O tom co nejde naprogramovat

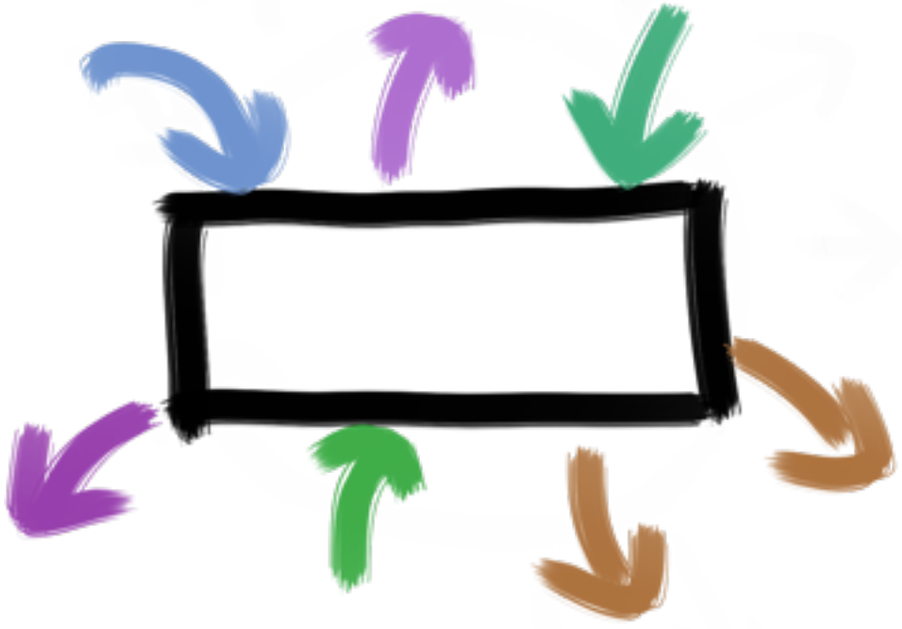
Ada Böhm
ada@kreatrix.org

Algorytmy

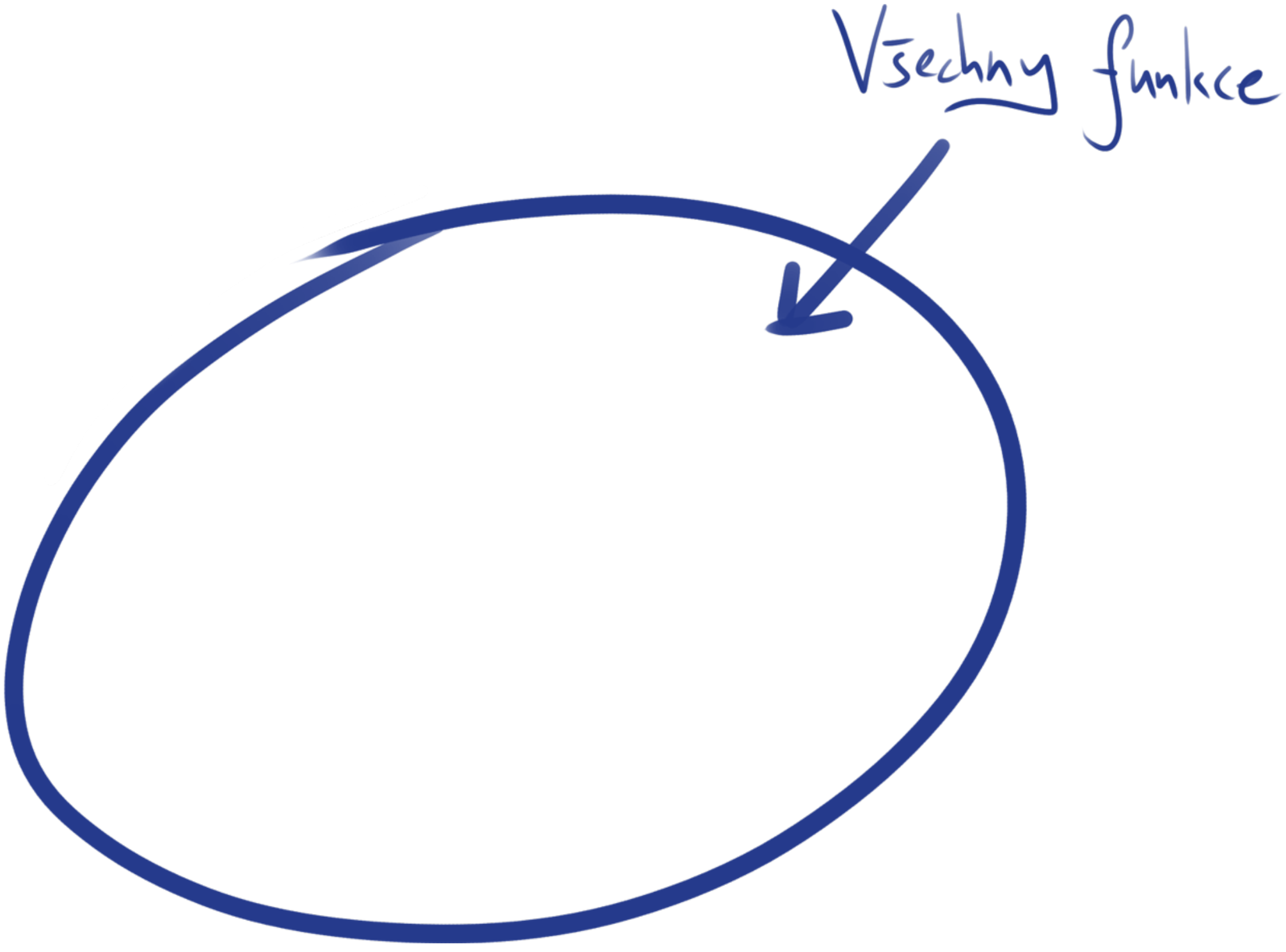
Algoritmy

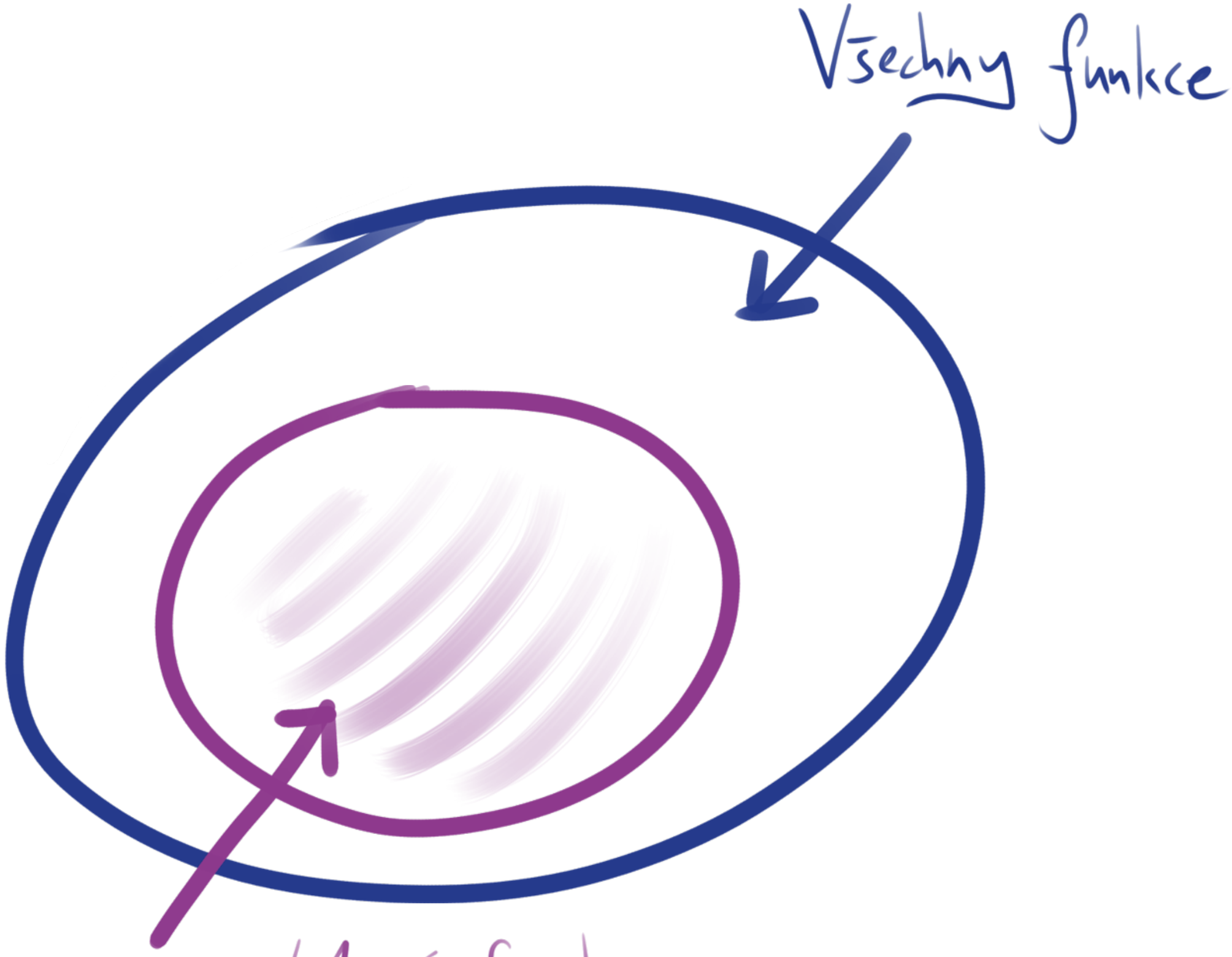
```
def muj_program(i):
    if i > 42:
        return i + 10
    else:
        return i * 10
```

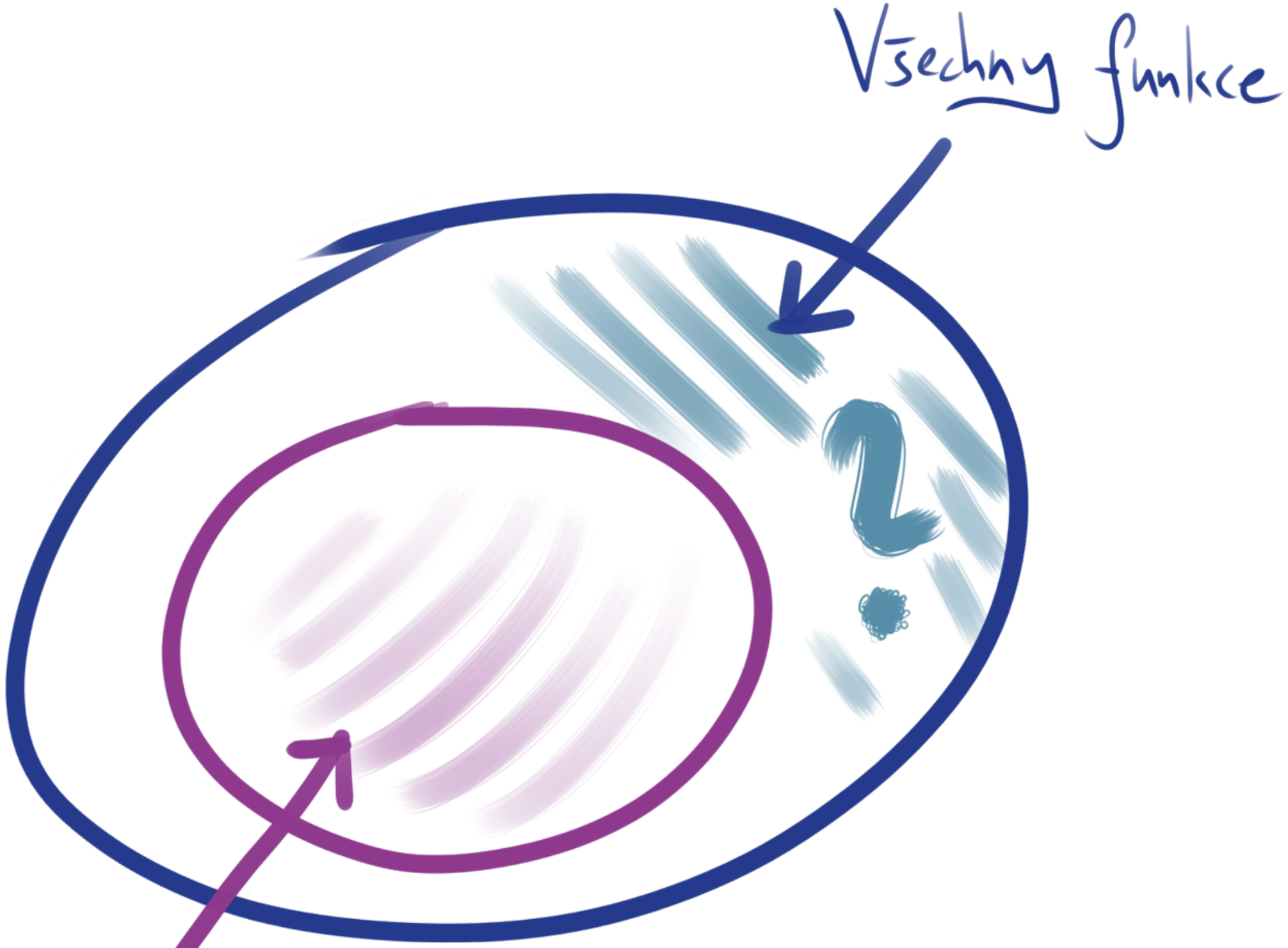
Algoritmy



Reaktivní systémy


Všechny funkce


Všechny funkce
Naprogramovatelné funkce


Všechny funkce
Naprogramovatelné funkce

A

$A :=$

a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x
y	z	0	1	2	3	4	5
6	7	8	9	.	,	?	␣
(	)	[	]	{	}	<	>
=	!	'	"	+	-	*	/

{"ahoj", "123", "", "x = y + 5", ...

```
{"ahoj", "123", "", "x = y + 5", ...
          "s = 'Hello'
           print(s)", ... }
```

$A :=$

a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x
y	z	0	1	2	3	4	5
6	7	8	9	.	,	?	␣
(	)	[	]	{	}	<	>
=	!	'	"	+	-	*	/

A =

a 01	b 02	c 03	d 04	e 05	f 06	g 07	h 08
i 09	j 10	k 11	l 12	m 13	n 14	o 15	p 16
q 17	r 18	s 19	t 20	u 21	v 22	w 23	x 24
y 25	z 26	0 27	1 28	2 29	3 30	4 31	5 32
6 33	7 34	8 35	9 36	. 37	, 38	? 39	␣ 40
(41	) 42	[43	] 44	{ 45	} 46	< 47	> 48
= 49	! 50	' 51	" 52	+ 53	- 54	* 55	/ 56

A =

a 01	b 02	c 03	d 04	e 05	f 06	g 07	h 08
i 09	j 10	k 11	l 12	m 13	n 14	o 15	p 16
q 17	r 18	s 19	t 20	u 21	v 22	w 23	x 24
y 25	z 26	0 27	1 28	2 29	3 30	4 31	5 32
6 33	7 34	8 35	9 36	. 37	, 38	? 39	␣ 40
(41	) 42	[43	] 44	{ 45	} 46	< 47	> 48
= 49	! 50	' 51	" 52	+ 53	- 54	* 55	/ 56

print

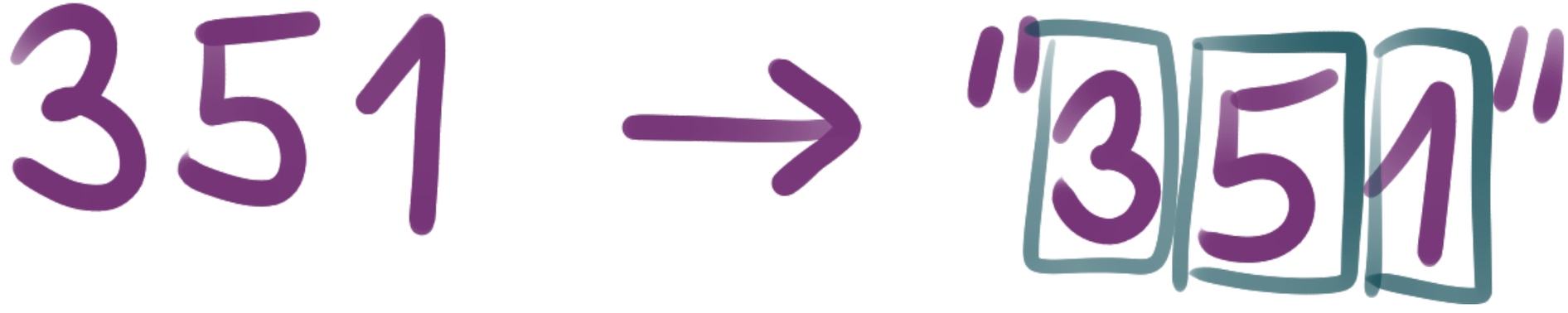
A =

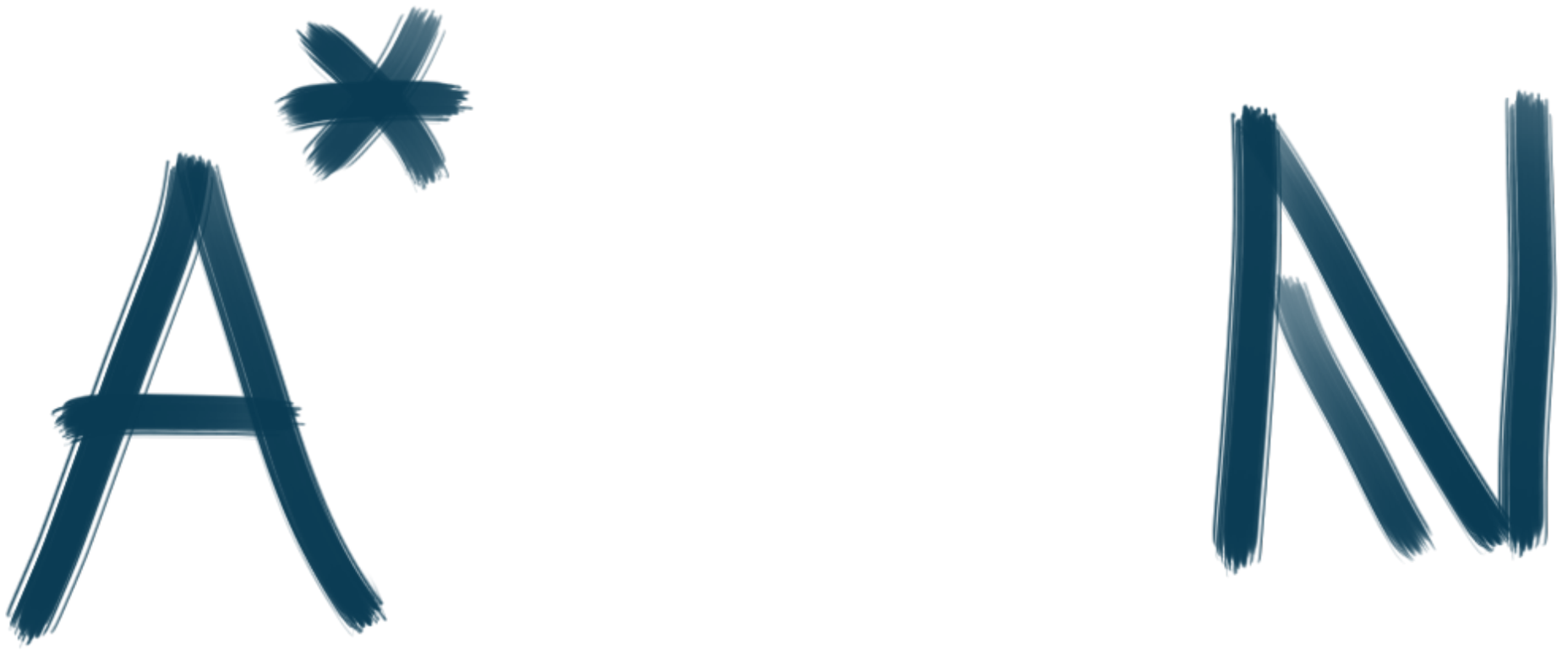
a 01	b 02	c 03	d 04	e 05	f 06	g 07	h 08
i 09	j 10	k 11	l 12	m 13	n 14	o 15	p 16
q 17	r 18	s 19	t 20	u 21	v 22	w 23	x 24
y 25	z 26	0 27	1 28	2 29	3 30	4 31	5 32
6 33	7 34	8 35	9 36	. 37	, 38	? 39	␣ 40
(41	) 42	[43	] 44	{ 45	} 46	< 47	> 48
= 49	! 50	' 51	" 52	+ 53	- 54	* 55	/ 56

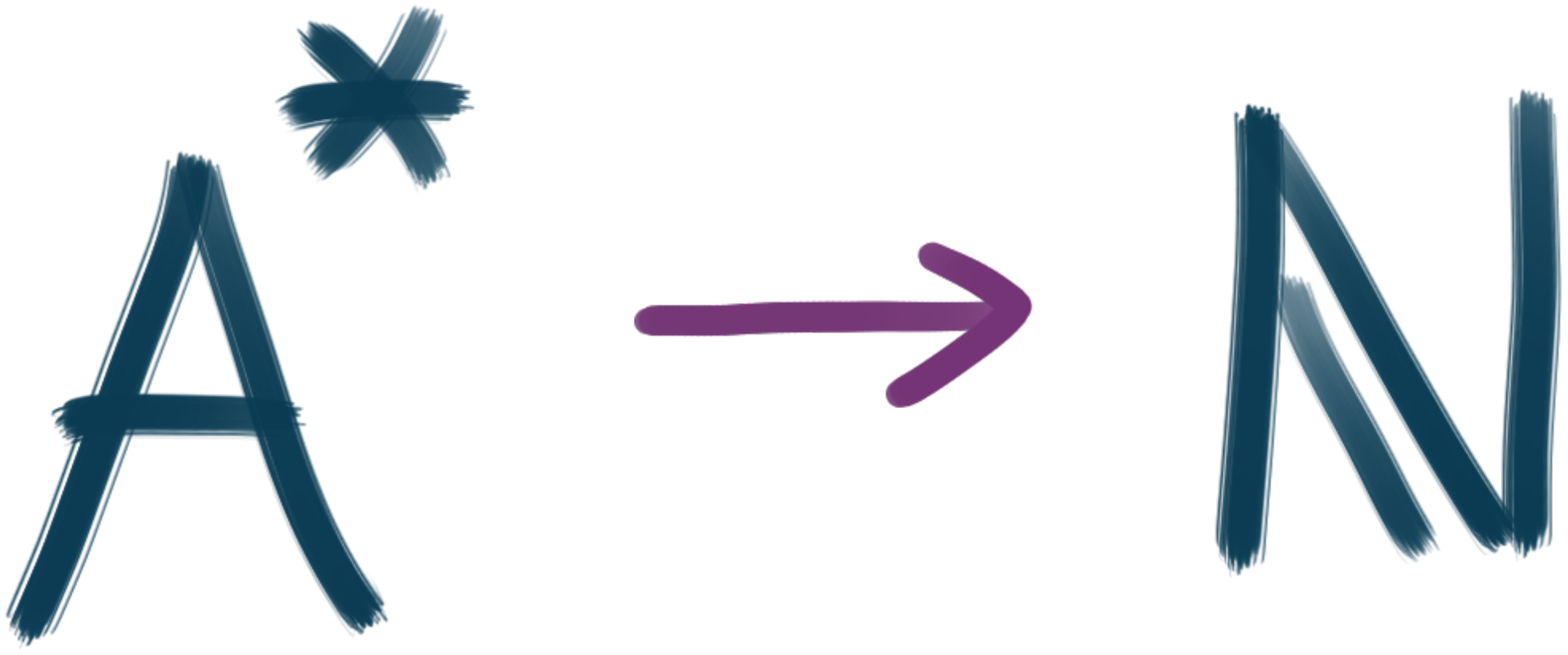
print
16 18 09 14 20

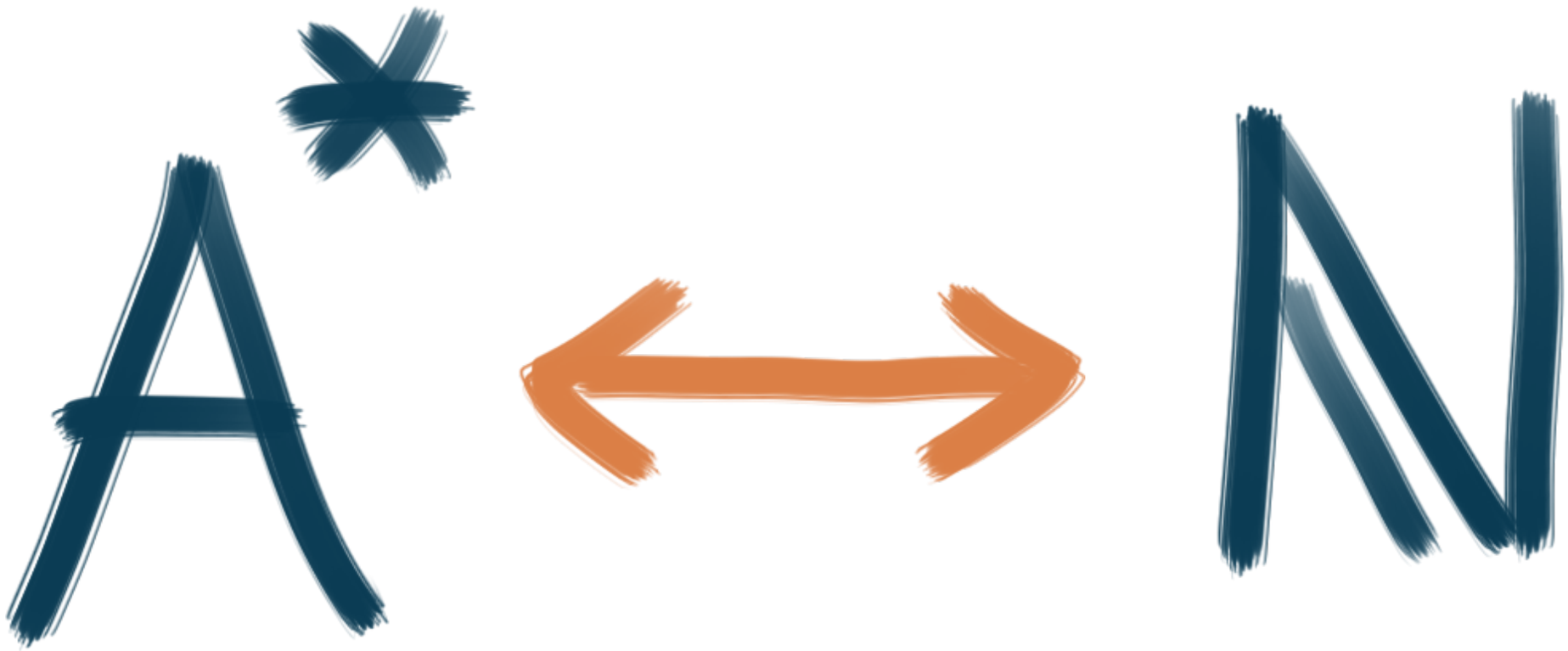
1,618,091,420

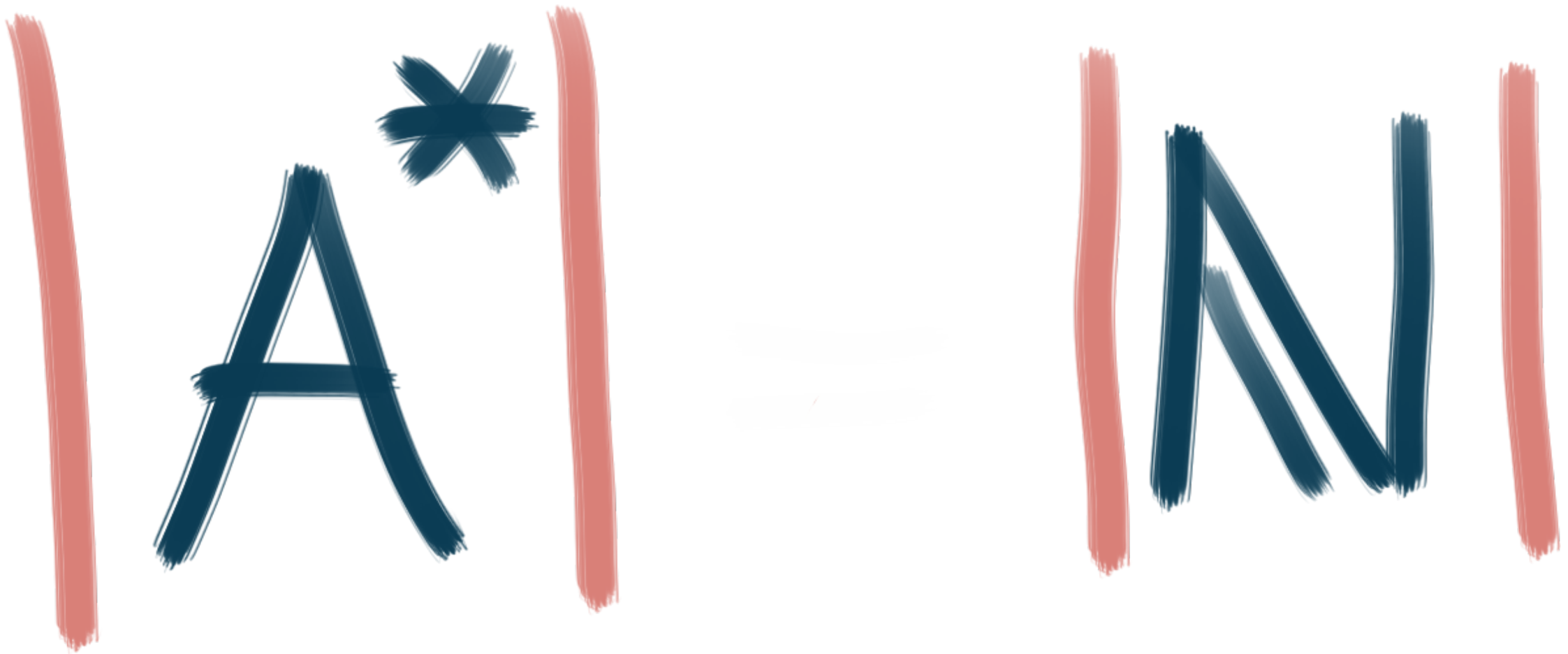
351 → "351"

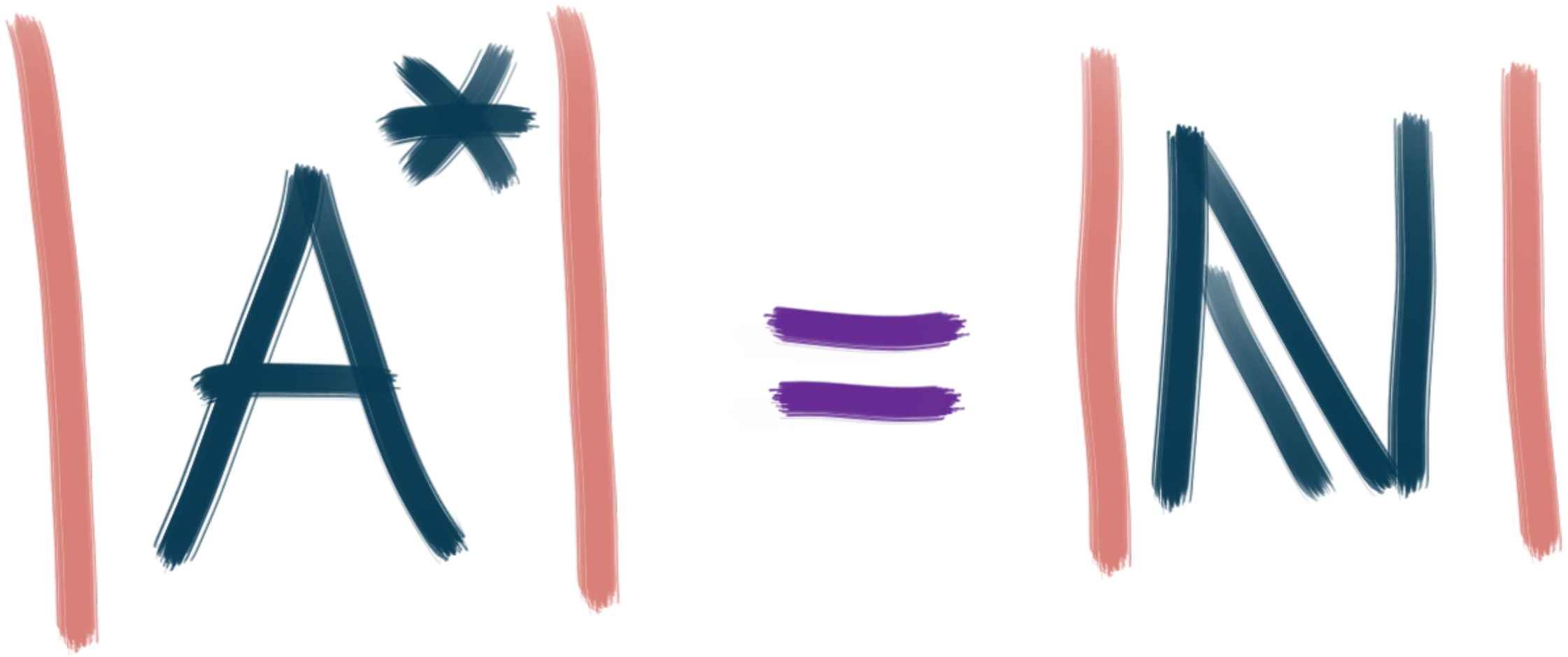

351
"351"

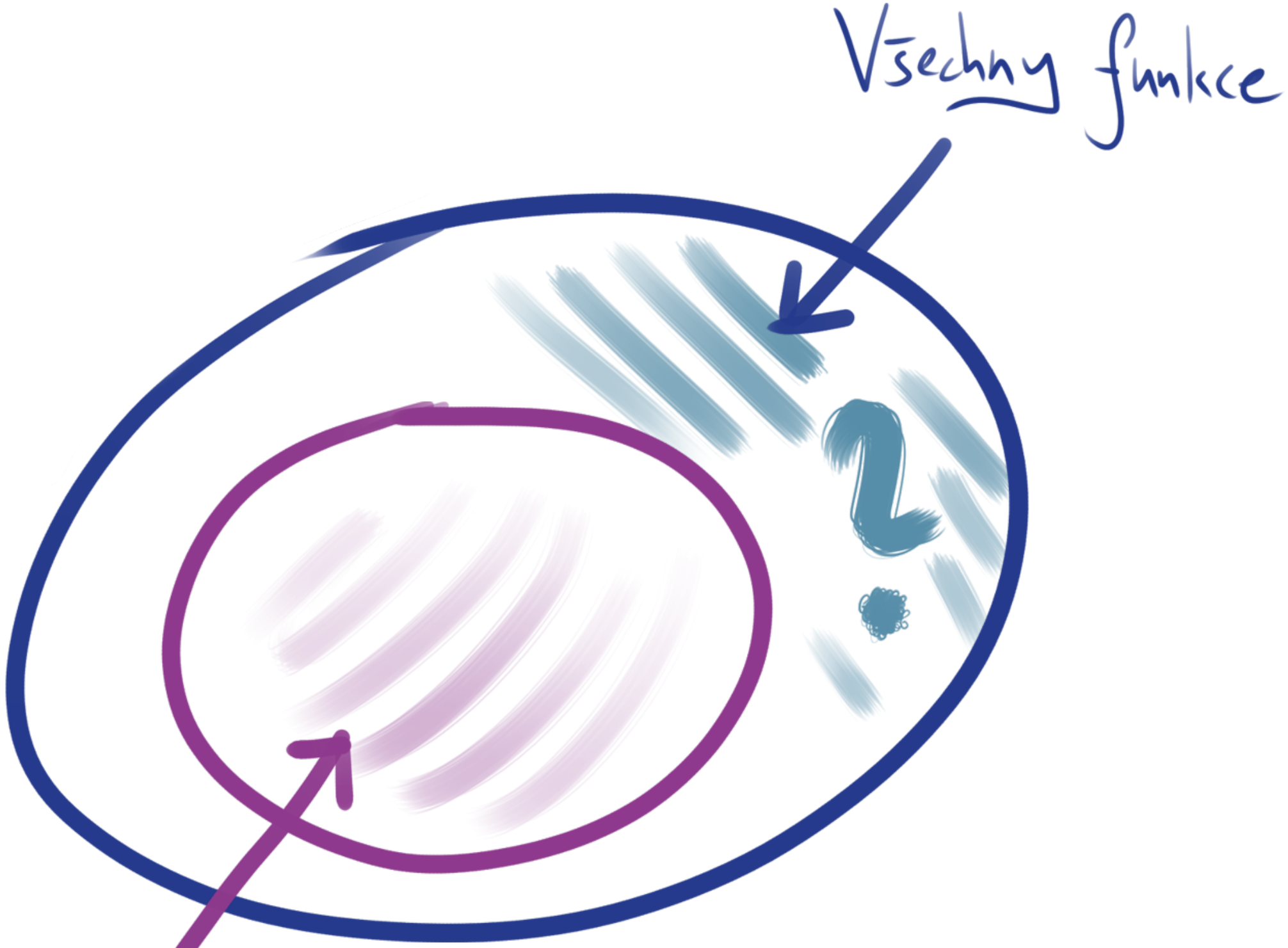









A*
=
N


Všechny funkce
Naprogramovatelné funkce

$$\S : A \to B$$

$$f : A \to B$$

$$f' : \mathbb{N} \to \{0,1\}$$

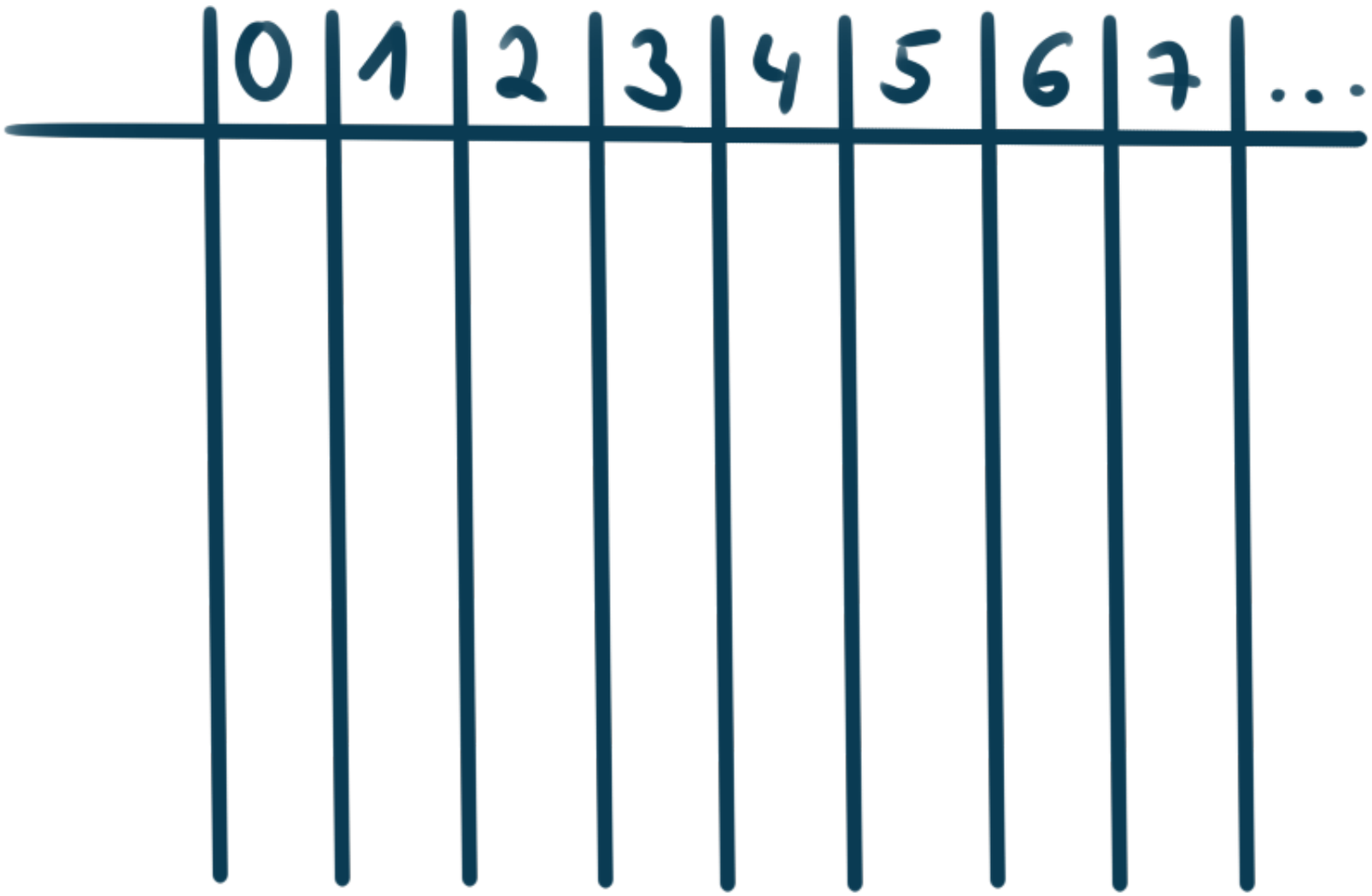
$$f : A \to B$$

$$f' : \mathbb{N} \to \{\times, \checkmark\}$$

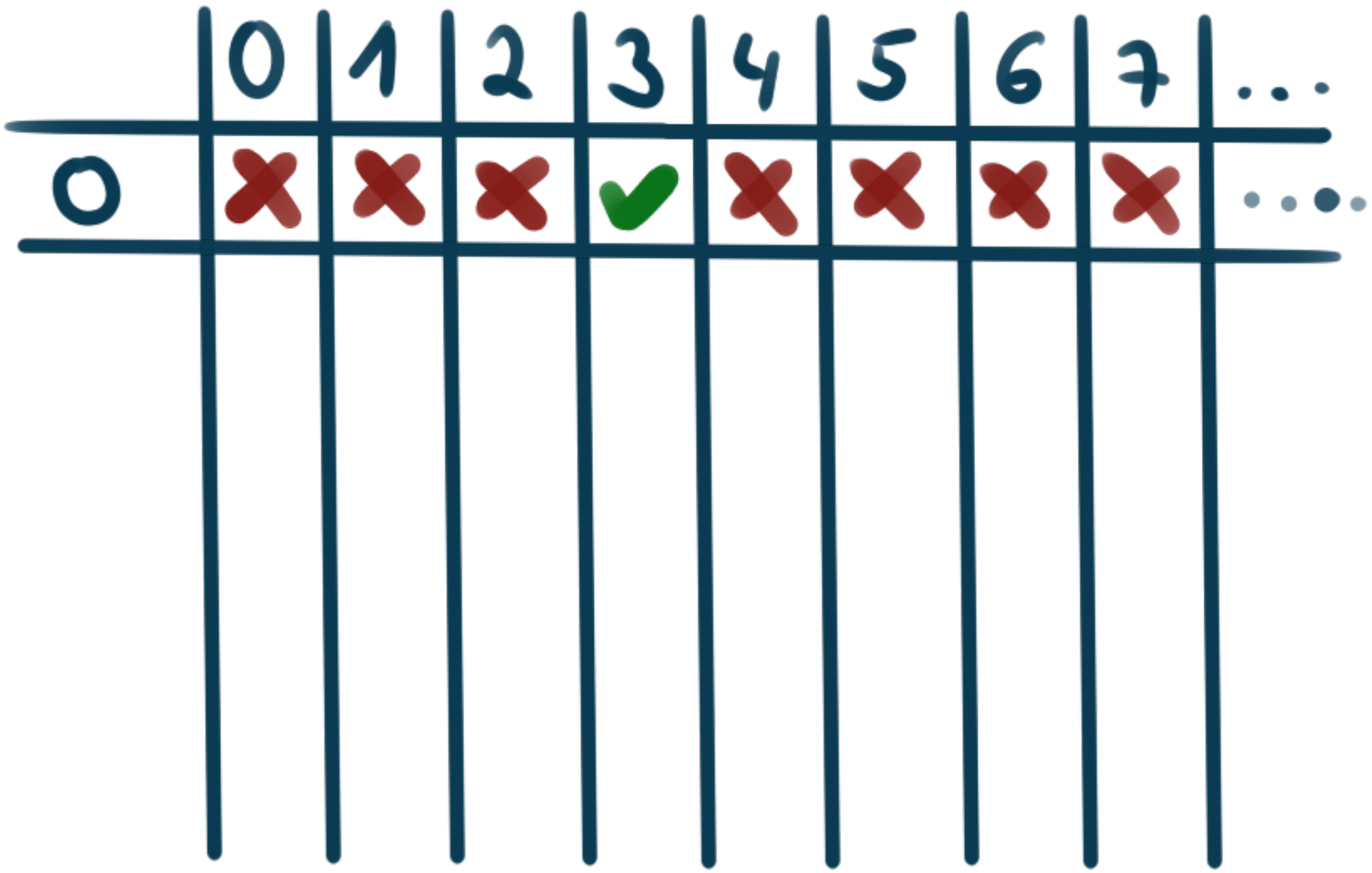
$$f : A \to B$$

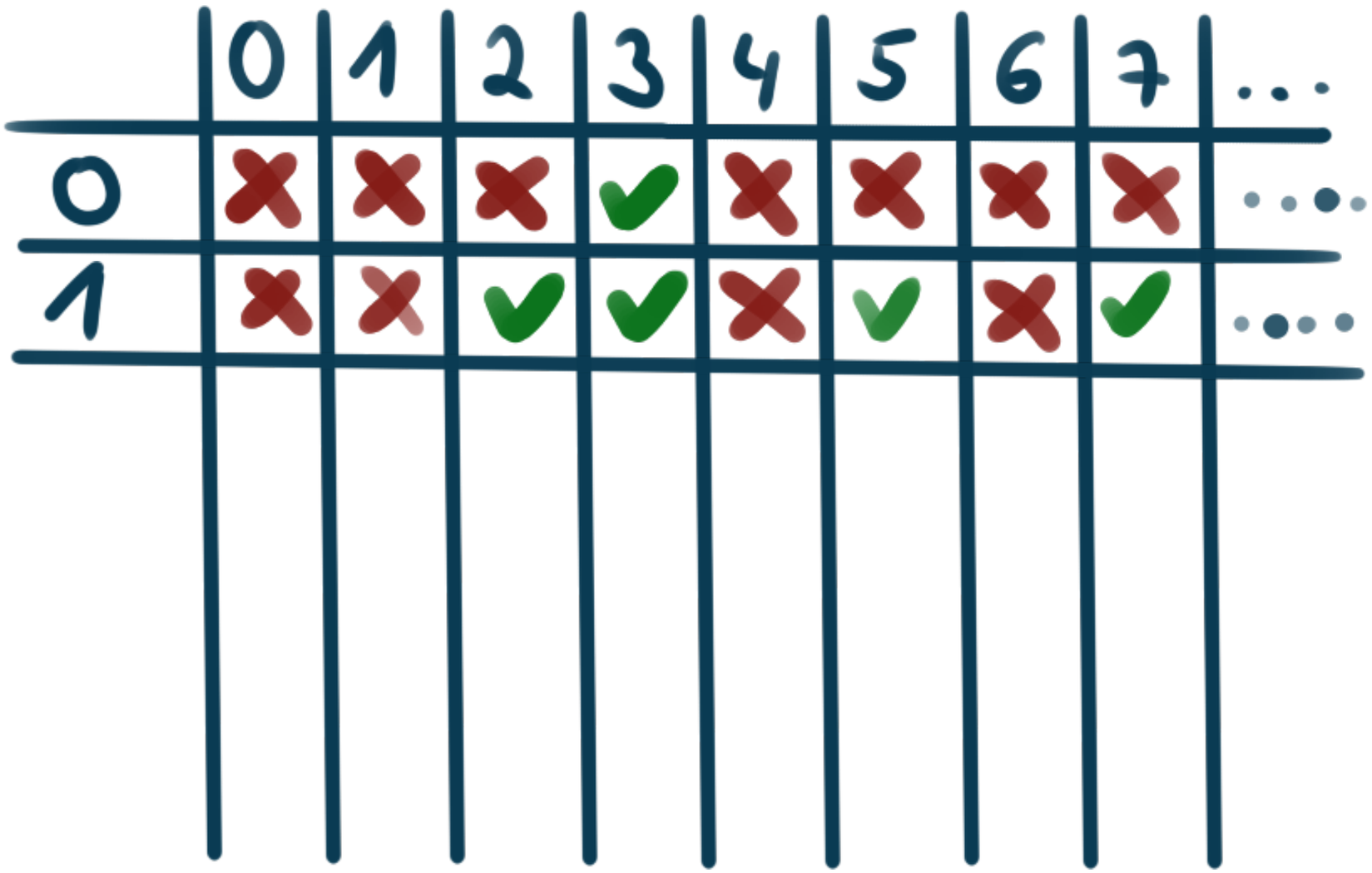
$$f' : \mathbb{N} \to \{\times, \checkmark\}$$

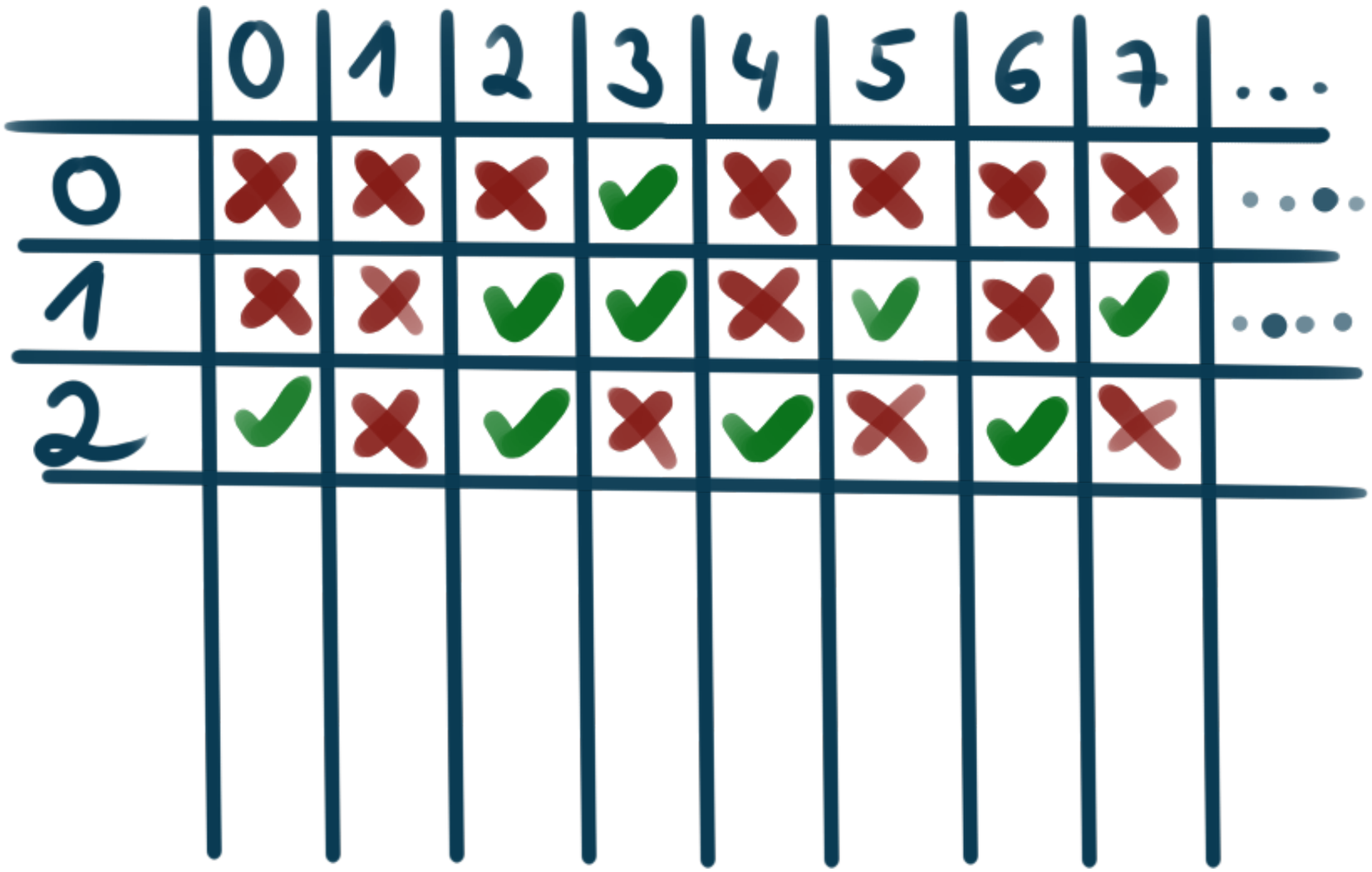
x	0	1	2	3	4	5	6	...
$f'(x)$	✓	×	✓	×	×	✓	×	...

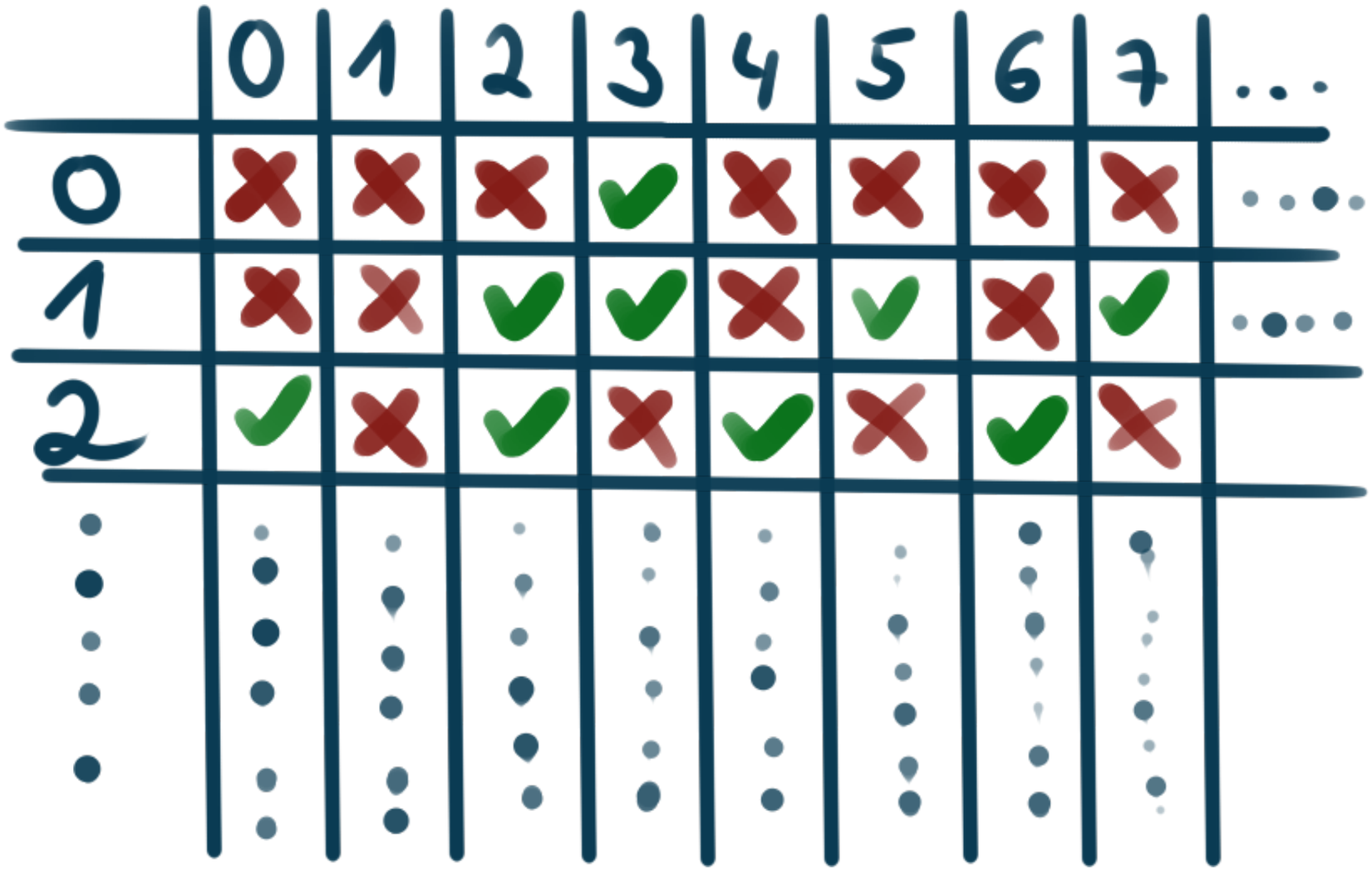

0
1
2
3
4
5
6
7
...

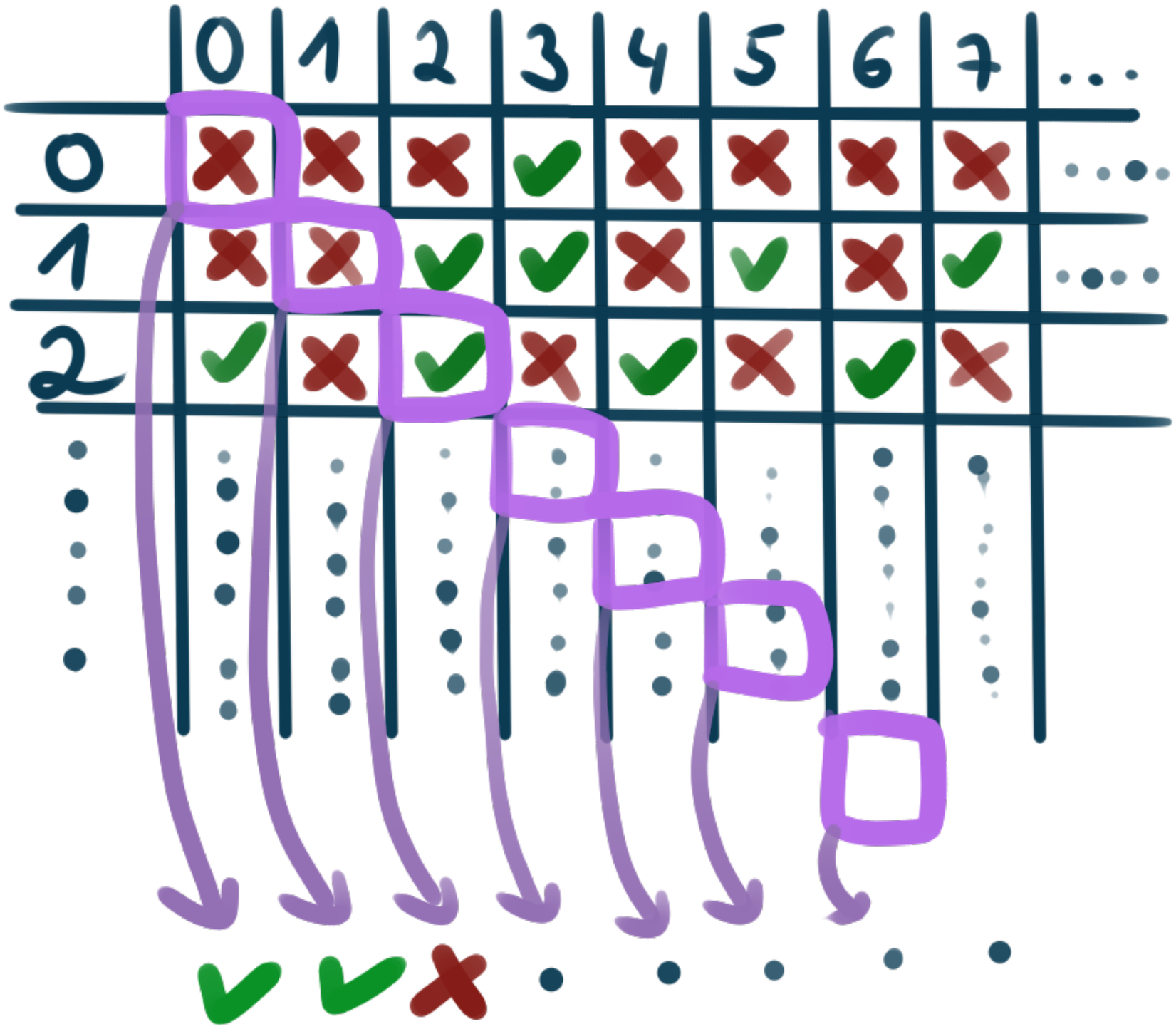
	0	1	2	3	4	5	6	7	...
0									

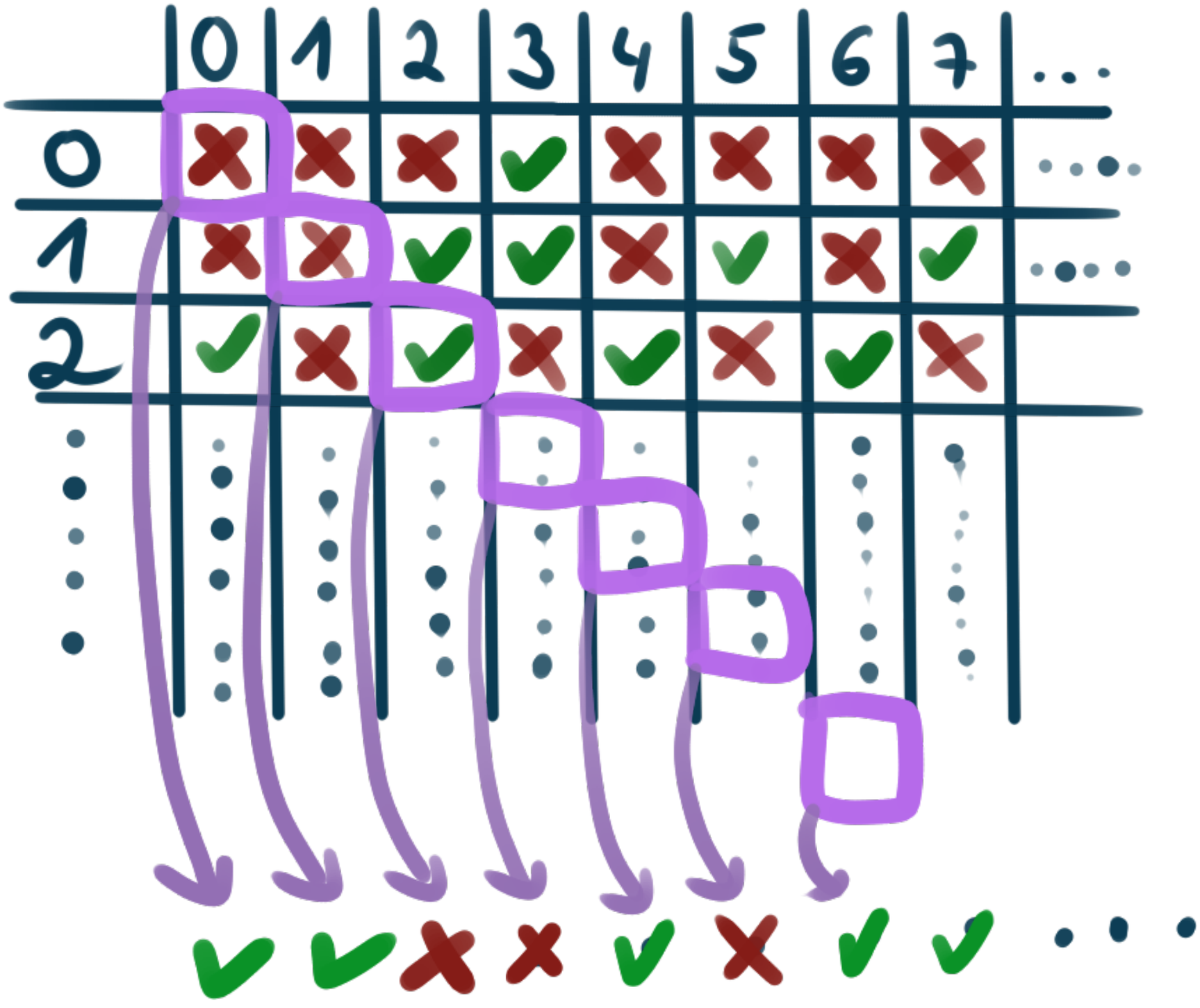

0 1 2 3 4 5 6 7 ...
0
...

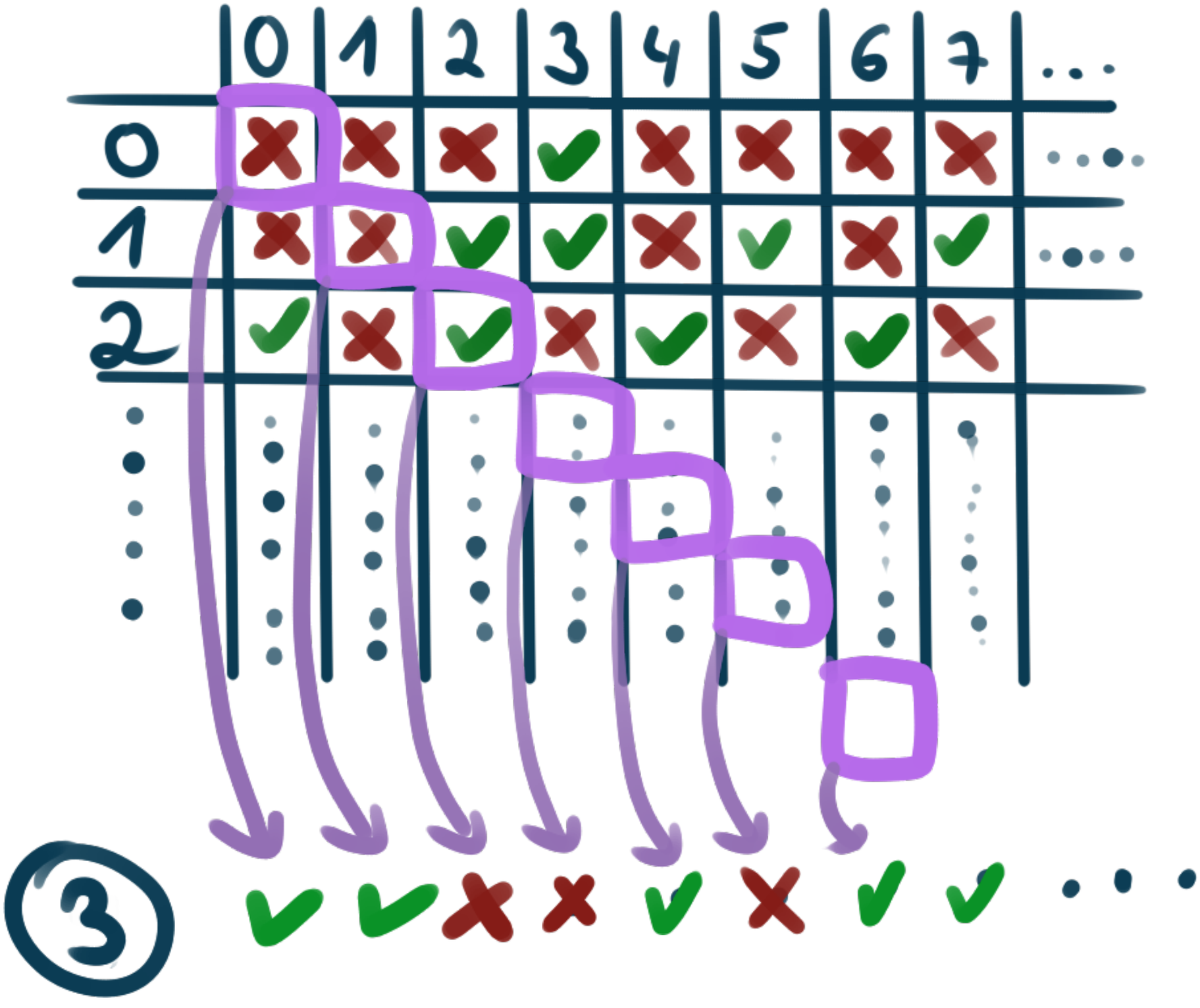

0 1 2 3 4 5 6 7 ...
0
1

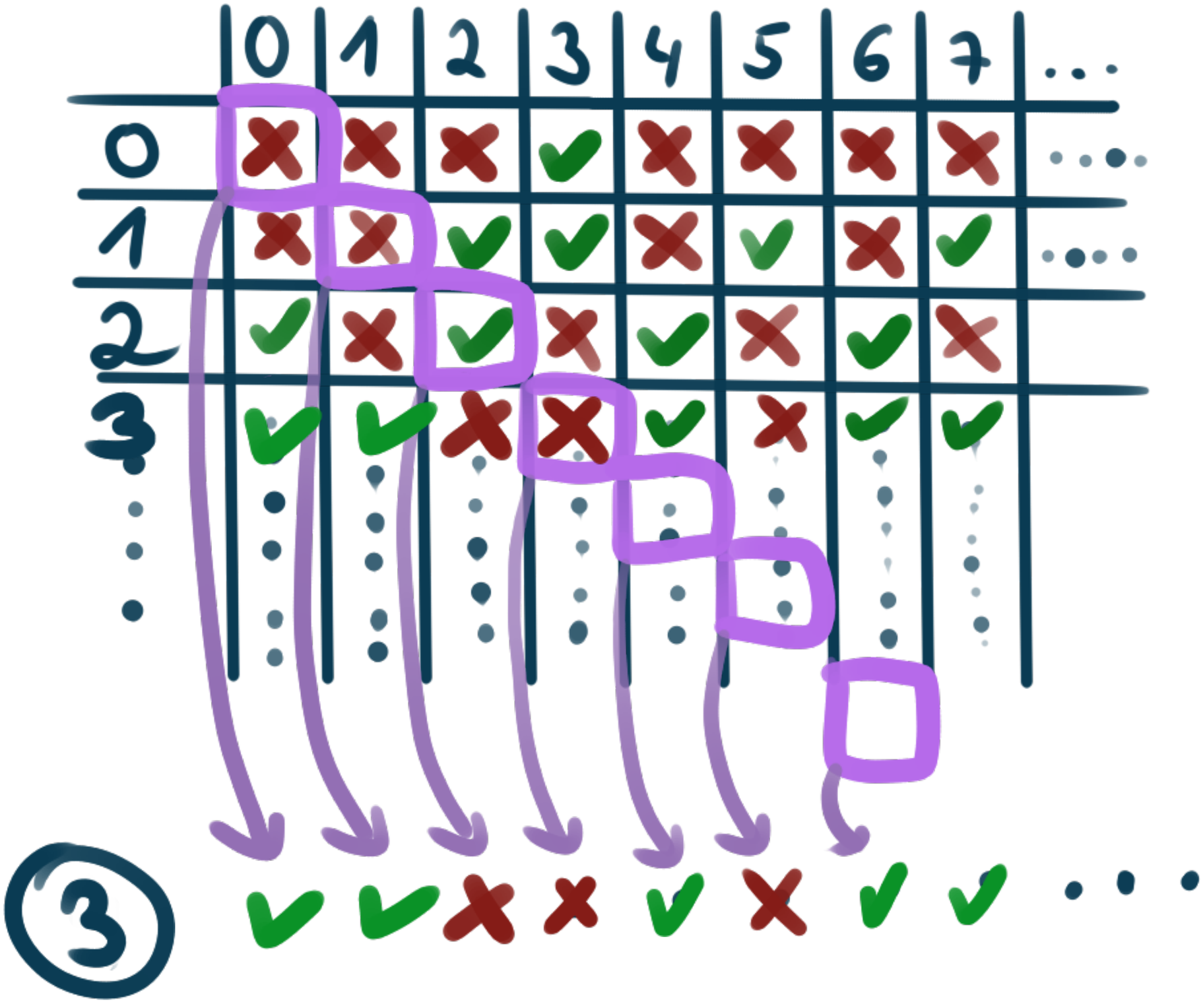

0 1 2 3 4 5 6 7 ...
0
1
2


0 1 2 3 4 5 6 7 ...
0
1
2


0
1
2
3
4
5
6
7
...
0
1
2


0
1
2
3
4
5
6
7
...
0
1
2

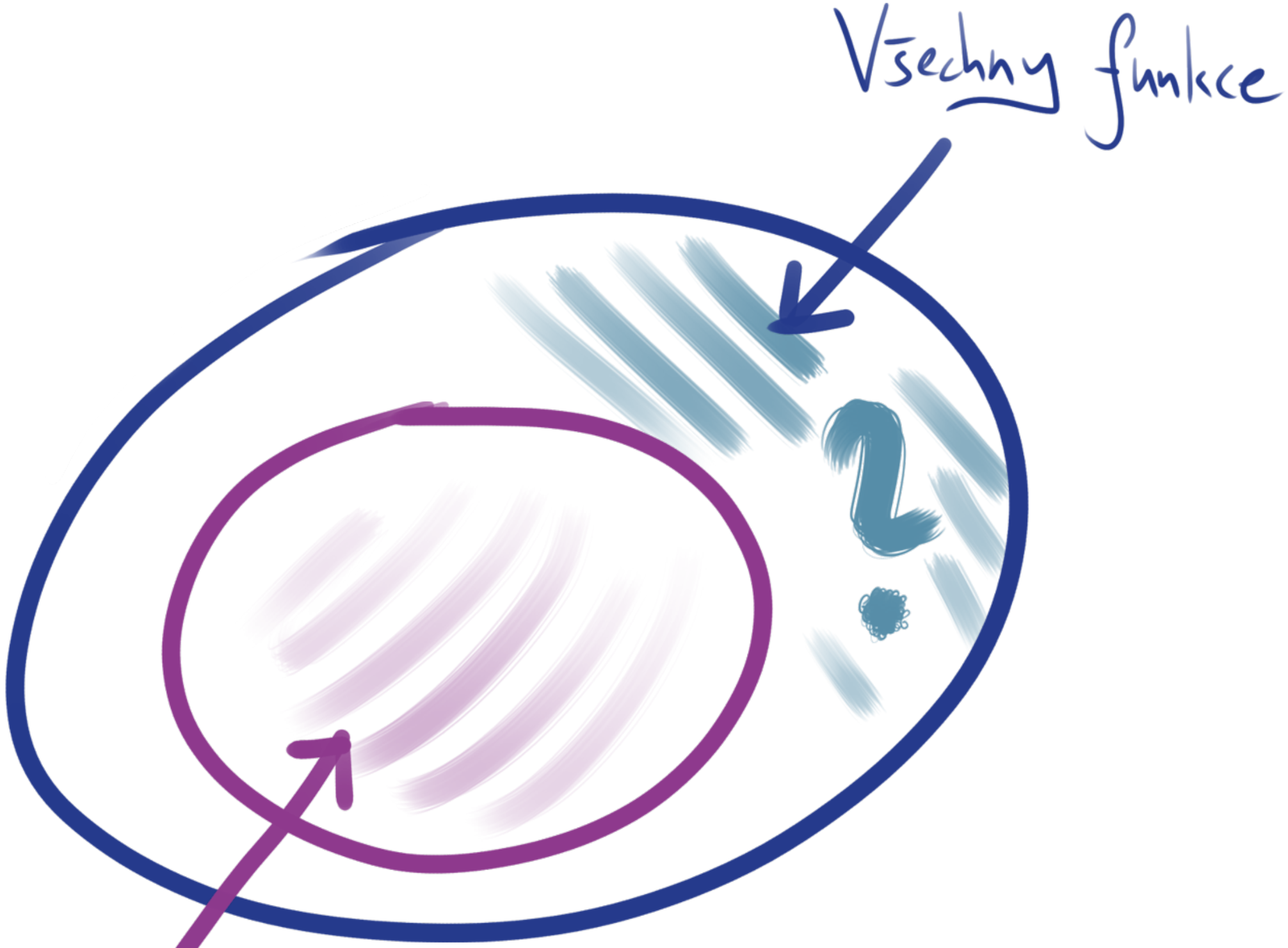

0 1 2 3 4 5 6 7 ...
0
1
2
3


0 1 2 3 4 5 6 7 ...
0
1
2
3
3

$$|N| < |F|$$

$$|A^*| = |\mathbb{N}| < |F|$$

$$|P| \leq |A^*| = |\mathbb{N}| < |F|$$


Všechny funkce
Naprogramovatelné funkce

```
def muj_program(x):
    x = int(x)
    r = 1
    while x != 0:
        r *= x
        x -= 1
    return r
```

```
program = """
    def muj_program(x):
        x = int(x)
        r = 1
        while x != 0:
            r *= x
            x -= 1
        return r
"""
```

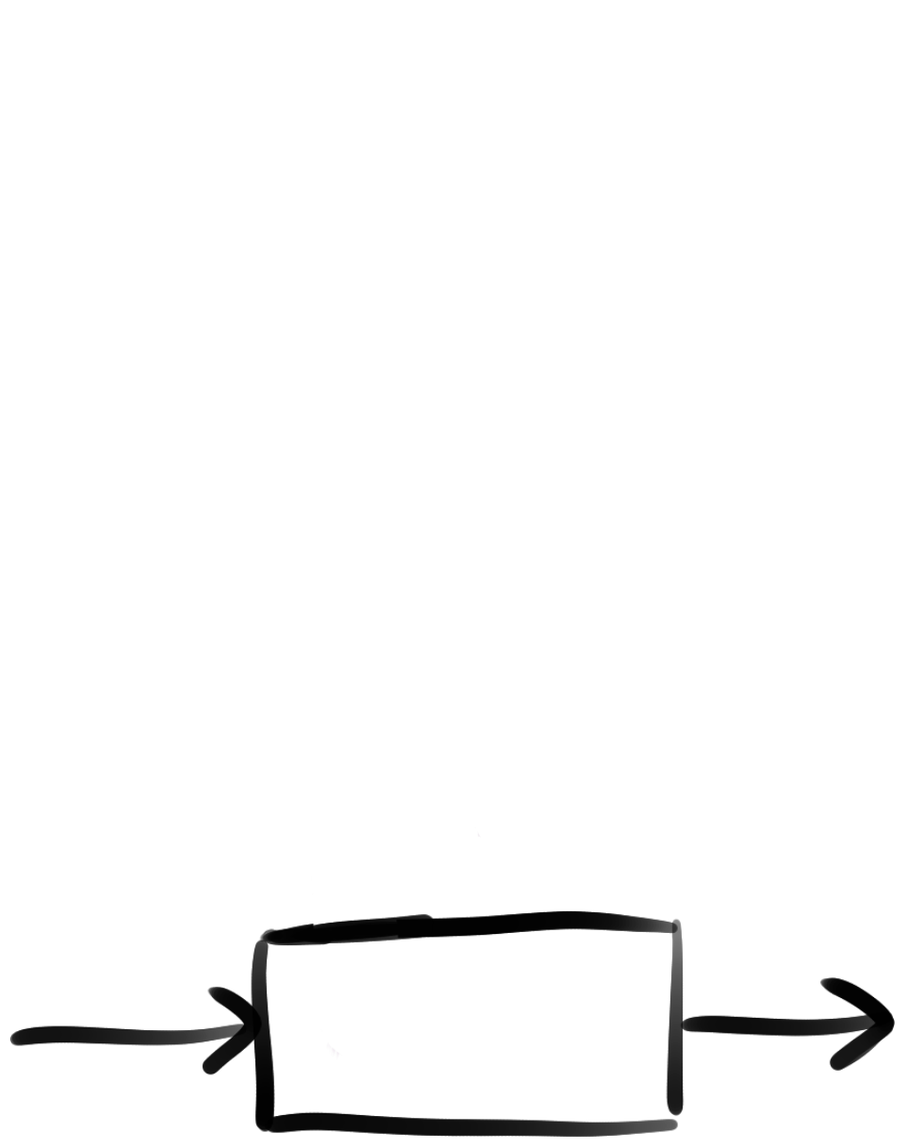
```
program = """
    def muj_program(x):
        x = int(x)
        r = 1
        while x != 0:
            r *= x
            x -= 1
        return r
"""

zastavi_se(program, "3")   # Vrati: True
```

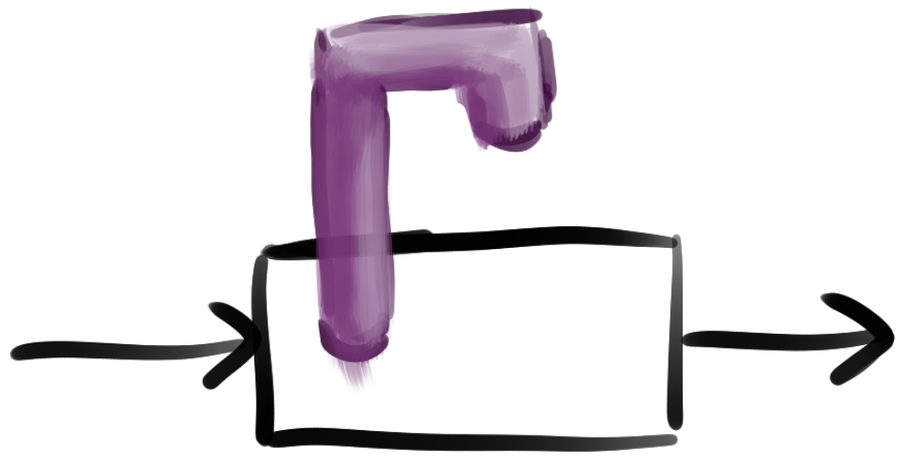
```
program = """
    def muj_program(x):
        x = int(x)
        r = 1
        while x != 0:
            r *= x
            x -= 1
        return r
"""

zastavi_se(program, "3")   # Vrati: True

zastavi_se(program, "-5")  # Vrati: False
```

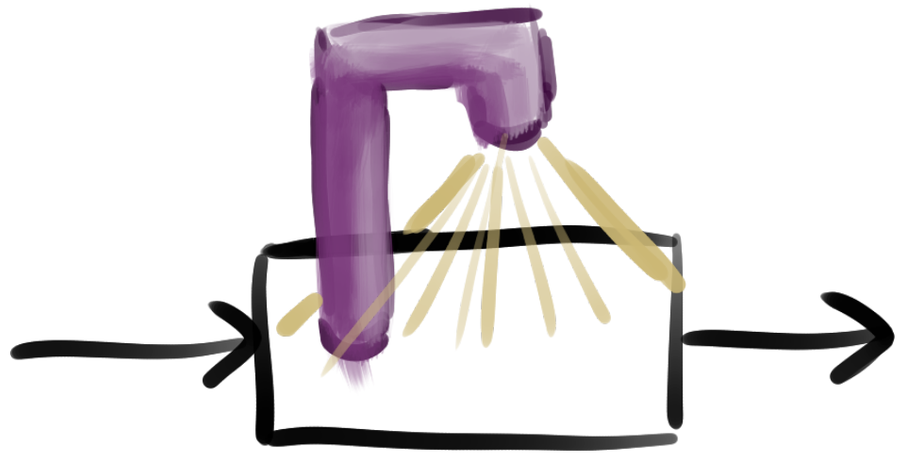
```
def zajimavy_program(p):
    if zastavi_se(p, p):
        while True:
            pass
    else:
        return
```

```python
def zajimavy_program(p):
    if zastavi_se(p, p):
        while True:
            pass
    else:
        return

program = """
    def zajimavy_program(p):
        if zastavi_se(p, p):
            while True:
                pass
        else:
            return
"""
```

```
def zajimavy_program(p):
    if zastavi_se(p, p):
        while True:
            pass
    else:
        return

program = """
    def zajimavy_program(p):
        if zastavi_se(p, p):
            while True:
                pass
        else:
            return
"""

zajimavy_program(program)
```

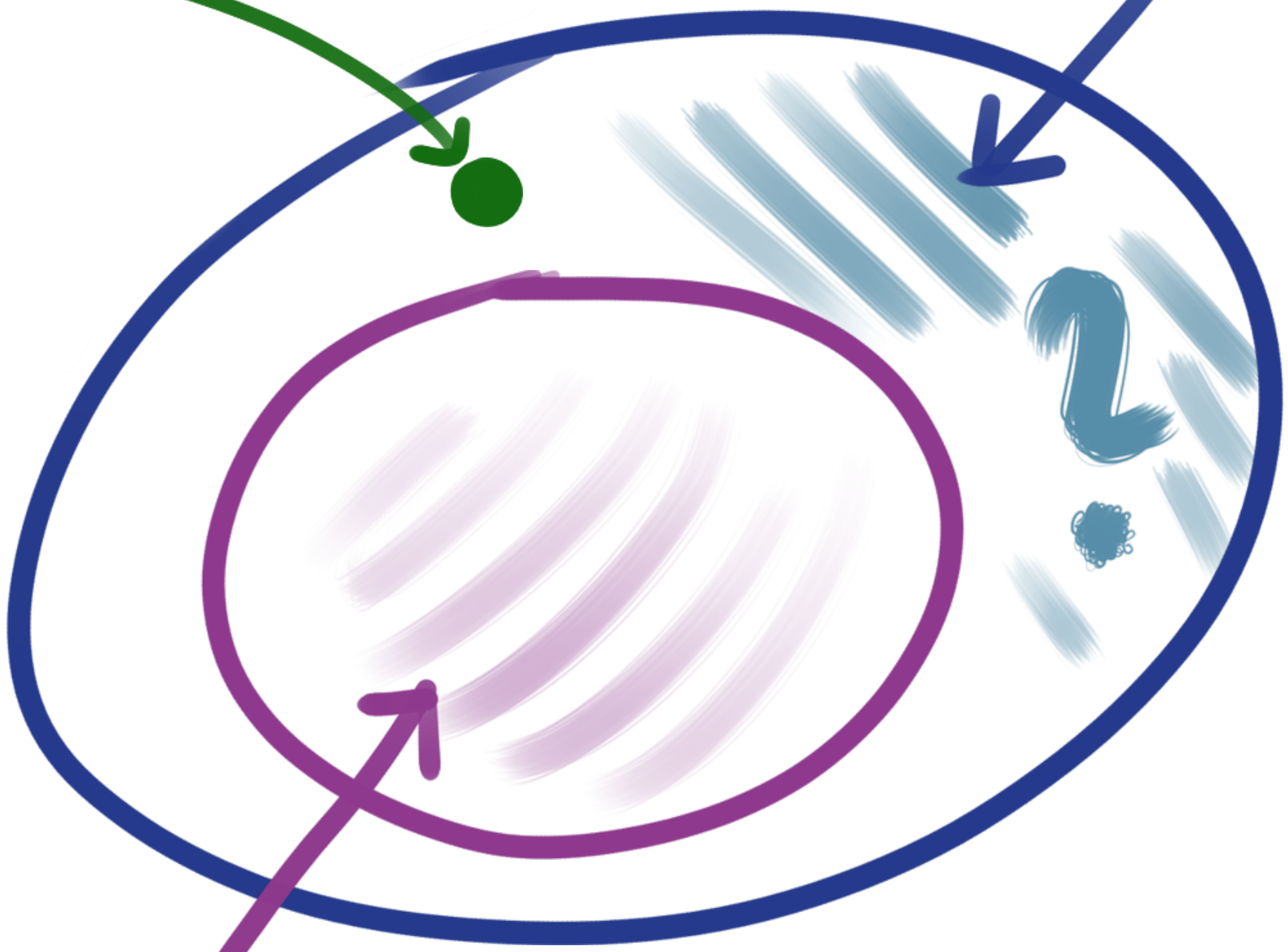






zastaví se

Všechny funkce



Naprogramovatelné funkce

```
def muj_program(x):
    x = int(x)
    r = 1
    while x != 0:
        r *= x
        x -= 1
    return r
```

```python
def muj_program(x):
    x = int(x)
    r = 1
    while x != 0:
        r *= x
        x -= 1
    return r
```

```python
def muj_program(x):
    x = int(x)
    r = 1
    while x != 0:
        r *= x
        x -= 1
    return r
```

x = 5

r = 320

$$6 = 3+2+1$$

$$28 = 14+7+4+2+1$$

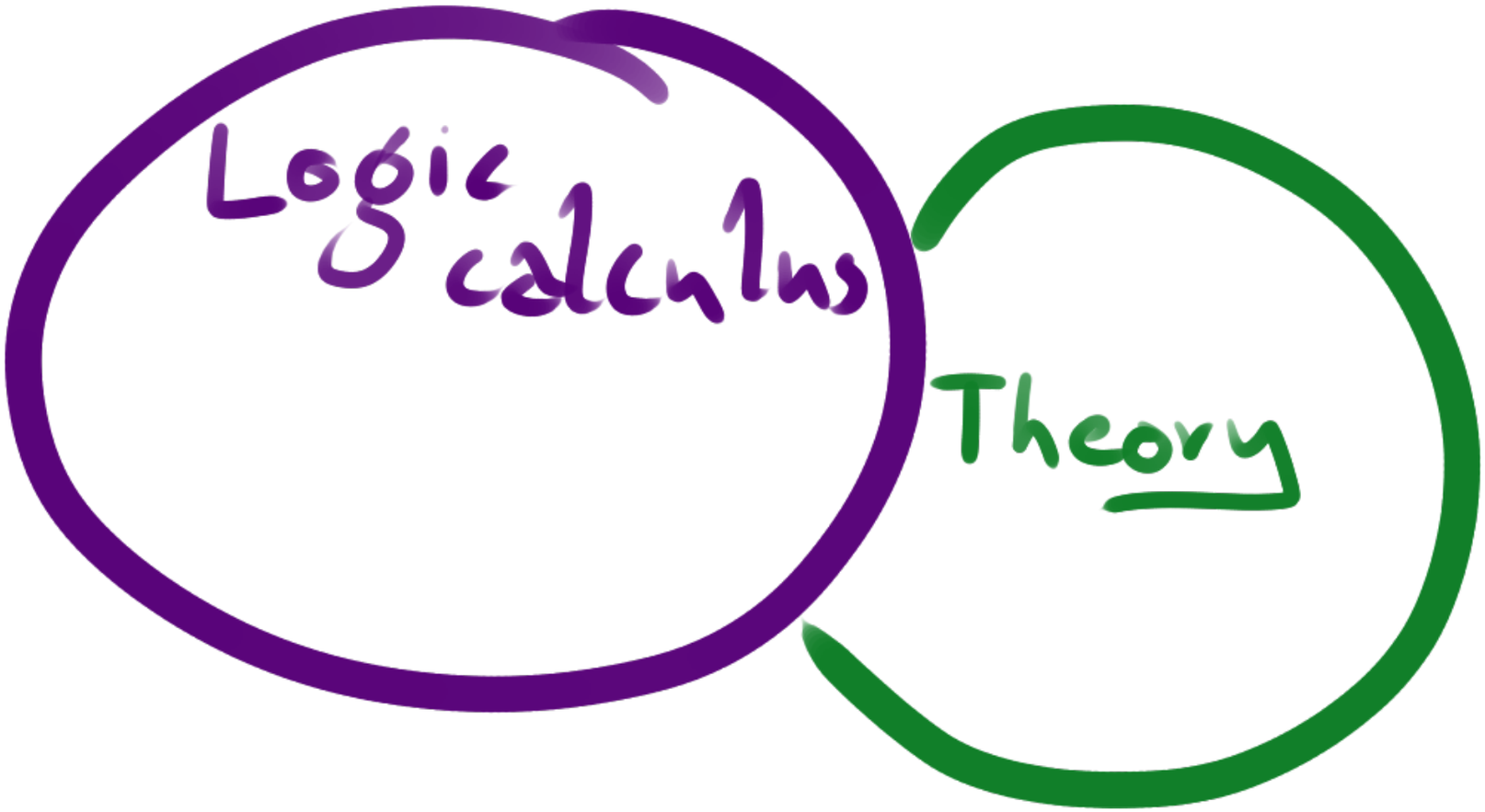
$$10 \neq 5+2+1$$

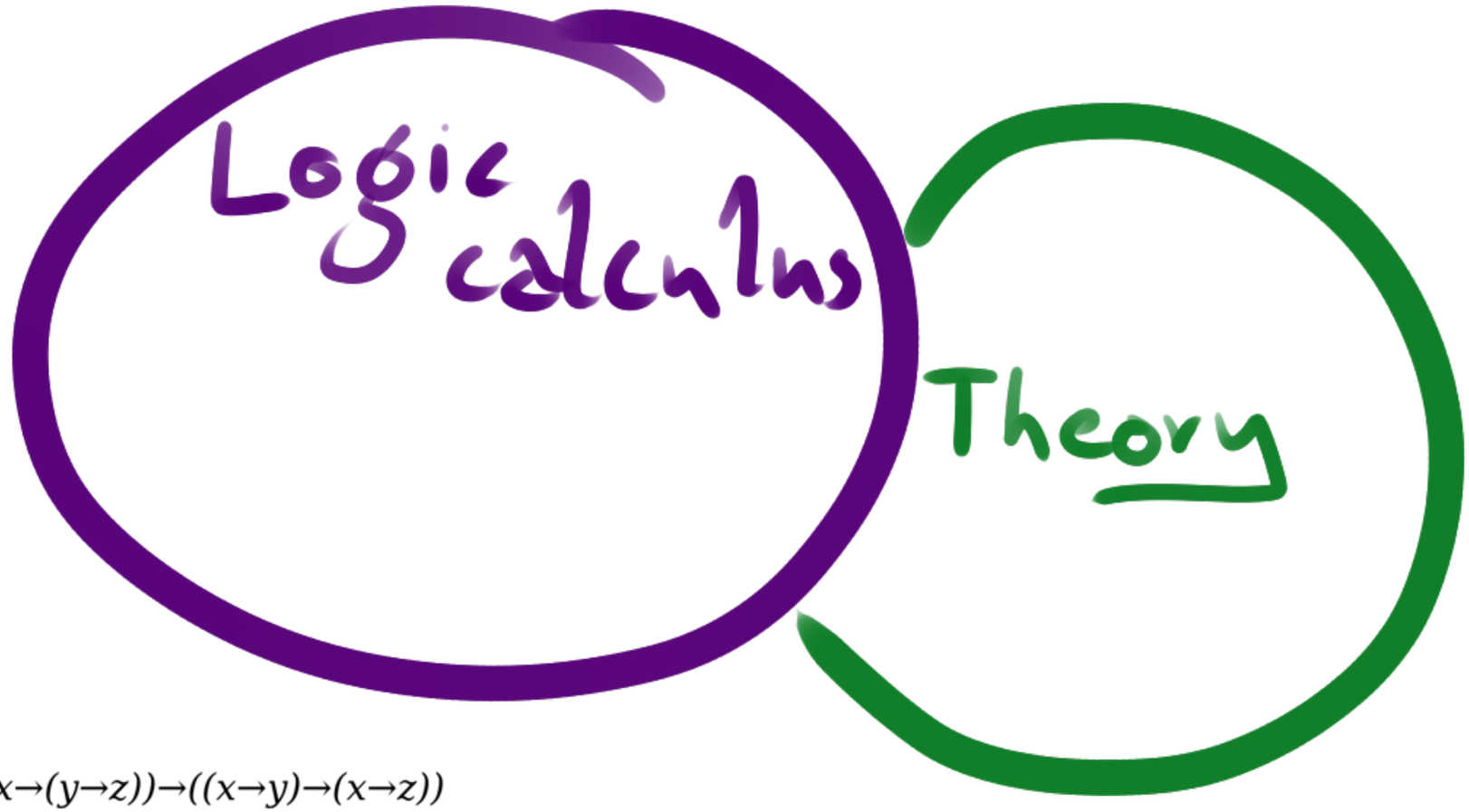
$$6 = 3 + 2 + 1$$

$$28 = 14 + 7 + 4 + 2 + 1$$

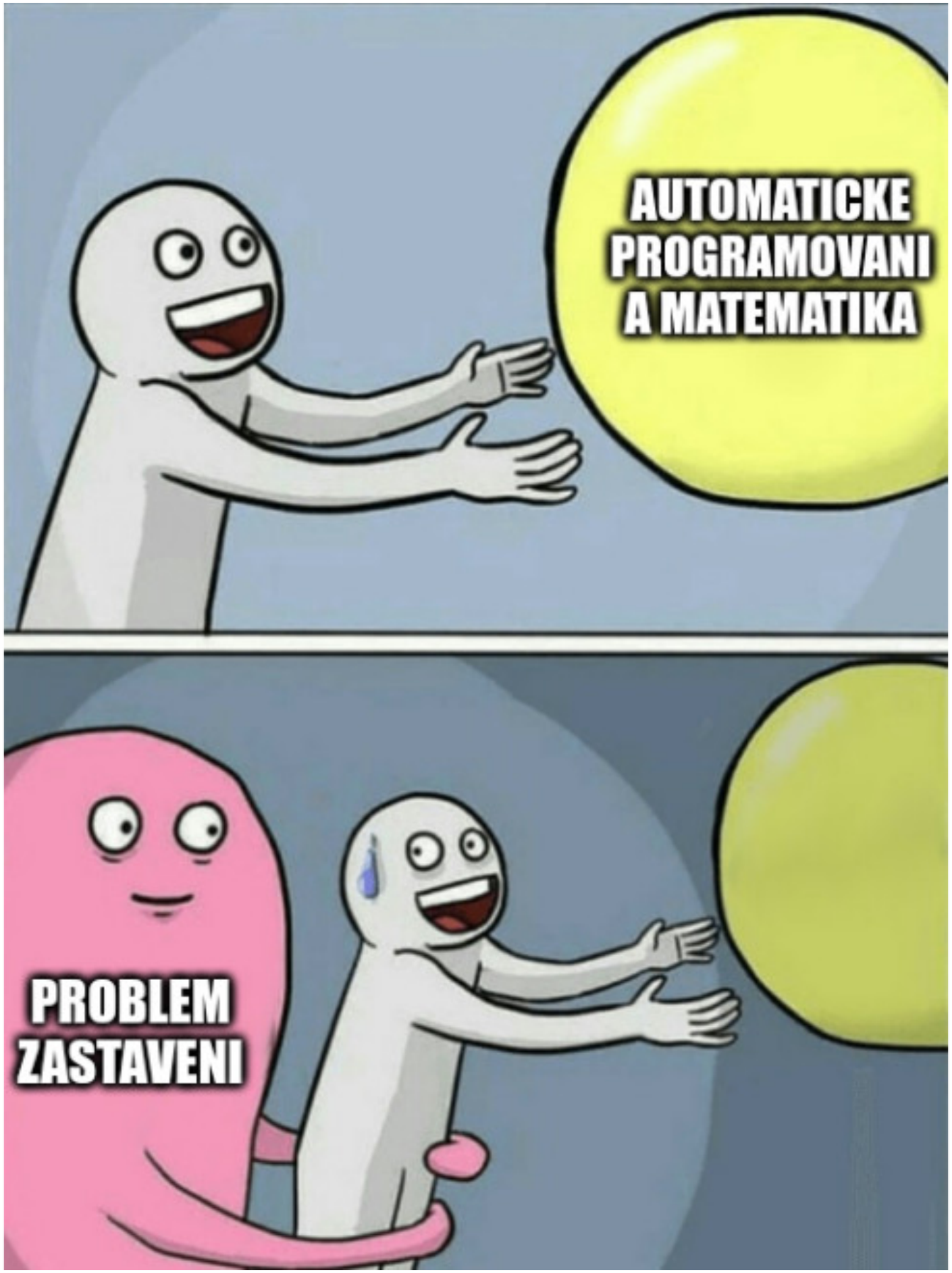
$$10 \neq 5 + 2 + 1$$

```
x = 1
while True:
    s = 0
    for y in range(1, x):
        if x % y == 0:
            s += y
    if s == x:
        break
    x += 2
```

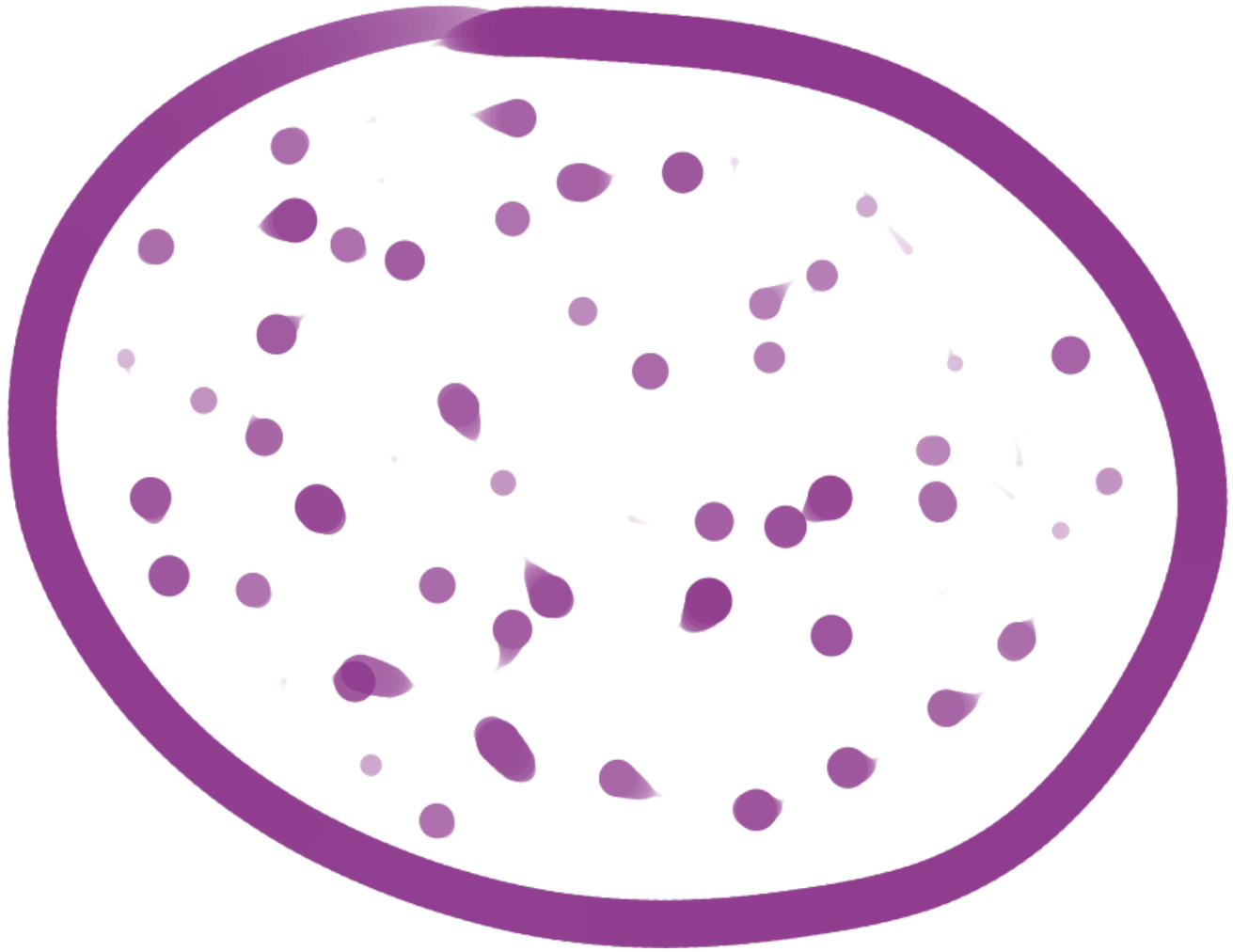

Logic calculus
Theory

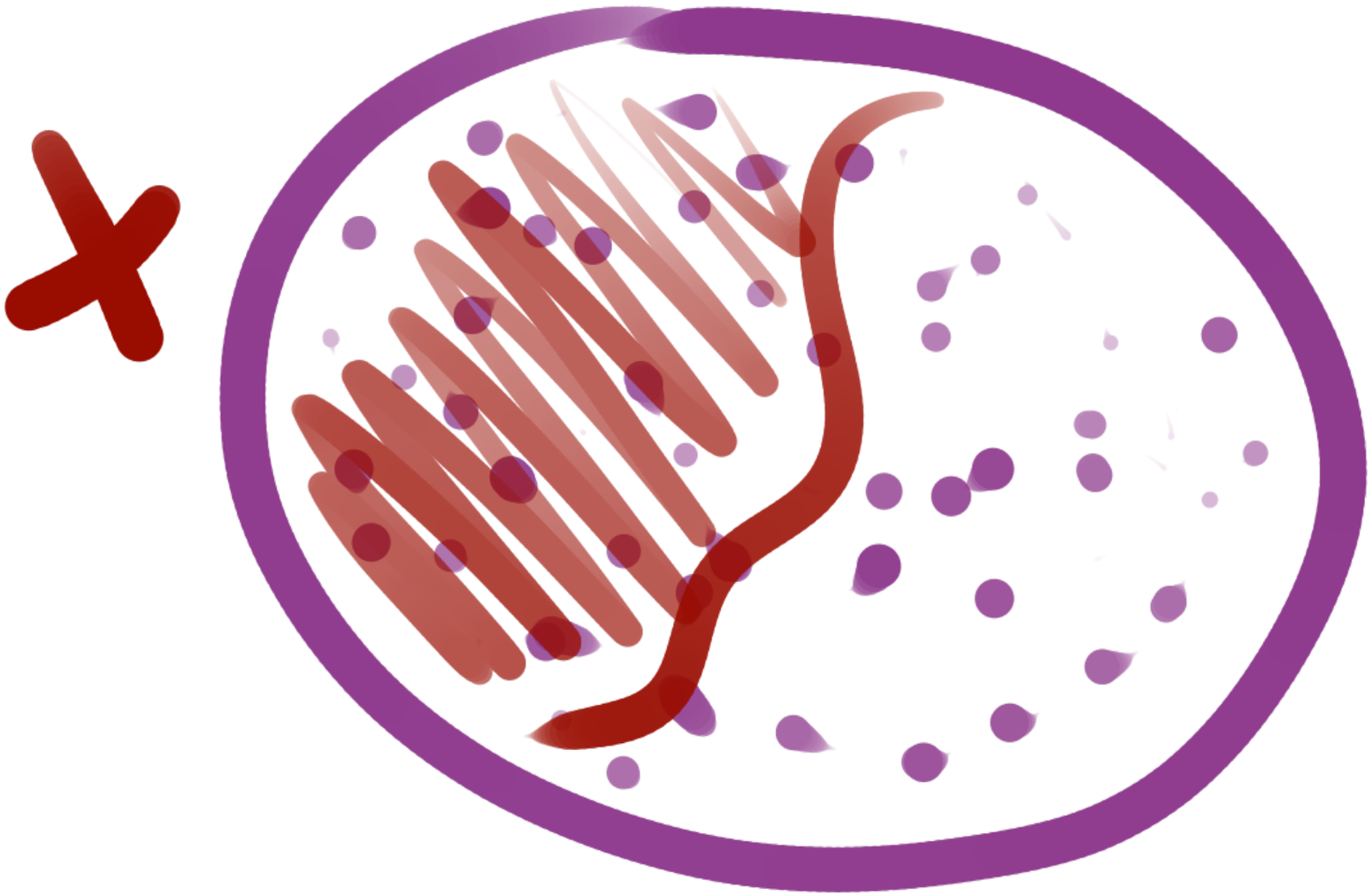


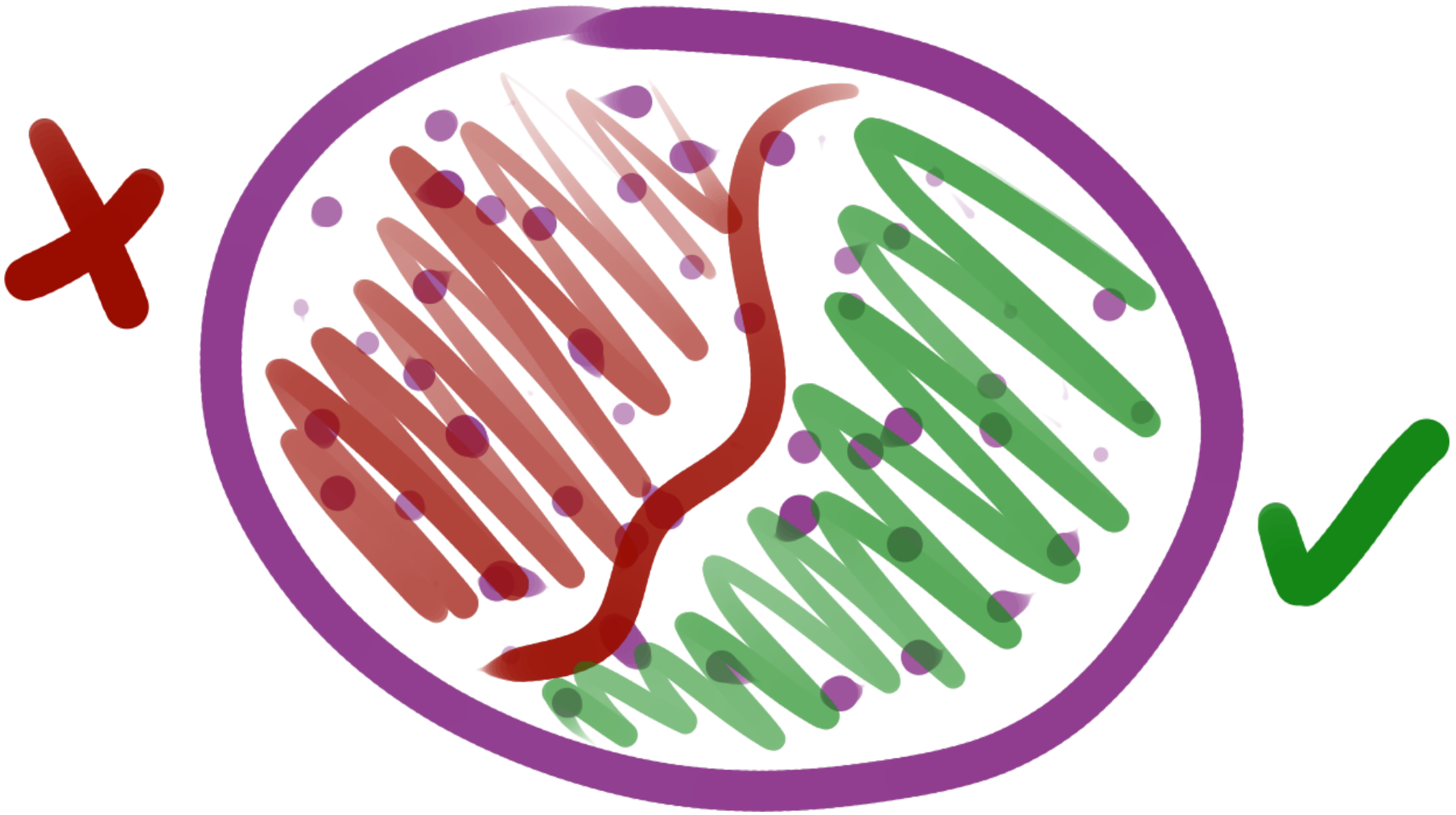
1. $(x \to (y \to z)) \to ((x \to y) \to (x \to z))$
2. $(w \to ((w \to w) \to w)) \to ((w \to (w \to w)) \to (w \to w))$
3. $x \to (y \to x)$
4. $w \to ((w \to w) \to w)$
5. $(w \to (w \to w)) \to (w \to w)$
6. $w \to (w \to w)$
7. $w \to w$
8. $x \to x$

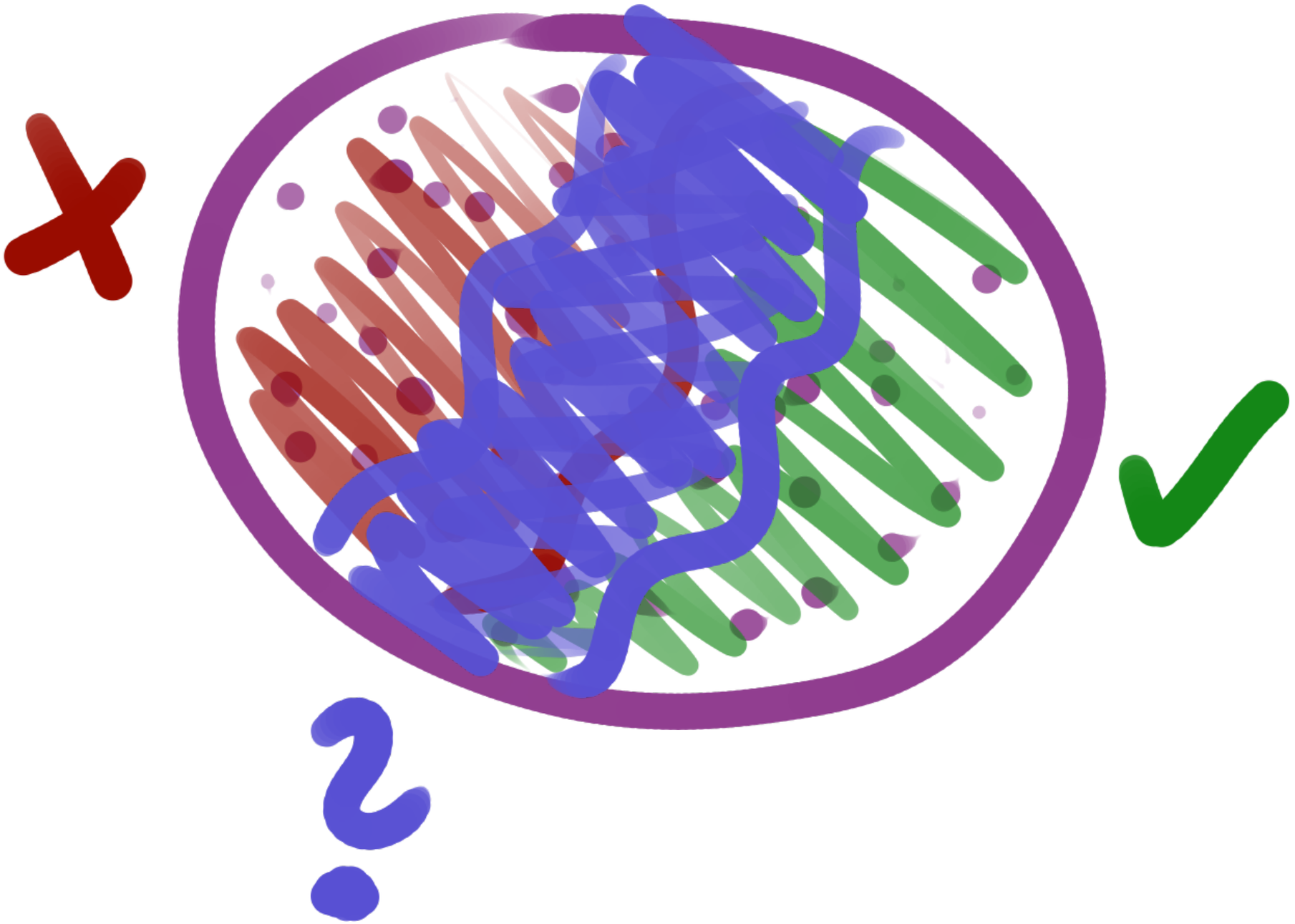

AUTOMATICKE
PROGRAMOVANI
A MATEMATIKA
PROBLEM
ZASTAVENI

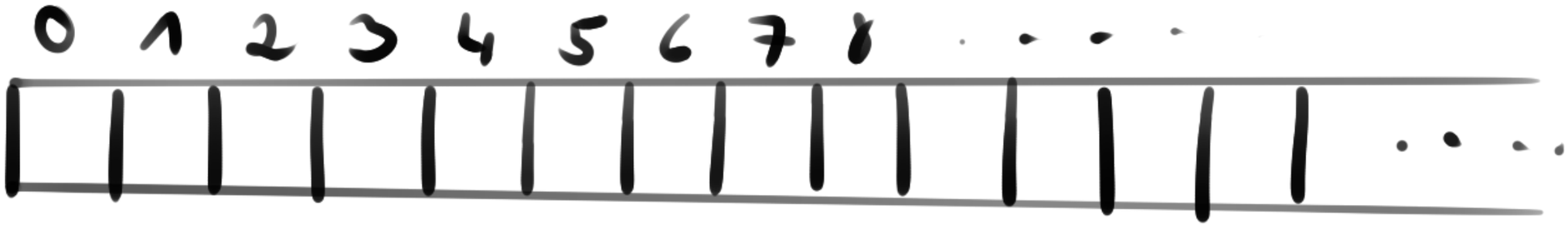

SOOO
WHAT NOW
makeameme.org



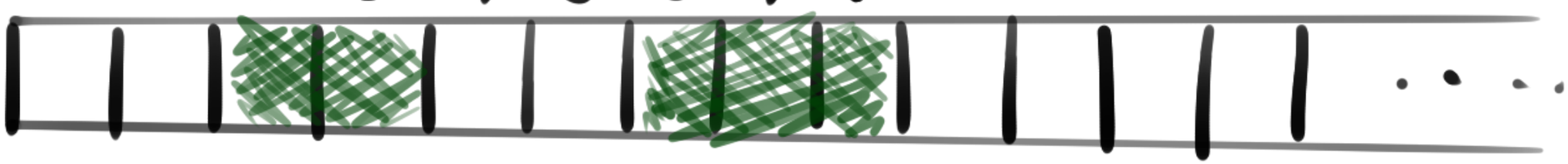


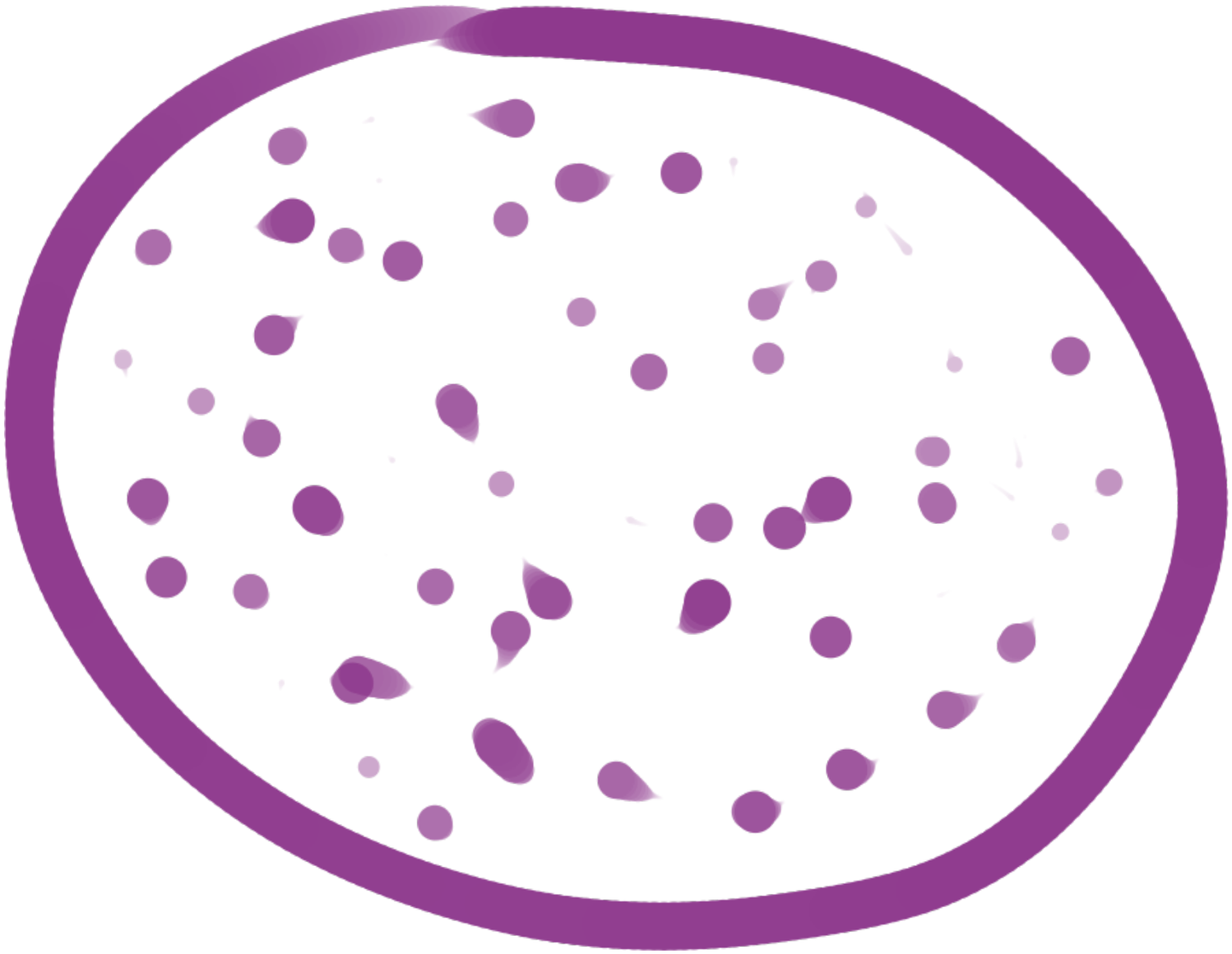


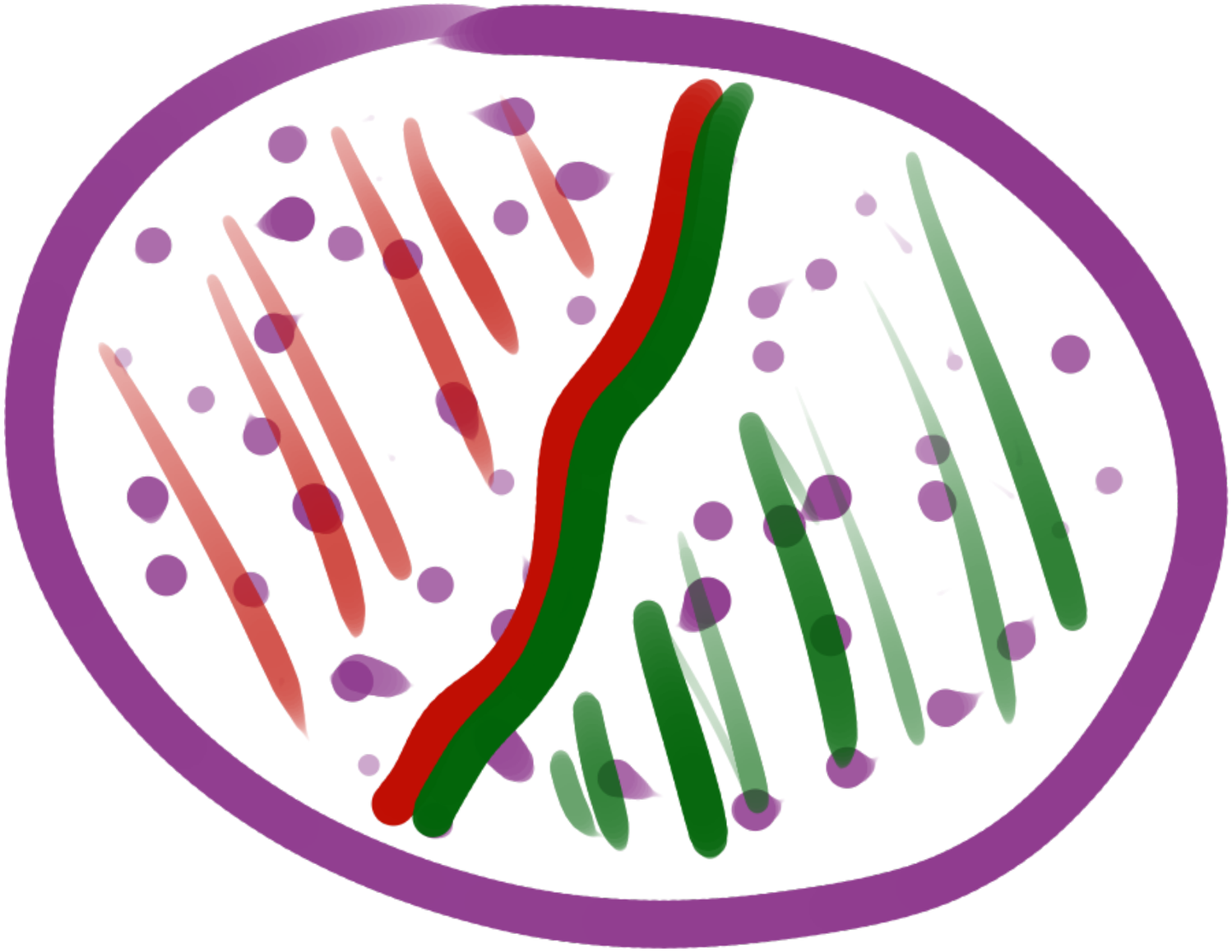


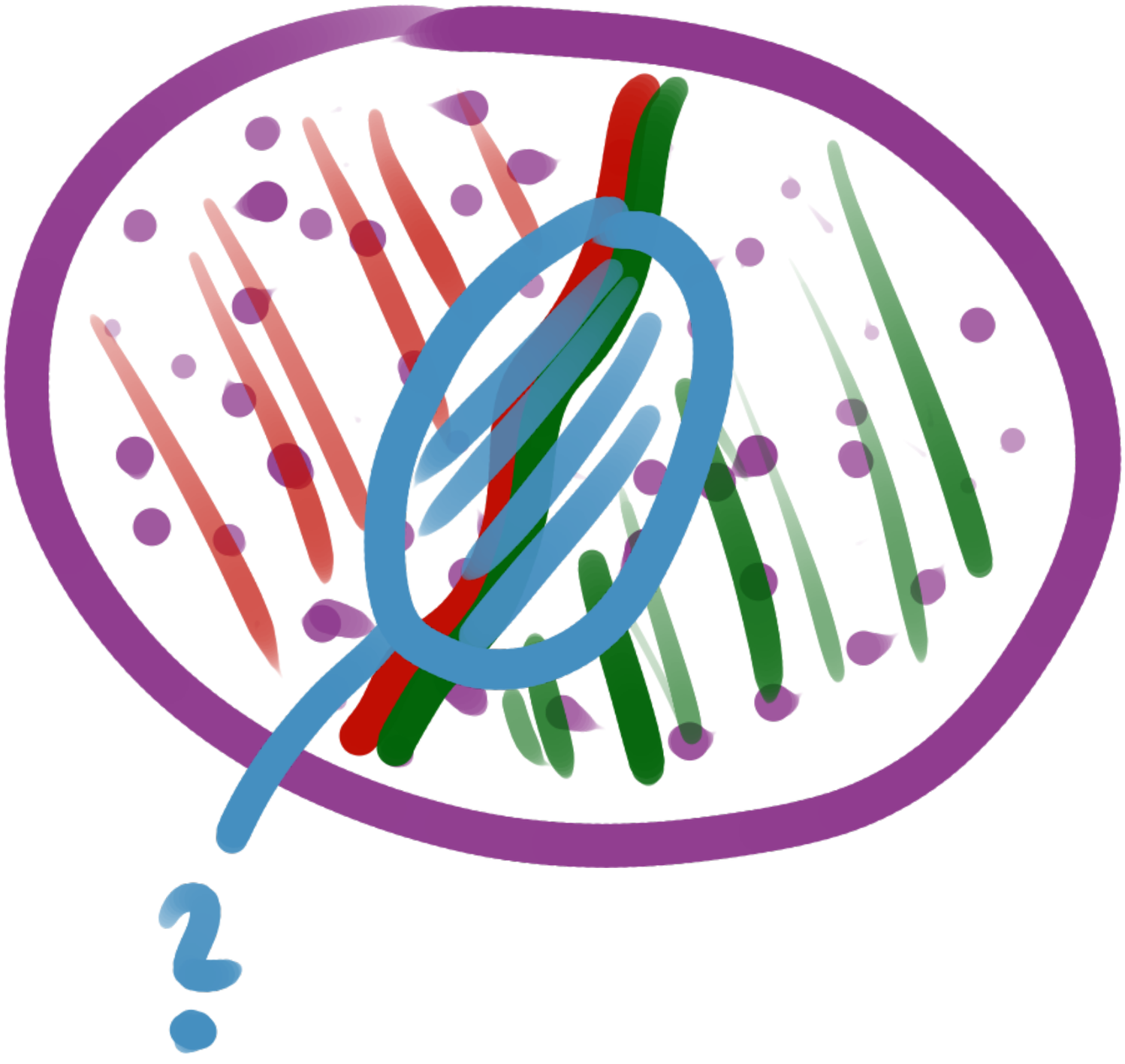


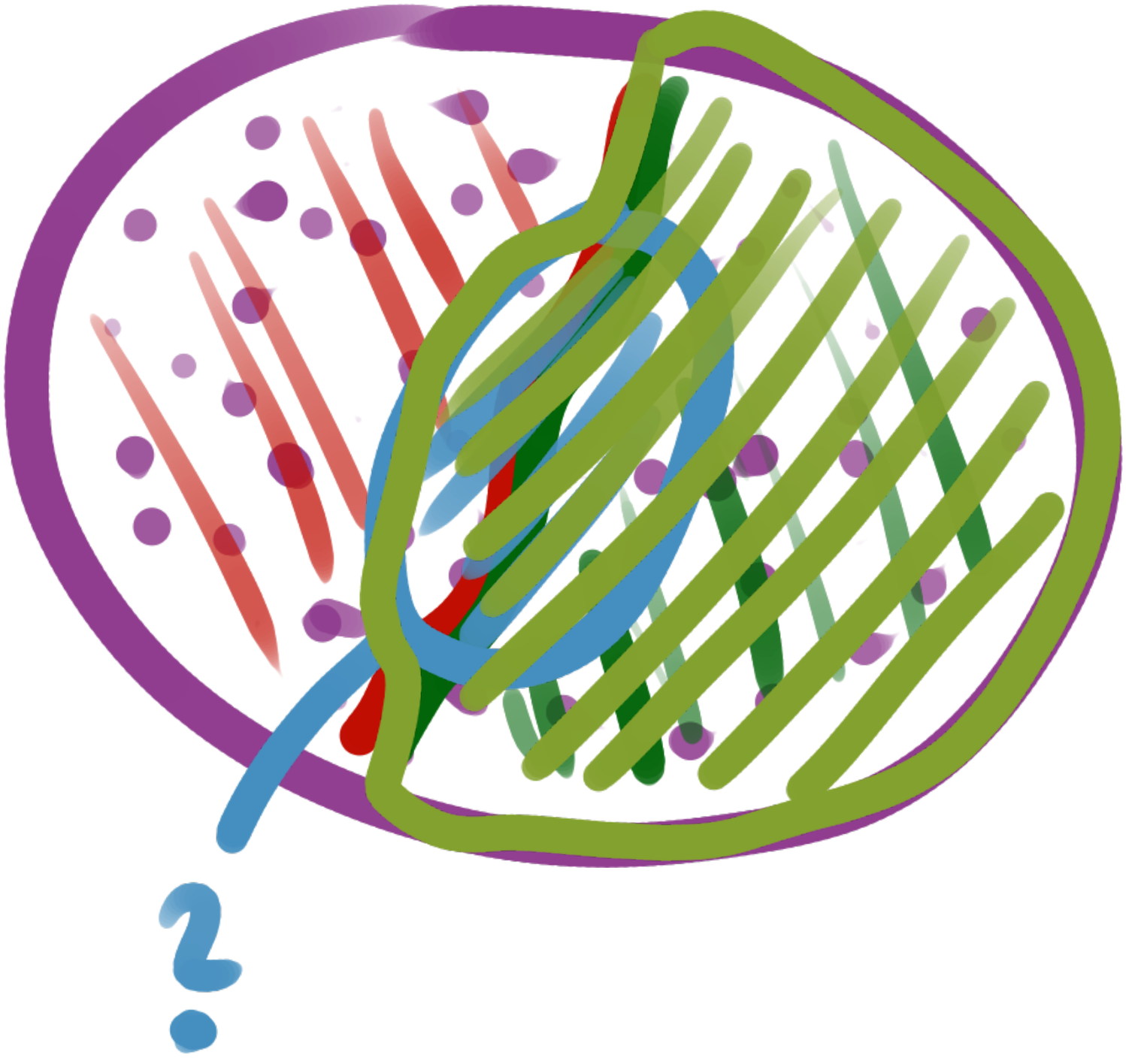
0 1 2 3 4 5 6 7 8 . . . .

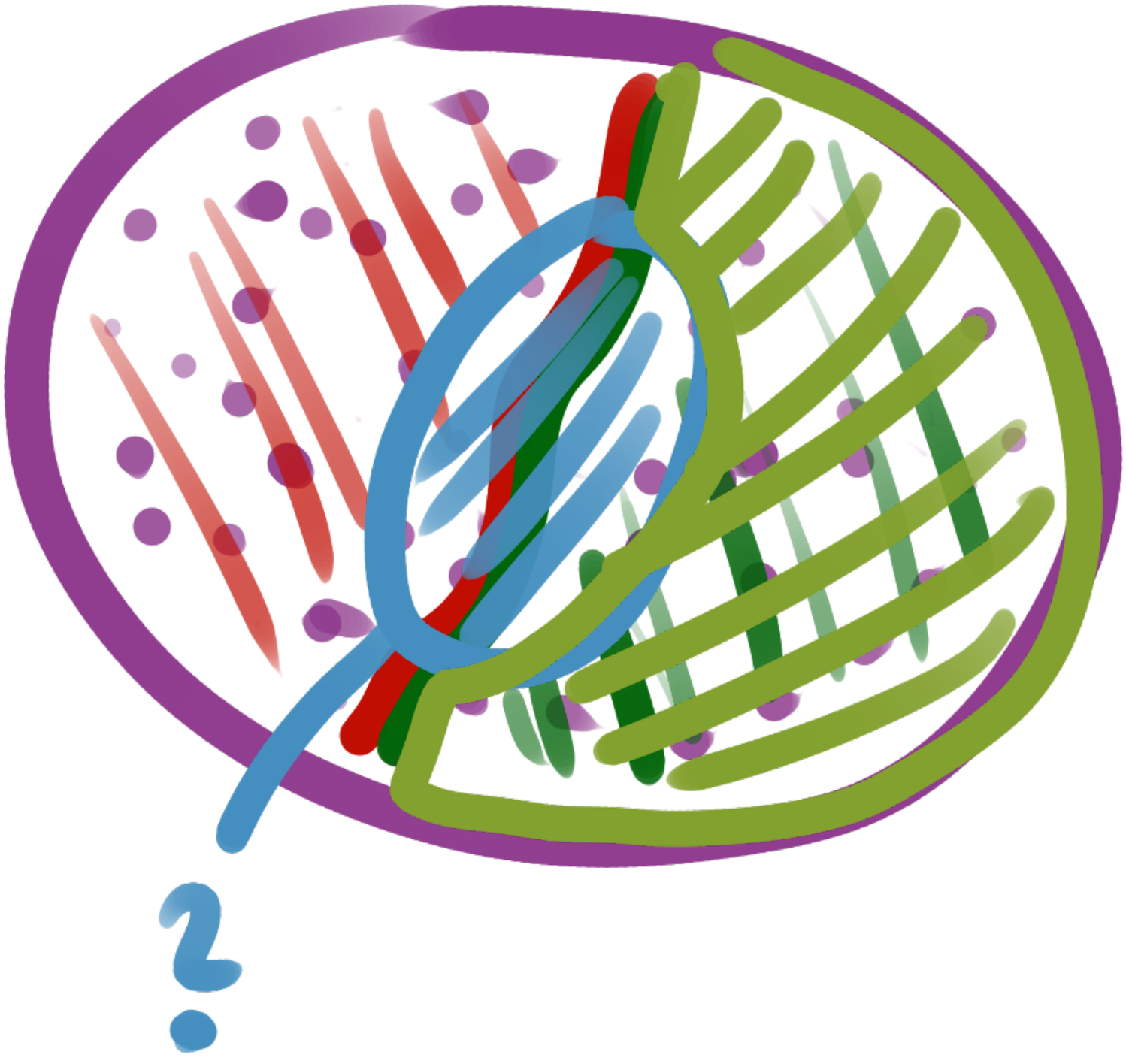


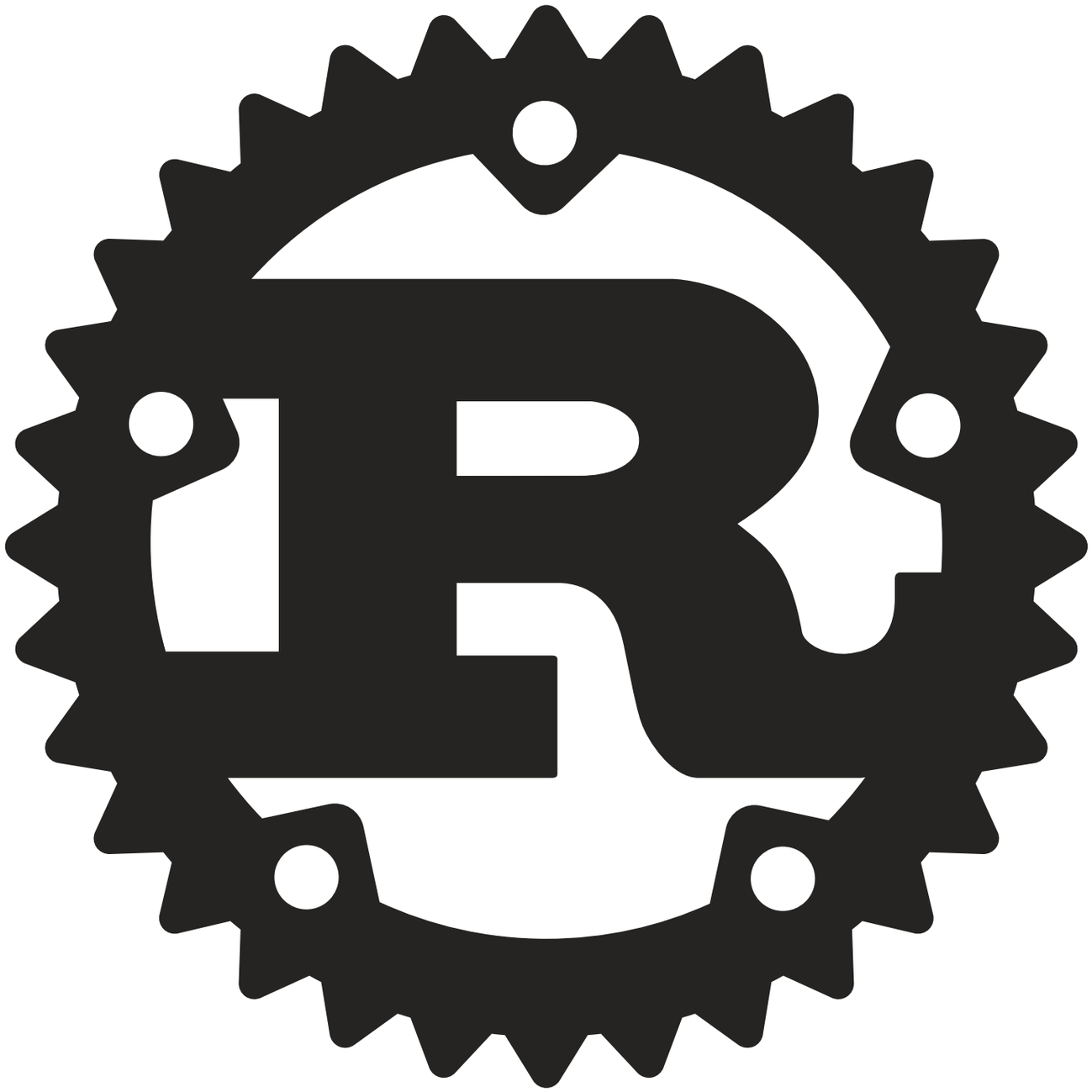


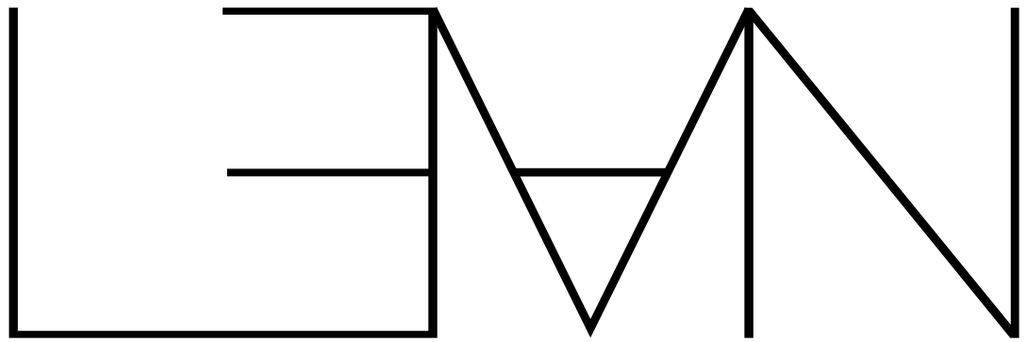


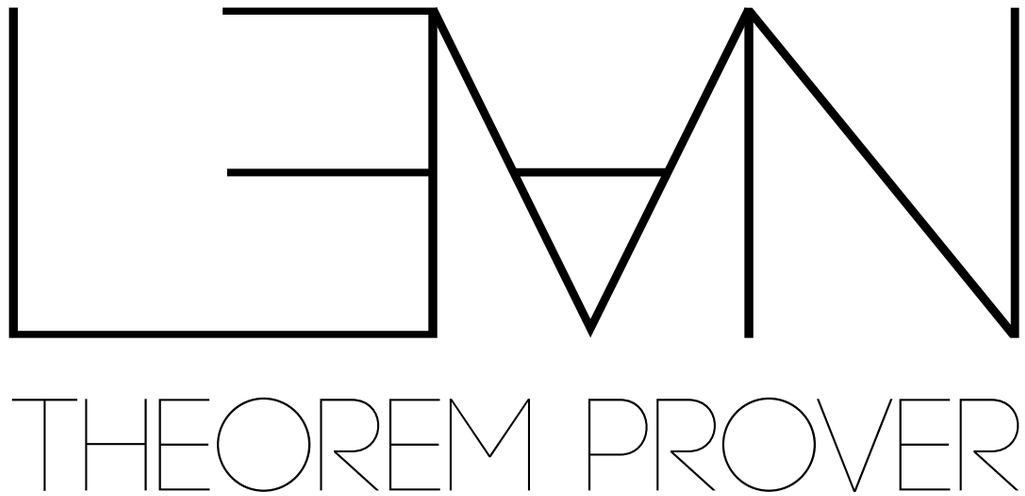









THEOREM PROVER




```
theorem and_commutative (p q : Prop) : p ∧ q → q ∧ p :=
  fun hpq : p ∧ q =>
  have hp : p := And.left hpq
  have hq : q := And.right hpq
  show q ∧ p from And.intro hq hp
```

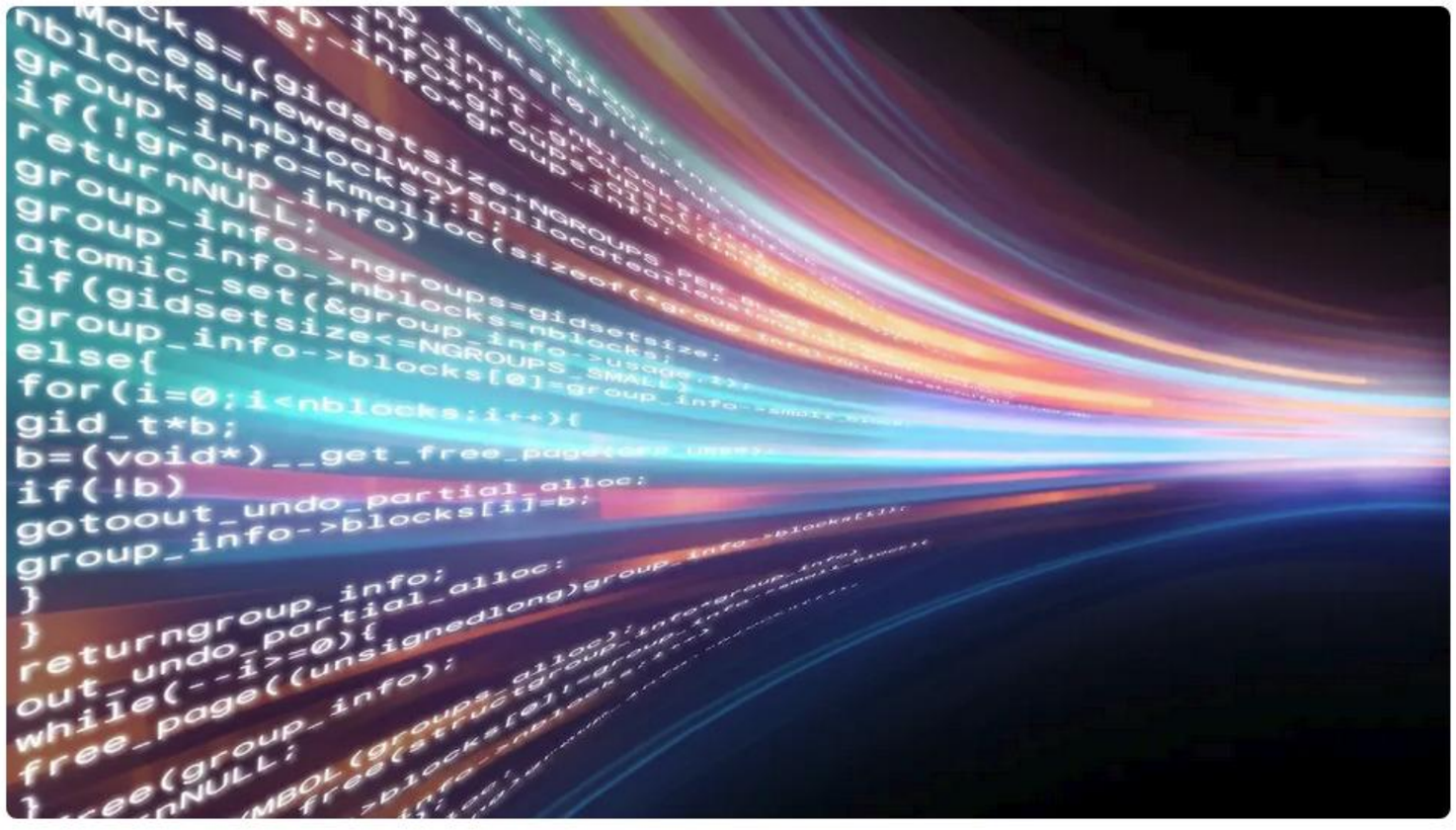
RESEARCH

FunSearch: Making new discoveries in mathematical sciences using Large Language Models

14 DECEMBER 2023

Alhussein Fawzi and Bernardino Romera Paredes

Share



O tom co nejde naprogramovat

Ada Böhm
ada@kreatrix.org