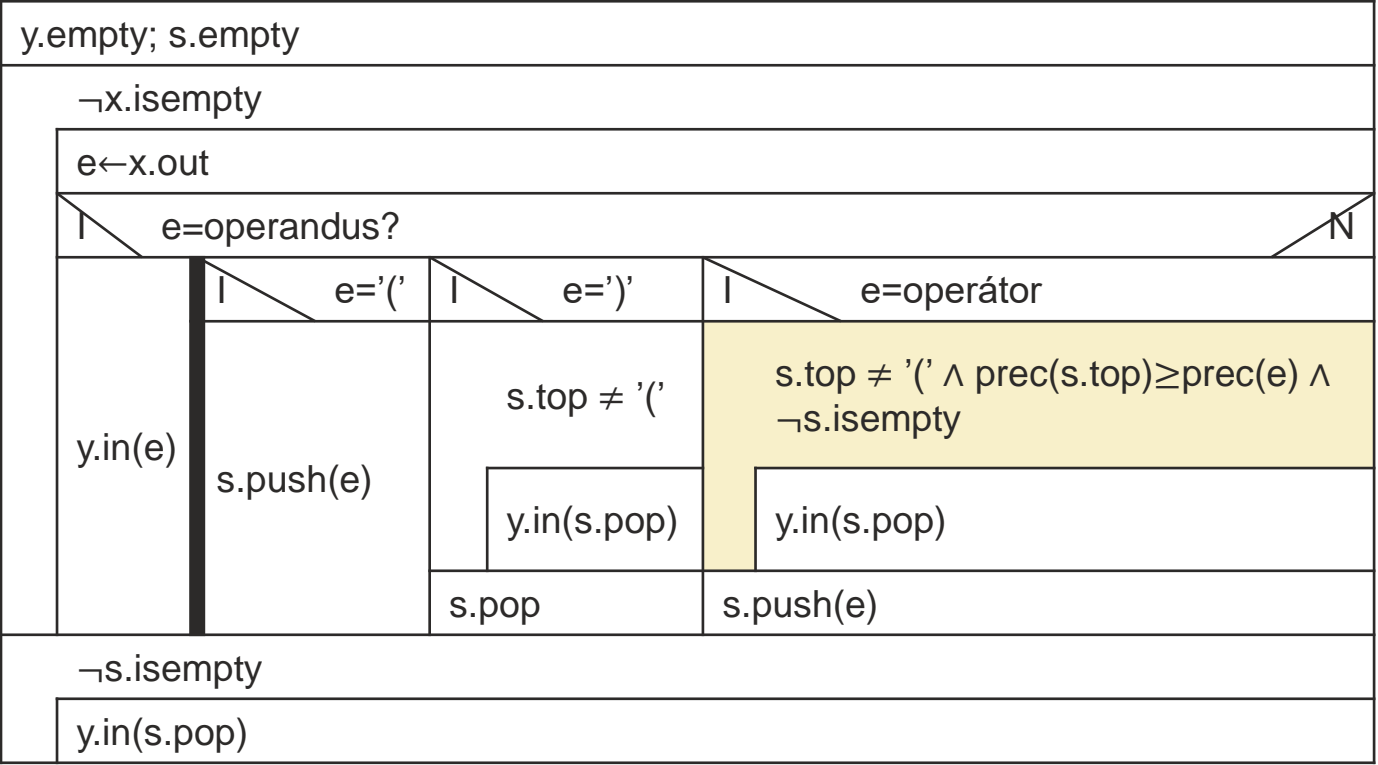
A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.

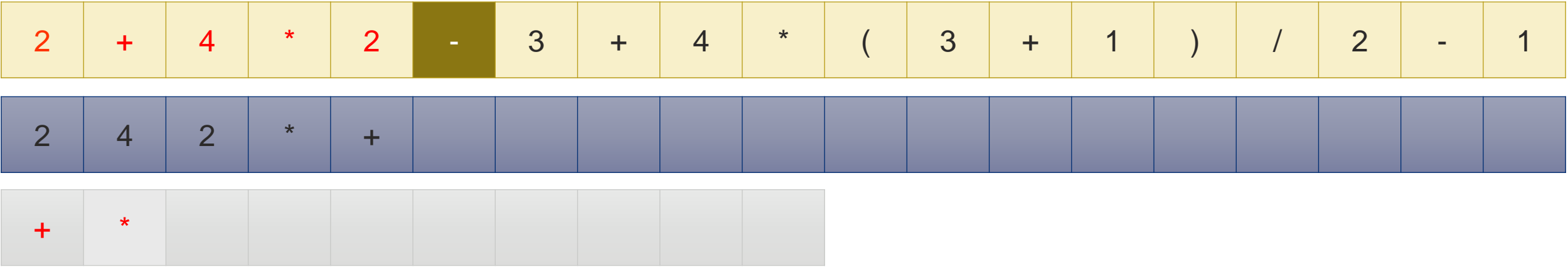




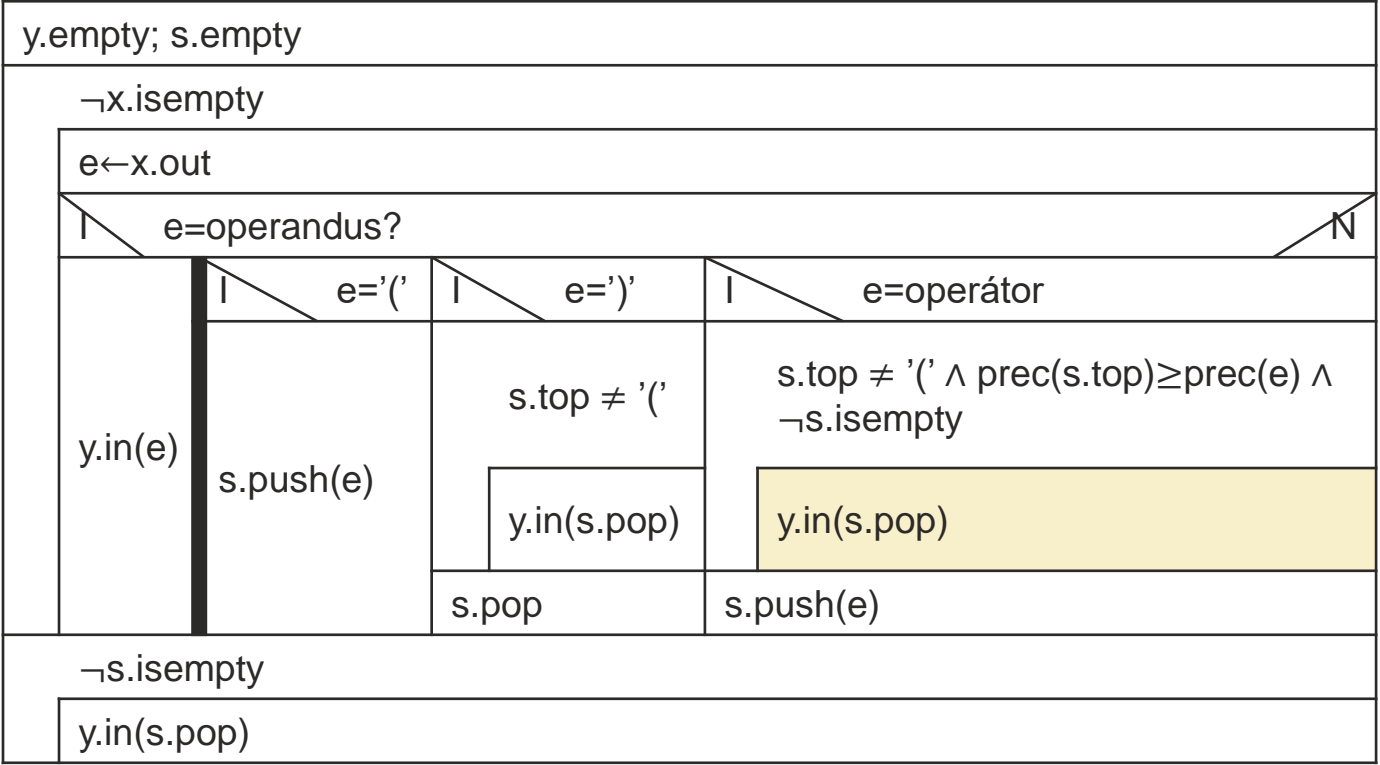
A veremben levő elemek száma csökkent.

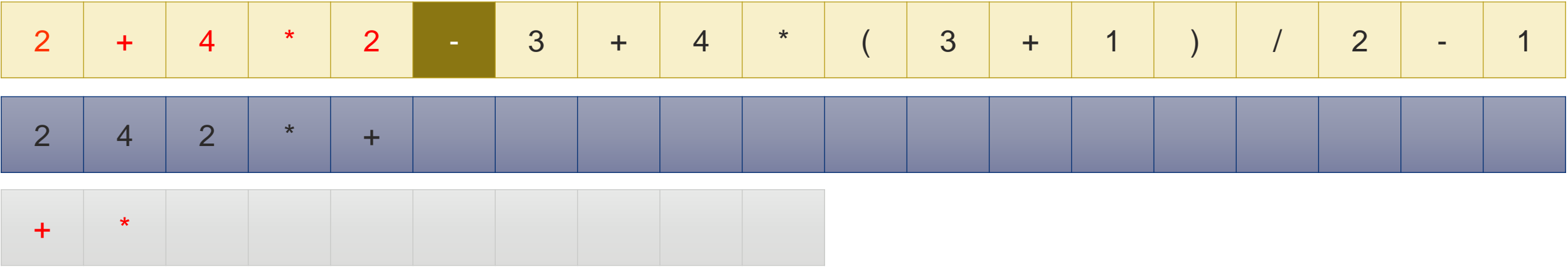
A verem tetején levő művelet (+) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.



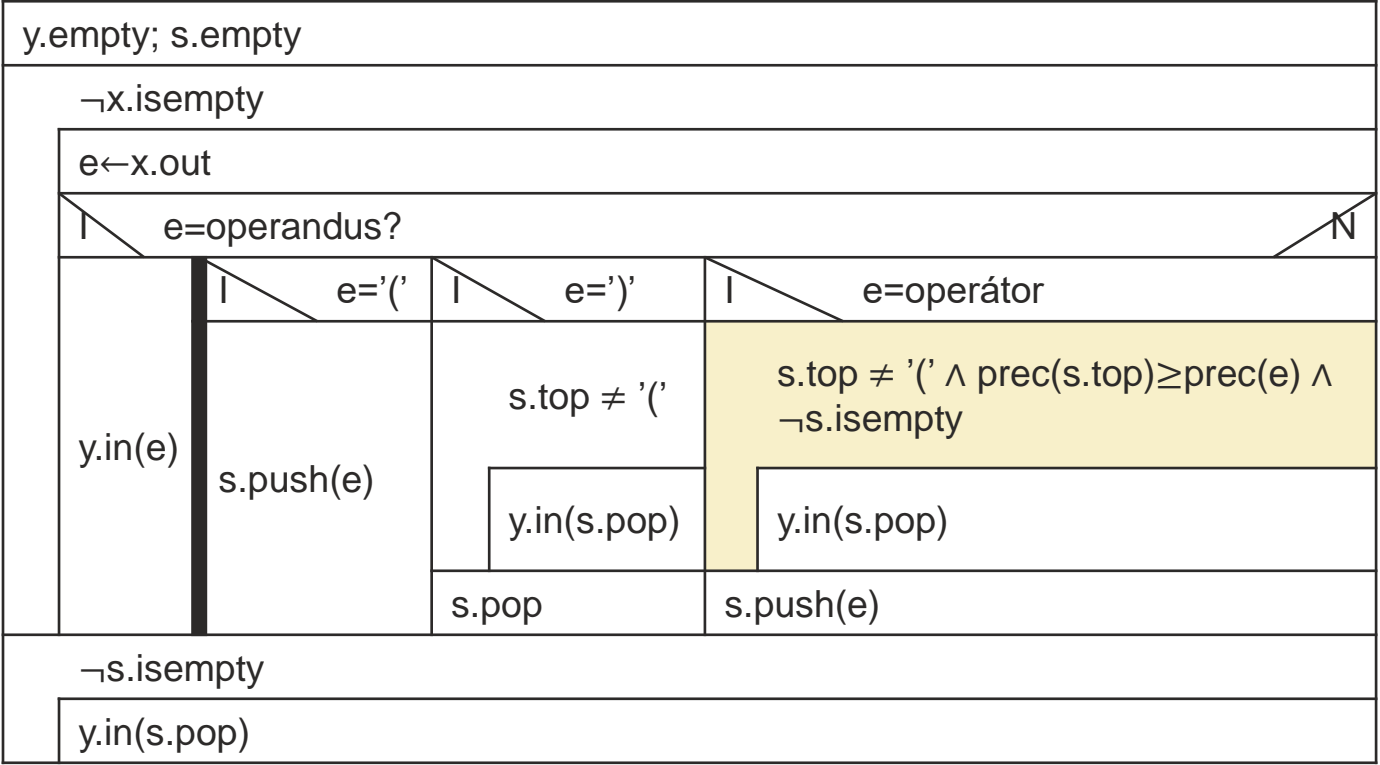


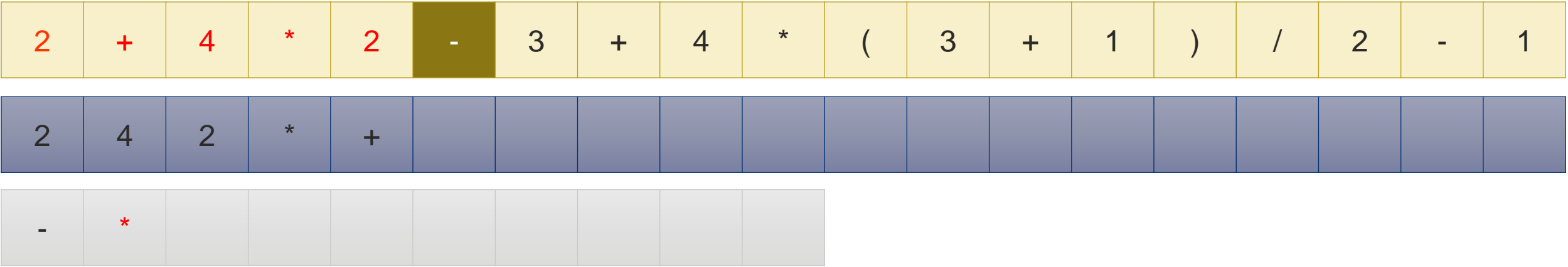
- A veremben levő elemek száma csökkent.
- A verem tetején levő művelet (+) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.
- A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.





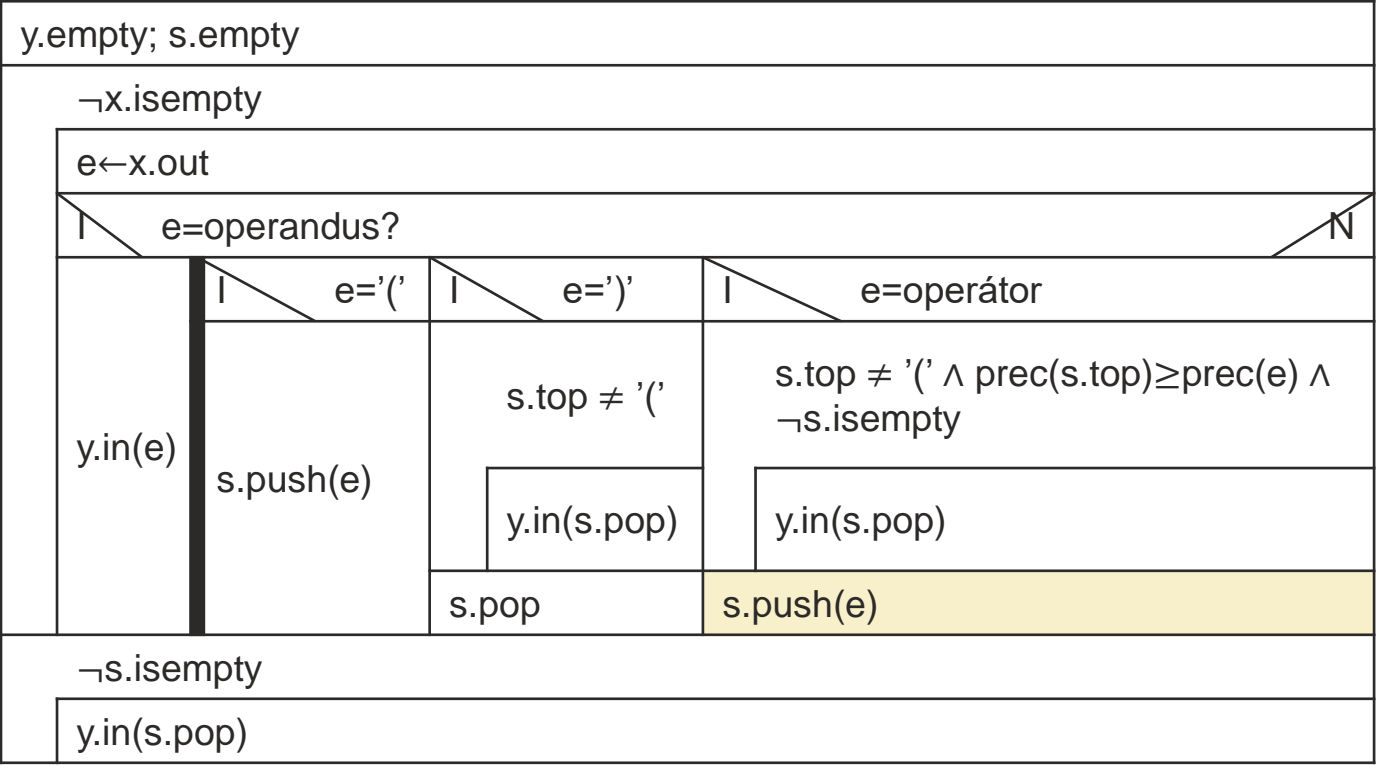
A verem kiürült, így a ciklusfeltétel nem teljesül.

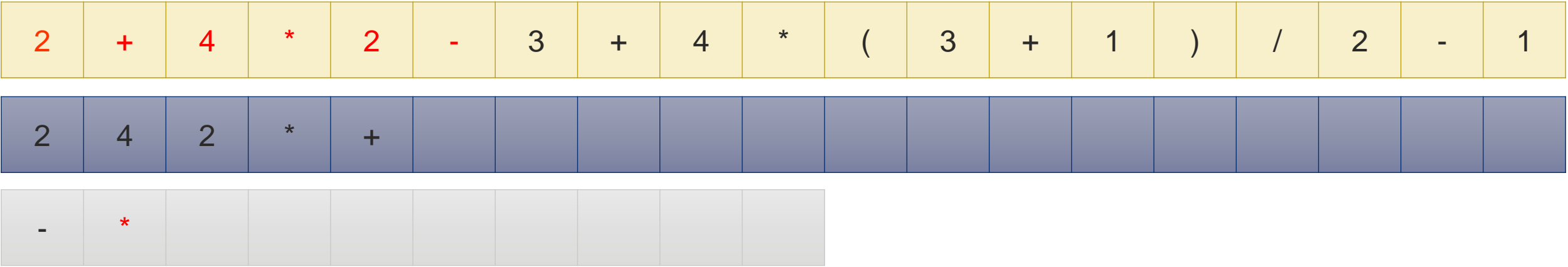




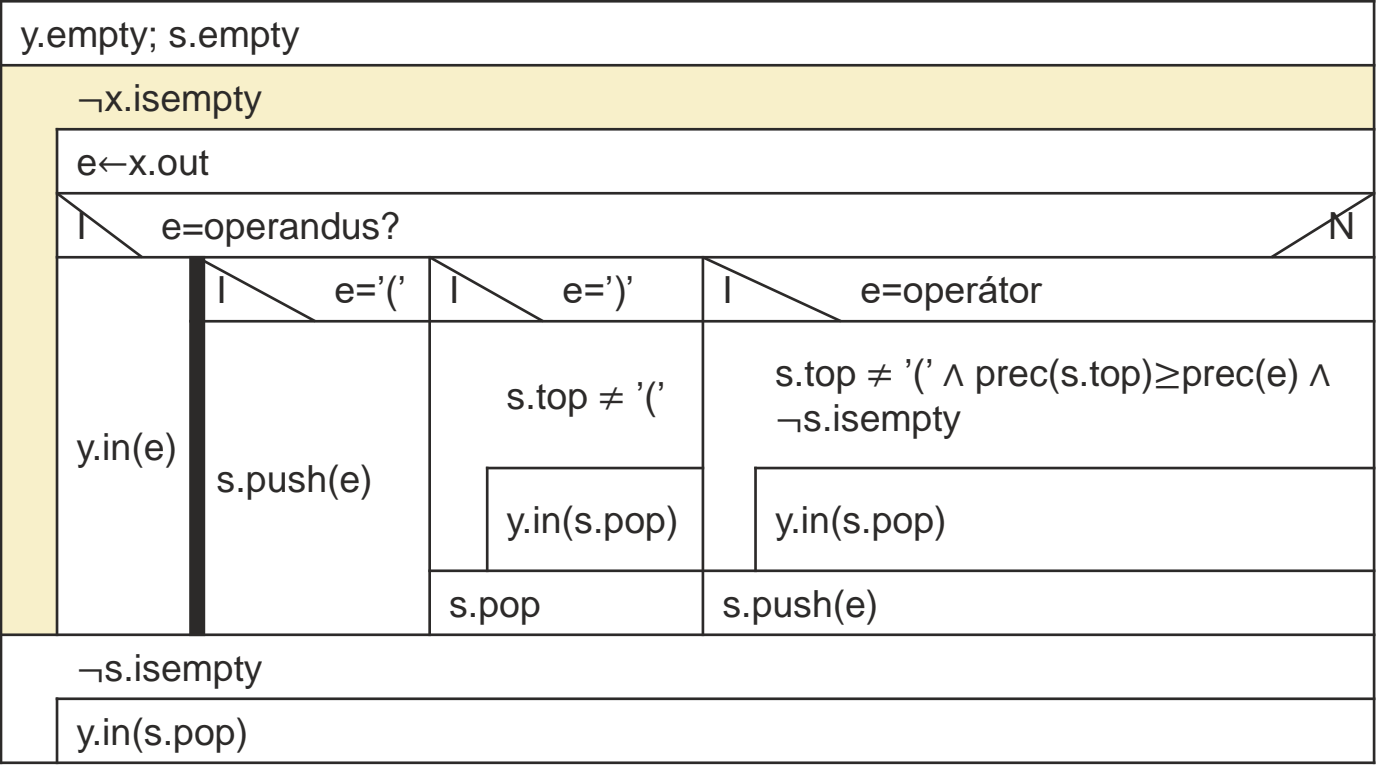
A verem kiürült, így a ciklusfeltétel nem teljesül.

Végül az aktuális operátort a verem tetejére tesszük.





Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+														
-	*																	

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

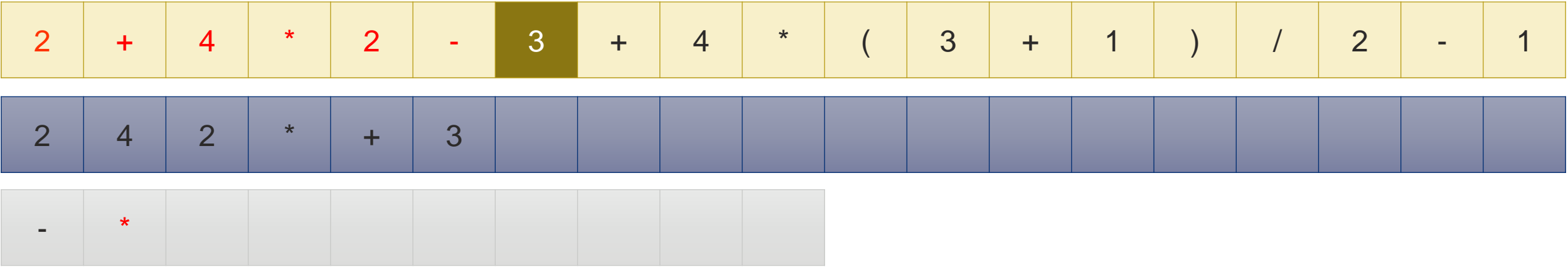
Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I e=operandus? N				
y.in(e)	I e='('		I e=')'	I e=operátor
	s.push(e)		s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
			y.in(s.pop)	y.in(s.pop)
			s.pop	s.push(e)
¬s.isempty				
y.in(s.pop)				

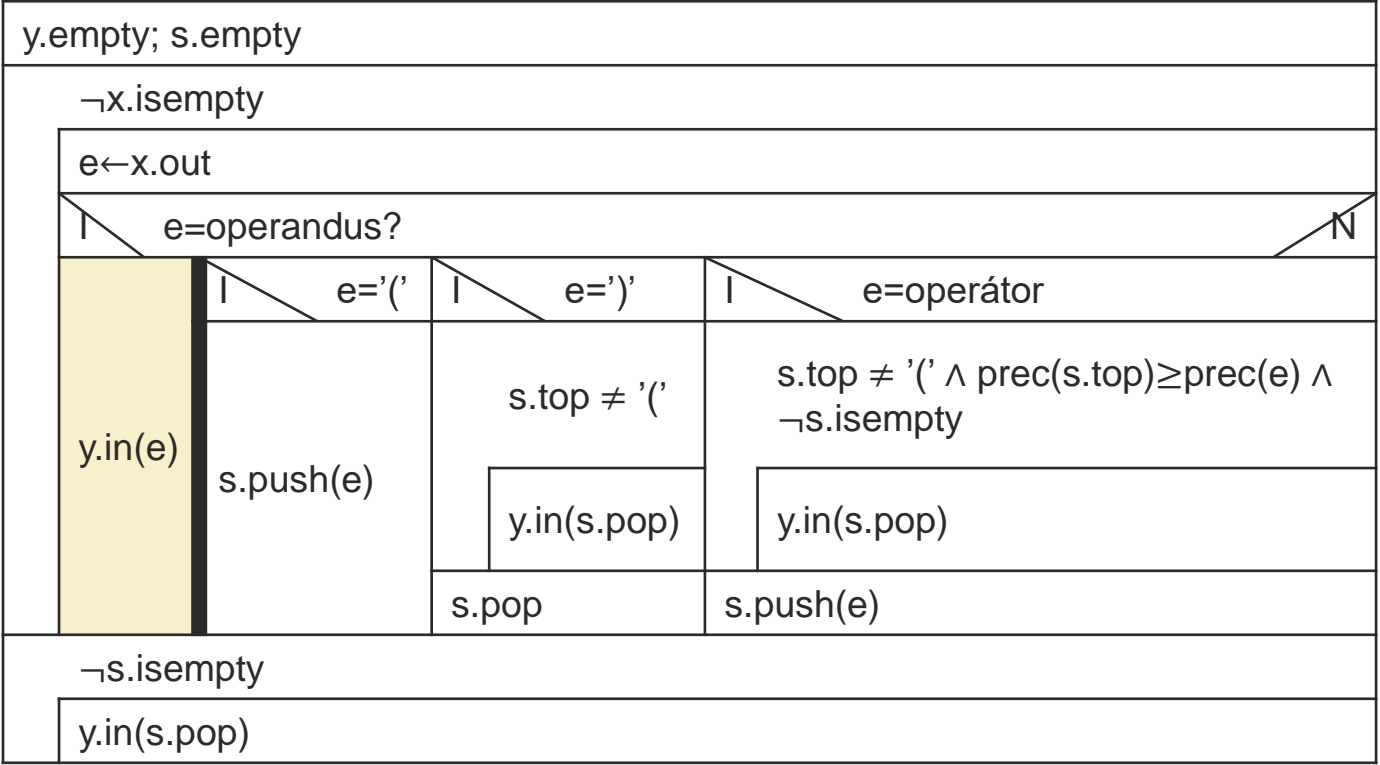
2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+														
-	*																	

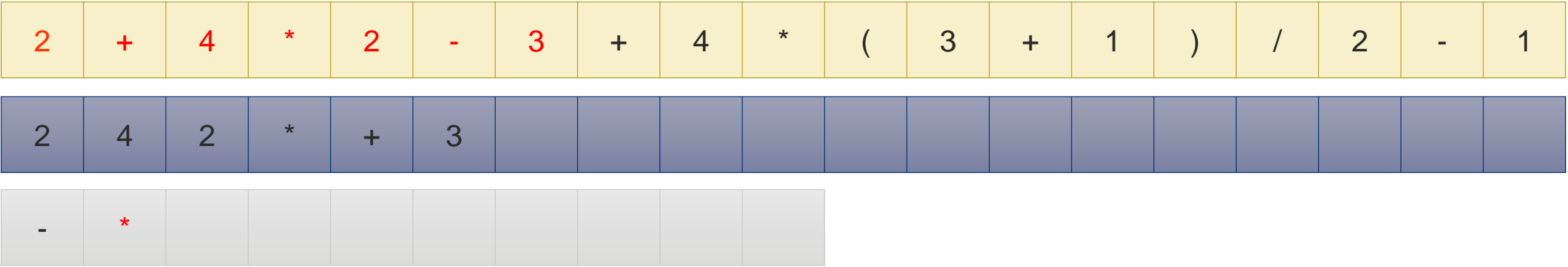
A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
y.in(e)	e=operandus?		
	e='('	e=')'	e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

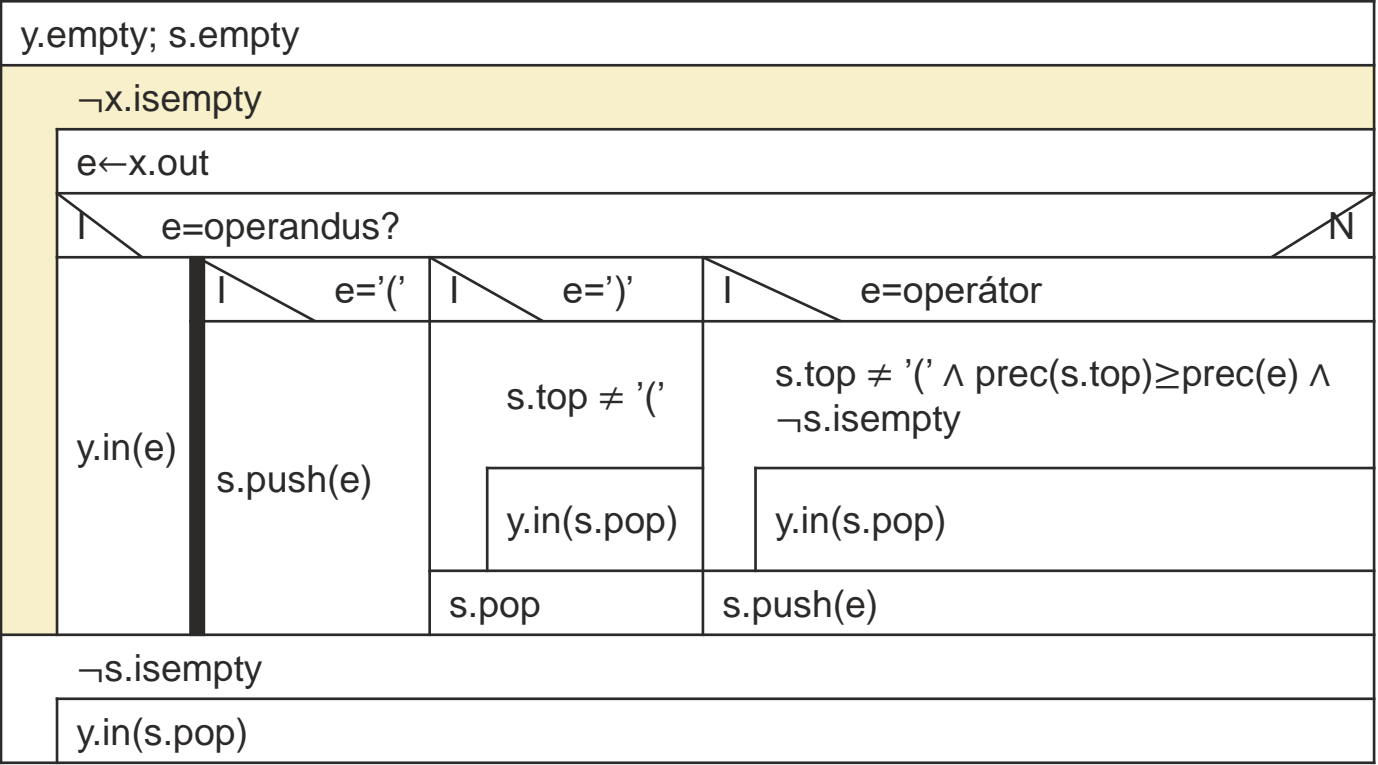


A kivett elem operandus?
Igen → így azt az y sorba tesszük.





Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

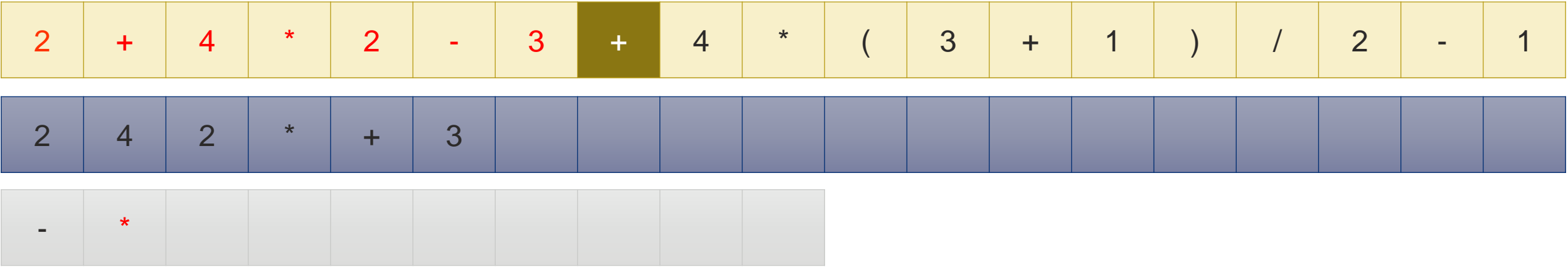


2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3													
-	*																	

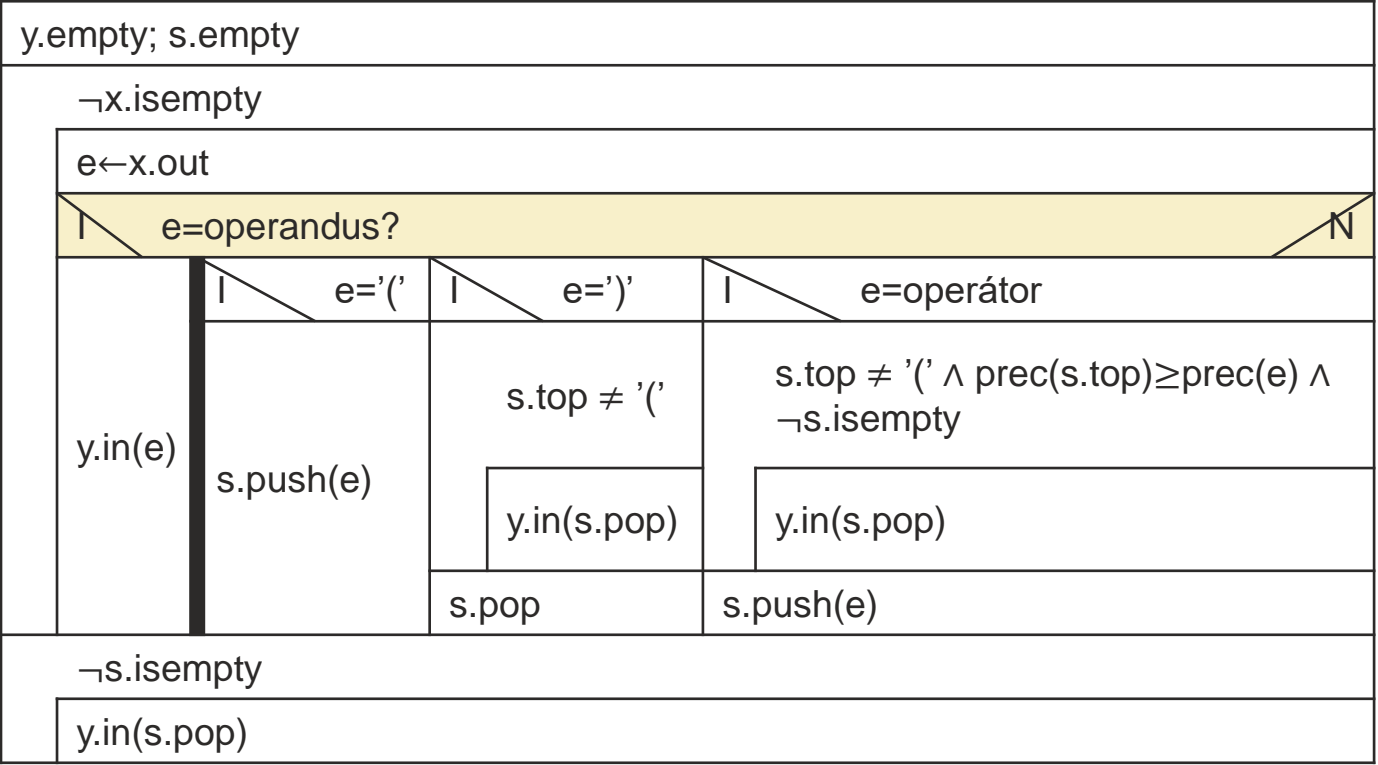
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

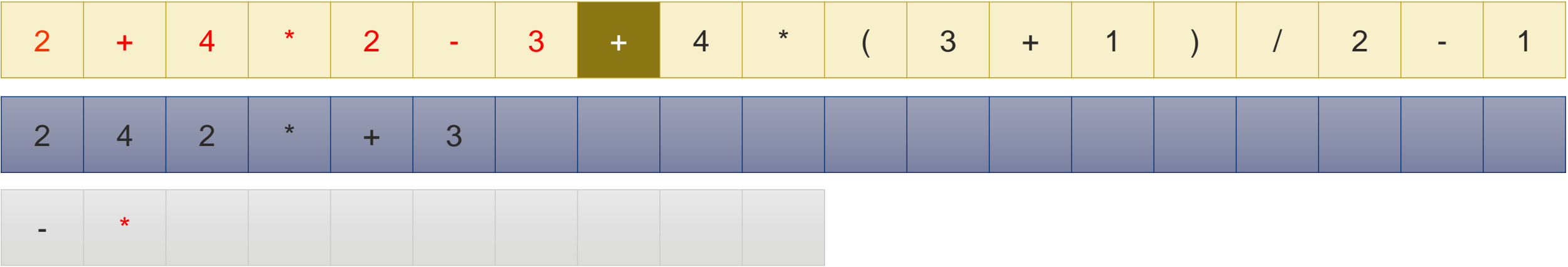
Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('		I / e=')'	
	s.push(e)		I / e=operátor	
			s.top ≠ '('	
			s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
			y.in(s.pop)	
		s.pop		s.push(e)
¬s.isempty				
y.in(s.pop)				

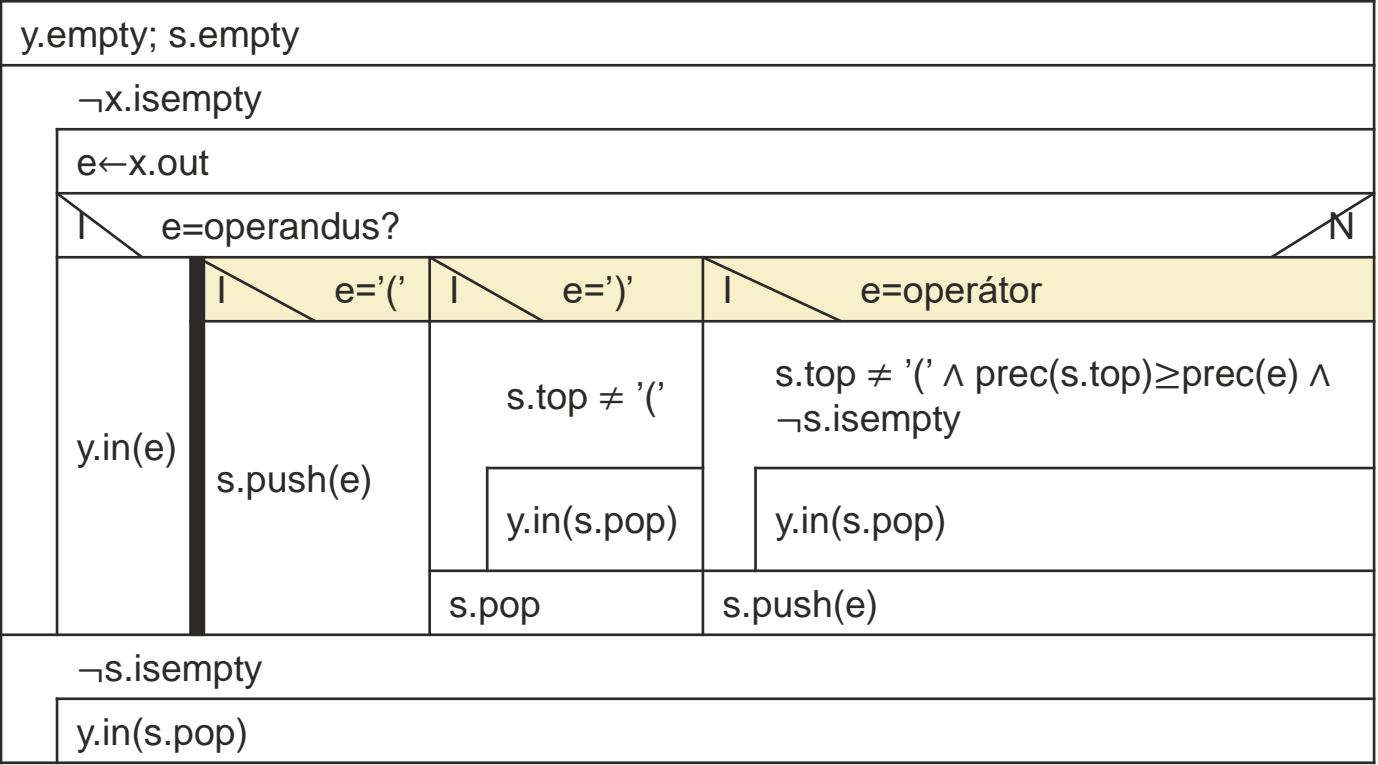


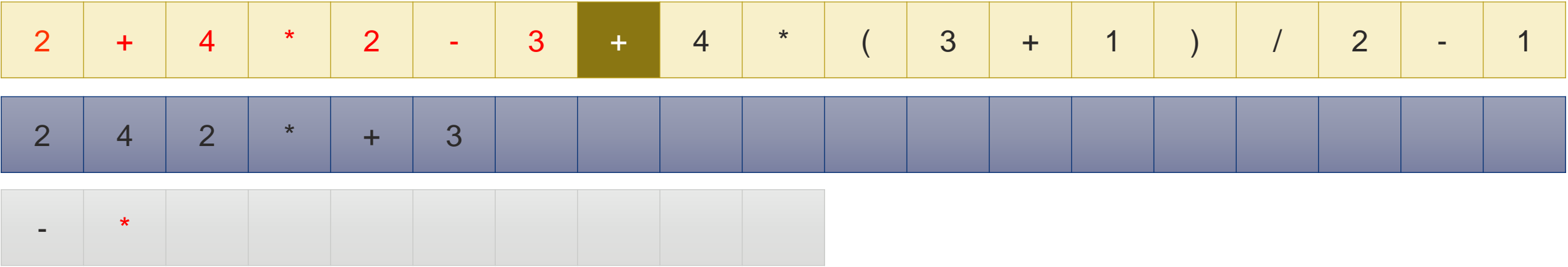
A kivett elem operandus?





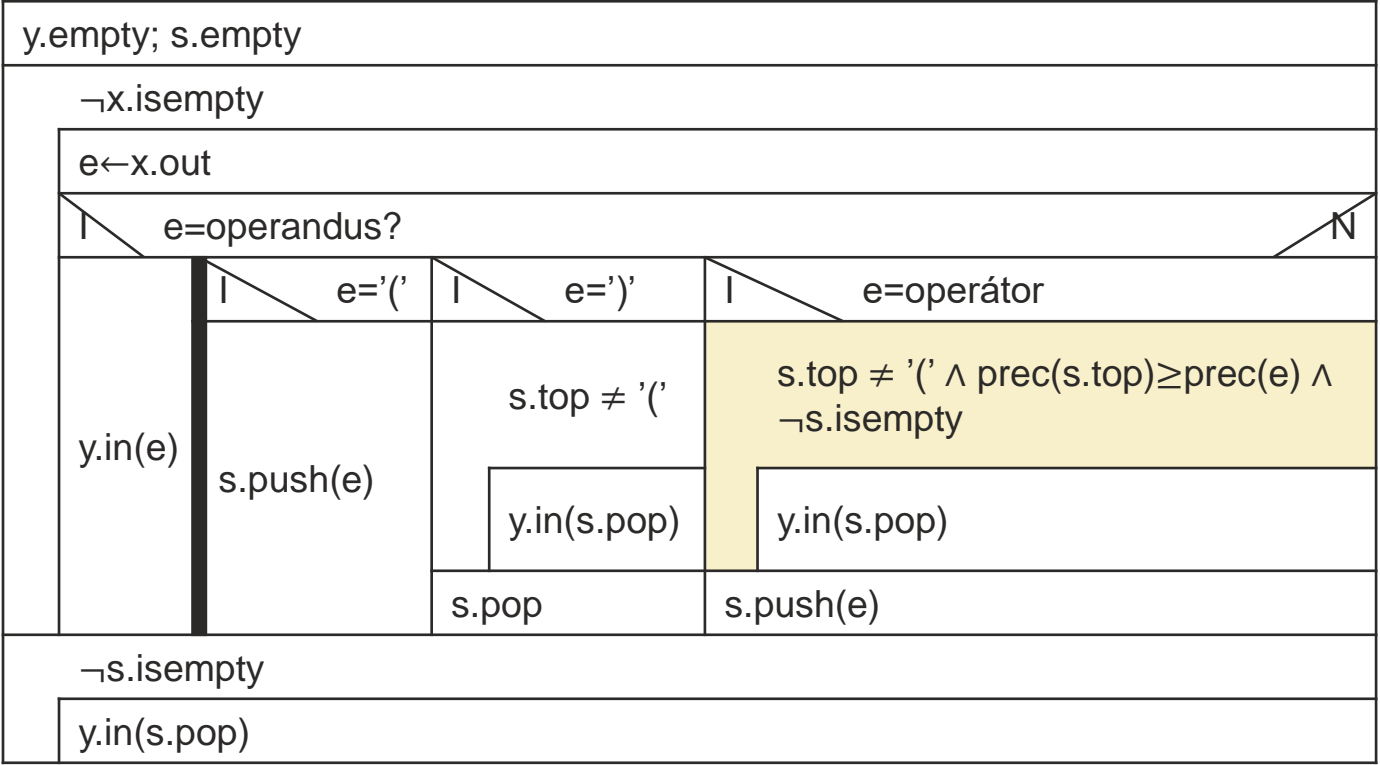
A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?

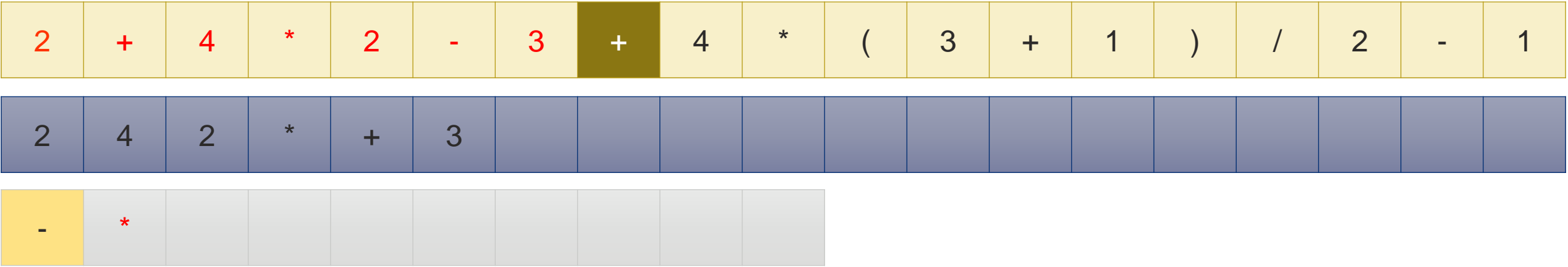




Nem → A kivett elem operátor, vagy valamely zárójel?

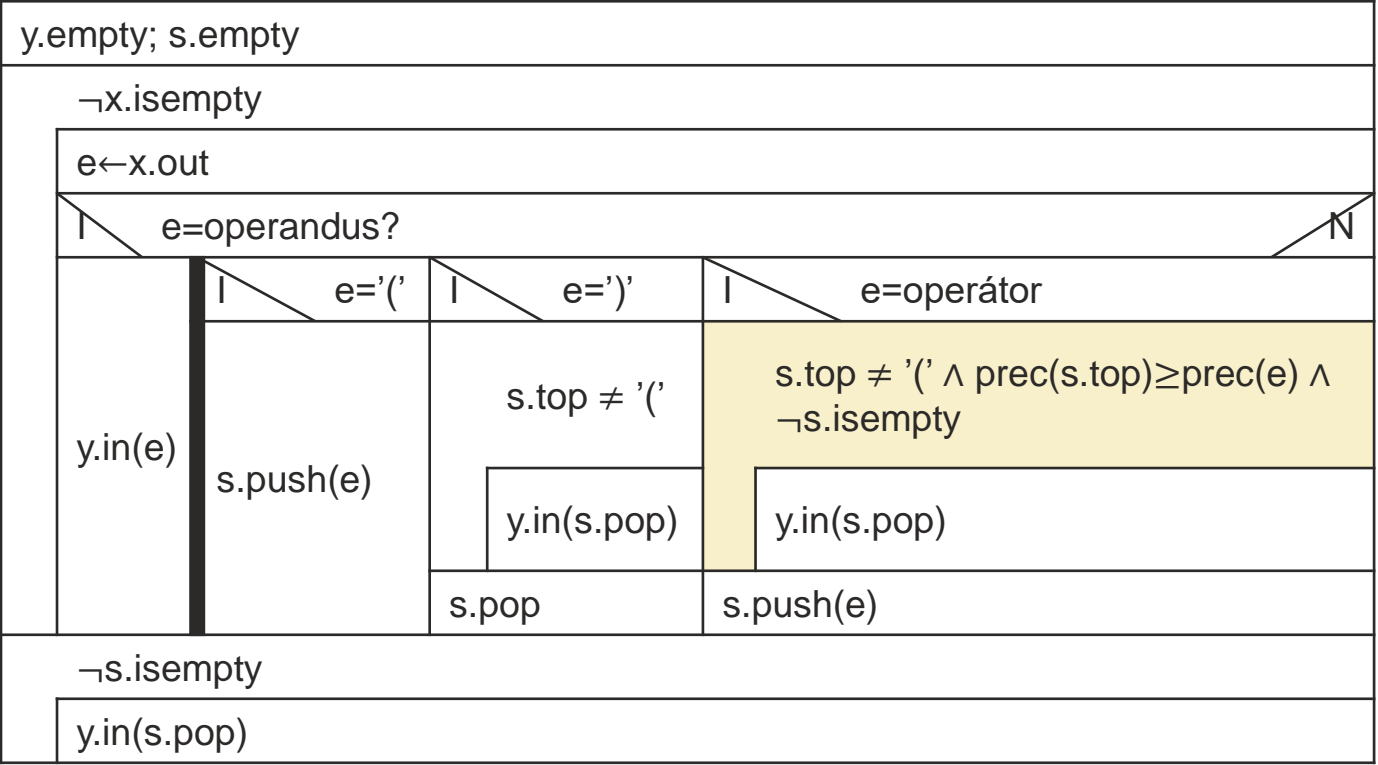
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.





Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

A verem tetején levő művelet (-) precedenciája nem kisebb az aktuálisnál (+), ezért a ciklusba belépünk.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-												
-	*																	

A verem tetején levő művelet (-) precedenciája nem kisebb az aktuálisnál (+), ezért a ciklusba belépünk.

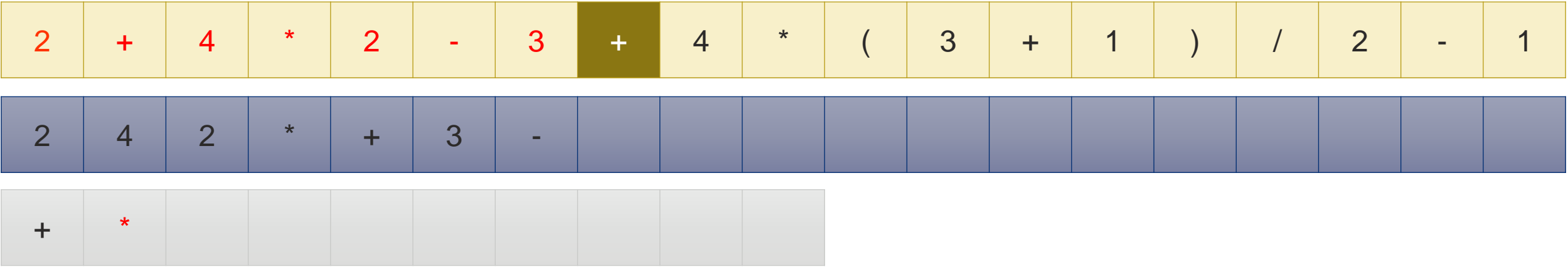
A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-												
-	*																	

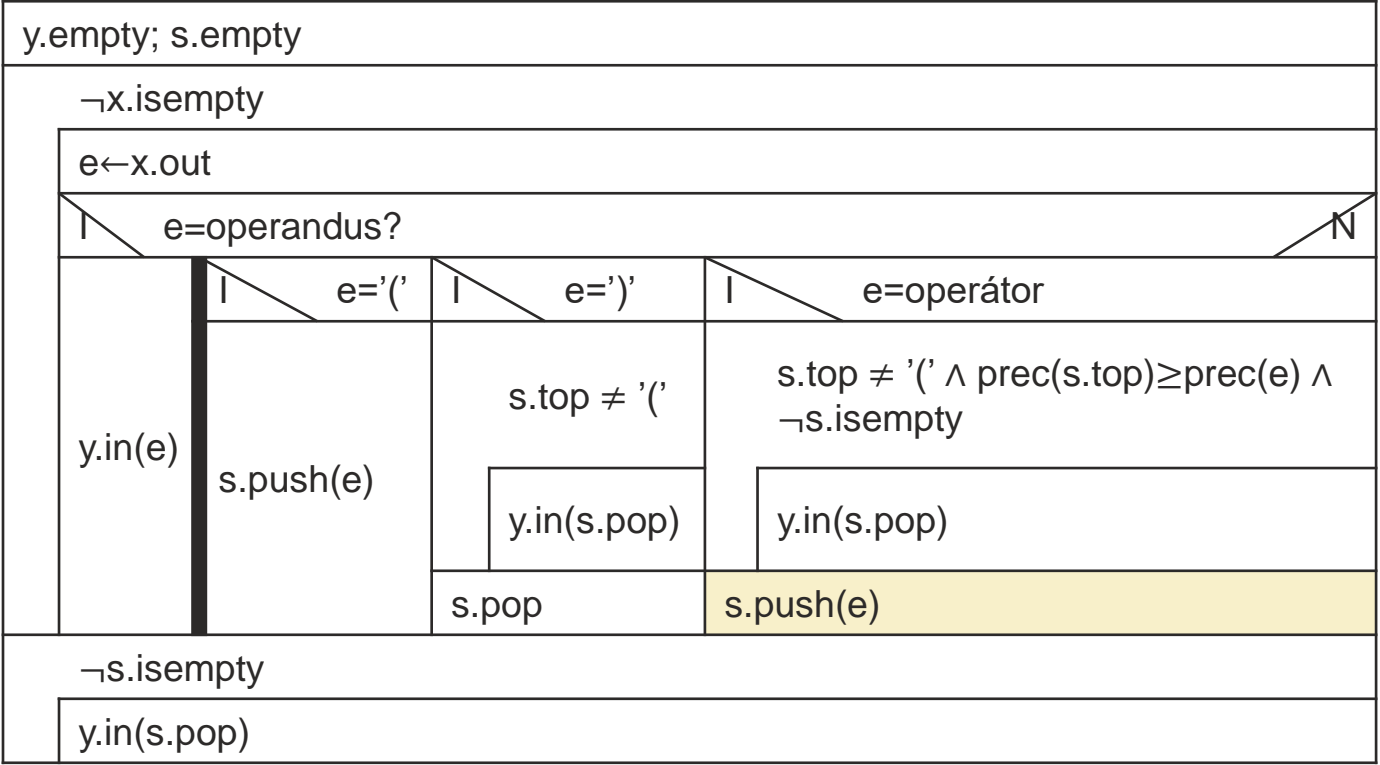
A verem kiürült, így a ciklusfeltétel nem teljesül.

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div>e=operandus?</div> <div> <div>I</div> <div>N</div> </div> </div>			
y.in(e)	<div> <div>e='('</div> <div> <div>s.push(e)</div> </div> </div>		<div> <div>e=')'</div> <div> <div>s.top ≠ '('</div> <div> <div>y.in(s.pop)</div> <div>s.pop</div> </div> </div> </div>
			<div> <div>e=operátor</div> <div> <div>s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty</div> <div> <div>y.in(s.pop)</div> <div>s.push(e)</div> </div> </div> </div>
	¬s.isempty		
	y.in(s.pop)		



A verem kiürült, így a ciklusfeltétel nem teljesül.

Végül az aktuális operátort a verem tetejére tesszük.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-												
+	*																	

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('		I / e=')'	
	s.push(e)		I / e=operátor	
			s.top ≠ '('	
	y.in(s.pop)		s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
	s.pop		y.in(s.pop)	
s.push(e)				
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-												
+	*																	

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

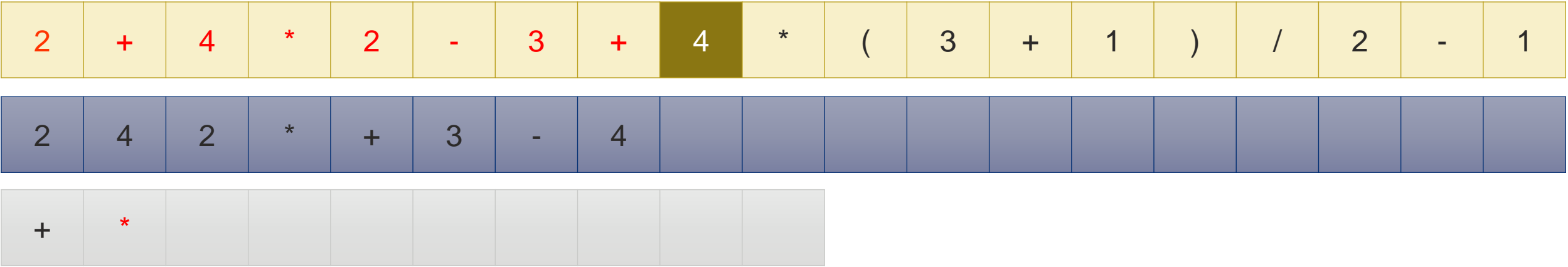
Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div></div> <div>e=operandus?</div> <div>N</div> </div>			
y.in(e)	<div> <div></div> <div>e='('</div> </div>		<div> <div></div> <div>e=')'</div> </div>
	s.push(e)		<div> <div>s.top ≠ '('</div> <div> <div>y.in(s.pop)</div> <div>s.pop</div> </div> </div>
			<div> <div>e=operátor</div> <div> <div>s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty</div> <div> <div>y.in(s.pop)</div> <div>s.push(e)</div> </div> </div> </div>
	<div> <div>¬s.isempty</div> <div>y.in(s.pop)</div> </div>		

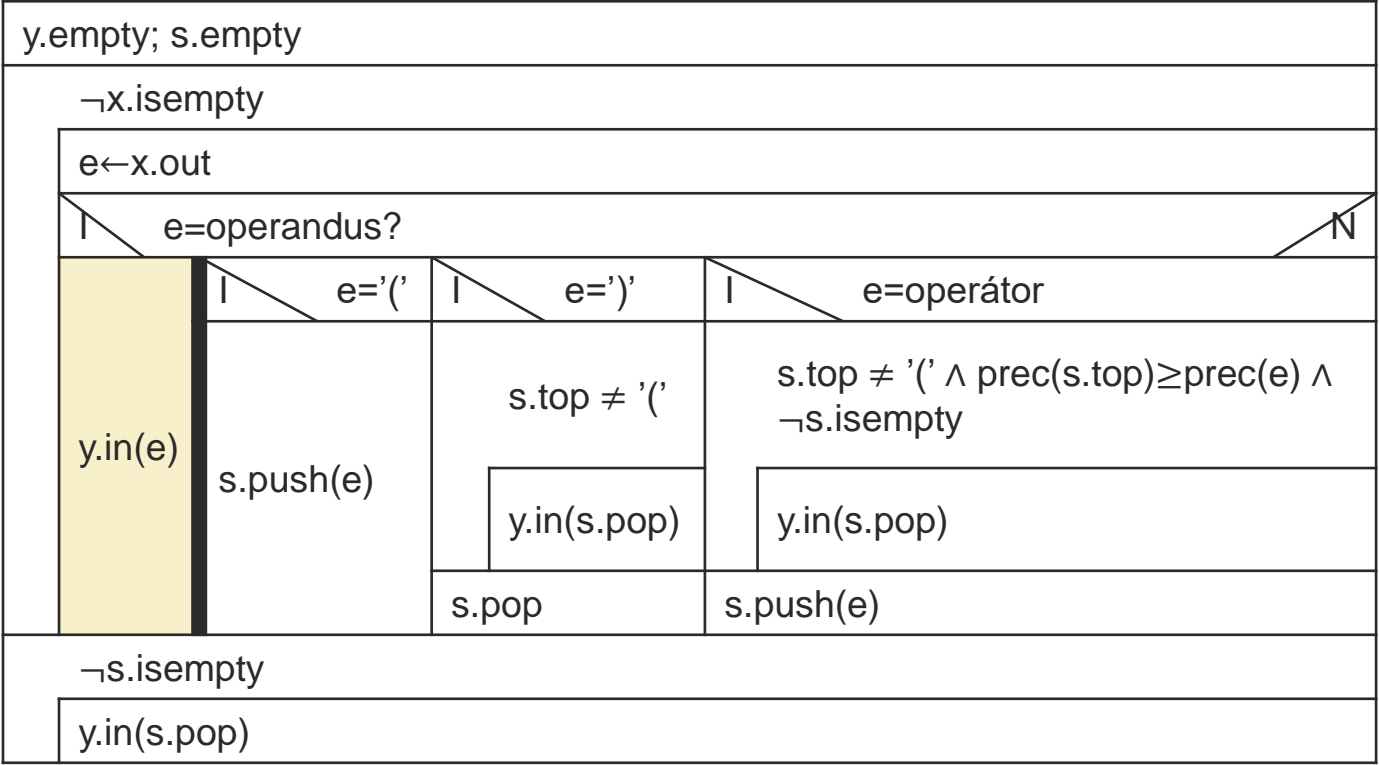
2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-												
+	*																	

A kivett elem operandus?

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				



A kivett elem operandus?
Igen → így azt az y sorba tesszük.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*																	

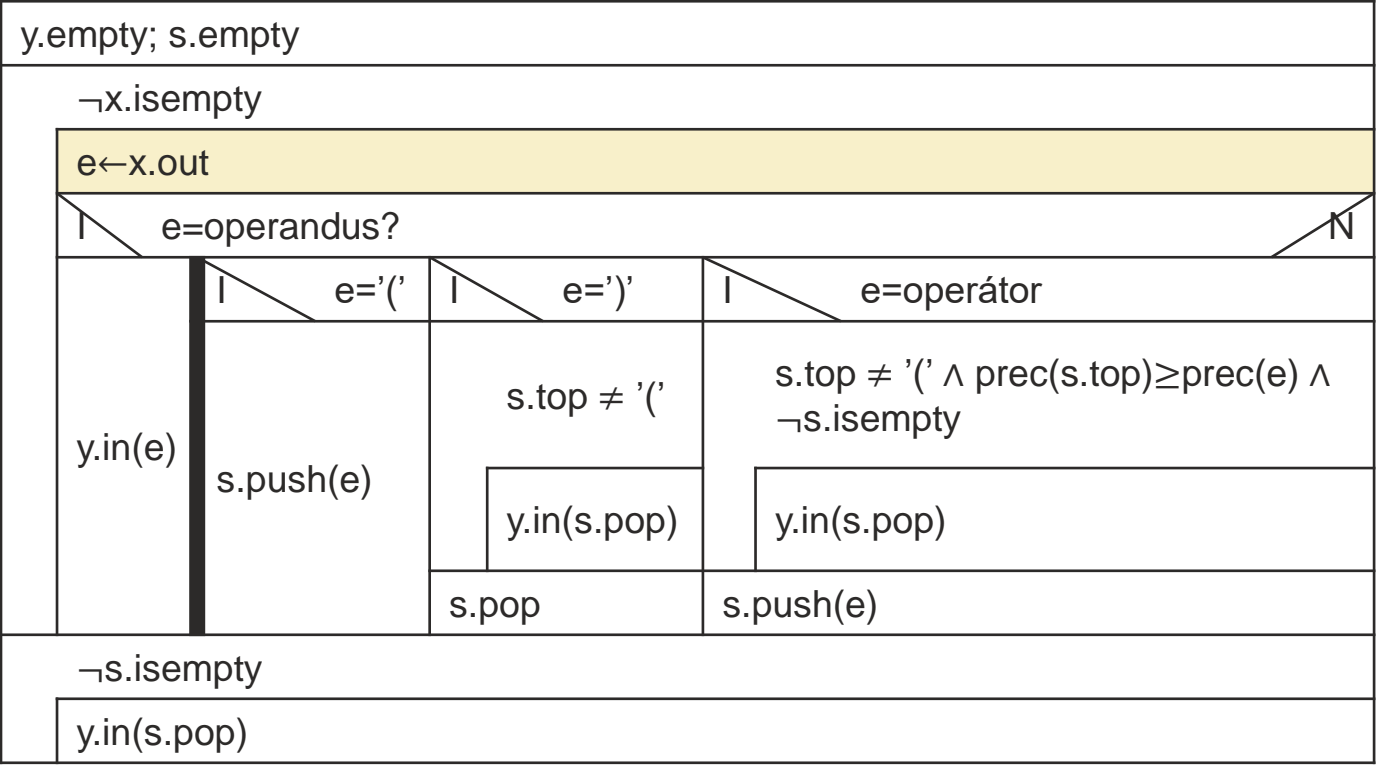
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

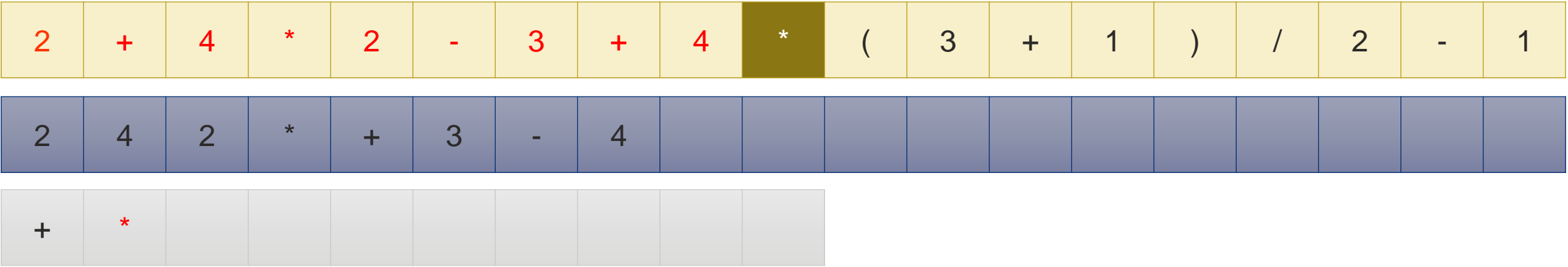
y.empty; s.empty				
¬x.isempty				
e←x.out				
I \ e=operandus? / N				
y.in(e)	I \ e='(' /	I \ e=')' /	I \ e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				



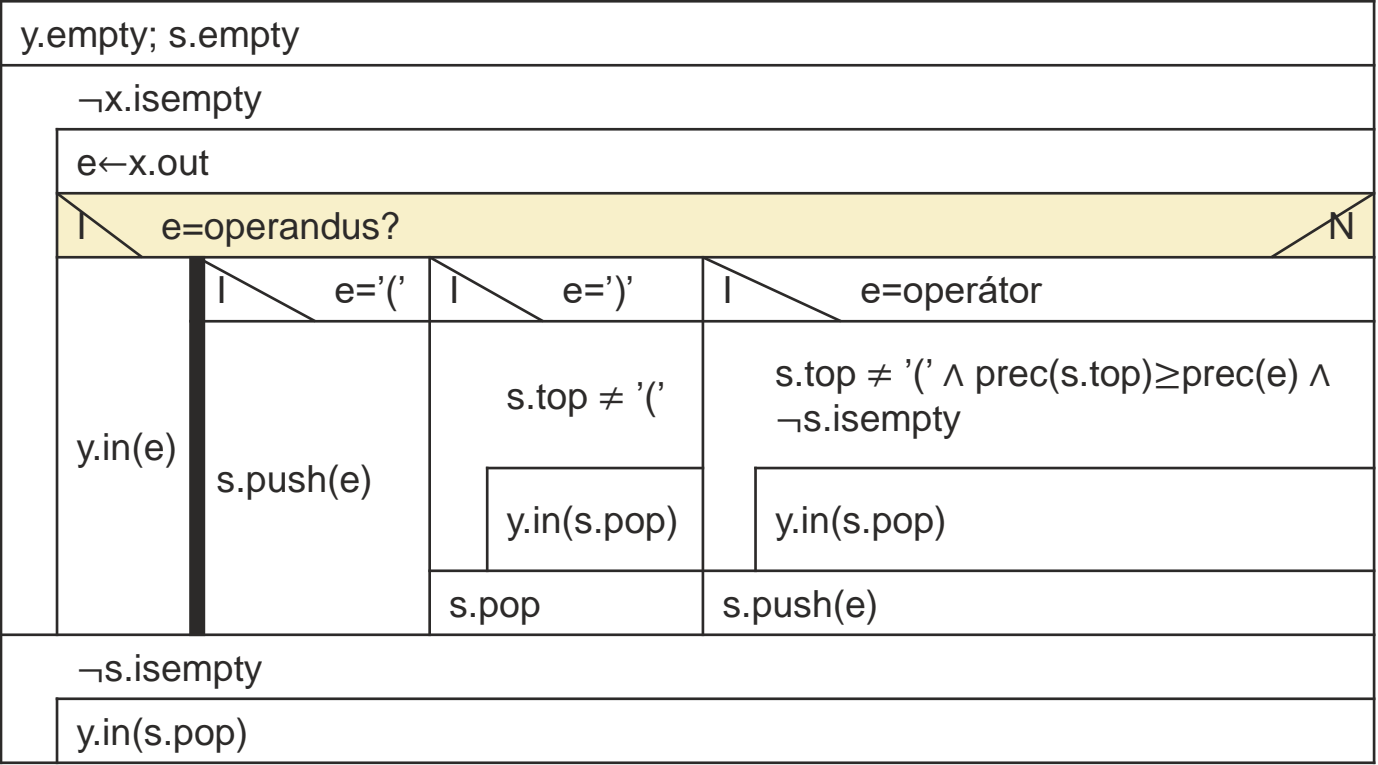
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

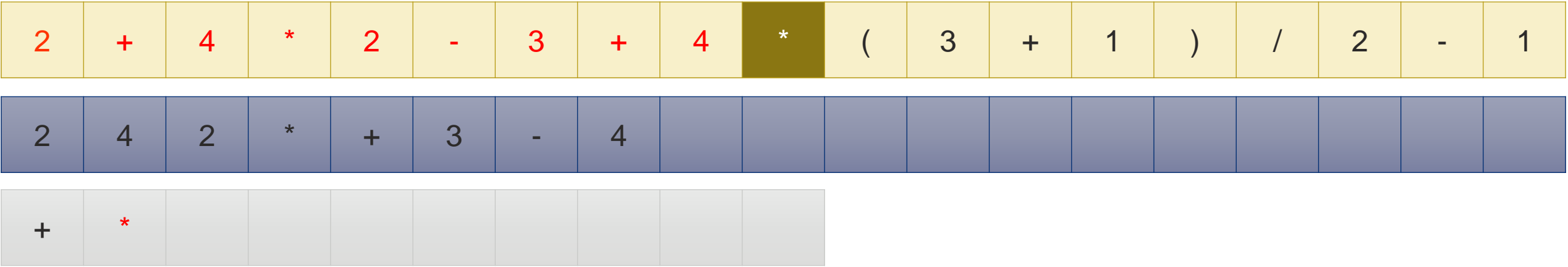
Az x sor következő elemét kivesszük a sorból.



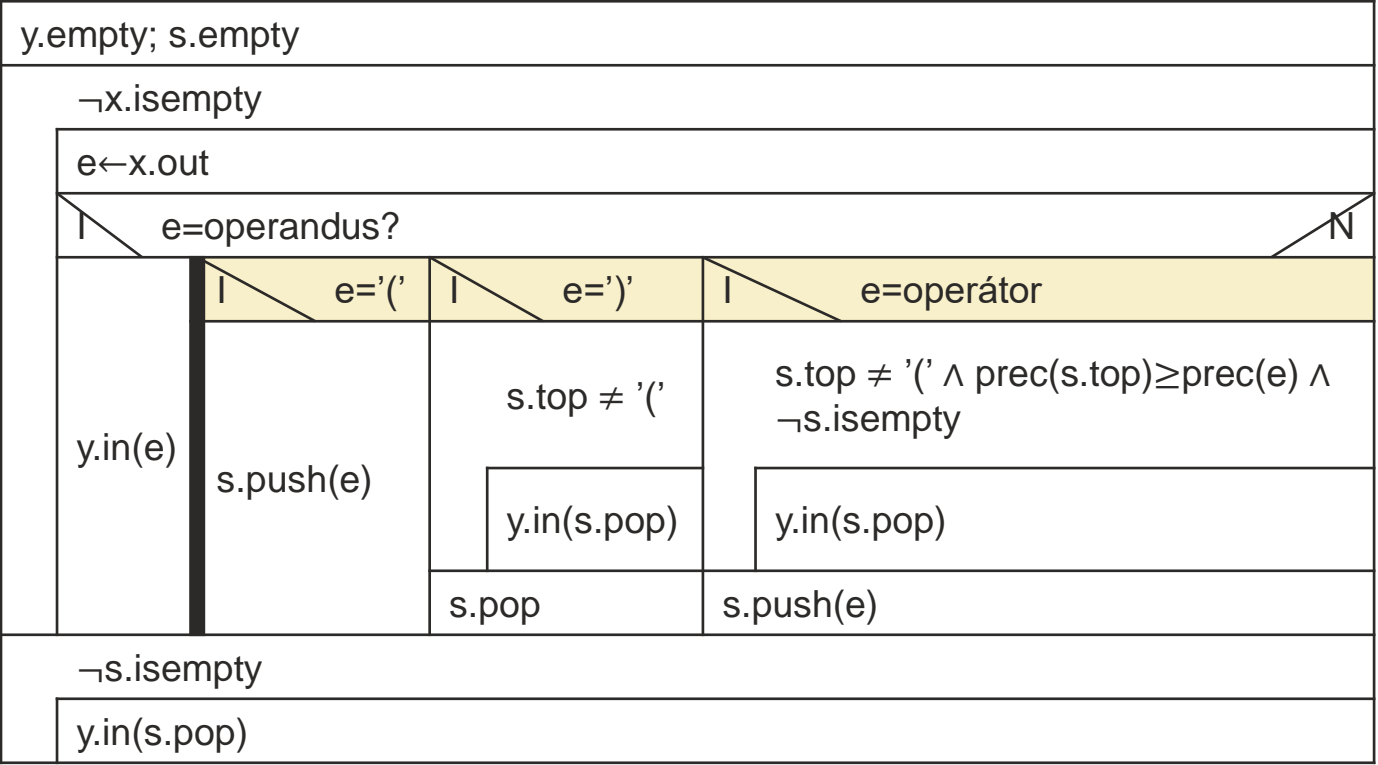


A kivett elem operandus?





A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*																	

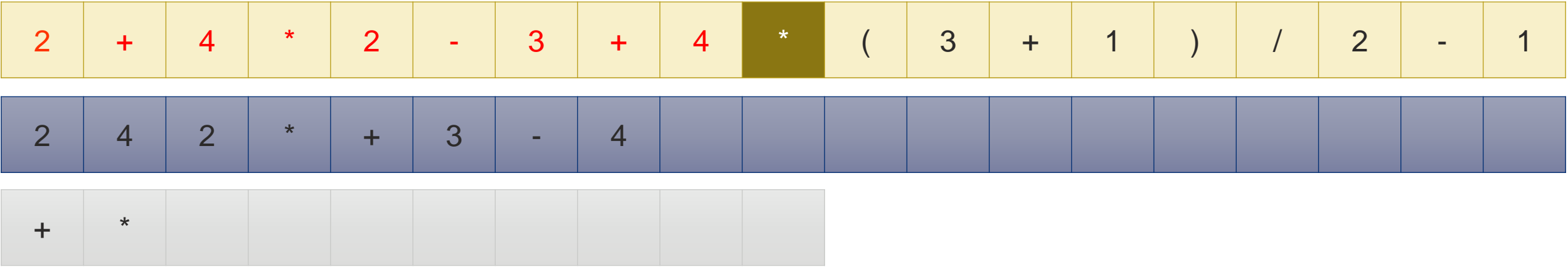
Nem → A kivett elem operátor, vagy valamely zárójel?

Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárójelet találtunk belépünk a ciklusba.

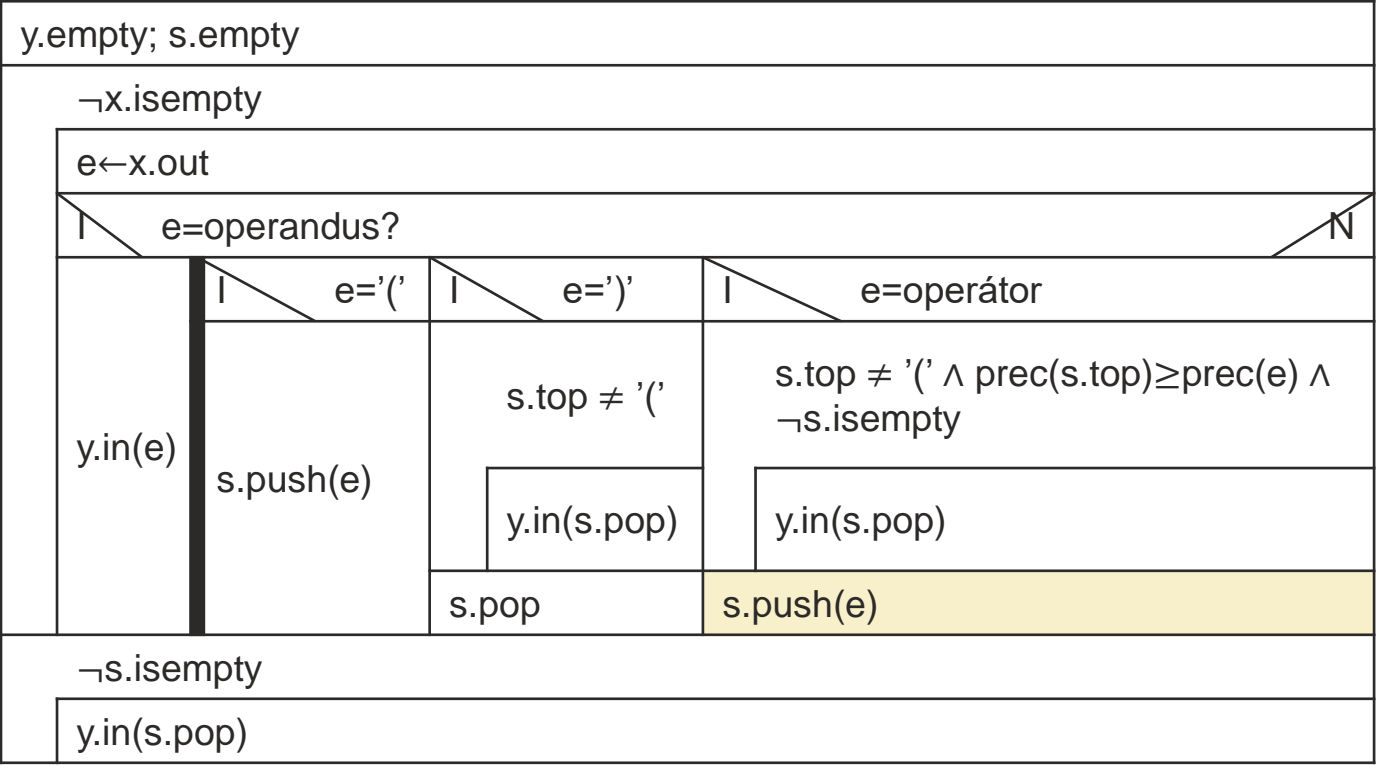
y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

A verem tetején levő operátor precedenciája alacsonyabb.





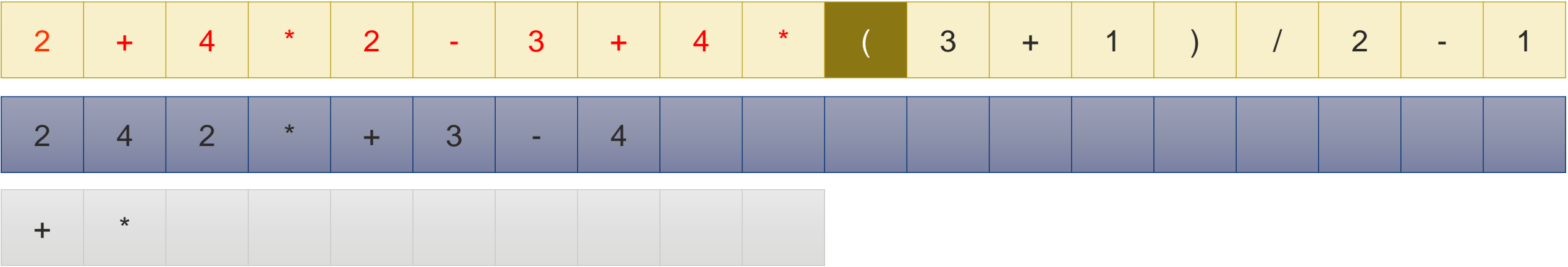
A verem tetején levő operátor precedenciája alacsonyabb.
Végül az aktuális operátort a verem tetejére tesszük.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*																	

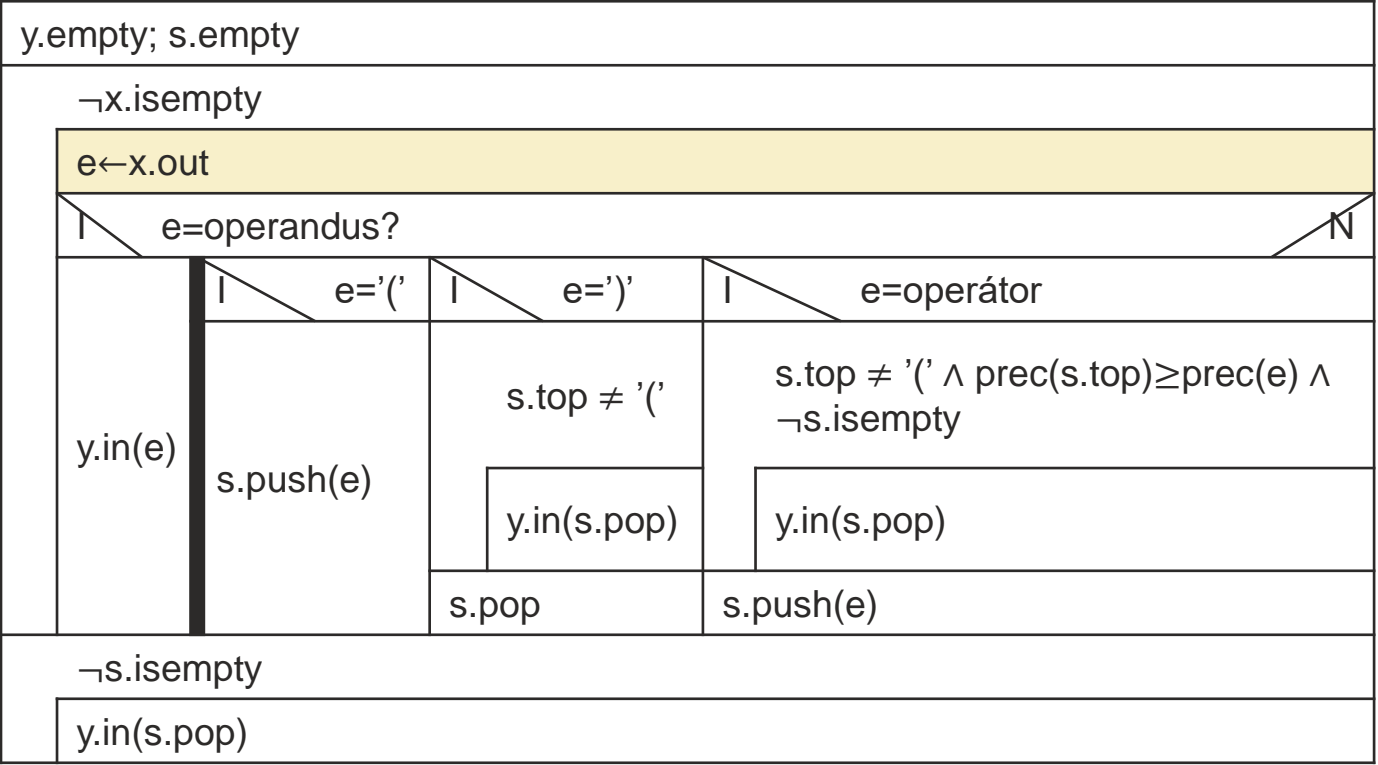
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

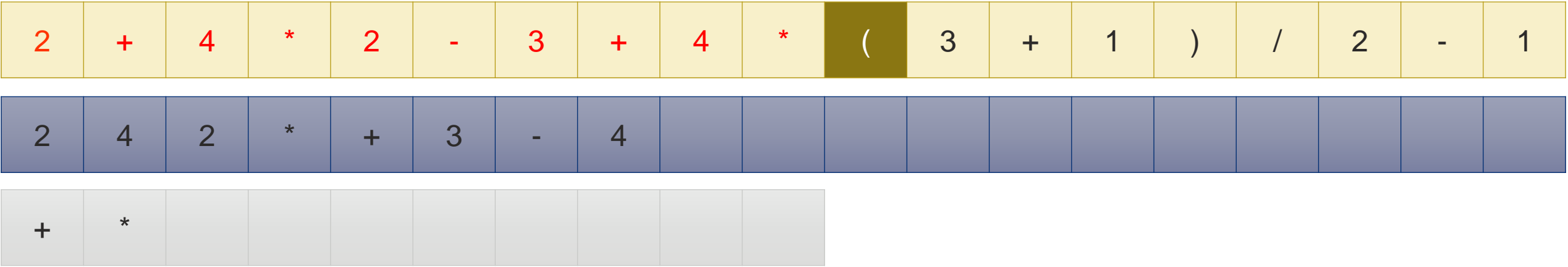
y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				



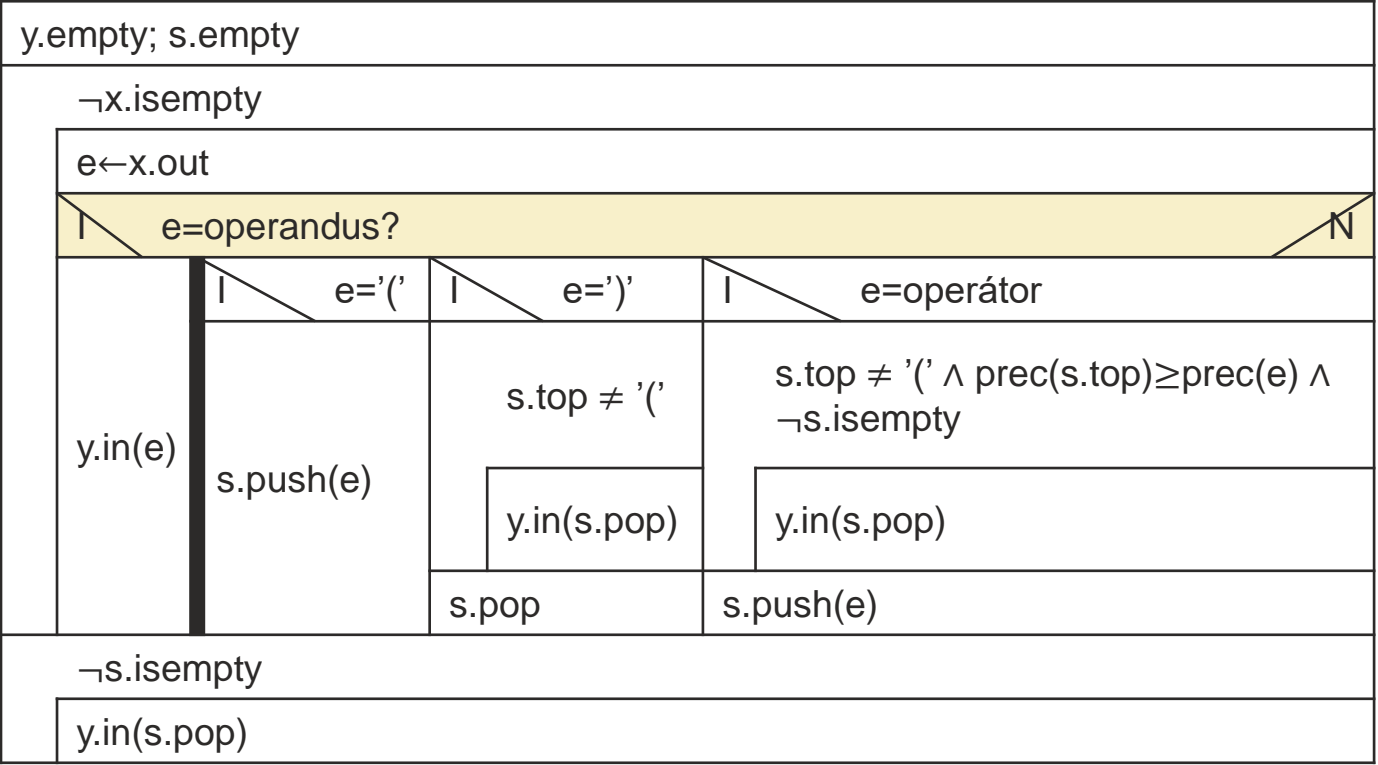
Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

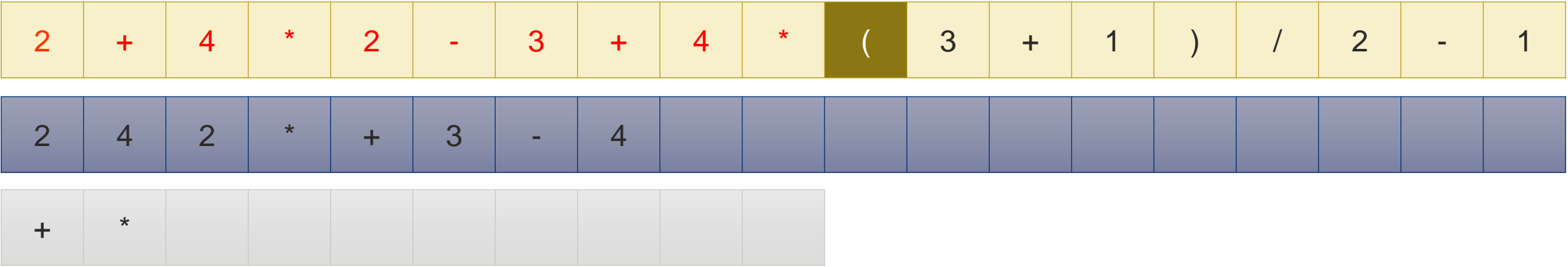
Az x sor következő elemét kivesszük a sorból.



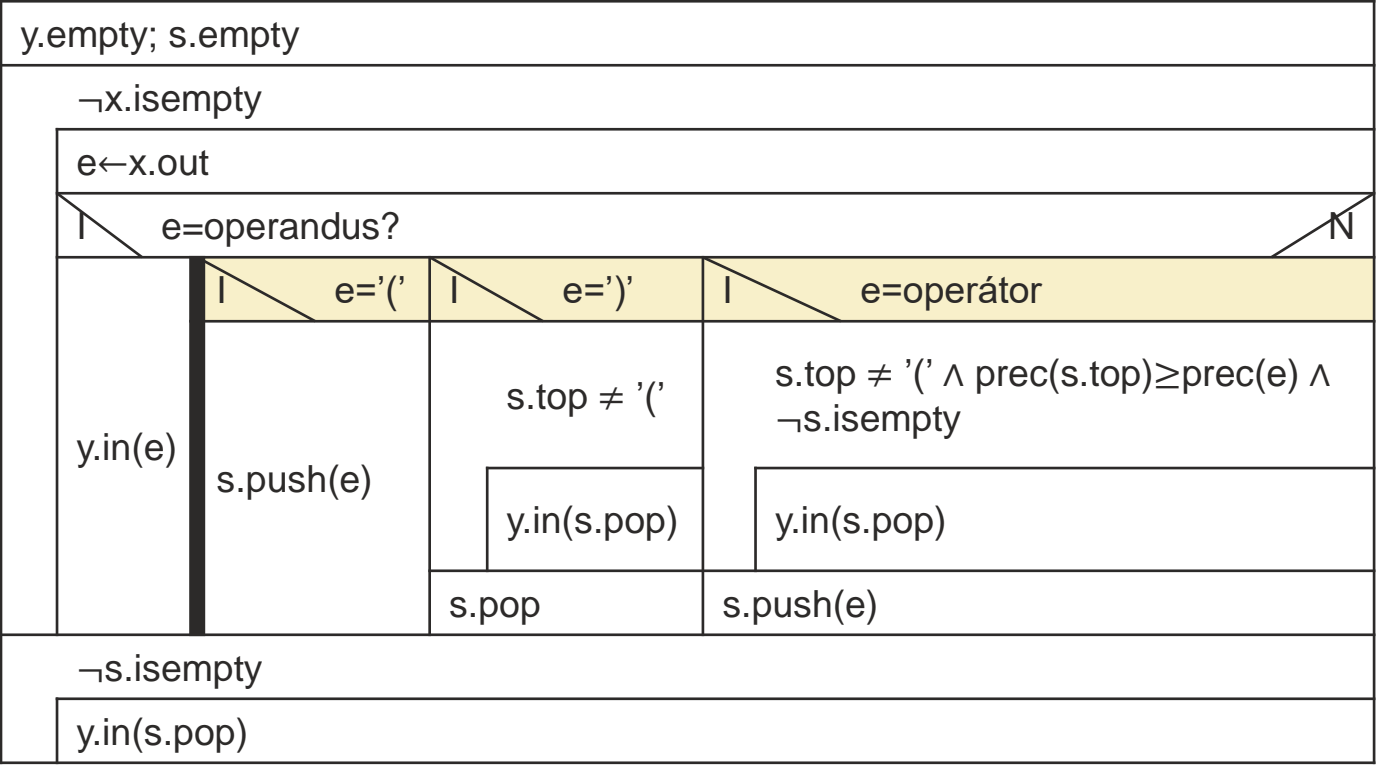


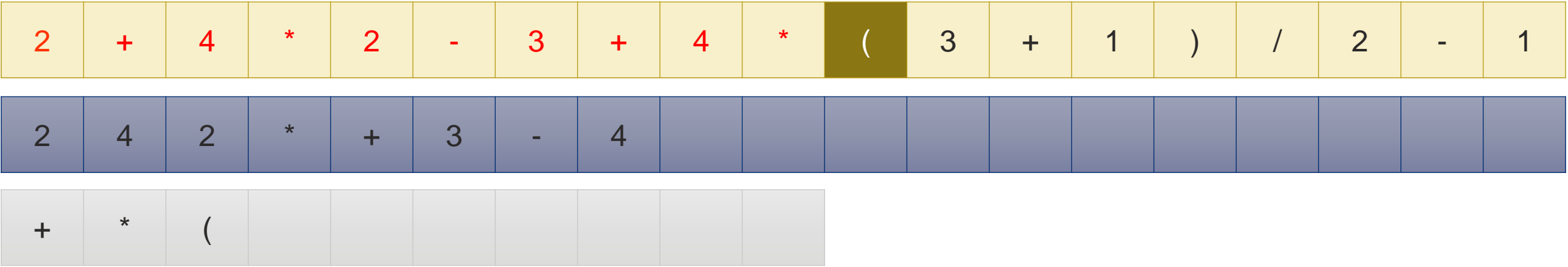
A kivett elem operandus?





A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?





A kivett elem operandus?
Nem → A kivett elem operátor, vagy
valamely zárójel?
Nyitózárójel → Betesszük a
verembe.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*	(

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*	(

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4											
+	*	(

A kivett elem operandus?

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

A kivett elem operandus?
 Igen → így azt az y sorba tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /		I / e=')' /	
	s.push(e)		I / e=operátor /	
			s.top ≠ '('	
			s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
	y.in(s.pop)		y.in(s.pop)	
s.pop		s.push(e)		
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? \ N				
y.in(e)	I / e='(' \	I / e=')' \	I / e=operátor \	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div>I</div> <div>e=operandus?</div> <div>N</div> </div>			
y.in(e)	<div> <div>I</div> <div>e='('</div> </div>		<div> <div>I</div> <div>e=')'</div> </div>
	<div> <div>I</div> <div>e=operátor</div> </div>		
	<div> <div>I</div> <div>s.top ≠ '('</div> </div>		<div> <div>I</div> <div>s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty</div> </div>
	<div> <div>I</div> <div>s.push(e)</div> </div>		<div> <div>I</div> <div>y.in(s.pop)</div> </div>
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

A kivett elem operandus?
 Nem → A kivett elem operátor, vagy
 valamely zárójel?

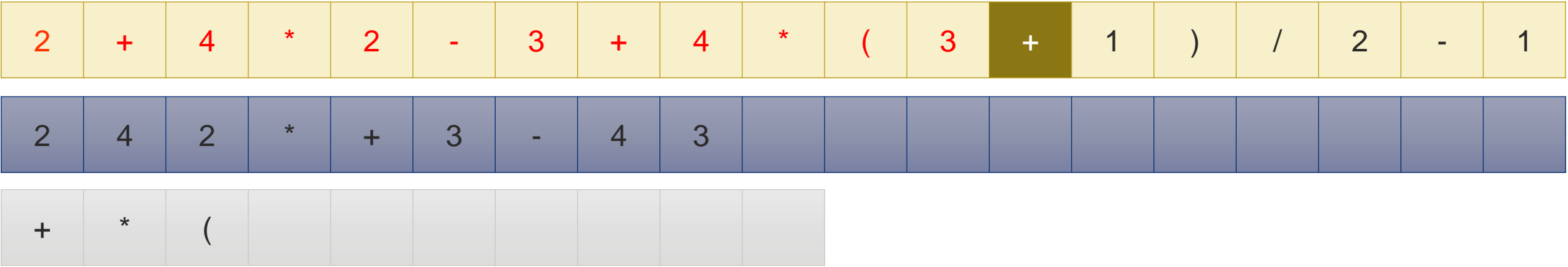
y.empty; s.empty			
¬x.isempty			
e←x.out			
┌────────── e=operandus? ───────────┐			
y.in(e)	┌────────── e='(' ───────────┐		┌────────── e=')' ───────────┐
	┌────────── e=operátor ───────────┐		
	s.top ≠ '('		s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
	y.in(s.pop)		y.in(s.pop)
s.push(e)		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(

Nem → A kivett elem operátor, vagy valamely zárójel?

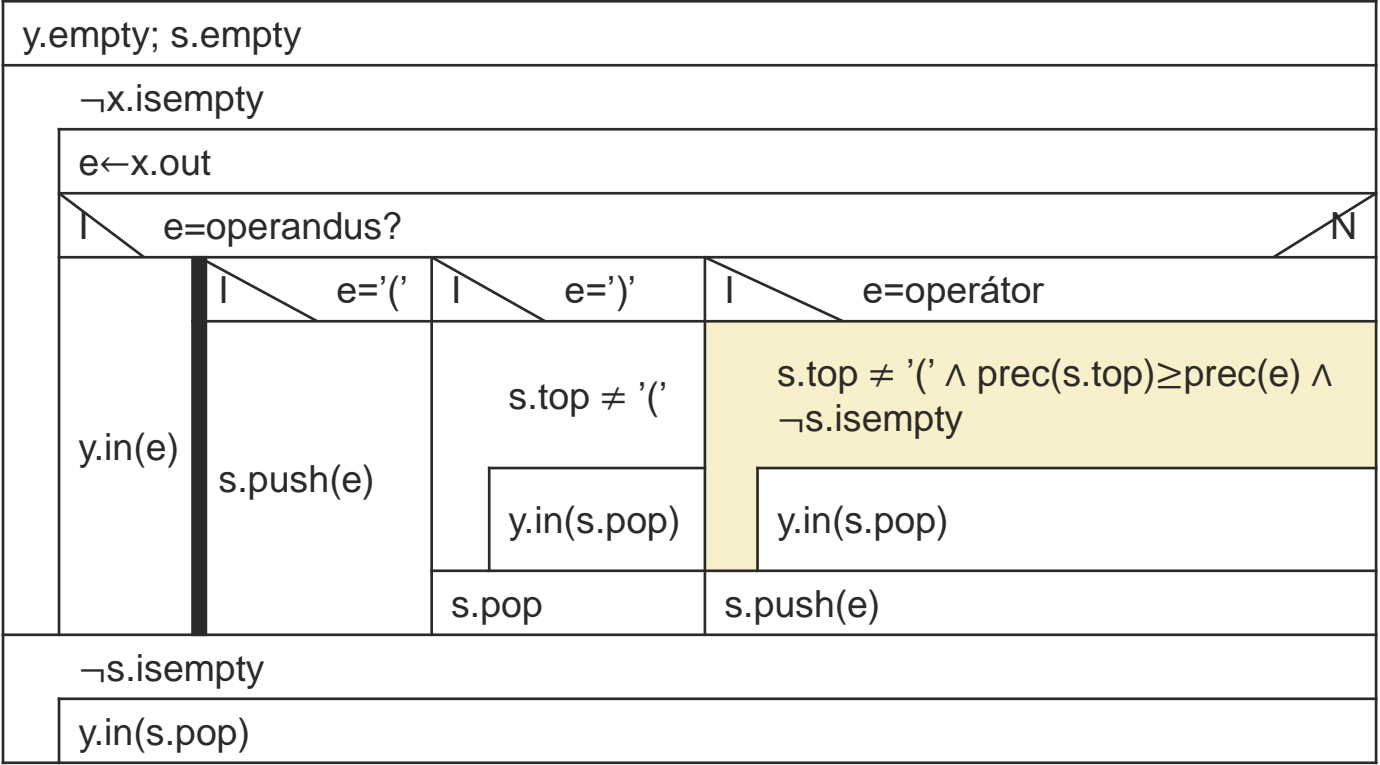
Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárójelet találtunk belépünk a ciklusba.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			



Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

Nyitózárojelet találunk így nem lépünk a ciklusba.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(+															

Nyitózárójelet találunk így nem lépünk a ciklusba.

Végül az aktuális operátort a verem tetejére tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3										
+	*	(+															

A kivett elem operandus?

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

A kivett elem operandus?
 Igen → így azt az y sorba tesszük.

y.empty; s.empty						
¬x.isempty						
e←x.out						
I / e=operandus? / N						
y.in(e)	I / e='(' /		I / e=')' /		I / e=operátor /	
	s.push(e)		s.top ≠ '('		s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
			y.in(s.pop)		y.in(s.pop)	
			s.pop		s.push(e)	
	¬s.isempty					
y.in(s.pop)						

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I \ e=operandus? / N				
y.in(e)	I \ e='(' /	I \ e=')' /	I \ e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty						
¬x.isempty						
e←x.out						
I / e=operandus? / N						
y.in(e)	I / e='(' /		I / e=')' /		I / e=operátor /	
	s.push(e)		s.top ≠ '('		s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
			y.in(s.pop)		y.in(s.pop)	
			s.pop		s.push(e)	
¬s.isempty						
y.in(s.pop)						

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
y.in(e)	e=operandus?		
	e='('	e=')'	e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

A kivett elem operandus?
 Nem → A kivett elem operátor, vagy
 valamely zárójel?

y.empty; s.empty				
¬x.isempty				
e←x.out				
┌────────── e=operandus? ───────────┐				
y.in(e)	┌────────── e='(' ───────────┐		┌────────── e=')' ───────────┐	
	┌────────── e=operátor ───────────┐			
	s.top ≠ '('		s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
	y.in(s.pop)		y.in(s.pop)	
s.pop		s.push(e)		
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1									
+	*	(+															

A kivett elem operandus?

Nem → A kivett elem operátor, vagy
valamely zárójel?

Csukózárójel → Ciklus, amíg a
verem tetején nem nyitózárójelet
találunk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

- Csukózárójel → Ciklus, amíg a verem tetején nem nyitózárójelet találunk.
- Nem nyitózárójel ezért belépünk a ciklusba.
- A verem tetején levő operátort az y sorba írjuk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

A verem mérete csökkent.

A következő szimbólum nyitózárójel ezért nem lépünk be a ciklusba.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

- A verem mérete csökkent.
- A következő szimbólum nyitózárójel ezért nem lépünk be a ciklusba.
- A veremből kivesszük a nyitózárójelet és eldobjuk.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div>I</div> <div>e=operandus?</div> <div>N</div> </div>			
y.in(e)	<div> <div>I</div> <div>e='('</div> </div>		<div> <div>I</div> <div>e=')'</div> </div>
	s.push(e)		<div> <div>s.top ≠ '('</div> <div> <div>y.in(s.pop)</div> <div>s.pop</div> </div> </div>
			<div> <div>e=operátor</div> <div> <div>s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty</div> <div> <div>y.in(s.pop)</div> <div>s.push(e)</div> </div> </div> </div>
	<div> <div>¬s.isempty</div> <div>y.in(s.pop)</div> </div>		

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
	¬s.isempty		
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

A kivett elem operandus?
 Nem → A kivett elem operátor, vagy
 valamely zárójel?

y.empty; s.empty			
¬x.isempty			
e←x.out			
┌────────── e=operandus? ───────────┐			
y.in(e)	┌────────── e='(' ───────────┐		┌────────── e=')' ───────────┐
	┌────────── e=operátor ───────────┐		
	s.top ≠ '('		s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
	y.in(s.pop)		y.in(s.pop)
s.push(e)		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

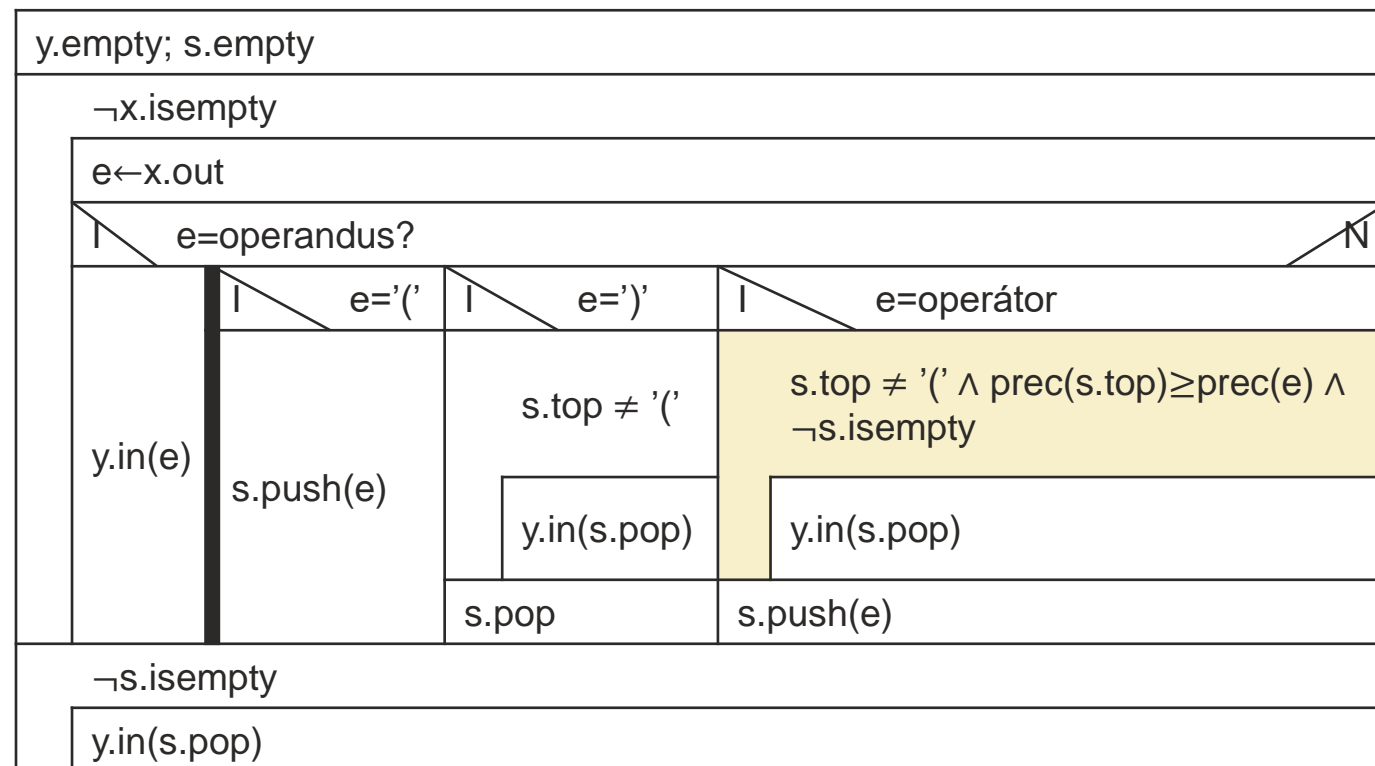
2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+								
+	*	(+															

Nem → A kivett elem operátor, vagy valamely zárójel?

Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárójelet találtunk belépünk a ciklusba.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

A verem tetején levő művelet (*) precedenciája nem kisebb az aktuálisnál (/), ezért a ciklusba belépünk.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*							
+	*	(+															

A verem tetején levő művelet (*) precedenciája nem kisebb az aktuálisnál (/), ezért a ciklusba belépünk.

A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div></div> <div>e=operandus?</div> <div>N</div> </div>			
y.in(e)	<div> <div></div> <div>e='('</div> </div>		<div> <div></div> <div>e=operátor</div> </div>
	s.push(e)		<div> <div>s.top ≠ '('</div> <div> <div>y.in(s.pop)</div> <div>y.in(s.pop)</div> </div> </div>
	s.pop		s.push(e)
	<div> <div>¬s.isempty</div> <div>y.in(s.pop)</div> </div>		

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*							
+	*	(+															

A verem mérete csökkent.

A verem tetején levő következő operátor precedenciája alacsonyabb.

y.empty; s.empty			
¬x.isempty			
e←x.out			
<div> <div>e=operandus?</div> <div> <div> <div> <div>e='('</div> <div> <div>y.in(e)</div> <div>s.push(e)</div> </div> </div> <div> <div>e=')'</div> <div> <div>s.top ≠ '('</div> <div> <div>y.in(s.pop)</div> <div>s.pop</div> </div> </div> <div> <div>e=operátor</div> <div> <div> <div>s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty</div> <div> <div>y.in(s.pop)</div> <div>s.push(e)</div> </div> </div> </div> </div> </div> </div></div></div>			
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*							
+	/	(+															

A verem mérete csökkent.

A verem tetején levő következő operátor precedenciája alacsonyabb.

Végül az aktuális operátort a verem tetejére tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*							
+	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*							
+	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('		I / e=')'	
	s.push(e)		I / e=operátor	
			s.top ≠ '('	
	y.in(s.pop)		s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
	s.pop		y.in(s.pop)	
s.push(e)				
¬s.isempty				
y.in(s.pop)				

2+4*2-3+4*(3+1)/2-1

242*+3-431+*

+ / (+

A kivett elem operandus?

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

A kivett elem operandus?
 Igen → így azt az y sorba tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

A kivett elem operandus?

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

A kivett elem operandus?
 Nem → A kivett elem operátor, vagy
 valamely zárójel?

y.empty; s.empty				
¬x.isempty				
e←x.out				
┌────────── e=operandus? ───────────┐				
y.in(e)	┌────────── e='(' ───────────┐		┌────────── e=')' ───────────┐	
	s.push(e)		s.top ≠ '('	
			y.in(s.pop)	
			s.pop	
┌────────── e=operátor ───────────┐				
s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty				
y.in(s.pop)				
s.push(e)				
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2						
+	/	(+															

Nem → A kivett elem operátor, vagy
valamely zárójel?

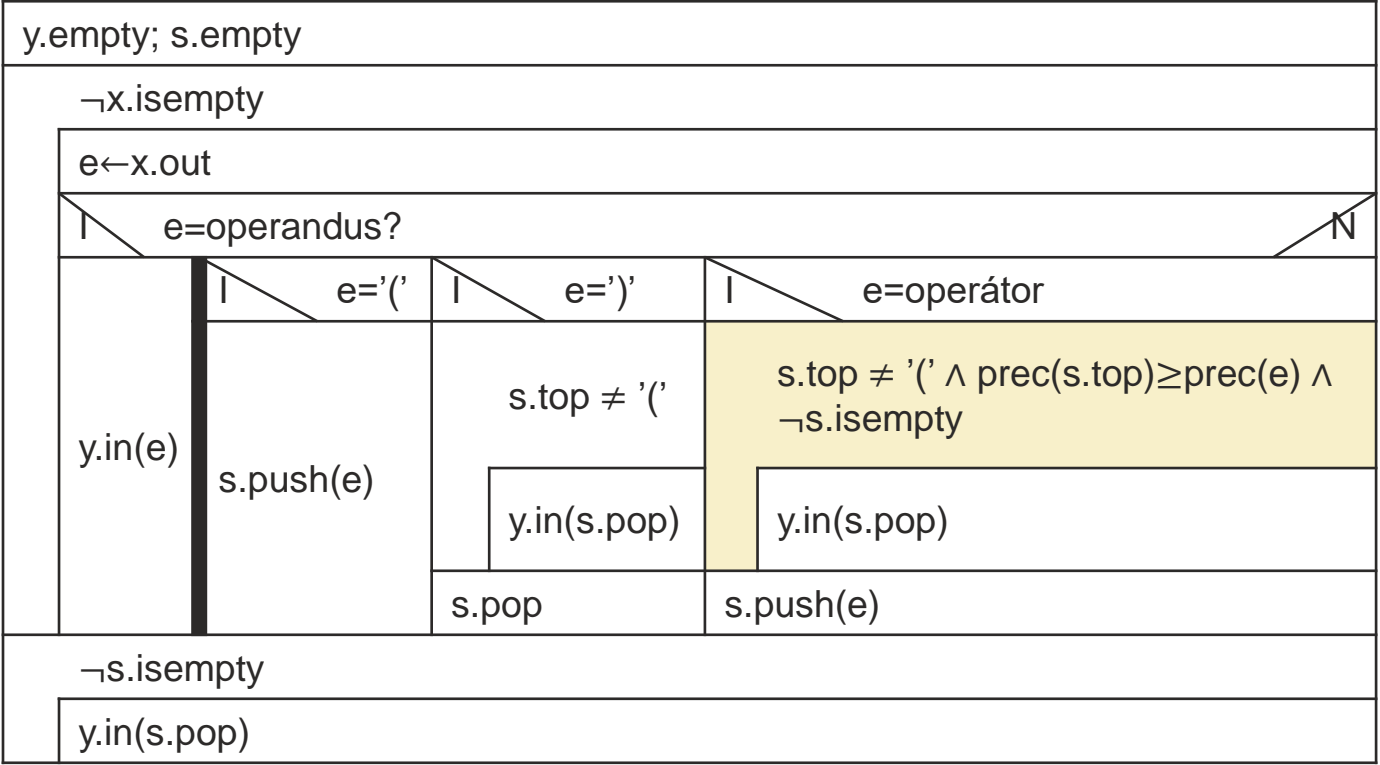
Operátor → Amíg a verem nem üres
és az aktuális operátor
precedenciája nem kisebb mint a
verem tetején levő és nem
nyitózárójelet találtunk belépünk a
ciklusba.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
	s.pop		s.push(e)
¬s.isempty			
y.in(s.pop)			



Operátor → Amíg a verem nem üres és az aktuális operátor precedenciája nem kisebb mint a verem tetején levő és nem nyitózárojelet találtunk belépünk a ciklusba.

A verem tetején levő művelet (/) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/					
+	/	(+															

A verem tetején levő művelet (/) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.

A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2	4	2	*	+	3	-	4	3	1	+	*	2	/					
---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--

+	/	(+															
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

A verem mérete csökkent.

A verem tetején levő művelet (+) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I e=operandus? N				
y.in(e)	I e='('	I e=')'	I e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+				
+	/	(+															

A verem mérete csökkent.

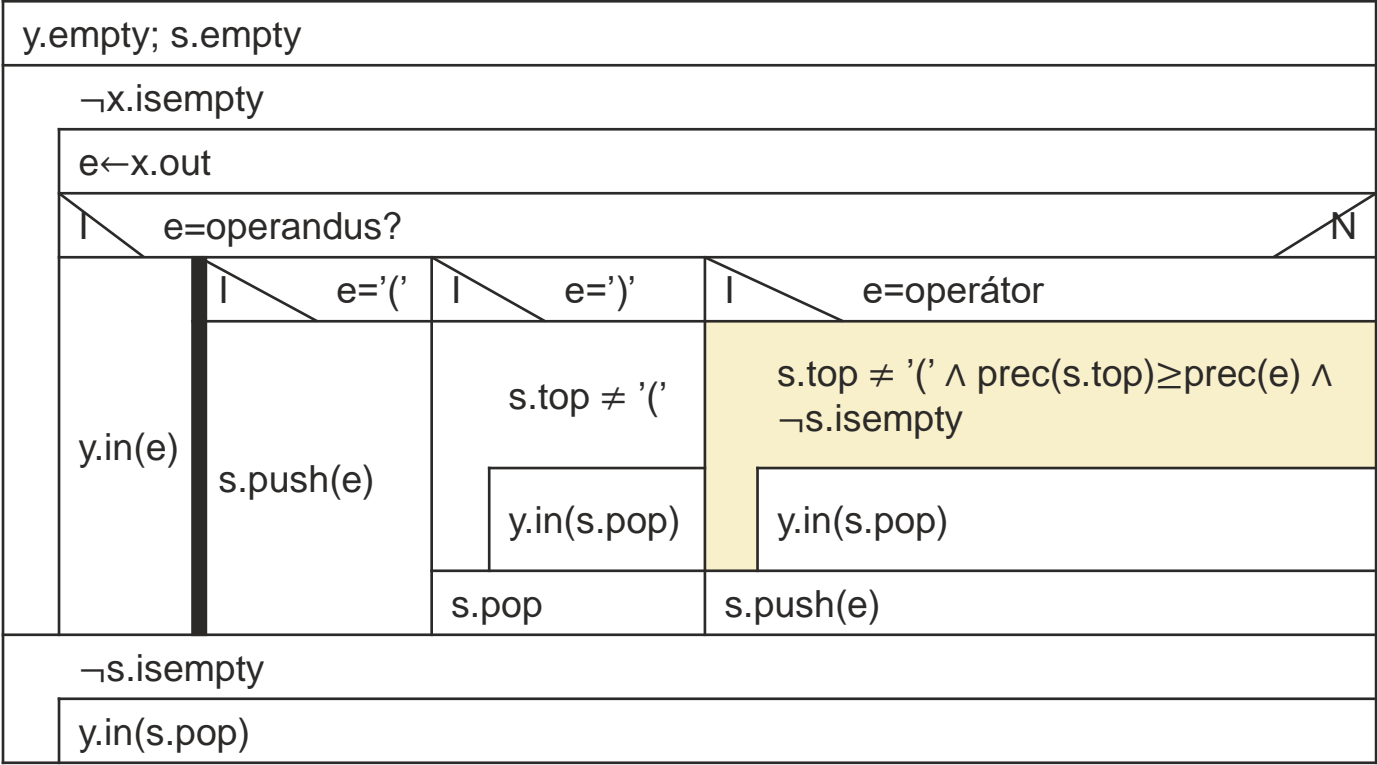
A verem tetején levő művelet (+) precedenciája nem kisebb az aktuálisnál (-), ezért a ciklusba belépünk.

A verem tetején levő operátort kivesszük a veremből és az y sorba írjuk.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

A verem mérete újra csökkent.

A verem kiürült, így a ciklusfeltétel nem teljesül.



2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+				
-	/	(+															

A verem mérete újra csökkent.

A verem kiürült, így a ciklusfeltétel nem teljesül.

Végül az aktuális operátort a verem tetejére tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+				
-	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+				
-	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Az x sor következő elemét kivesszük a sorból.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
	¬s.isempty			
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+				
-	/	(+															

A kivett elem operandus?

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='('	I / e=')'	I / e=operátor
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1			
-	/	(+															

A kivett elem operandus?
 Igen → így azt az y sorba tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /		I / e=')' / I / e=operátor /	
	s.push(e)		s.top ≠ '('	
			s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
	y.in(s.pop)		y.in(s.pop)	
	s.pop		s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1			
-	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1			
-	/	(+															

Ha az x nem üres, akkor a ciklusmag végrehajtásra kerül.

Mivel már kiürült, így a következő utasításra lép a program.

y.empty; s.empty			
¬x.isempty			
e←x.out			
I / e=operandus? / N			
y.in(e)	I / e='(' /	I / e=')' /	I / e=operátor /
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty
		y.in(s.pop)	y.in(s.pop)
		s.pop	s.push(e)
¬s.isempty			
y.in(s.pop)			

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1			
-	/	(+															

Végezetül a veremben található operátorok mindegyikét az y sorba tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? / N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top)≥prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	+	4	*	2	-	3	+	4	*	(3	+	1)	/	2	-	1
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
-	/	(+															

Végezetül a veremben található operátorok mindegyikét az y sorba tesszük.

y.empty; s.empty				
¬x.isempty				
e←x.out				
I / e=operandus? N				
y.in(e)	I / e='('	I / e=')'	I / e=operátor	
	s.push(e)	s.top ≠ '('	s.top ≠ '(' ∧ prec(s.top) ≥ prec(e) ∧ ¬s.isempty	
		y.in(s.pop)	y.in(s.pop)	
		s.pop	s.push(e)	
¬s.isempty				
y.in(s.pop)				

2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

A kész fordított lengyel forma!

2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

Input:
y – sor

2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

--	--	--	--	--	--	--	--	--	--

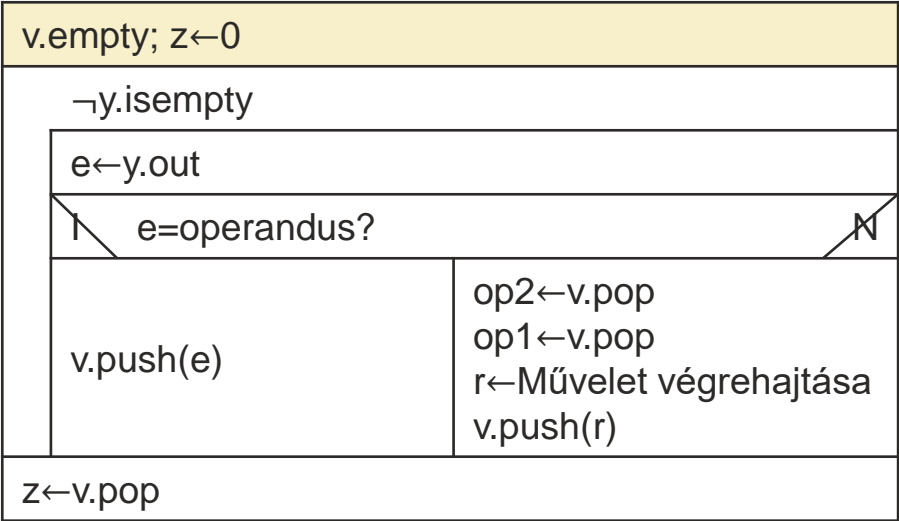
Input:
y – sor

Feldolgozás során használt tárolók:
v – verem (számítás)



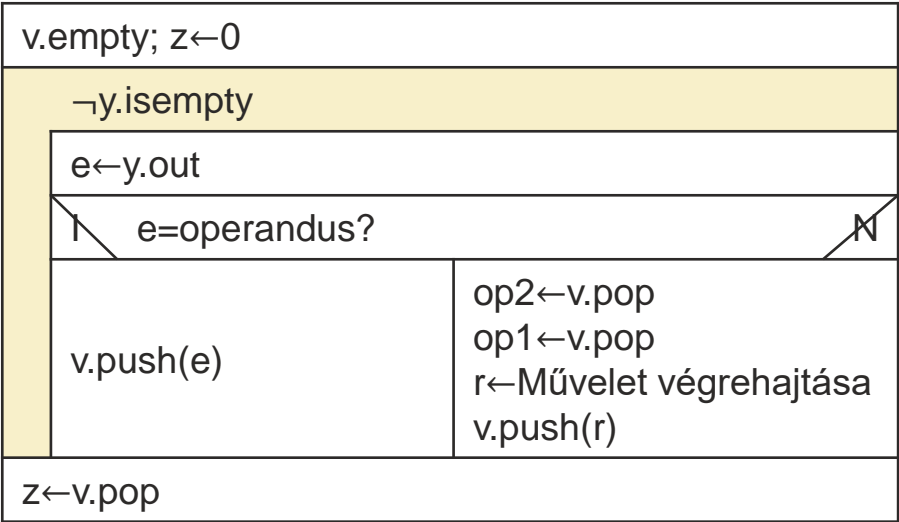
Kiértékelésnél az előző y sort használjuk.

A v vermet létrehozzuk és a z változót, amit az eredményt tartalmazza a végén nullázzuk.



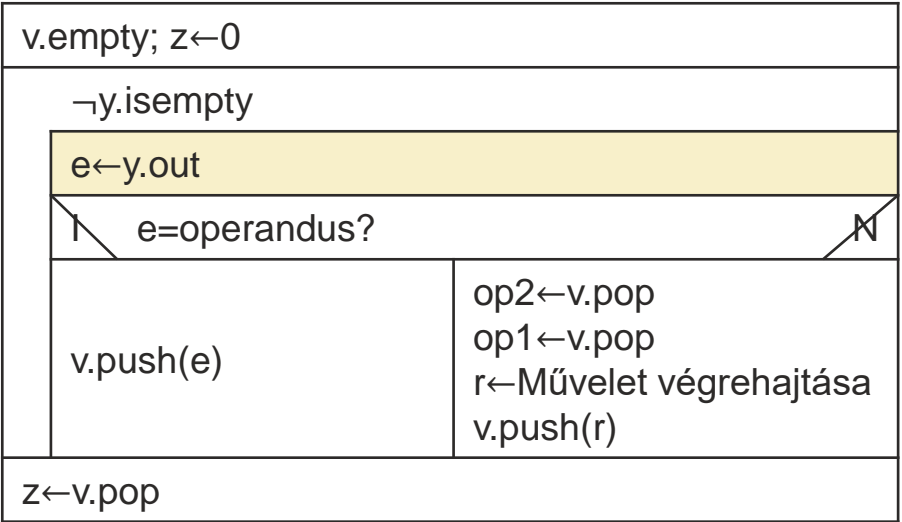


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



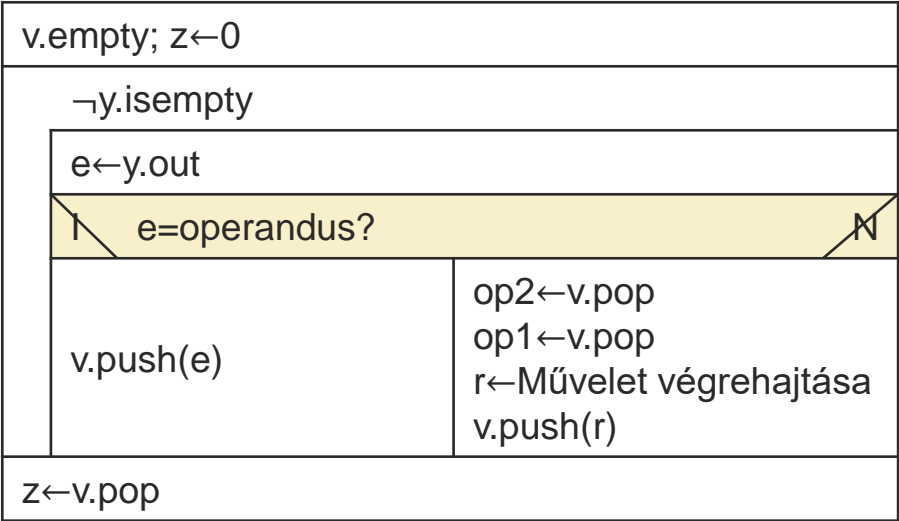


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



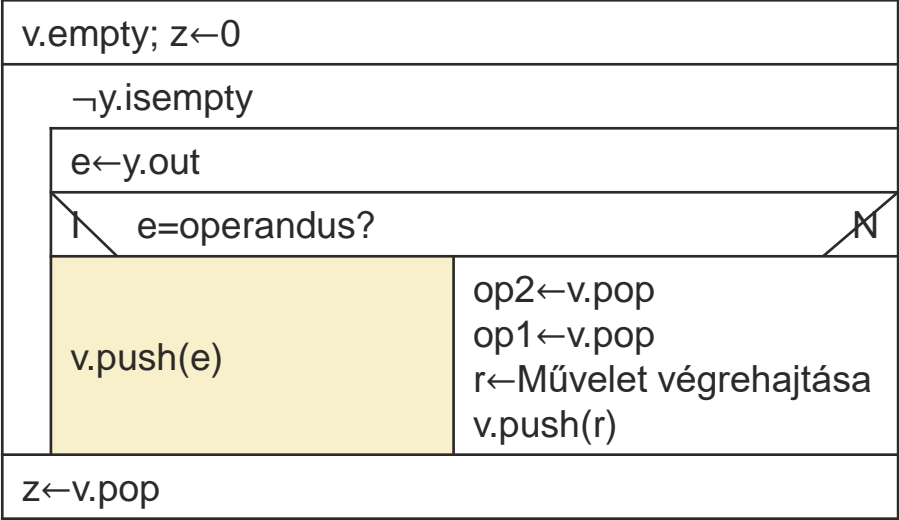


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



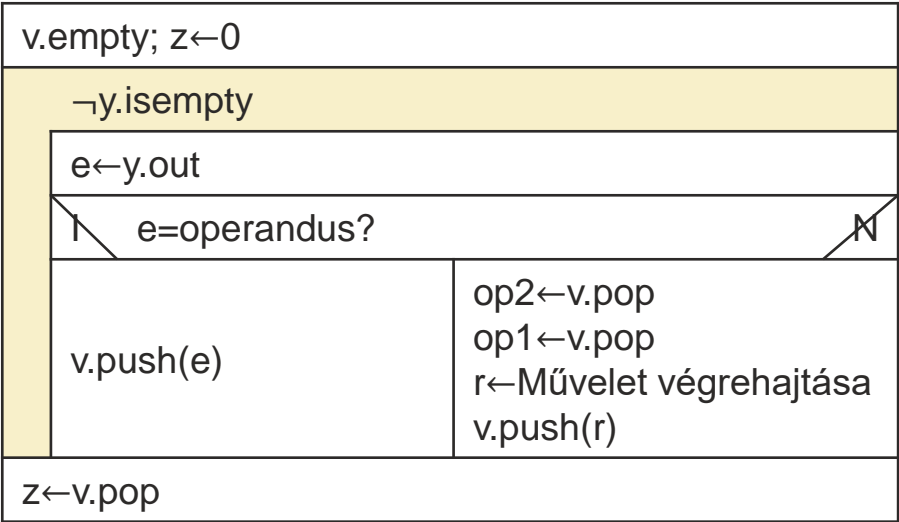


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?
Operandus → Betesszük a verembe.



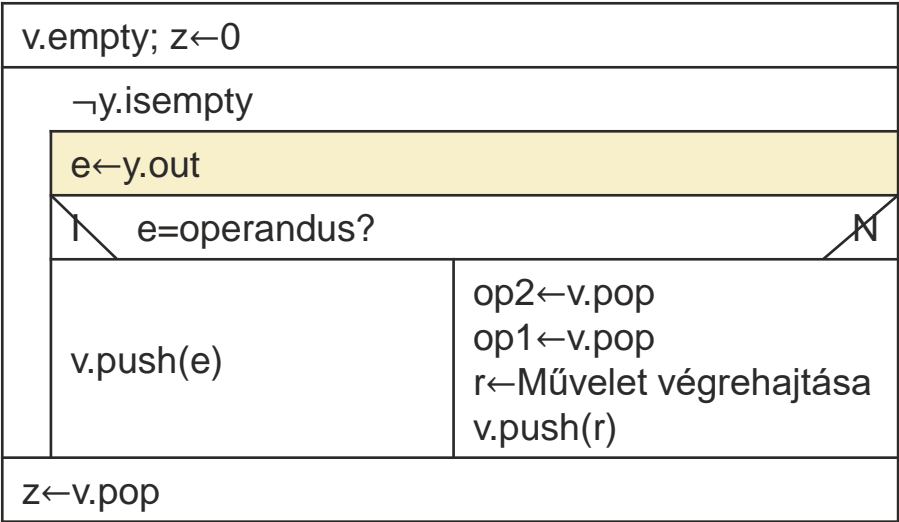


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



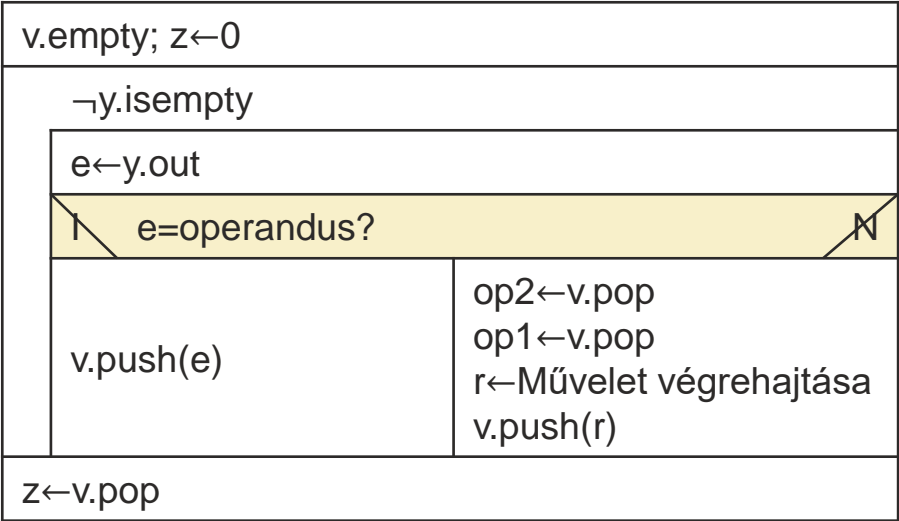


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



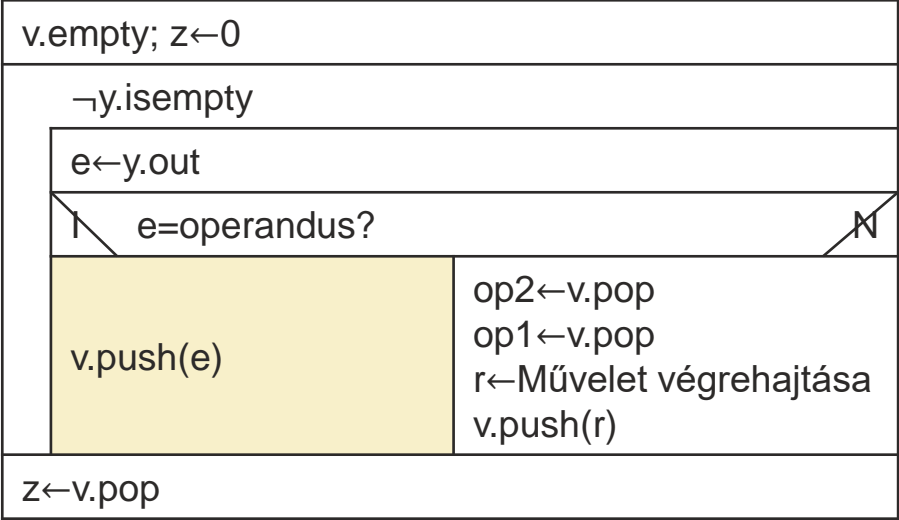


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



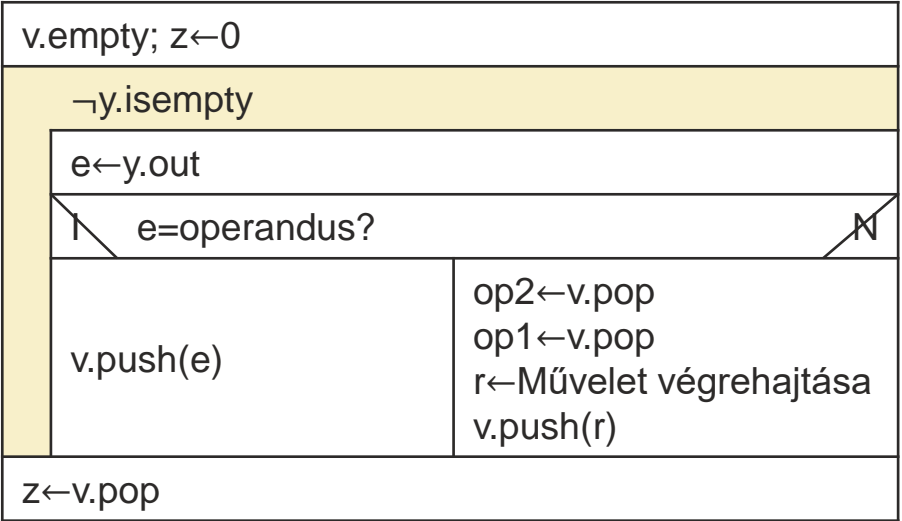


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?
Operandus → Betesszük a verembe.



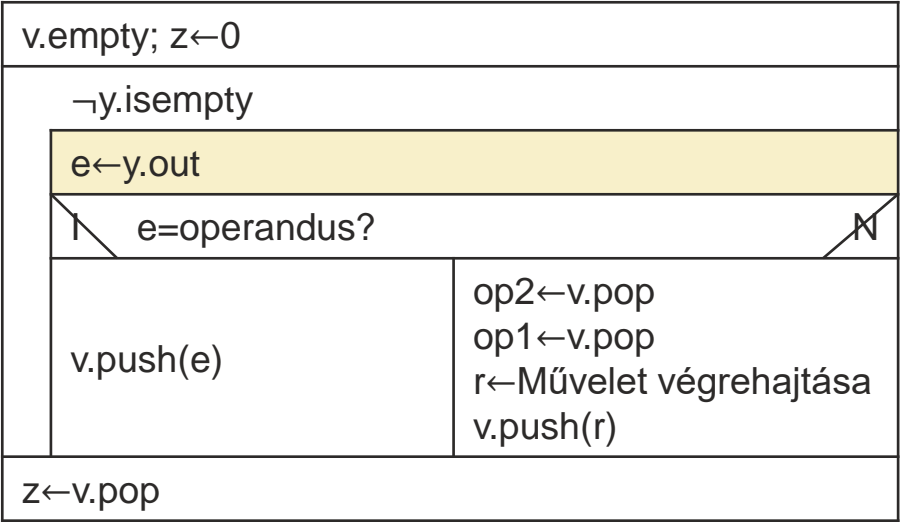


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



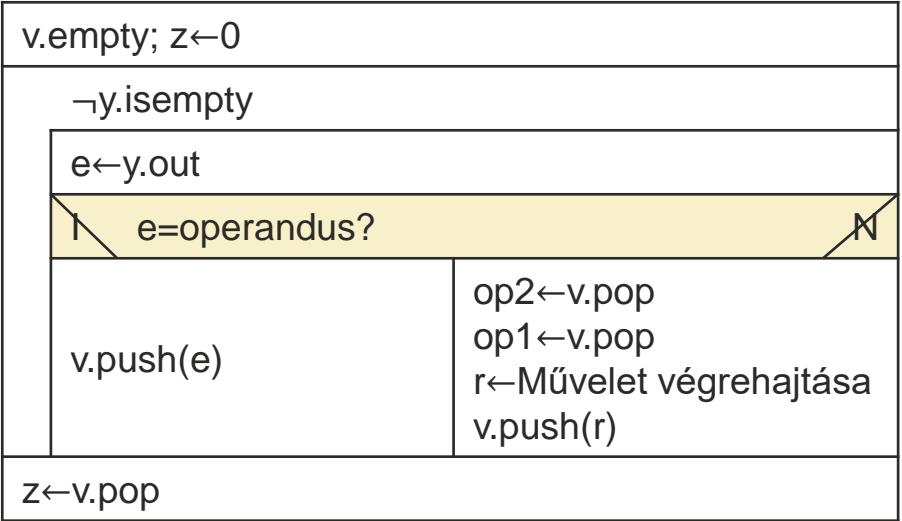


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



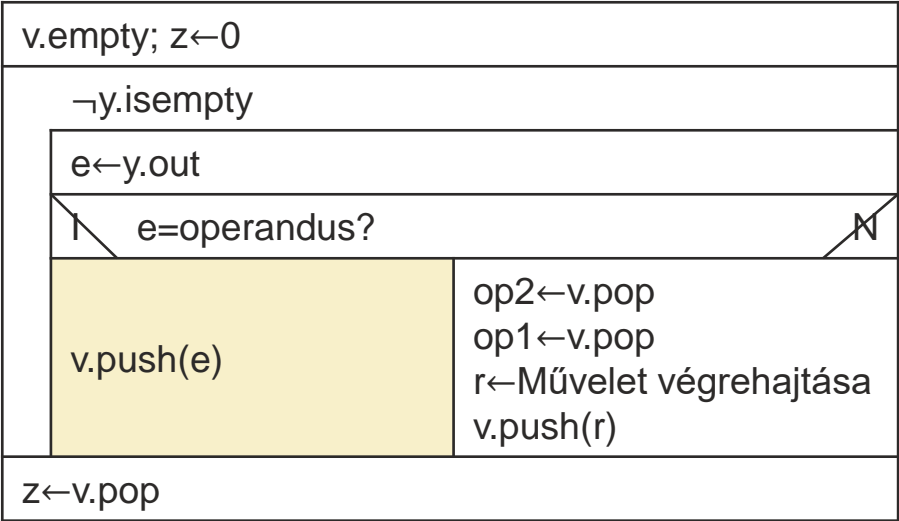


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



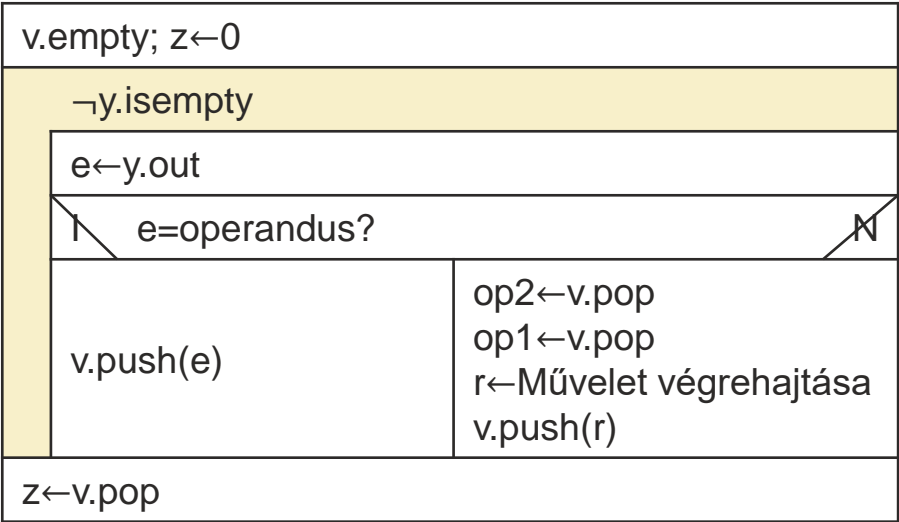


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?
 Operandus → Betesszük a verembe.



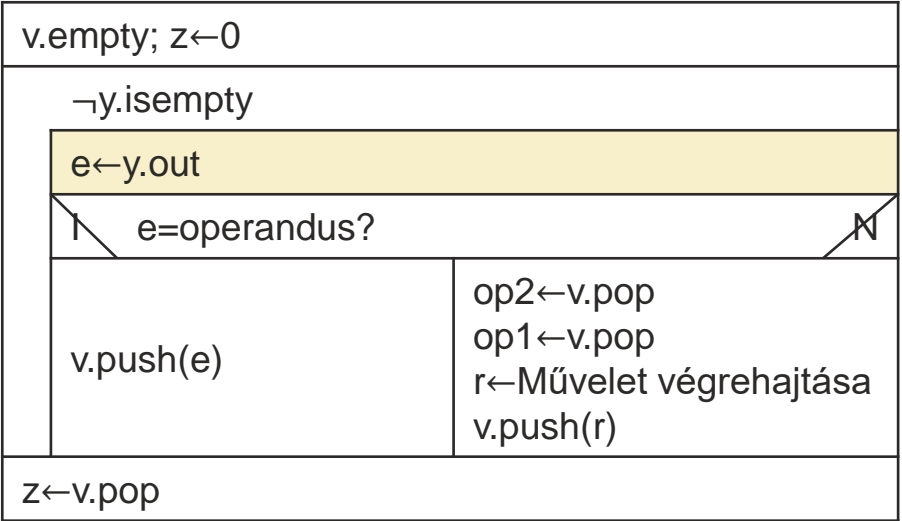
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
2	4	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



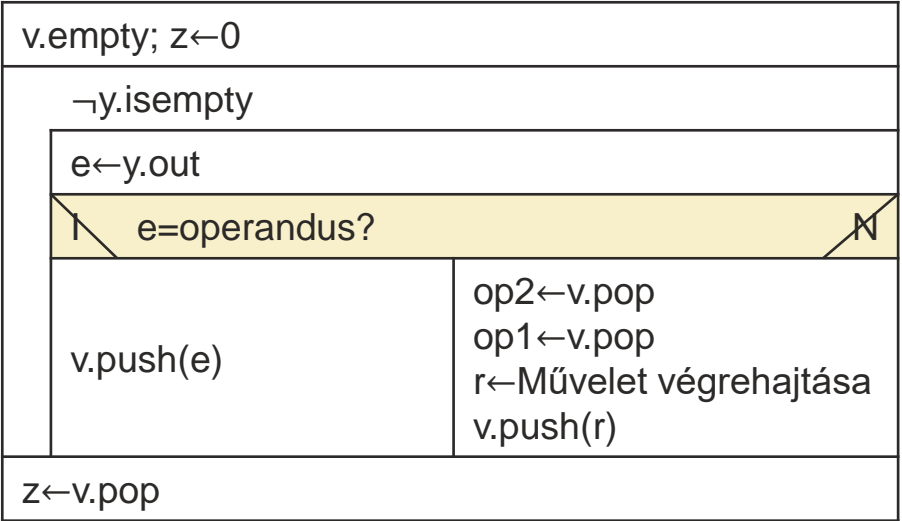


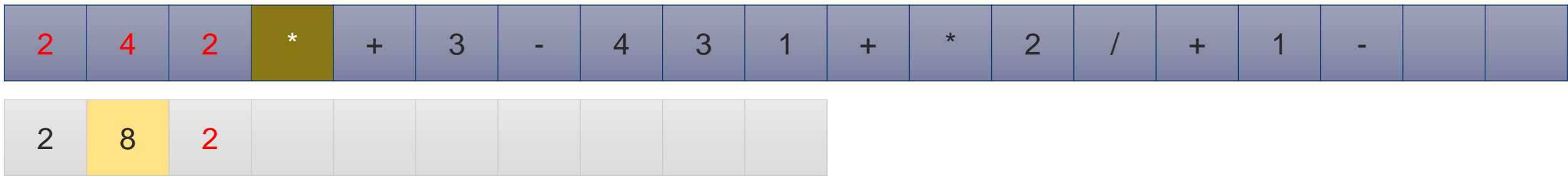
Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



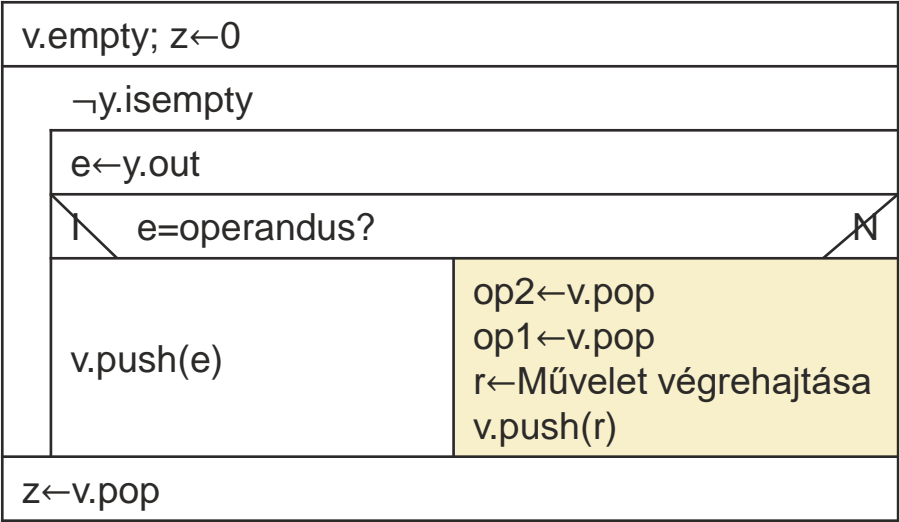


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

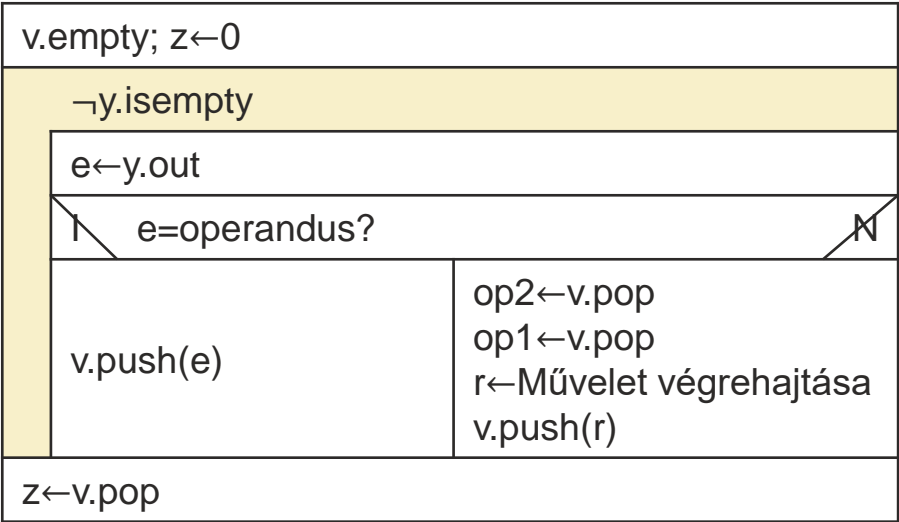
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



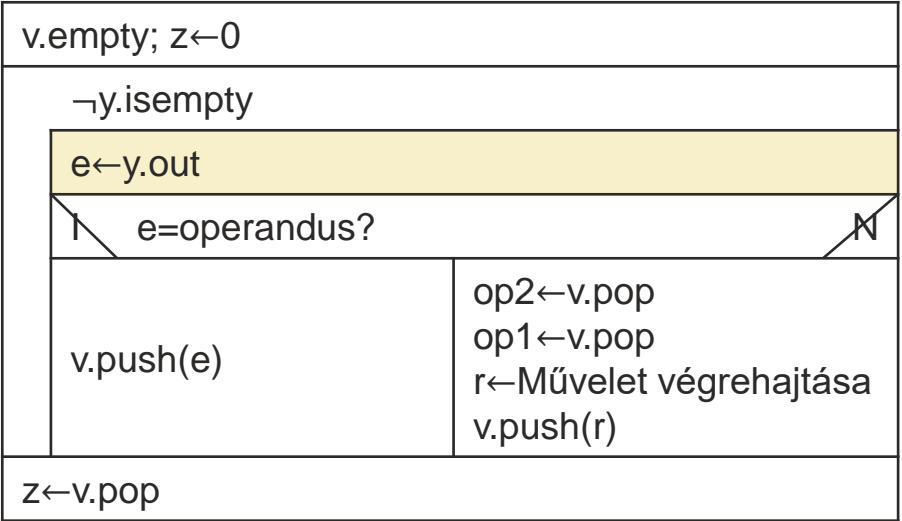
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
2	8	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



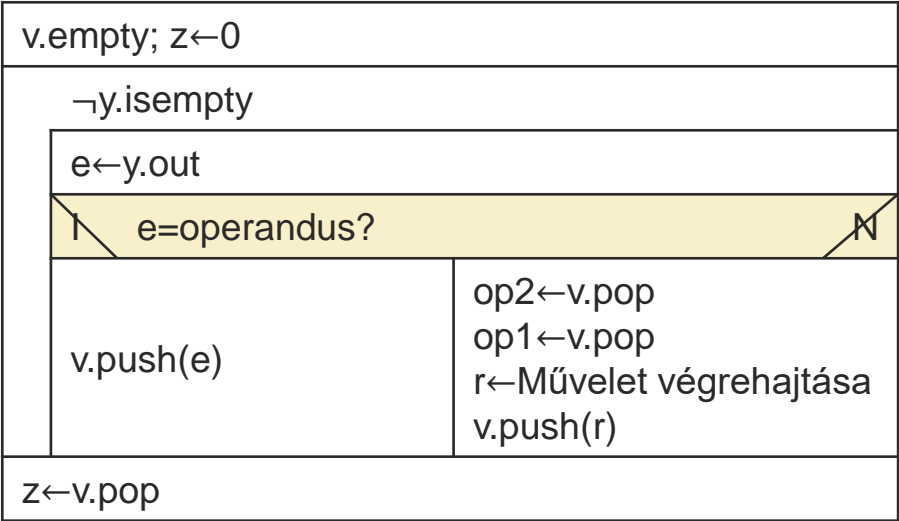


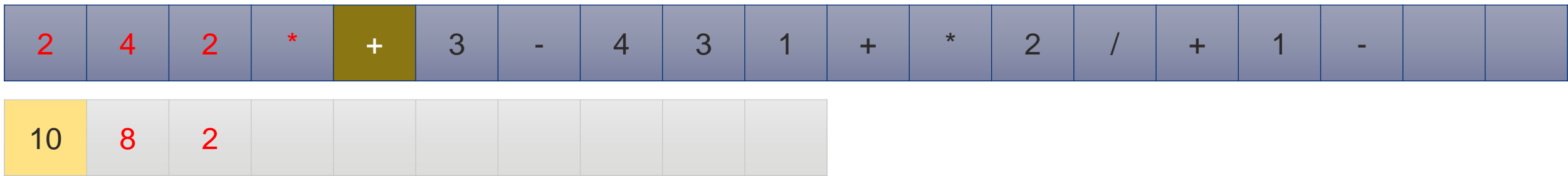
Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?



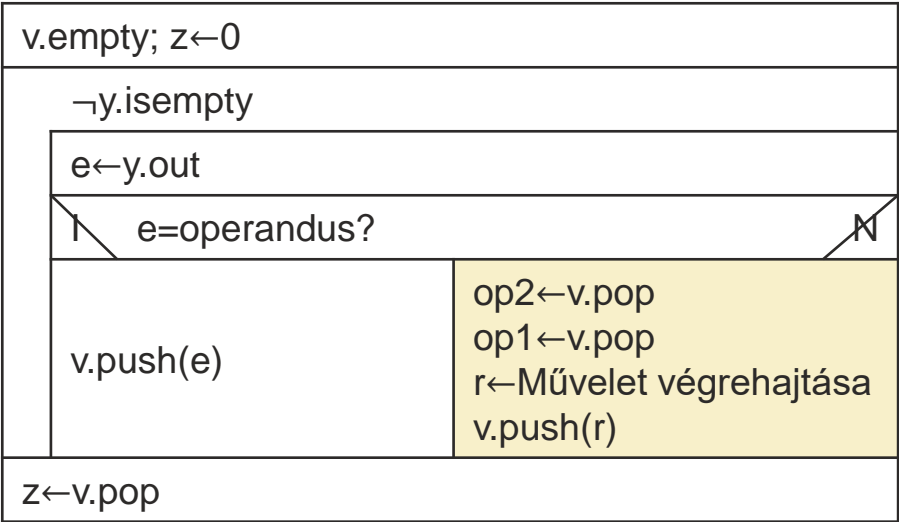


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

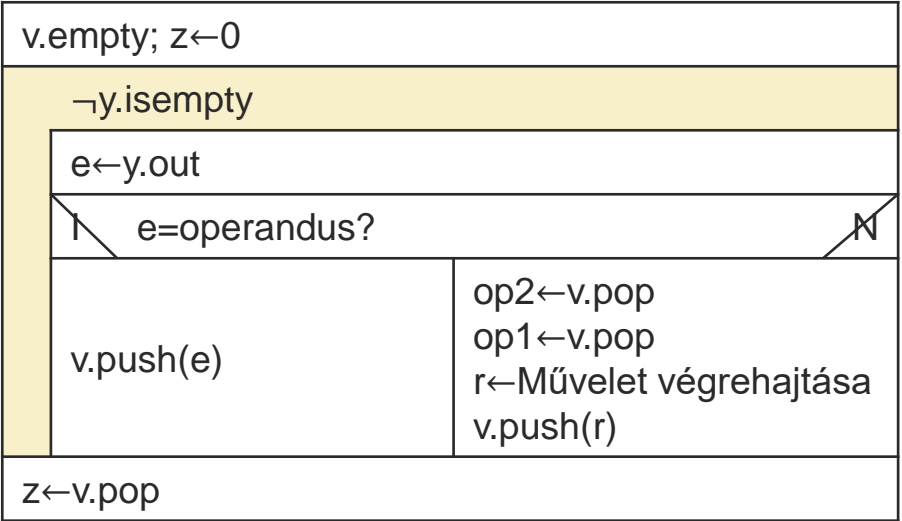
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



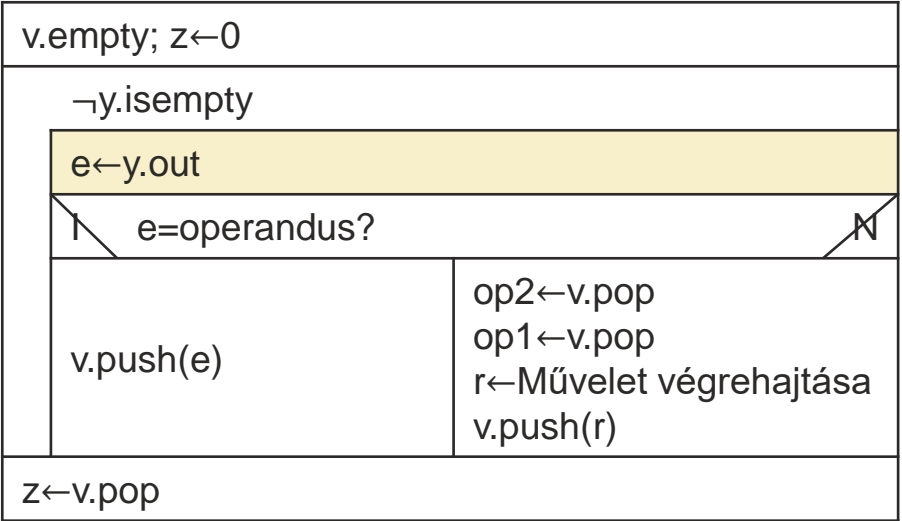
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
10	8	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



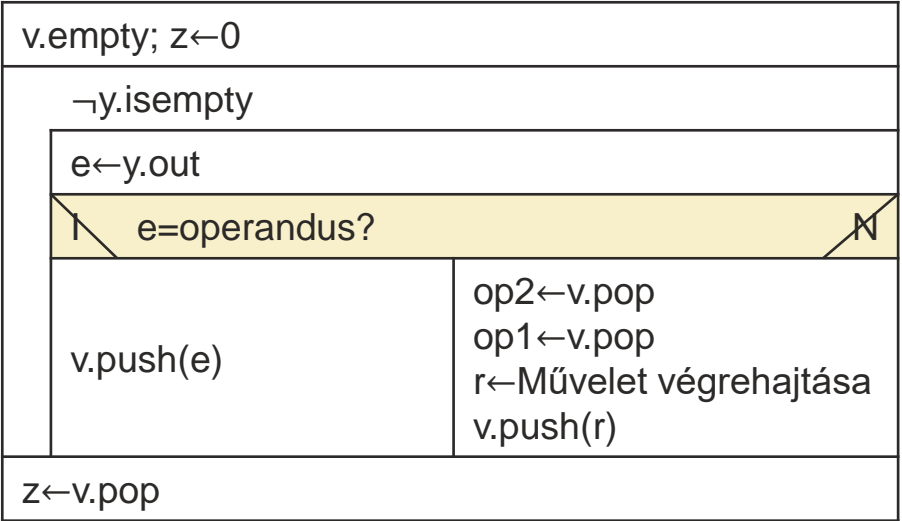


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
10	8	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?



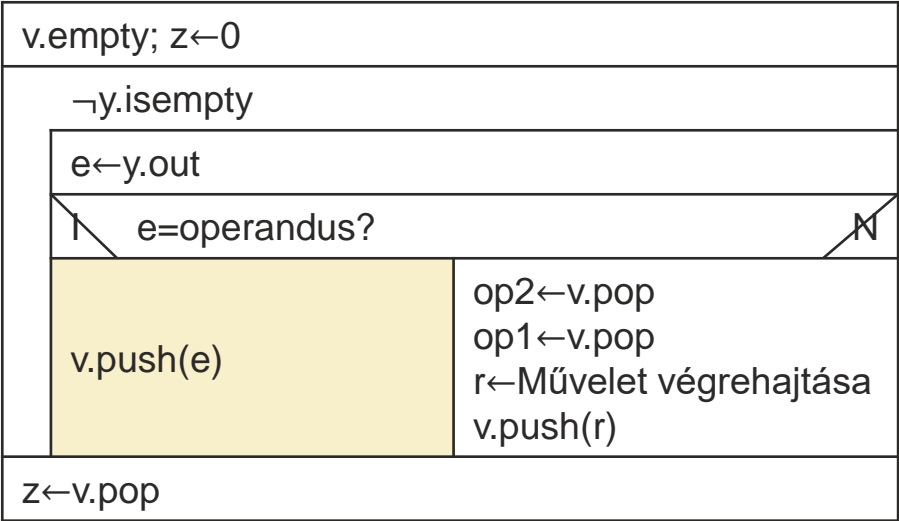


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

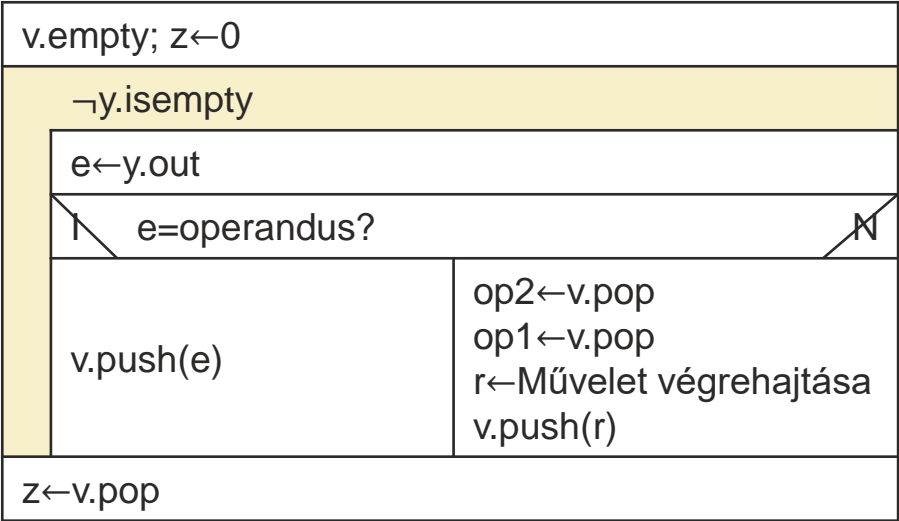
Operandus vagy operátor?

Operandus → Betesszük a verembe.



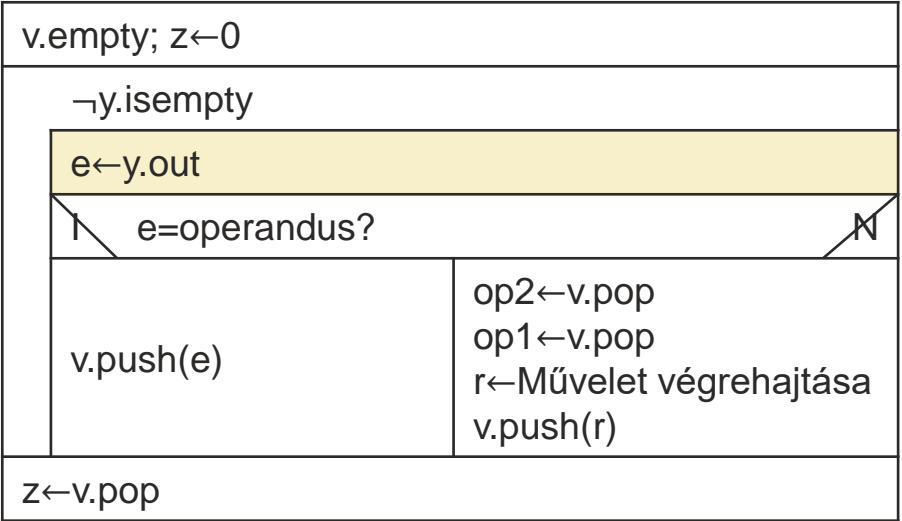
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
10	3	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



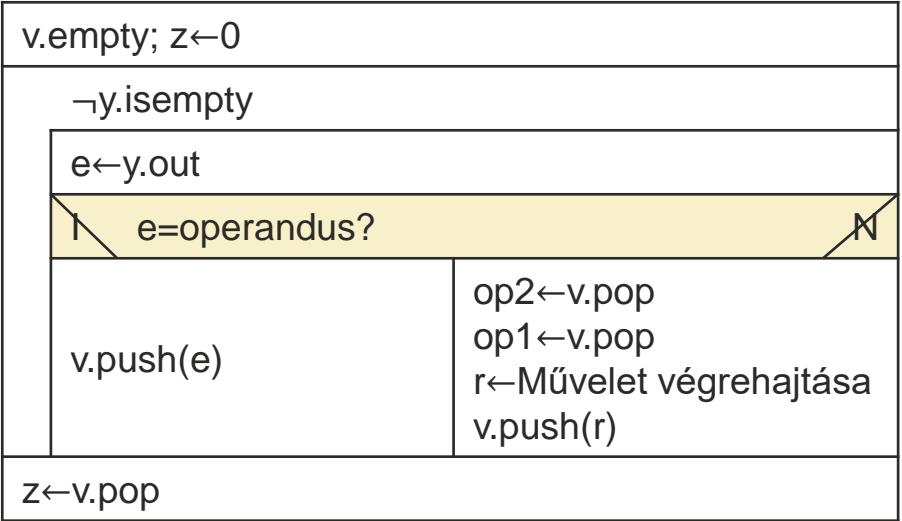


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
10	3	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?



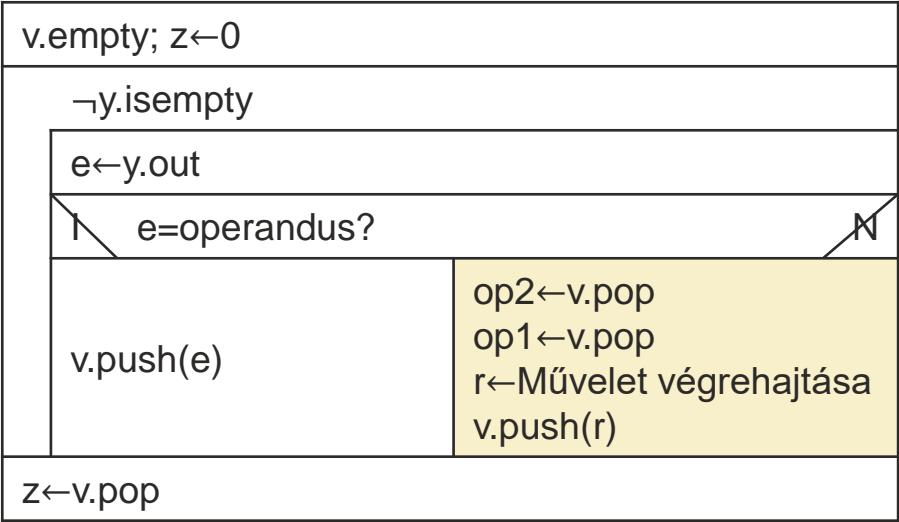


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

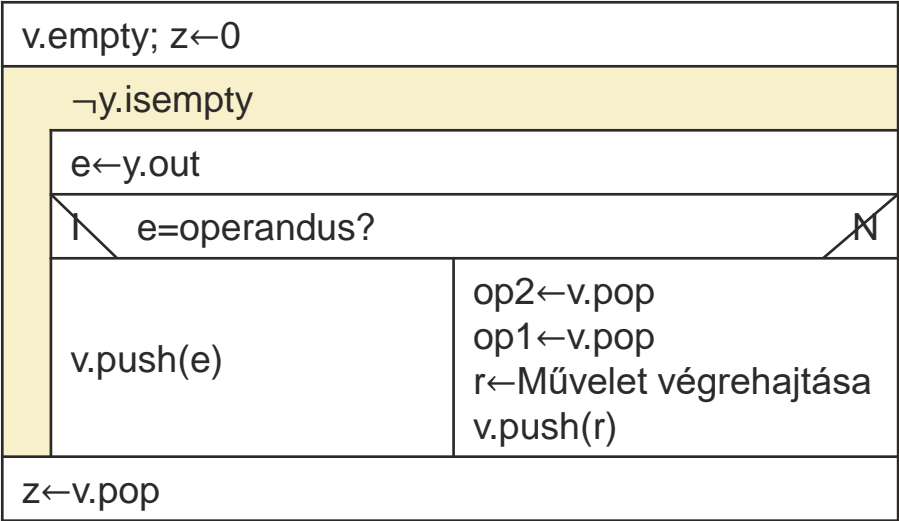
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



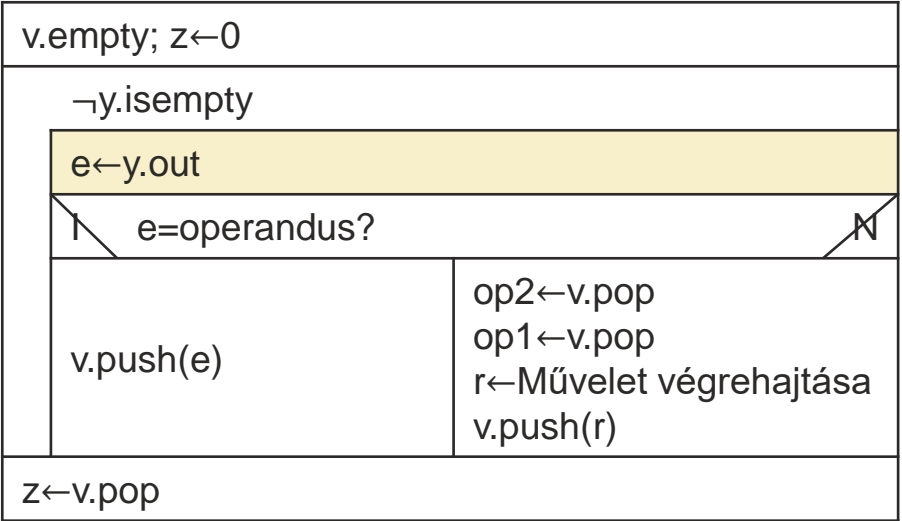
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	3	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



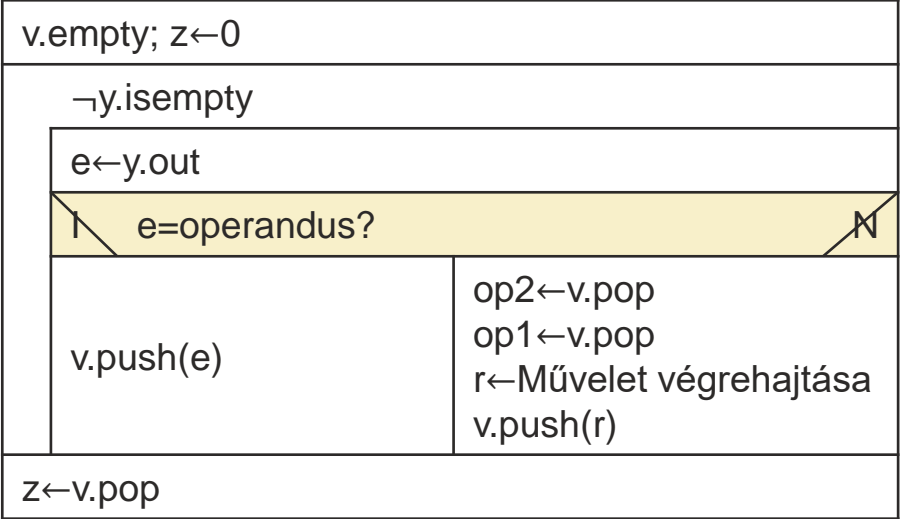


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



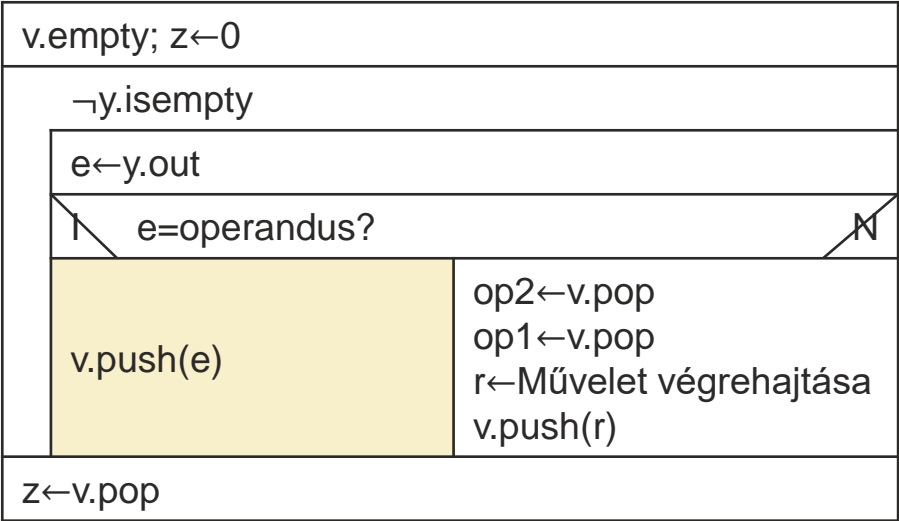


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

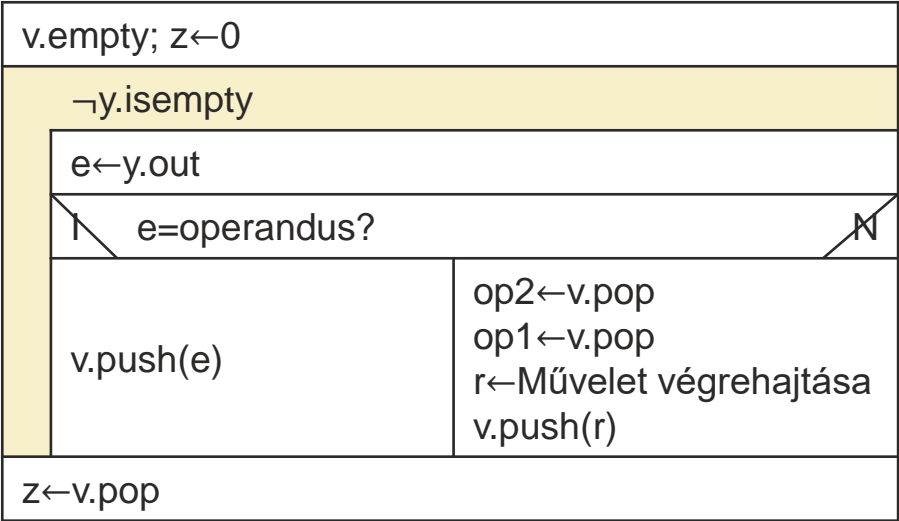
Operandus vagy operátor?

Operandus → Betesszük a verembe.



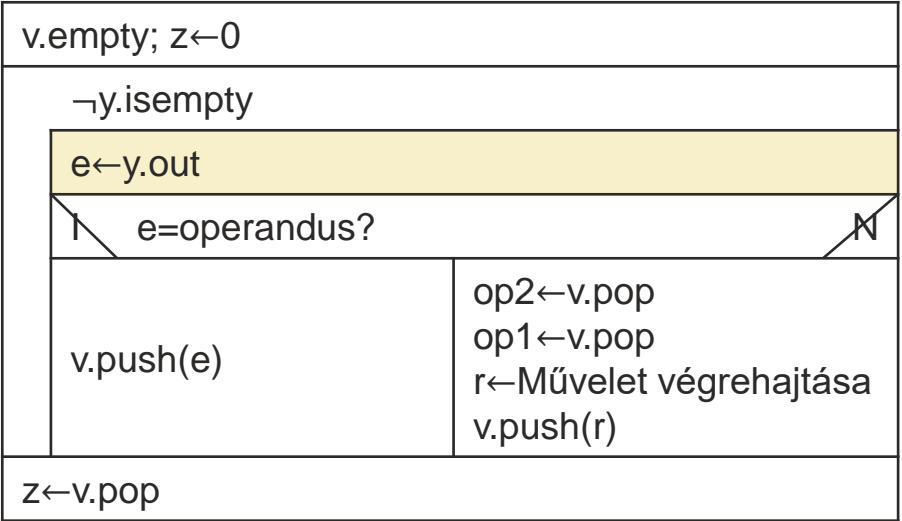
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	4	2																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



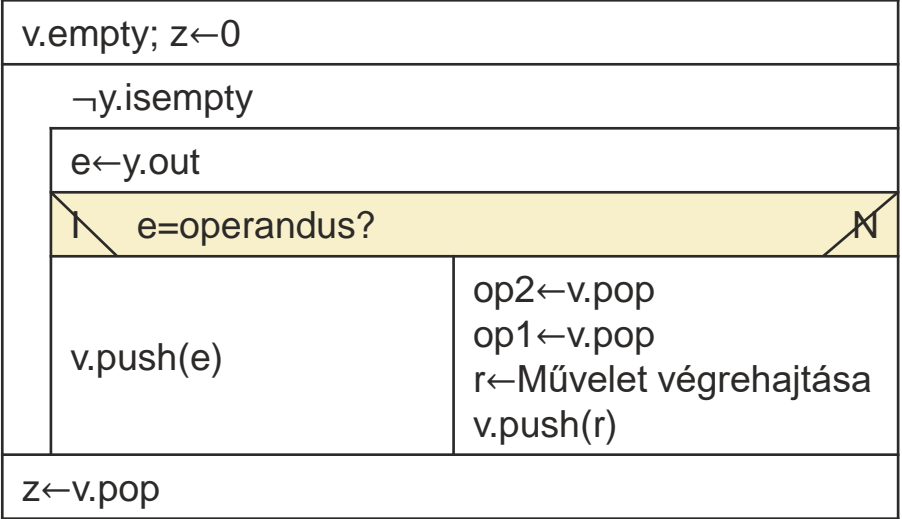


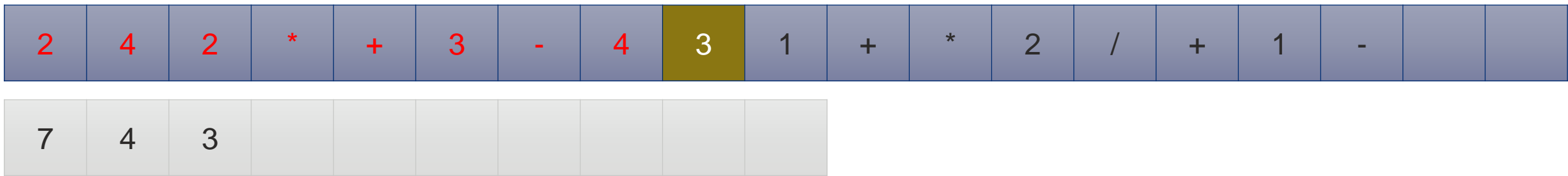
Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



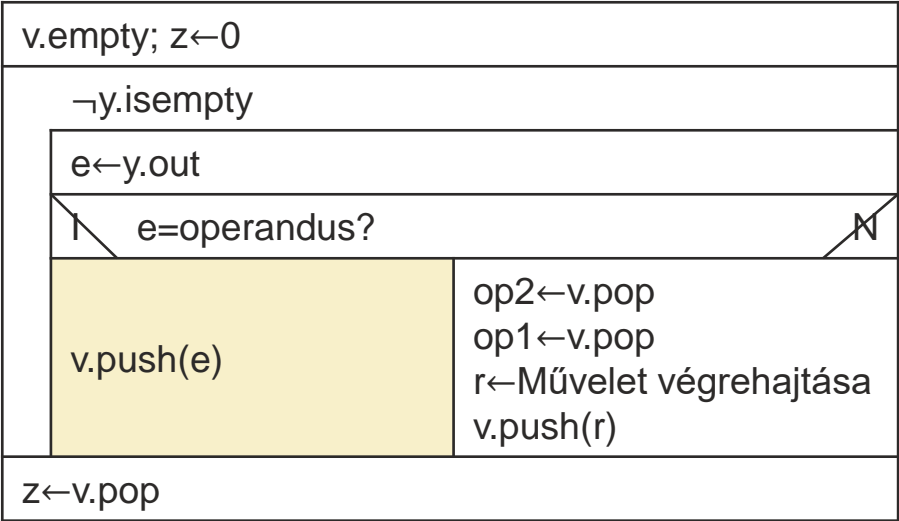


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

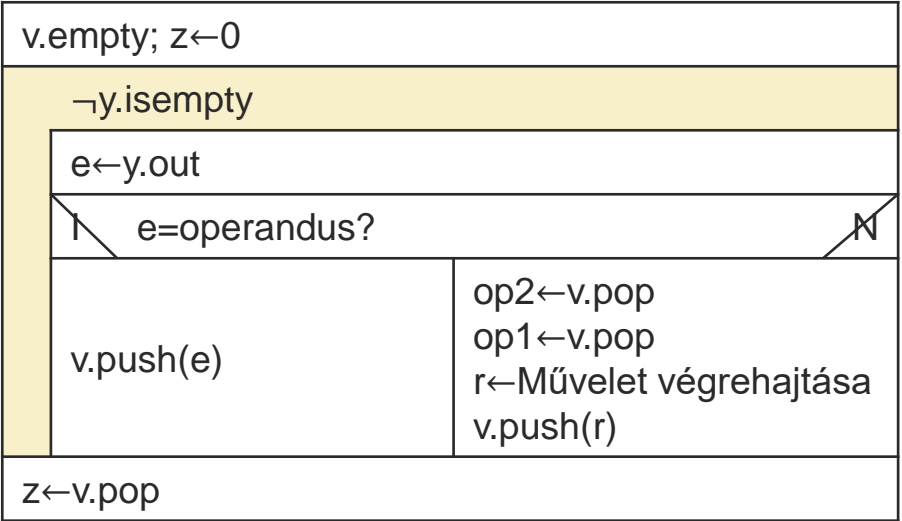
Operandus vagy operátor?

Operandus → Betesszük a verembe.



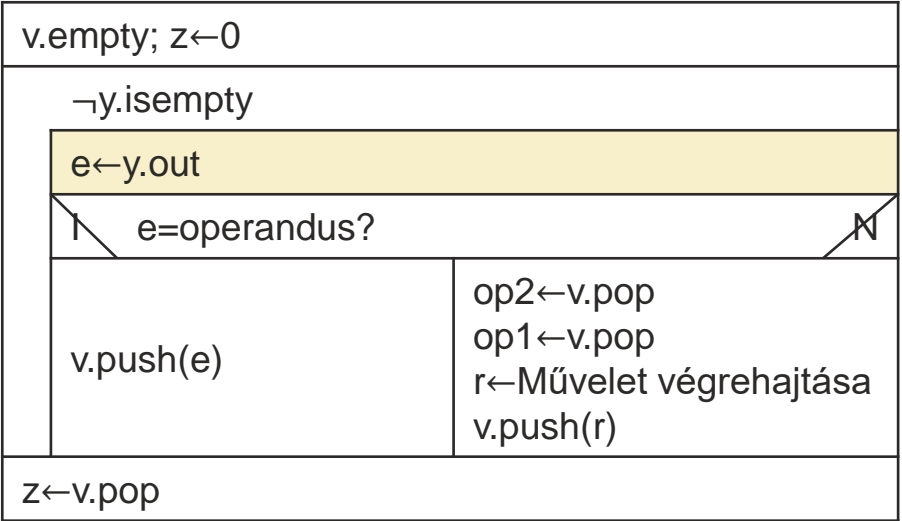
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	4	3																

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



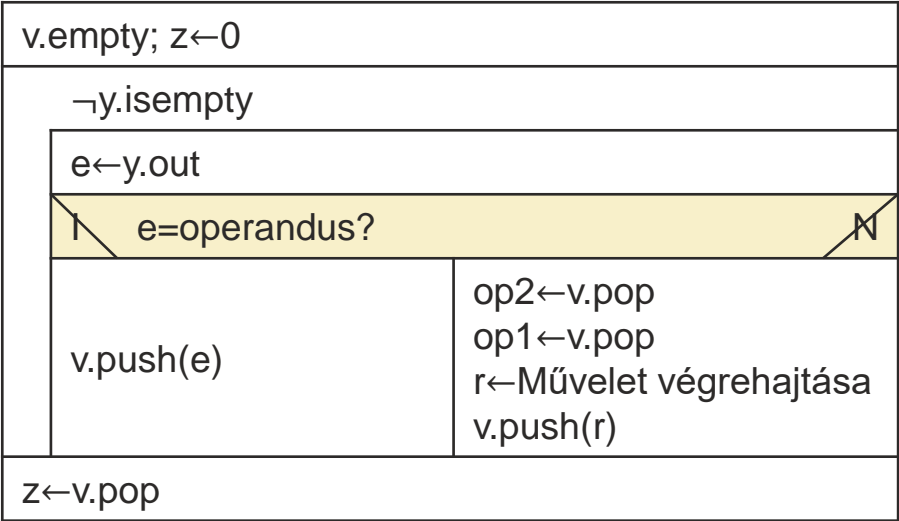


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



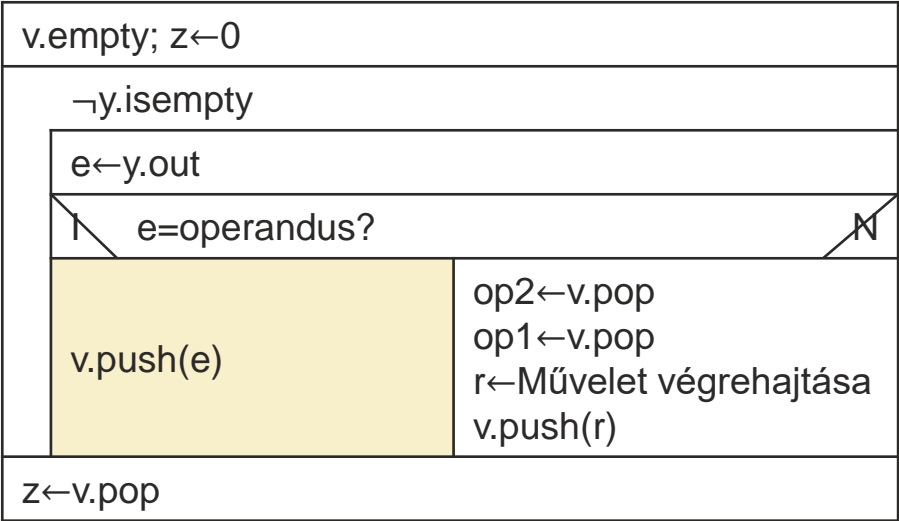


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?



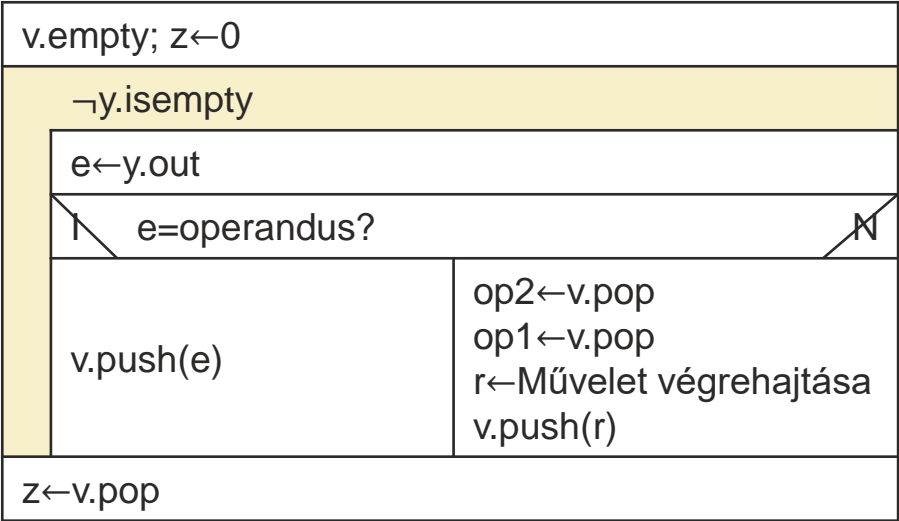


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?
 Operandus → Betesszük a verembe.



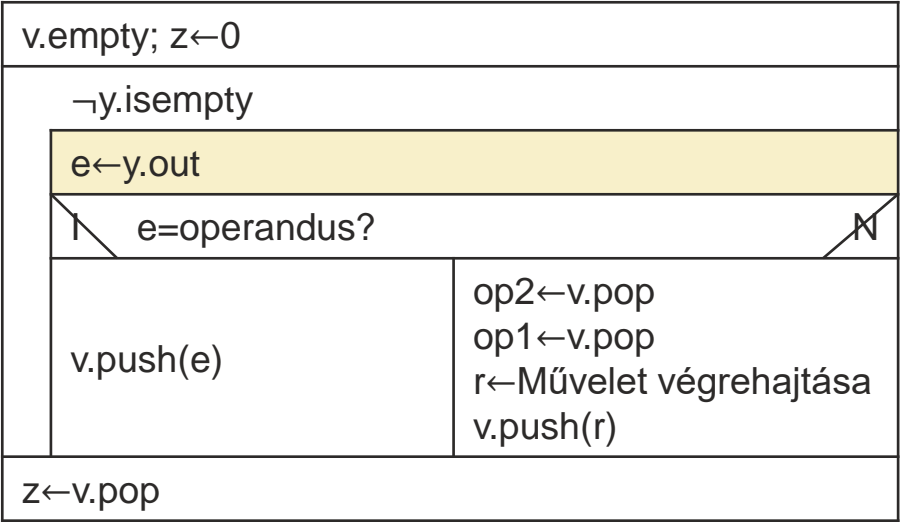
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	4	3	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



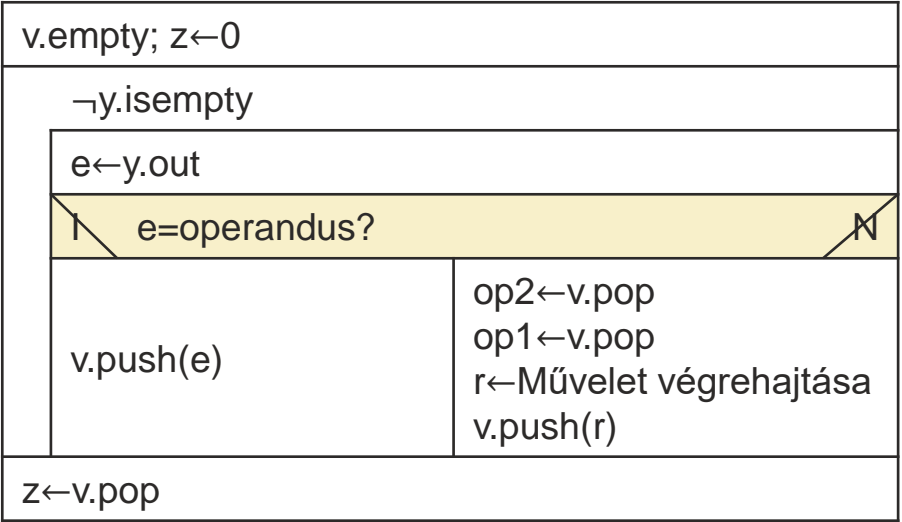


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



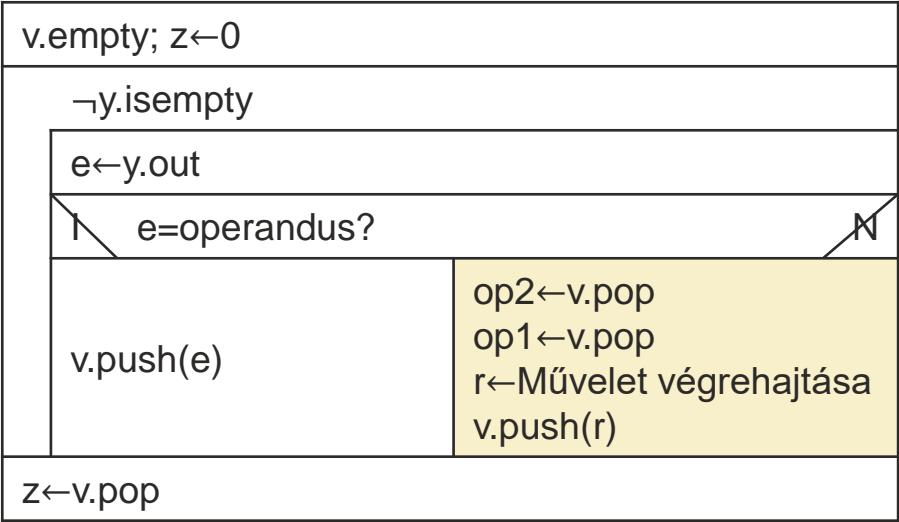


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

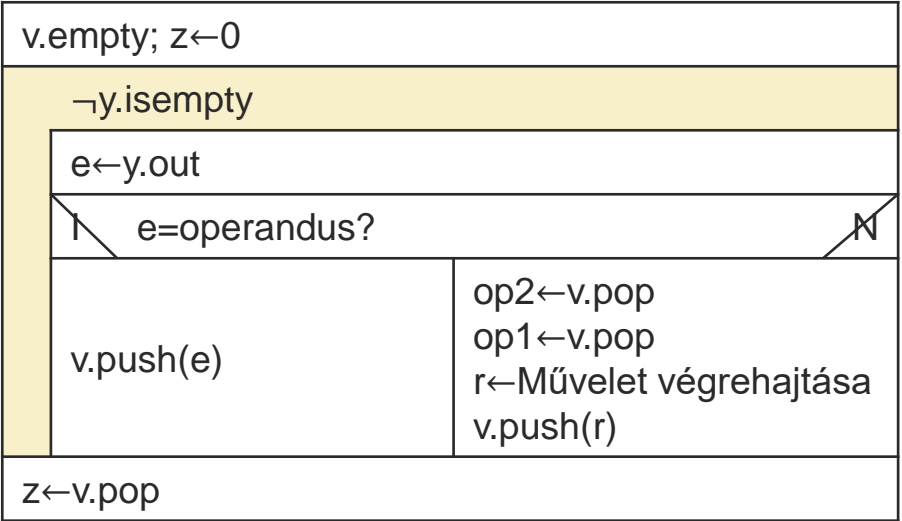
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



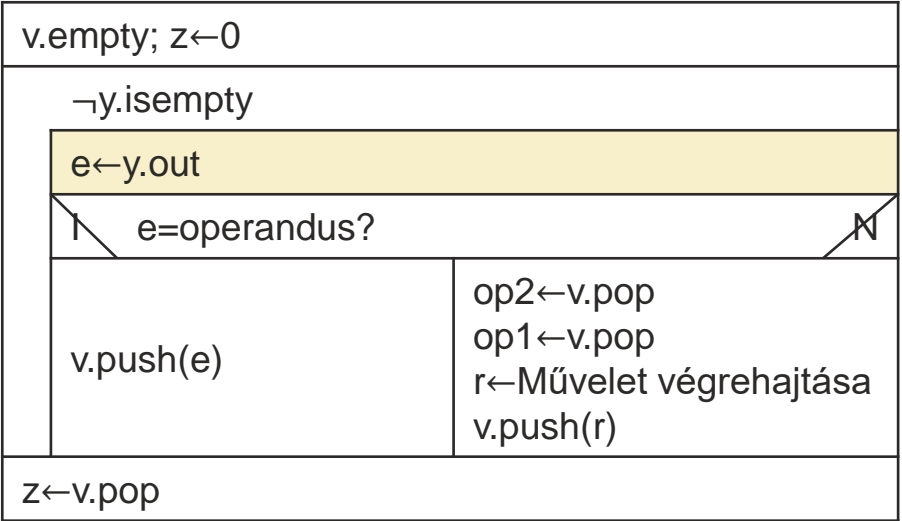
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	4	4	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



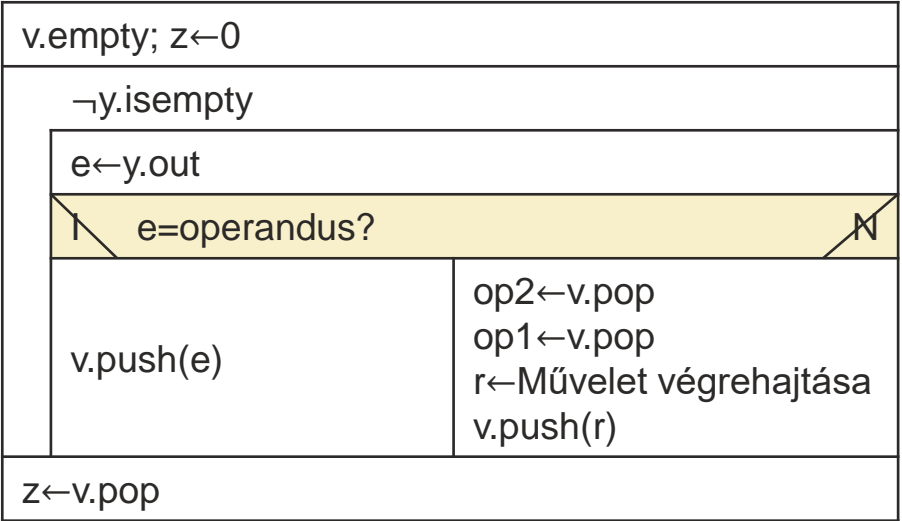


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



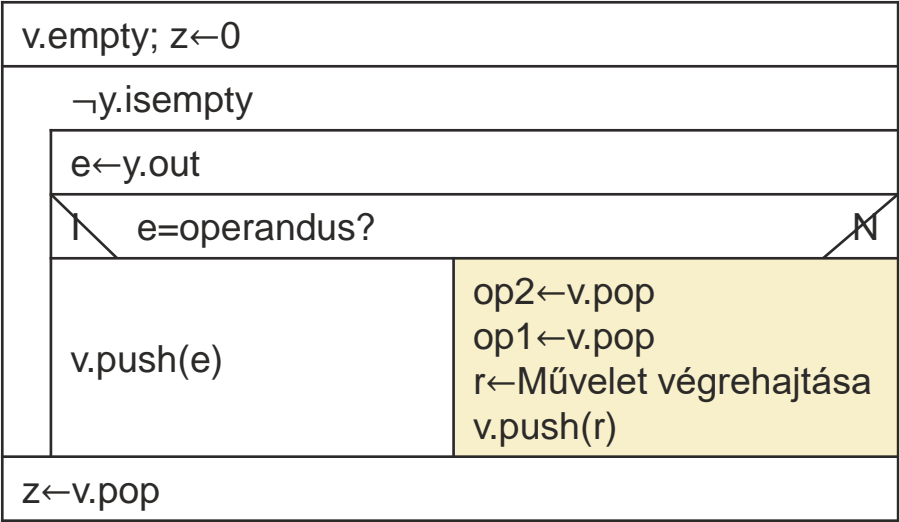


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

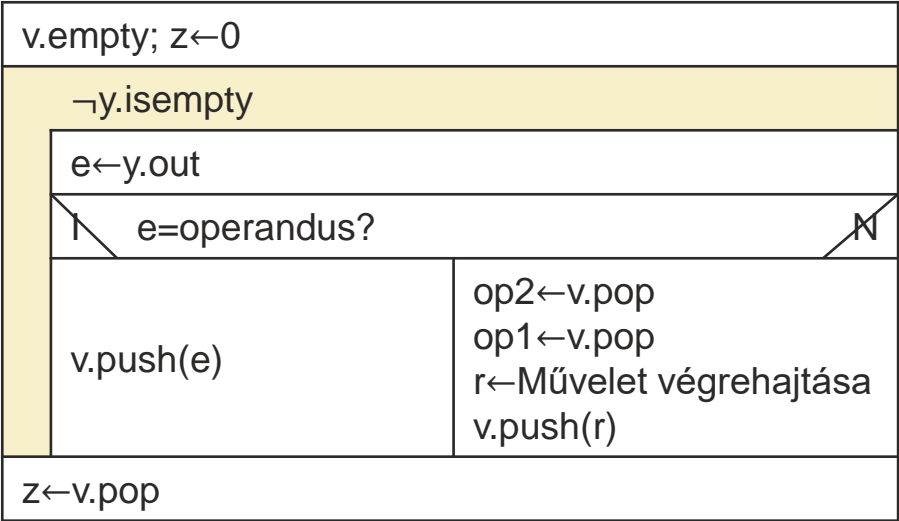
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



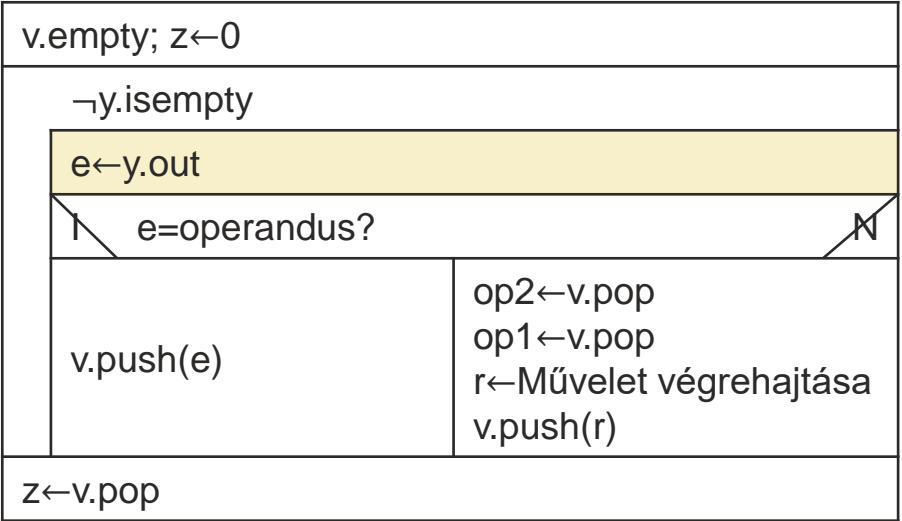
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	16	4	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



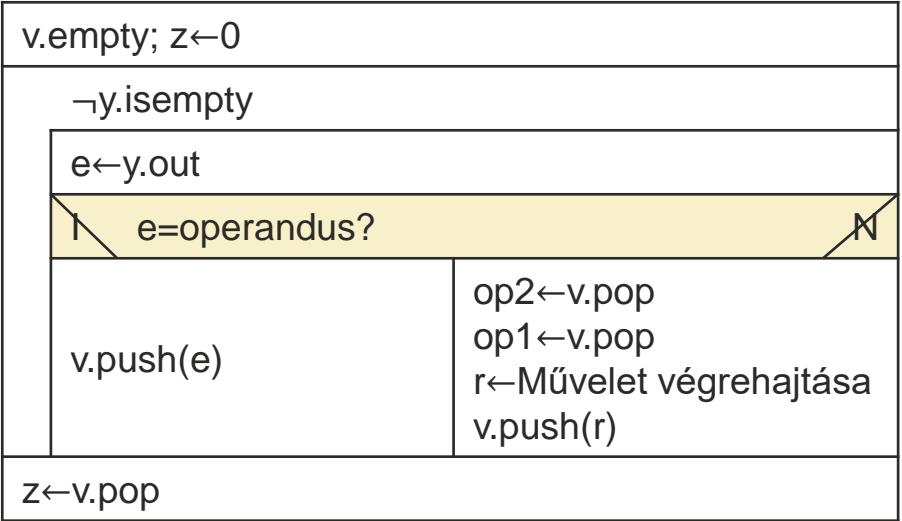
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	16	4	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



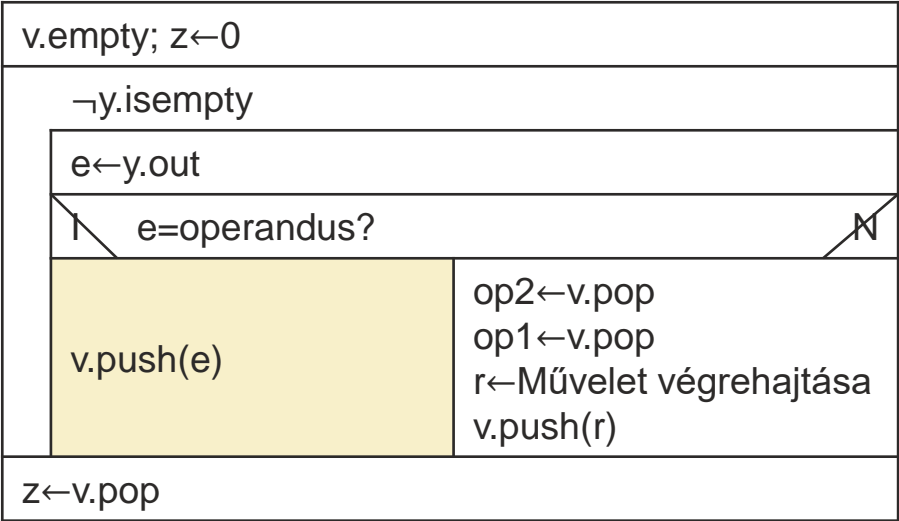
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	16	4	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



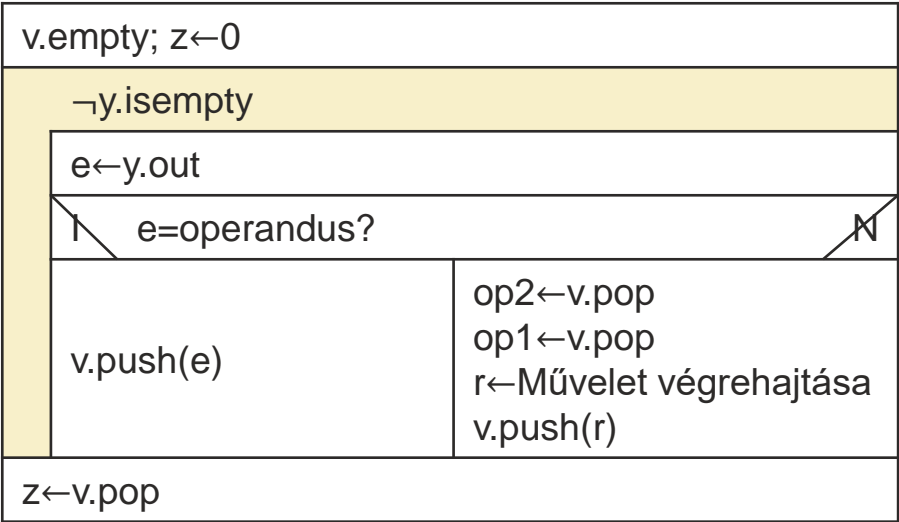


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?
 Operandus → Betesszük a verembe.



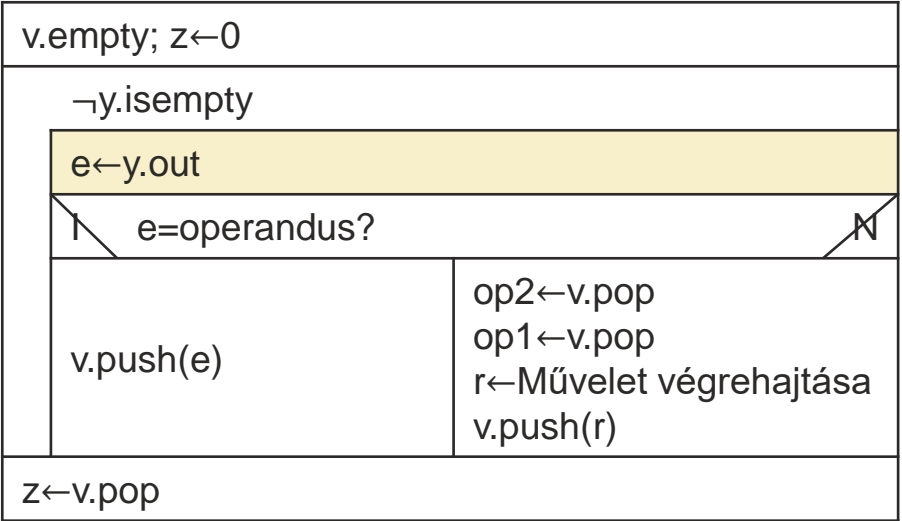
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	16	2	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



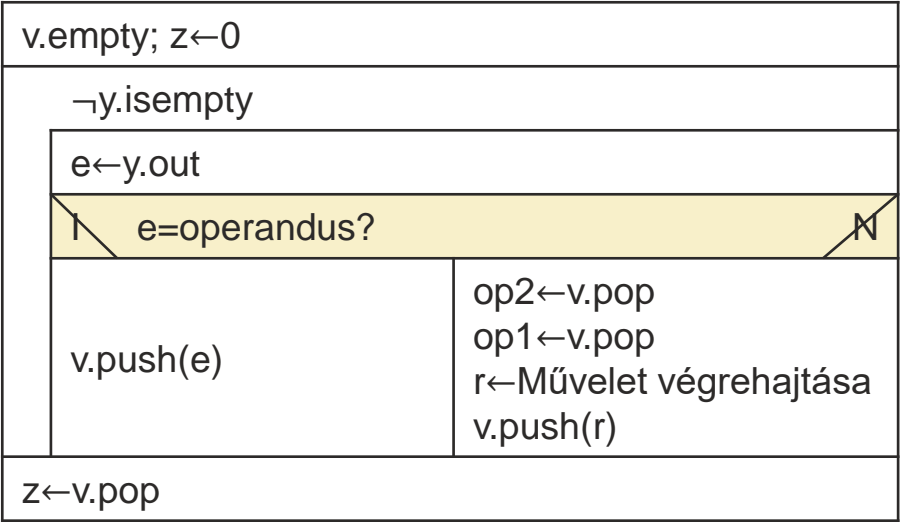


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



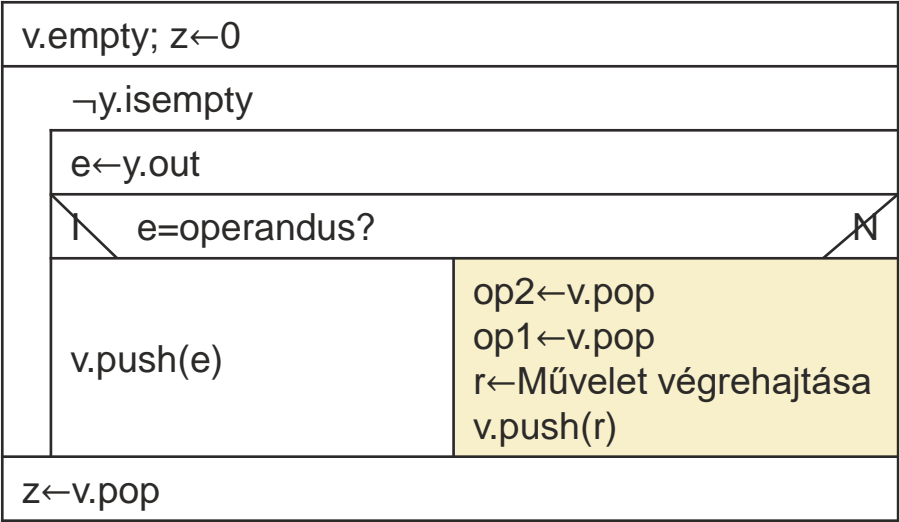


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

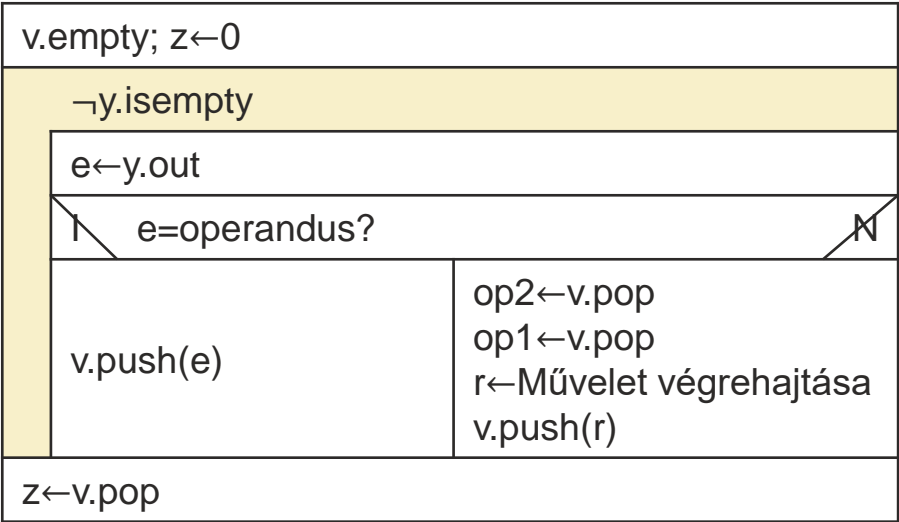
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



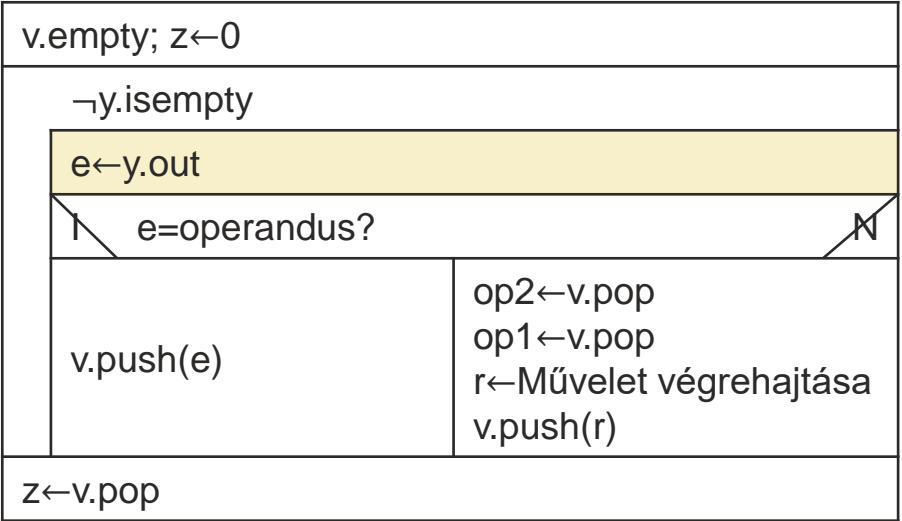
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
7	8	2	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



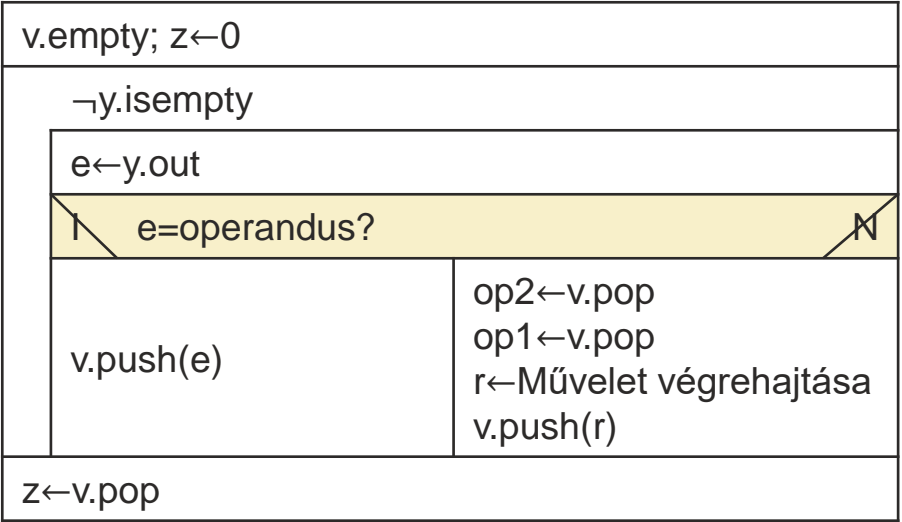


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



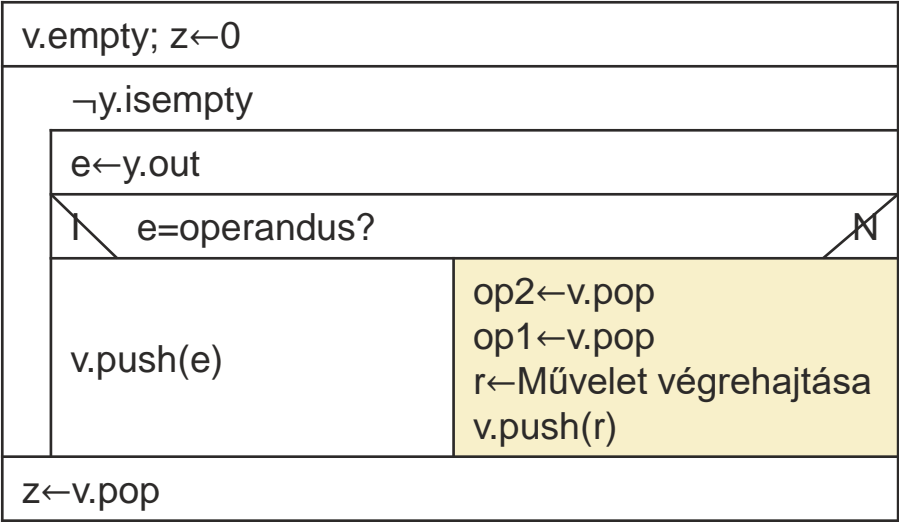


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

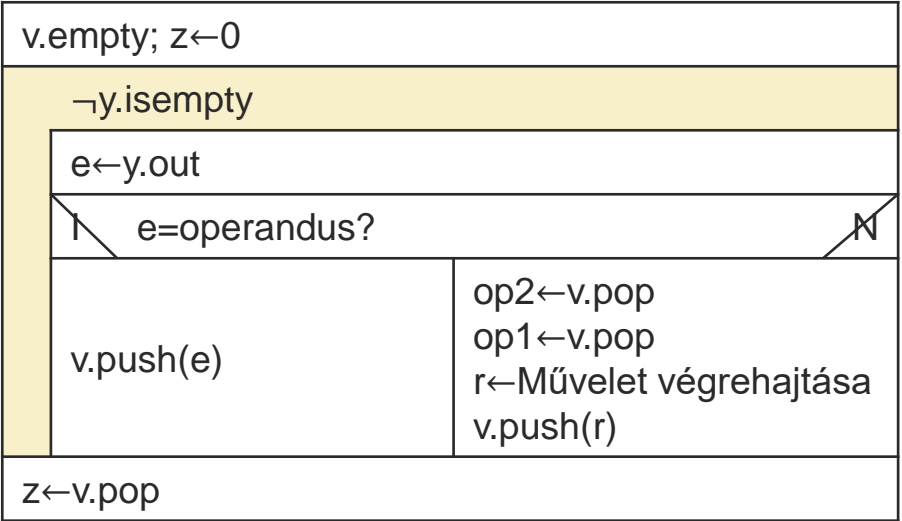
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.



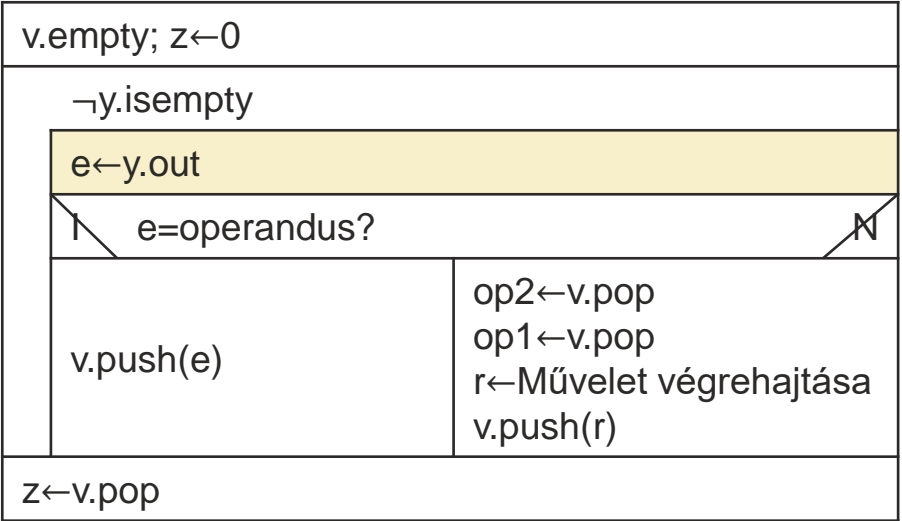
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
15	8	2	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



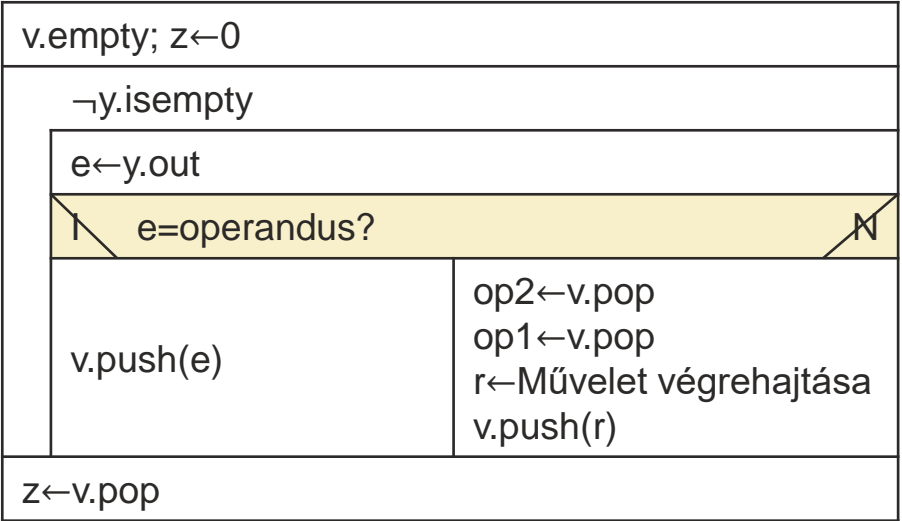


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.



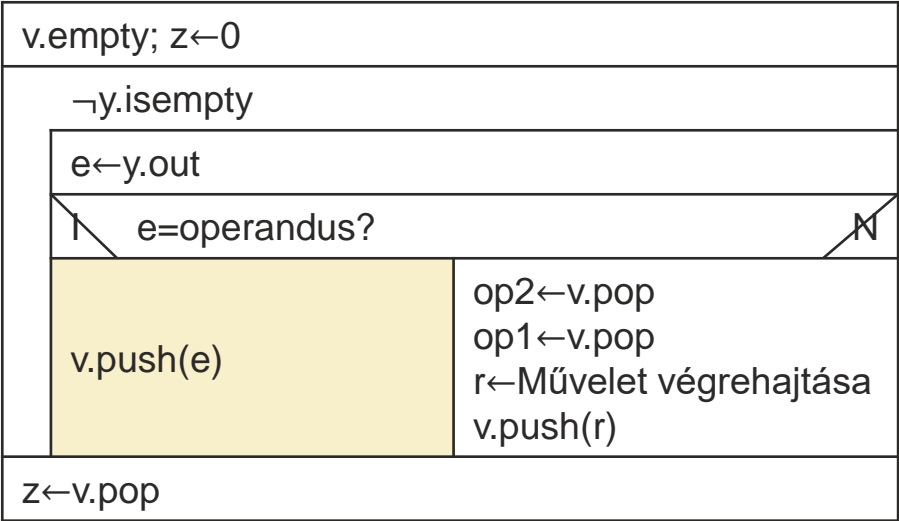


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?



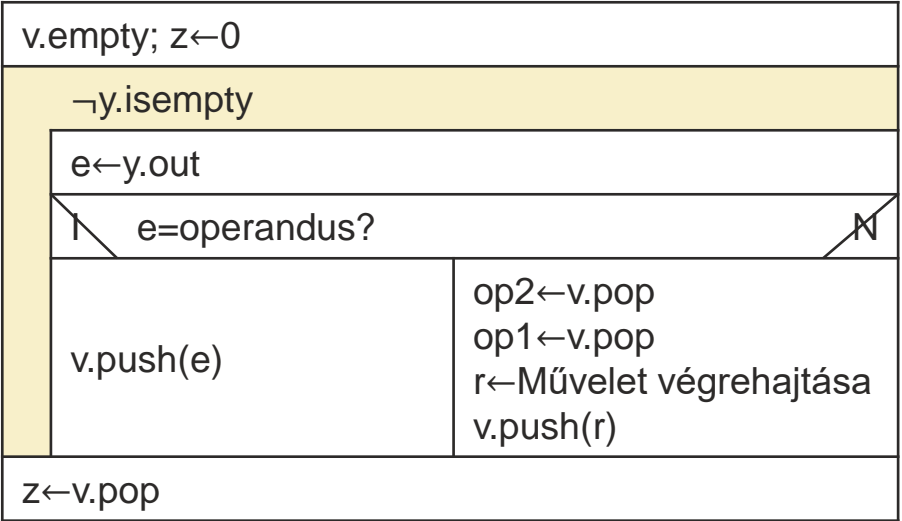


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.
Operandus vagy operátor?
Operandus → Betesszük a verembe.



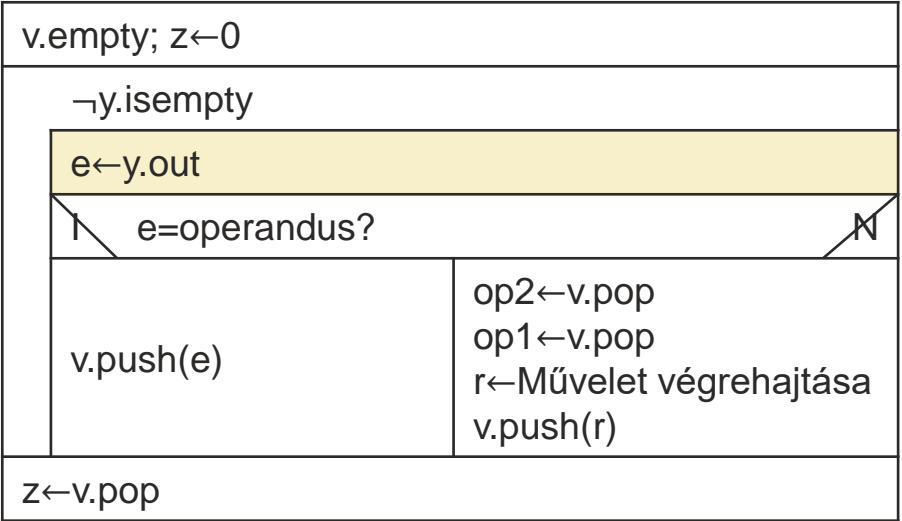
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
15	1	2	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.



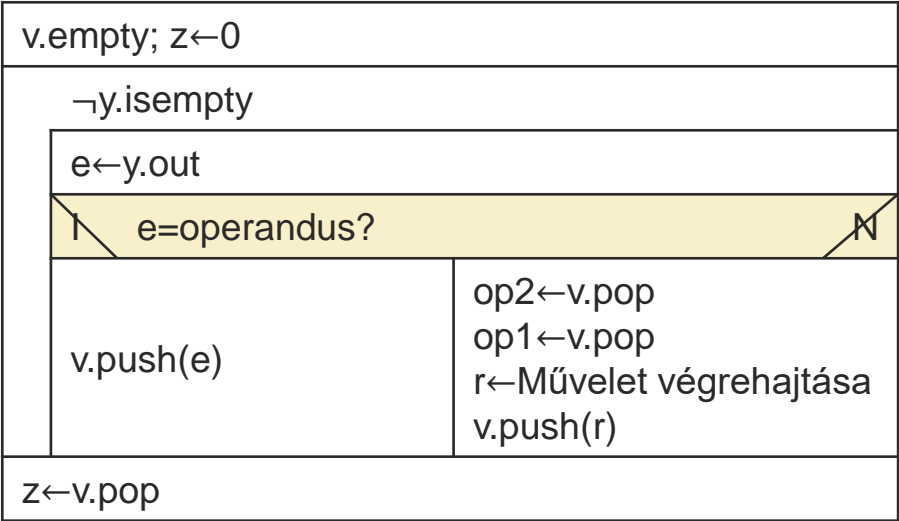


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Kivesszük a következő elemet.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
 Kivesszük a következő elemet.
 Operandus vagy operátor?



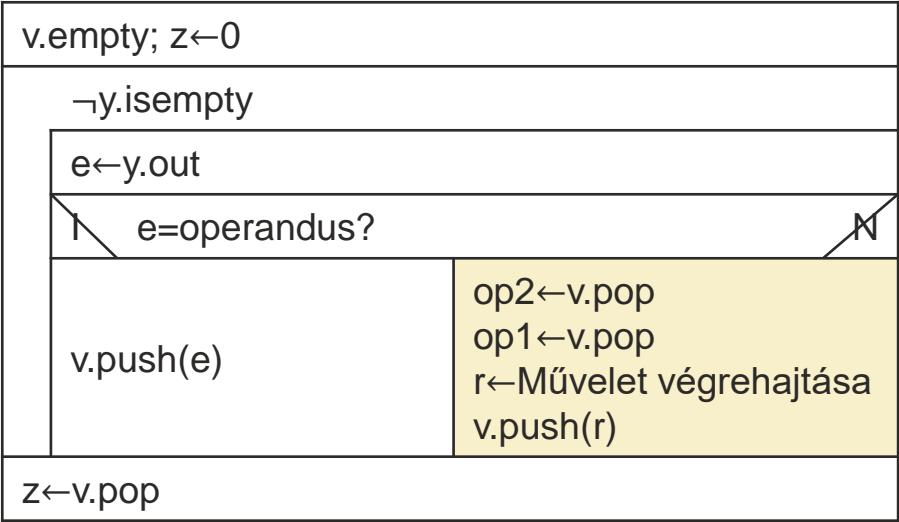


Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Kivesszük a következő elemet.

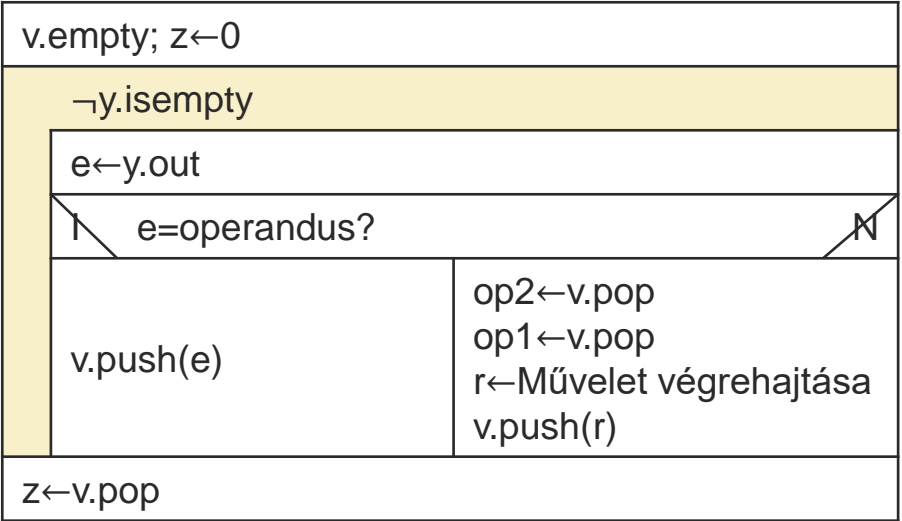
Operandus vagy operátor?

Operátor → kiveszünk két elemet a verem tetejéről és a végrehajtott művelet eredményét a verembe tesszük.





Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.
Mivel kiürült, így befejeződik a ciklus.



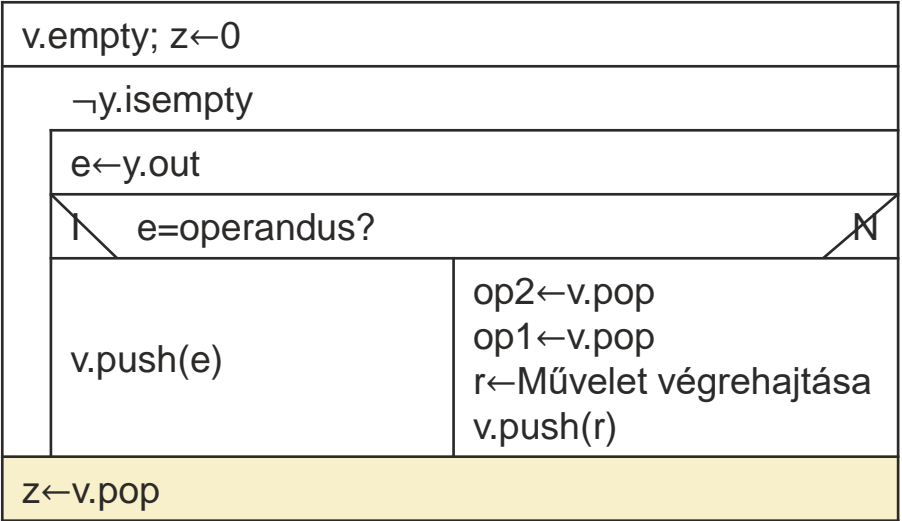
2	4	2	*	+	3	-	4	3	1	+	*	2	/	+	1	-		
14	1	2	1															

Ha a sor nem üres, akkor végrehajtjuk a ciklusmagot.

Mivel kiürült, így befejeződik a ciklus.

Az eredmény pedig a verem tetején található.

Amennyiben a veremben több elem van, akkor hiba történt.



Tömbök

Következő téma