

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



Informatika  
je tudi  
znanost

8. marec  
2014





# Predstavitev



Andrej (Andy) Brodnik



Uroš Čibej



Nataša Kristan



Jurij Mihelič





# Zakaj proučevati avtomate?

- Uporabnost v praksi
  - Obdelava in iskanje v besedilih
    - Opis programskih in naravnih jezikov
  - Programiranje in algoritmi
    - Razumevanje pravilnosti programov
  - Modeliranje realnih problemov
    - Postopki, protokoli, elektronska vezja, itd.





# Zakaj proučevati avtomate?

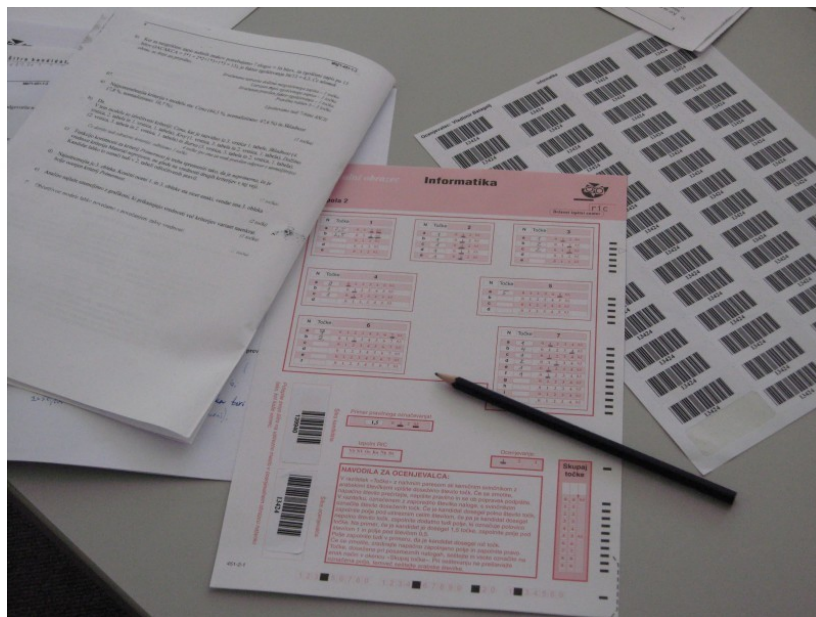
- Razumevanje omejitev računalnikov in programov
  - Kaj je možno izračunati v teoriji?
    - Neizračunljivi oz. neodločljivi problemi (*incomputable/undecidable problems*)
  - Kaj je možno izračunati v praksi?
    - Težko obvladljivi problemi (*intractable problems*)





# Zakaj teoretične osnove RIN in področje formalnih jezikov in avtomatov?

- PIK 2015
- Matura
- Bober
- ...



Ena od naslednjih štirih kombinacij je napačna. Katera?





# PIK 2015 – izpitni cilji

Pridobivanje in razvijanje  
**temeljnega znanja**  
iz informatike

Sposobnost uporabe IKT  
v povezavi z  
drugim znanjem

Razvoj digitalne in  
informacijske pismenosti

- Diskretne strukture
- Osnove programiranja
- **Algoritmi in zahtevnost (ACM kurikulum, 2008)**
- Arhitektura in organiziranost računalniških sistemov
- Operacijski sistemi
- Omrežno računalništvo
- Programski jeziki
- Vmesnik človek računalnik
- Grafično in vizualno računalništvo
- Inteligentni sistemi
- Upravljanje informacij
- Družbena in poklicna vprašanja
- Programsko inženirstvo
- Računska znanost







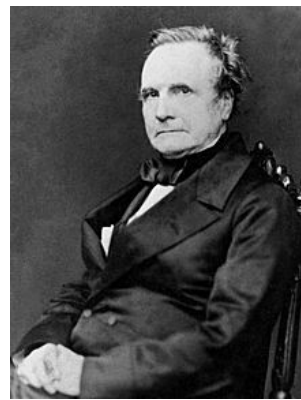
# Malce zgodovine



Pascal, 17. st



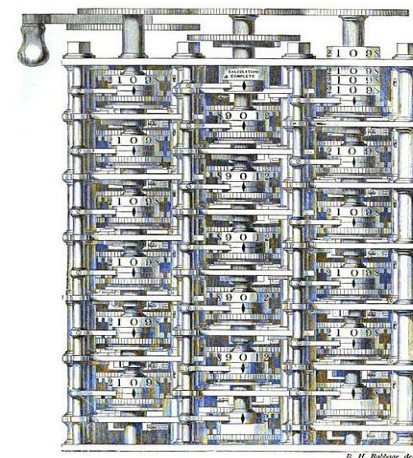
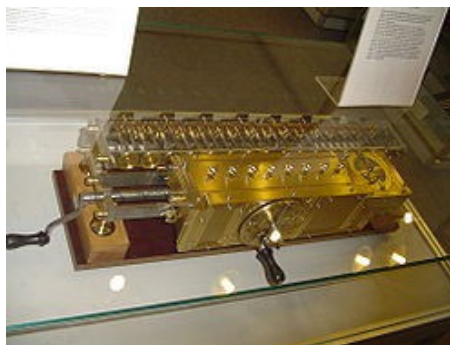
Leibniz, 17. st.



Babbage, 19. st.



Lovelace, 19. st.



*E. H. Babbage, del.*





# Malce zgodovine

- McCulloch in Pitts, 19{42,47}
  - nevrofiziologija
- Mealey in Moore, 19{55,56}
- Myhill in Nerode, 1958
- Rabin, Scott, 1959





Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



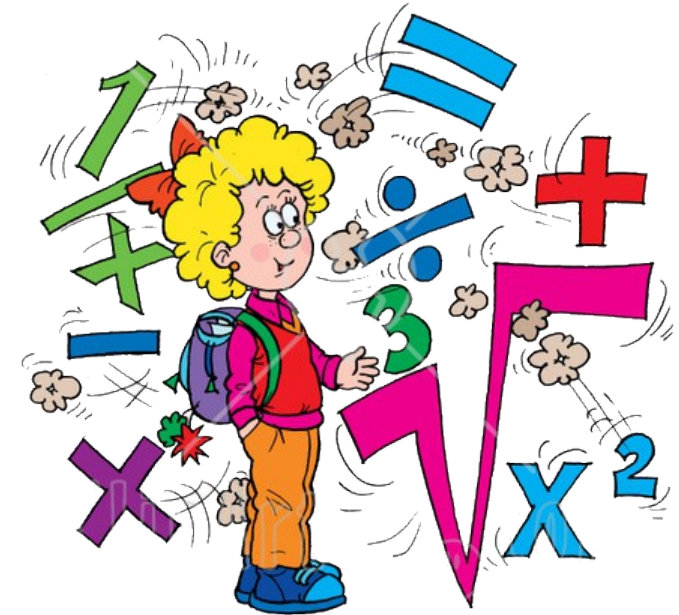
# Ogrevanje za končne avtomate

8. marec  
2014



# Končni avtomat

- Avtomat?
  - Formalni (matematični) model
  - Opis računalnika oz. računskega stroja
  - Odločitveni (da/ne) stroj
  - Dovolj preprost, da je matematično obvladljiv





# Končni avtomat

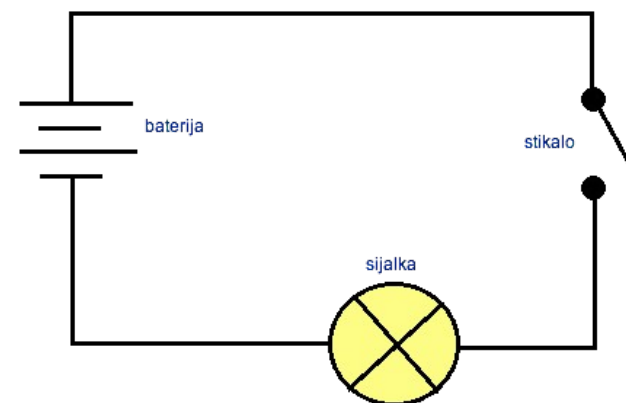
- Končni?
  - Opis avtomata je končen
    - Za opis ne porabimo preveč papirja in črnila
    - Opisan s končno mnogo simboli
  - **Pomni lahko le končno mnogo podatkov**
  - Kljub temu lahko opiše neskončnost





# Stikalo za luč

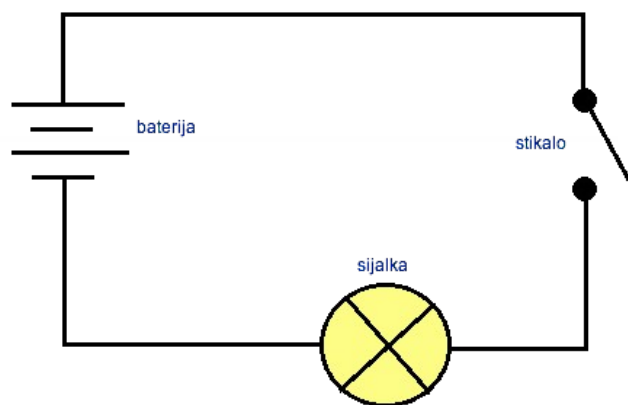
- Vezje sestavljeno iz:
  - baterije, stikala in luči.
- Stikalo pritisnemo  $n$ -krat:
  - Kdaj luč sveti?



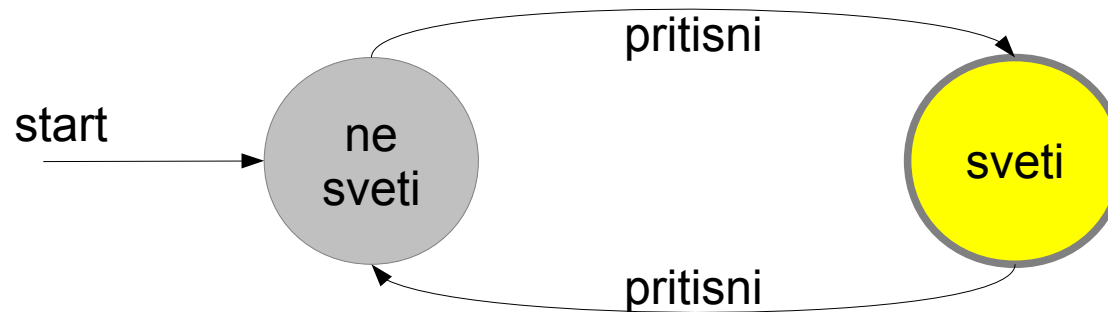




# Stikalo za luč

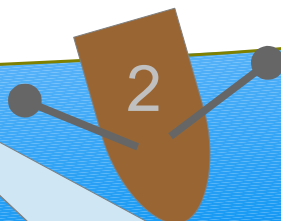


Razpoznava  
zaporedja pritiskov stikala  
lihe dolžine.





# Brodníkov problem – volk, koza, zelje

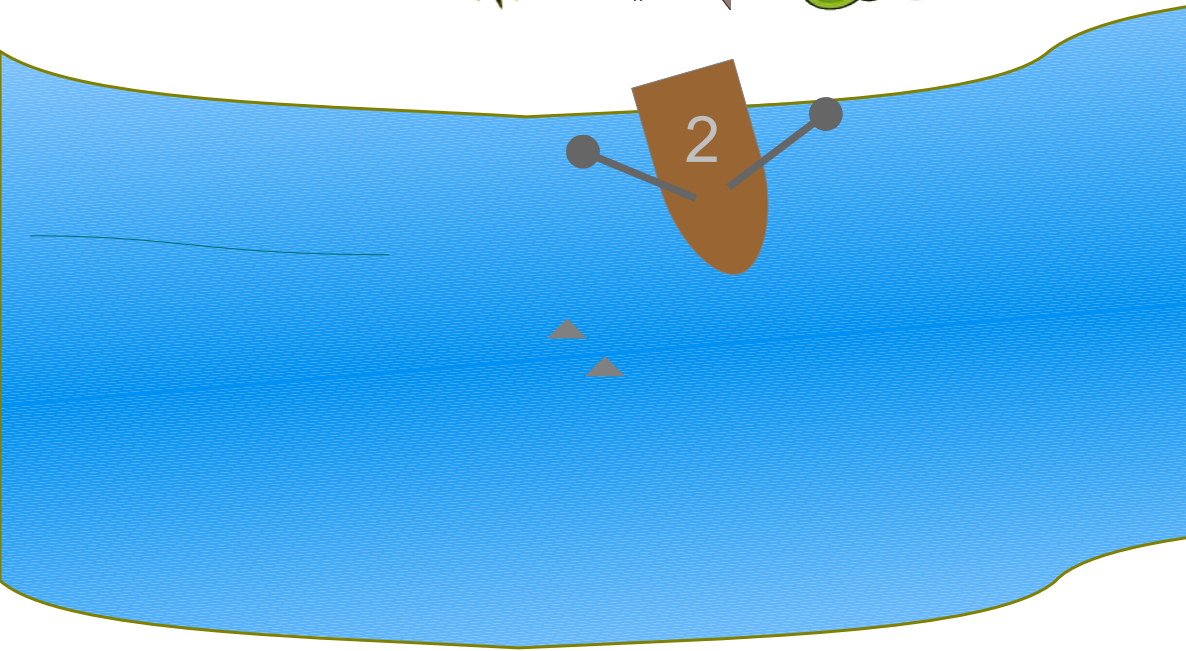


- Čoln le za dva
- Brodník vedno vesla
- Volk (brez brodníka) poje kozo
- Koza (brez brodníka) poje zelje





# Brodníkov problem



PROBLEM:

THE BOAT ONLY HOLDS TWO, BUT YOU CAN'T LEAVE THE GOAT WITH THE CABBAGE OR THE WOLF WITH THE GOAT.



SOLUTION:

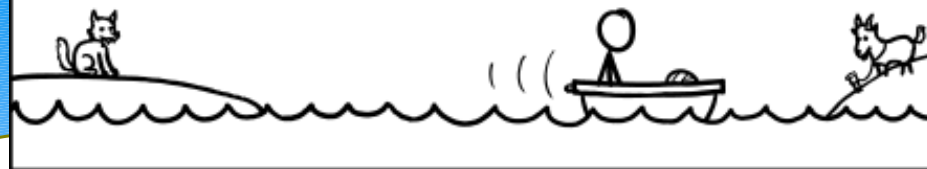
1. TAKE THE GOAT ACROSS.



2. RETURN ALONE.



3. TAKE THE CABBAGE ACROSS.



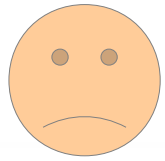
4. LEAVE THE WOLF.

WHY DID YOU HAVE A WOLF?



Na kateri  
strani reke  
je čoln?

# Brodníkov problem



b



v



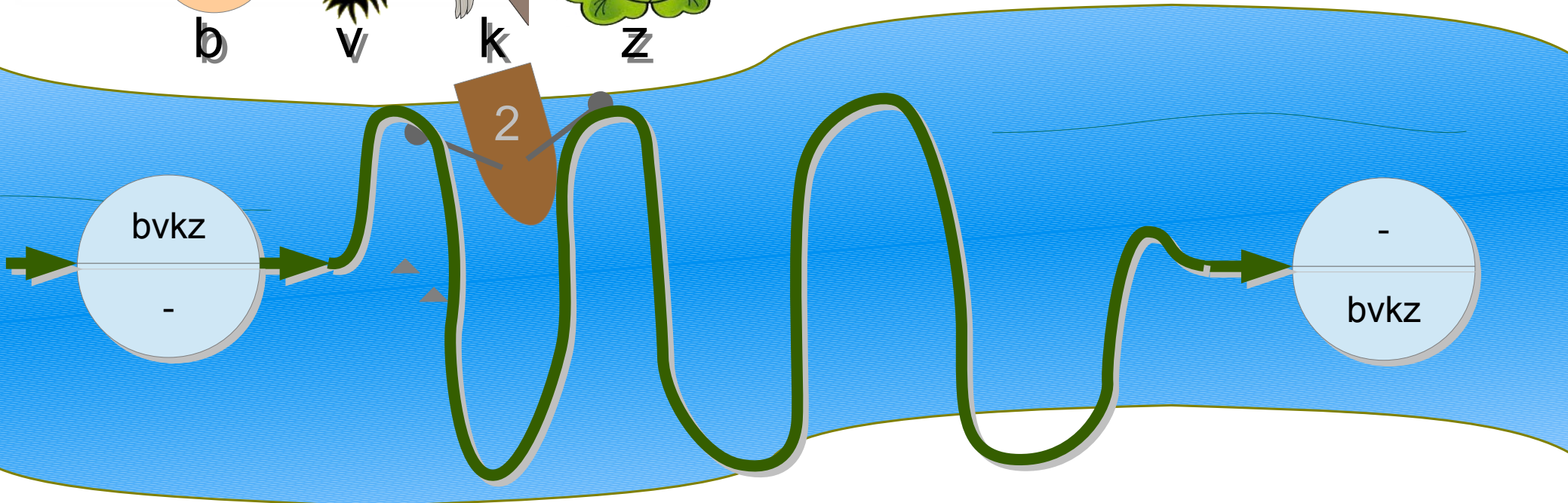
k



z

Kdo se pelje v čolnu?

- b – samo brodnik
- v – volk in brodnik
- k – koza in brodnik
- z – zelje in brodnik

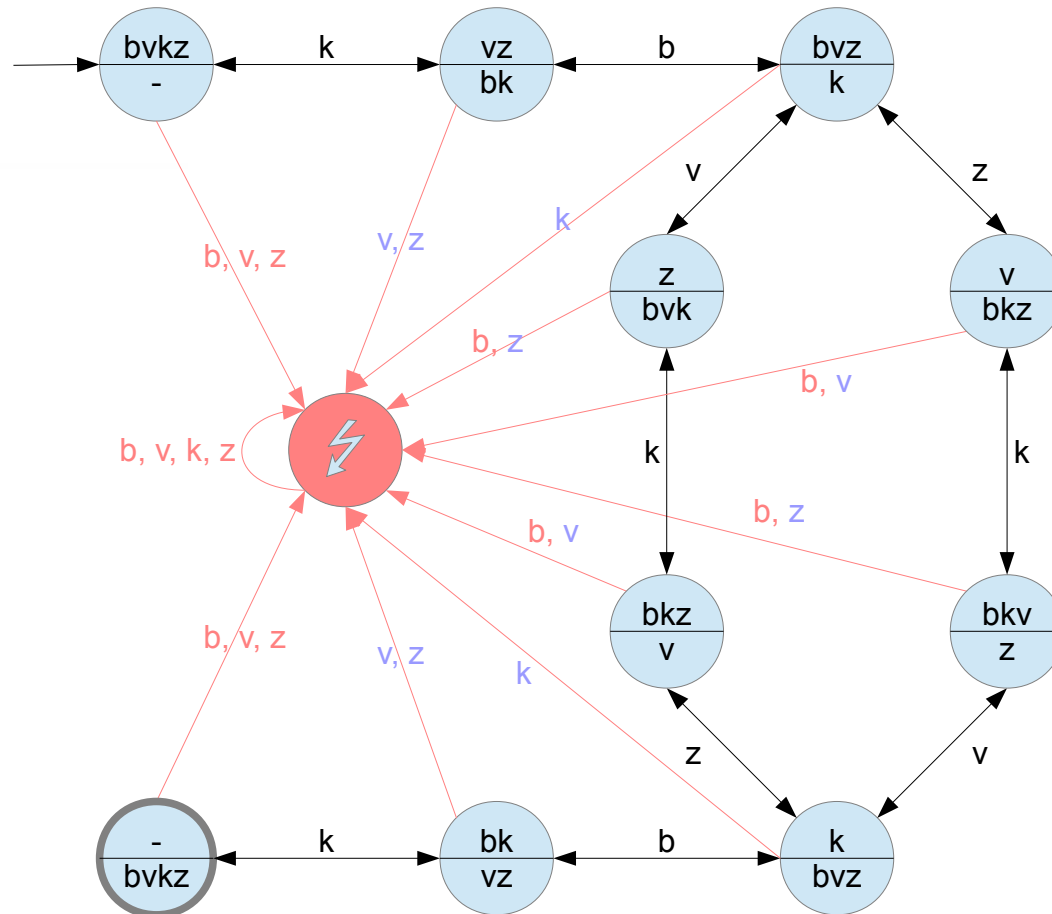






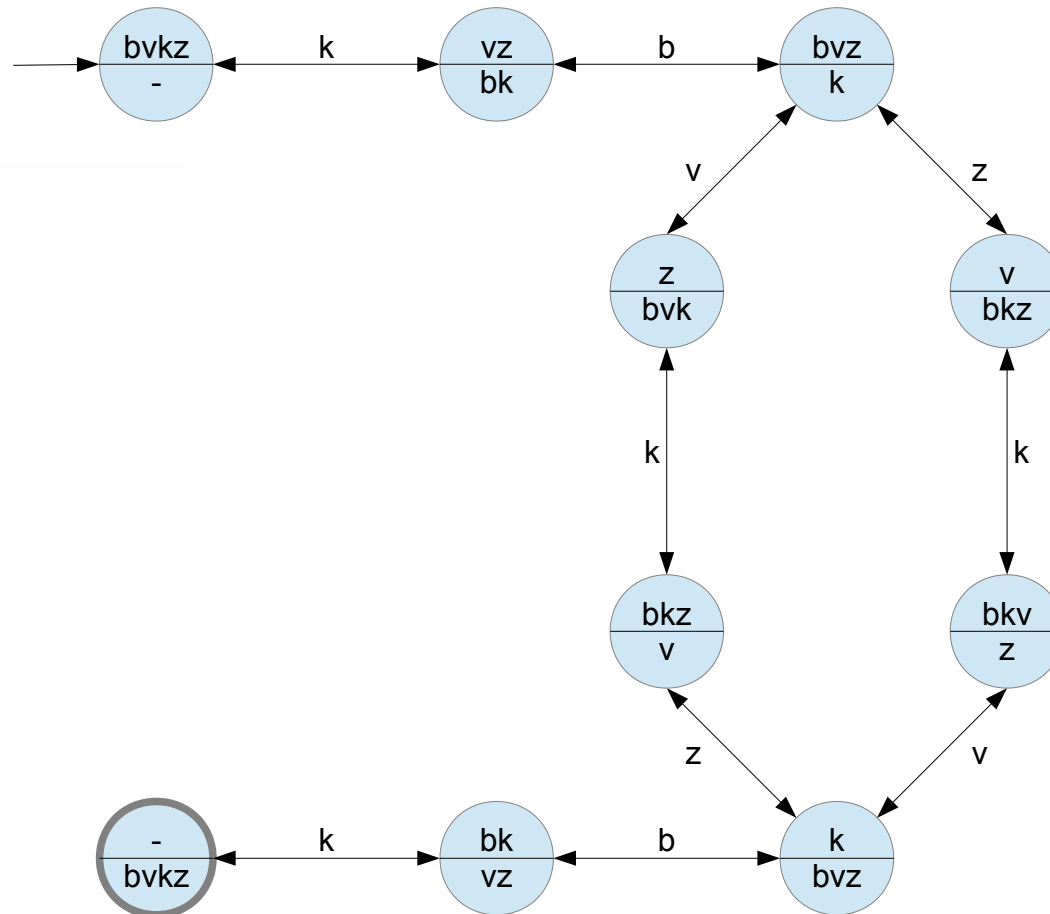


# Brodníkov problem



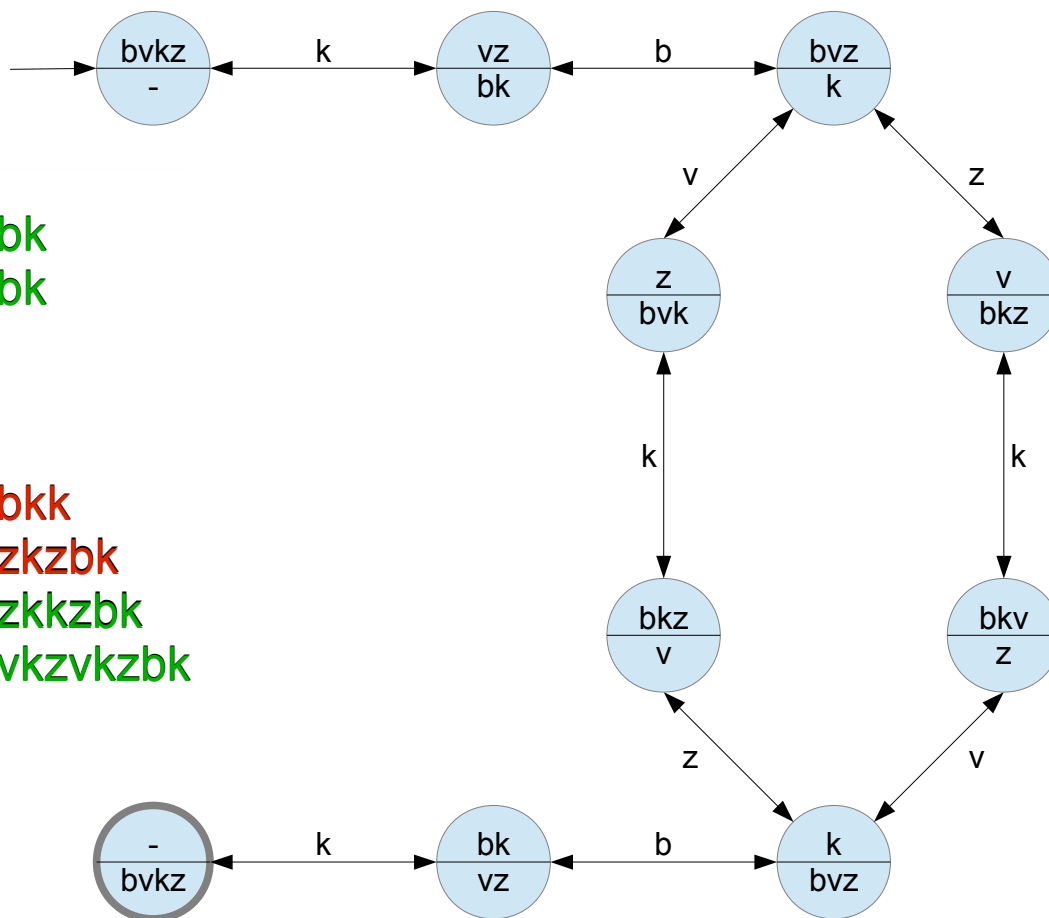


# Brodníkov problem





# Brodníkov problem



- **kbvkz**bk
- **kbz**kv**bk**
- **kbv**
- **kvz**
- **zzz**
- **kbvkz**bkk
- **kbz**kv**z**kbk
- **kbz**kv**z**kk**z**bk
- **kbv**k**z**v**k**z**v**k**z**bk





Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



Množice

8. marec  
2014



# Množice

- Množica je zbirka *različnih* objektov
  - $S = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
  - $A = \{ a, b, c, \dots, \check{z} \}$
  - $\{ 1, 2 \} = \{ 2, 1 \} = \{ 1, 1, 2, 2, 1, 2, 1 \}$
- Relacija pripadnosti
  - $1 \in S, \check{z} \in A, \check{z} \notin S$

0  
1 2  
3 4 5  
6 7 8 9



# Množice

- Množica je zbirka *različnih* objektov
  - $S = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
  - $A = \{ a, b, c, \dots, ž \}$
  - $\{ 1, 2 \} = \{ 2, 1 \} = \{ 1, 1, 2, 2, 1, 2, 1 \}$
- Moč množice (število elementov v množici)
  - $|S| = 10, |A| = 25$
  - $|\{ 1, 2 \}| = 2, |\{ 1, 1, 2, 2, 1, 2, 1 \}| = 2$





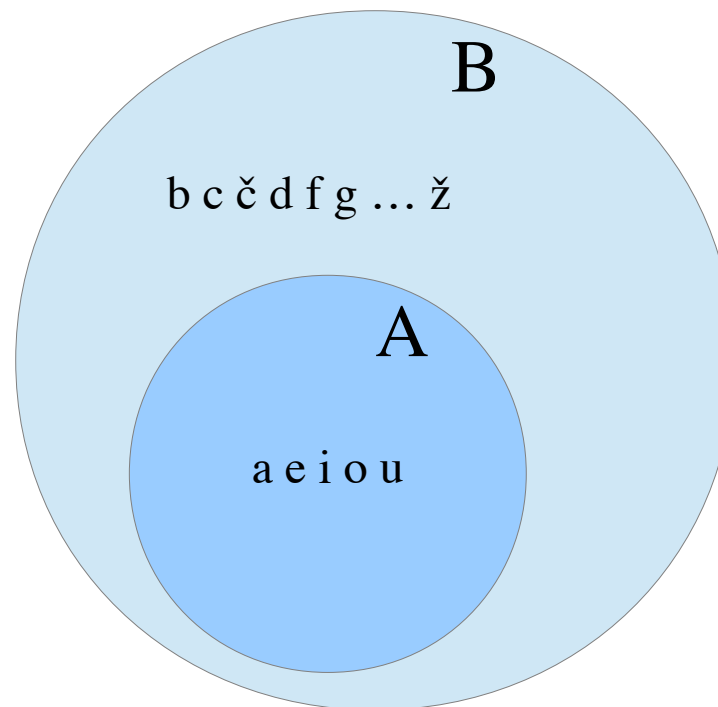
# Množice

- Podmnožice

- $A \subseteq B$ : vsak element mn. A je tudi element mn. B
- $A \subset B$ : stroga vsebovanost

- Nadmnožice

- $B \supseteq A \equiv A \subseteq B$
- $B \supset A \equiv A \subset B$







# Množice

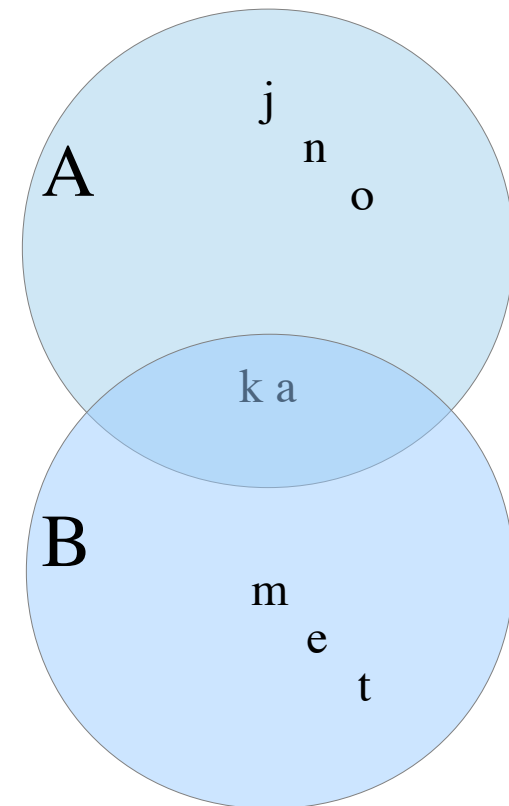
- Potenčna množica
  - Vsebuje vse možne podmnožice neke množice  $A$
  - Oznaka:  $P(A)$  ali  $2^A$
- $A = \{ a, b, c \}$
- $P(A) = \{ \emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\} \}$





# Množice

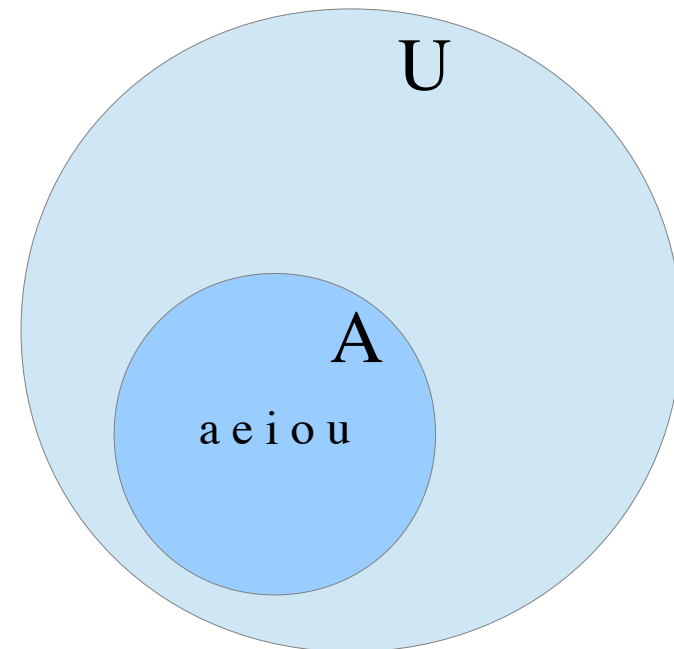
- Operacije nad množicami
  - $A = \{ j, a, n, k, o \}$
  - $B = \{ m, e, t, k, a \}$
  - Unija:  $A \cup B = \{ j, a, n, k, o, m, e, t \}$
  - Presek:  $A \cap B = \{ a, k \}$
  - Razlika:  $A \setminus B = \{ j, n, o \}$   
 $B \setminus A = \{ m, e, t \}$





# Množice

- Univerzalna množica  $U$ 
  - množica vseh objektov o kateri razpravljamo v okviru nekega problema
- Komplement:  $A^C = U \setminus A$ 
  - $U$  = črke abecede
  - $A$  = samoglasniki
  - $A^C$  = soglasniki





# Množice

- Urejeni par in  $n$ -terka

- $(s, t), (t, s), (9, 8, 7), (8, 9, 7), (1, 3, 3, t)$

Vsi  
možni  
pari

- Kartezični produkt množic

- $A \times B = \{ (x, y) : x \in A \text{ in } y \in B \}$

- $A = \{ 1, 2, 3 \}, B = \{ a, b \}$

- $A \times B = \{ (1, a), (2, a), (3, a), (1, b), (2, b), (3, b) \}$



Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# Abece in nizi

8. marec  
2014





# Abeceda

- Abeceda je *končna* množica simbolov
- Oznaka  $\Sigma$ , tudi  $A$ .
  - slovenska abeceda:  $\{ a, b, c, \dots, \check{z} \}$
  - ASCII, Unicode, ...
  - binarna abeceda:  $\{ 0, 1 \}$
  - genska abeceda:  $\{ a, c, t, g \}$
  - brodnikov problem:  $\{ b, v, k, z \}$





# Niz

- *Niz* ali *beseda* nad abecedo  $\Sigma$ 
  - zaporedje simbolov abecede  $\Sigma$ 
    - 011, 01010, 11001, 000, 1, 0
    - fri, abrakadabra, avtomat
    - kbvkzkb
- Pozor
  - 0 kot niz in 0 kot simbol
  - Prazni niz  $\varepsilon$  (včasih tudi  $\lambda$ )

- $\Sigma = \{0, 1\}$
- $\Sigma = \{0, 1, 2\}$
- $\Sigma = \{0, 1, \check{z}\}$

Oznake  
w, v, x, y, z



# Niz

- Dolžina niza
  - je število pozicij/simbolov v nizu
    - $|\varepsilon| = 0$
    - $|1| = 1$
    - $|10| = 2$
    - $|011| = 3$
    - $|0100| = 4$
    - $|\text{abrakadabra}| = 11$
    - $|\varepsilon\varepsilon\varepsilon\varepsilon\text{abra}\varepsilon\varepsilon\text{k}\varepsilon\text{d}\varepsilon\varepsilon\text{bra}\varepsilon\varepsilon\varepsilon\varepsilon\varepsilon\varepsilon| = 11$





# Niz

- *Stik oz. konkatencija nizov:*
  - $w = 100, x = 110$ 
    - $wx = 100110$
    - $xw = 110100$
    - $ww = w^2 = 100100$
    - $www = w^3 = 100100100$
    - $wxwx = (wx)^2 = 100110100110$
    - $\varepsilon w = w = 100$





# $\Sigma^*$ – sigma zvezdica

- Abeceda  $\Sigma$
- Množica  $\Sigma^*$ 
  - Vsi možni nizi nad  $\Sigma$
- Kako velika je  $\Sigma^*$ ?
- Znamo vse nize naštetih?

$\Sigma$  (0, 1)

$\Sigma^*$

$\epsilon$   
0, 1,  
00, 01, 10, 11,  
000, 001, 010, 011,  
100, 101, 110, 111,  
0000, 0001, 0010, 0011,  
0100, 0101, 0110, 0111,  
...





# $\Sigma^*$ – sigma zvezdica

- Vsi nizi brodníkovskega problema

$\Sigma$

b, v, k, z

$\Sigma^*$

$\varepsilon$

b, v, k, z,

bb, bv, bk, bz, vb, vv, vk, vz,

kb, kv, kk, kz, zb, zv, zk, zz,

bbb, bbv, bbk, bbz, bvb, bvv, bvk, bvz,

bkb, bkv, bkk, bkz, bzb, bzv, bzk, bzz,

vbb, vbv, vbk, vbz, vvb, vvv, vvk, vvz,

vkb, kvv, vkk, vkz, vzb, vzv, vzk, vzz,

...

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# Jeziki

8. marec  
2014



# Jezik

- Jezik  $L$  nad abecedo  $\Sigma$ 
  - $L$  je podmnožica množice  $\Sigma^*$
  - $L \subseteq \Sigma^*$
  - Jezik je množica dopustnih nizov
    - lahko končna ali
    - neskončna ali
    - celo prazna.

Jezik  
je torej  
v kontekstu  
avtomatov  
drug izraz  
za množico  
nizov





# Primeri jezikov

- Abeceda  $\Sigma = \{ 0, 1 \}$

$\Sigma$   
0, 1

- $L = \emptyset$
- $L = \{ 0 \}$
- $L = \{ 00, 01, 10, 11 \}$
- $L = \{ 1, 010, 00100, 0001000, \dots \}$
- $L = \{ 1, 10, 100, 1000, 10000, 100000, \dots \}$
- $L = \{ \varepsilon, 0, 1, 00, 01, 10, 000, 001, 010, 100, 101, 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010, 00000, \dots \}$





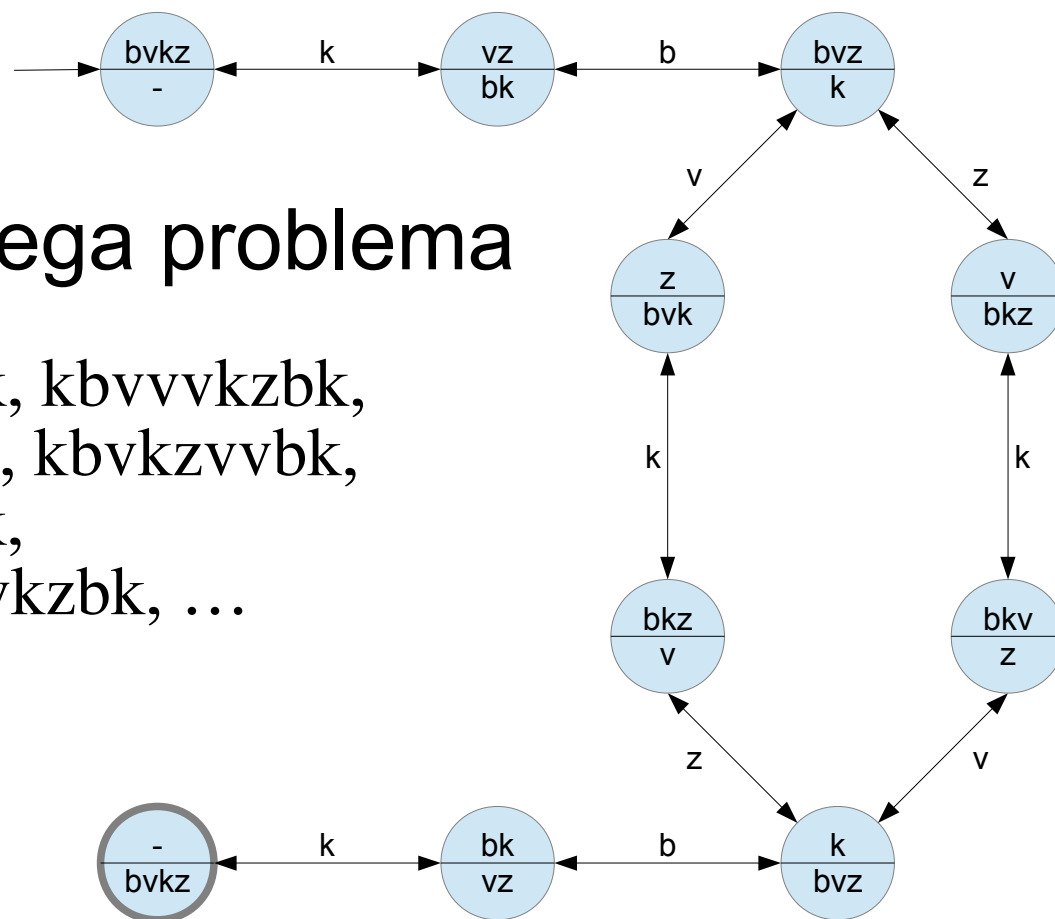
# Primeri jezikov

- Jezik lahko tudi opišemo
  - Dvojiški nizi, kjer je na sredini 1, ostalo je 0
  - Dvojiška števila, ki so potence števila 2
  - Nizi nad  $\{0,1\}$ , ki ne vsebujejo zaporednih 1
  - $L = \{0^n 1^n \mid n \geq 0\}$
  - $L = \{w \mid \#_0(w) = \#_1(w)\}$





# Primeri jezikov



## • Rešitve brodnikovega problema

- kbvkzbk, kbzkvbk,
- kkkbvkbk, kbbbvkbk, kbvvvkbk,  
kbvkkkbk, kbvkzzzbk, kbvkzvbk,  
kbvkzbbb, kbvkzbbk,
- kbbkkbvkbk, kkkbbbvkbk, ...





# Regularni izrazi

- Način opisovanja nizov oz. jezikov
- Podprti v večini programskih jezikov
- Formalni vs. praktični RI
- Jezike, ki jih je moč opisati z regularnimi izrazi imenujemo regularni jeziki.
- Obstajajo tudi jeziki, ki niso regularni.





$\Sigma$

0, 1

# Regularni izrazi

0	1
00000	0101010
000111000	
11110011001111	

izraz	jezik
$0 + 1$	0, 1
$00 + 01 + 001 + 101$	00, 01, 001, 101
$\epsilon + 10 + 100$	$\epsilon$ , 10, 100
$00(\epsilon + 0 + 1 + 11)11$	0011, 00011, 00111, 001111

izraz	jezik
$0^*$	$\epsilon$ , 0, 00, 000, 0000, 00000, 000000, 0000000, ...
$(11)^*$	$\epsilon$ , 11, 1111, 111111, 11111111, 1111111111, 111111111111, ...
$(0 + 1)^*$	$\epsilon$ , 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, ...
$(0 + 11)^*$	$\epsilon$ , 0, 11, 00, 011, 110, 1111, 000, 0011, 0110, ...
$(\epsilon + 0)^*$	$\epsilon$ , 0, 00, 000, 000, 0000, 00000, 000000, ...

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# Regularni izrazi (bonus)

8. marec  
2014



# Regularni izrazi

- Osnovni/atomični regularni izrazi
  - $\emptyset$  opisuje prazen jezik  $L(\emptyset) = \{ \}$
  - $\varepsilon$  opisuje jezik  $L(\varepsilon) = \{ \varepsilon \}$
  - $a \in \Sigma$  opisuje jezik  $L(a) = \{ a \}$





# Regularni izrazi

- Sestavljeni regularni izrazi
  - Jih tvorimo iz že obstoječih izrazov
  - $(p + r)$  opisuje unijo jezikov  $L(p + r) = L(p) \cup L(r)$
  - $(p r)$  opisuje stik jezikov  $L(p r) = L(p) L(r)$
  - $(p^*)$  opisuje iteracijo jezika  $L(p^*) = (L(p))^*$





# Jezik

- *Stik oz. konkatenacija* jezikov A in B
  - $AB =$  vse besede  $xy$ , kjer  $x \in A$  in  $y \in B$
  - $A = \{ 123, 456, 789 \}$ ,  $B = \{ \text{čšž}, \text{aeiou} \}$
  - $AB = \{ 123\text{čšž}, 456\text{čšž}, 789\text{čšž}, 123\text{aeiou}, 456\text{aeiou}, 789\text{aeiou} \}$







# Jezik

- Potenca jezika  $L^n$ 
  - Stik jezika samega s seboj
  - $L = \{ b, v, k, z \}$
  - $L^0 = \{ \varepsilon \}$
  - $L^1 = L$
  - $L^2 = \{ bb, bv, bk, bz, vb, vv, \dots, zk, zz \}$
  - $L^3 = \{ bbb, bbv, bbk, \dots, zzk, zzz \}$





# Jezik

- Iteracija  $L^*$

- $L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \dots$

- $L = \{ b, v, k, z \}$

- $L^* = \{ \varepsilon, b, v, k, z, bb, bv, \dots \}$





# Primeri

$$\begin{aligned} L(\emptyset) &= \{ \} \\ L(\varepsilon) &= \{ \varepsilon \} \\ L(b) &= \{ b \} \\ L(v) &= \{ v \} \\ L(k) &= \{ k \} \\ L(z) &= \{ z \} \end{aligned}$$

$$\begin{aligned} L(vk) &= L(v) L(k) = \{ vk \} \\ L(vkz) &= \{ vkz \} \\ L(v\emptyset) &= \{ \} \\ L(v\varepsilon) &= \{ v \} \end{aligned}$$

$$\begin{aligned} L(b^*) &= \{ \varepsilon, b, bb, bbb, bbbb, \dots \} \\ L(\varepsilon^*) &= \{ \varepsilon \} \\ L(\emptyset^*) &= \{ \} \end{aligned}$$

$$\begin{aligned} L(v + k) &= L(v) \cup L(k) = \{ v, k \} \\ L(v + k + z) &= \{ v, k, z \} \\ L(v + \emptyset) &= \{ v \} \\ L(v + \varepsilon) &= \{ \varepsilon, v \} \end{aligned}$$

$$\begin{aligned} L(vk + kz) &= \{ vk, kz \} \\ L((v + k)bb(k + z)) &= \{ vbbk, vbbz, kbbk, kbbz \} \\ L((\varepsilon + vv)bbb) &= \{ bbb, vvbbb \} \\ L((b + v + k + z)^*) &= \text{množica vseh nizov na } \Sigma \\ L(kb(v + k + z)^*bk) &= \{ kbbk, kbvbk, kbkkkzzzvkvkzvb, \dots \} \\ L((b + v + k + z)^*zz(b + v + k + z)^*zz(b + v + k + z)^*) &= 2x \text{ zaporedoma } zz \end{aligned}$$

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



8. marec  
2014

# Problemi



# Odločitveni problemi

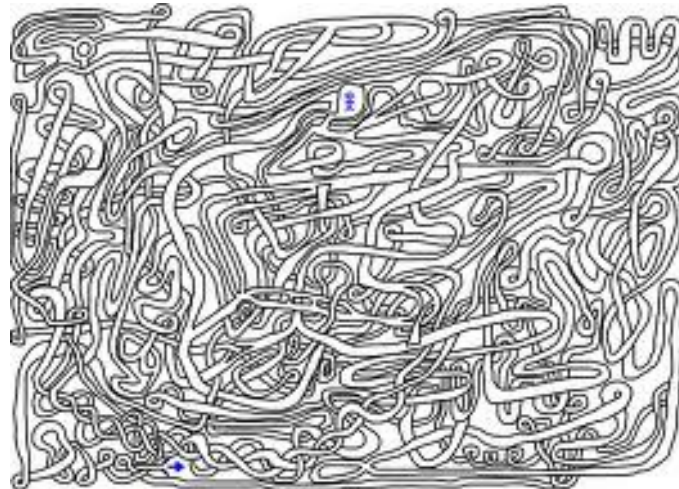
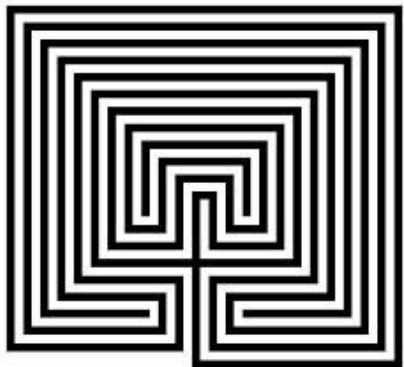
- Problem
  - Računski problem (angl. *computational problem*)
  - Problem, ki ga (lahko) rešuje računalnik
  - Ne gre (le) za aritmetično računanje
  - Primeri problemov:
    - iskanje najmanjšega elementa
    - urejanje zaporedja
    - iskanje poti v labirintu





# Odločitveni problemi

- Naloga oz. primerek problema
  - Za problem je možnih mnogo različnih nalog
  - Primeri nalog za problem iskanja poti v labirintu







# Odločitveni problemi

- Odločitveni problem
  - Računski problem katerega rešitev je lahko le odgovor: **da** ali **ne**
  - Gre torej za vprašanja



Naloga  $x$ , algoritem  $A$ , rešitev da/ne





# Odločitveni problemi

- Še več problemov
  - Se dani program zacikla/ustavi?
  - Ima dani sudoku/iskalec min/... rešitev?
  - Ima dana enačba ničlo?
  - Je mogoče dani zemljevid pobarvati s štirimi barvami?





# Odločitveni problemi

- Je dano število potenca števila 2?
  - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ..., 32, ..., 64, ..., 128, ..., 256, ..., 512, ..., 1024, ...
- Je dano dvojiško število potenca števila 2?
  - 1, 10, 100, 1000, 10000, 100000, 1000000, ...





# \*Odločitveni problemi

- Smo z **da/ne** problemi preveč omejeni?
  - Niti ne, če le znamo pametno spraševati
  - Lahko rešimo tudi optimizacijske probleme
- Problem najmanjše število zaporedja

Zaporedje 5 7 4 5 7 8 9 6 4 6 4 7 6 6 5 5 6 4 8 8 8 9 8 8

Lahko sprašujemo po vrsti

- Je 0 najmanjše število danega zaporedja?
- Je 1 najmanjše število danega zaporedja?
- Je 2 najmanjše število danega zaporedja?
- Je 3 najmanjše število danega zaporedja?
- Je 4 najmanjše število danega zaporedja?

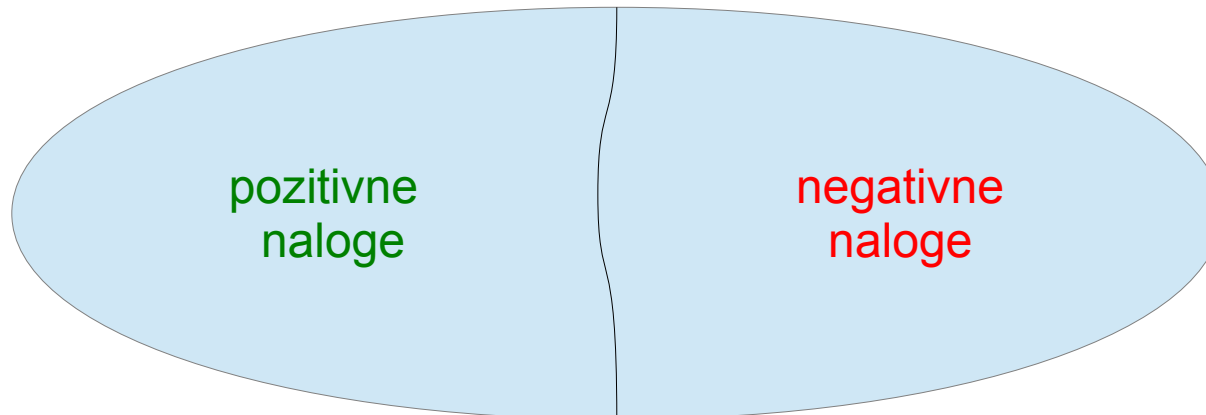
Ali pa uporabimo binarno iskanje

- Je najmanjše število danega zaporedja  $< 5$ ?
- Je najmanjše število danega zaporedja  $< 3$ ?



# Problemi in jeziki

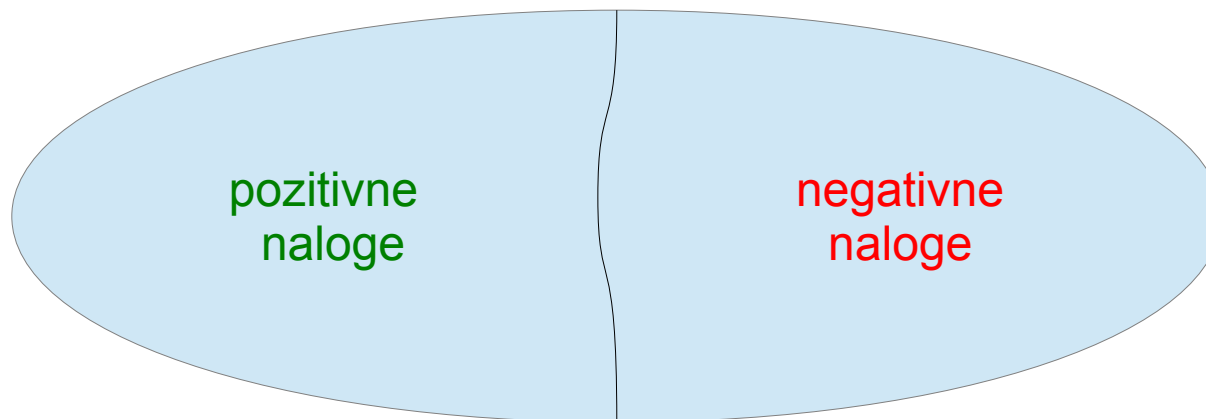
- Pozitivne naloge
  - So naloge problema, za katere je odgovor **da**.
- Negativne naloge
  - So naloge problema, za katere je odgovor **ne**.





# Problemi in jeziki

- Množica vseh pozitivnih nalogo je jezik
  - PS: Tudi množica negativnih nalog je jezik
- Reševanje problema je torej enakovredno razpoznavanju jezika
  - Ugotavljanje pripadnosti naloge jeziku





Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# Deterministični končni avtomat

8. marec  
2014



# Kaj je končni avtomat?

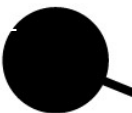
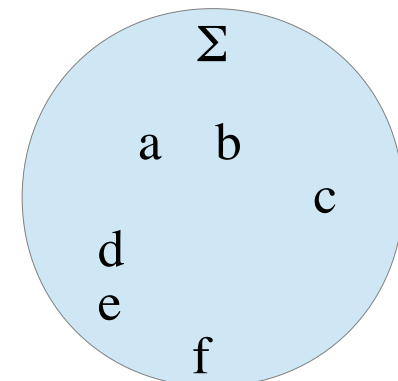
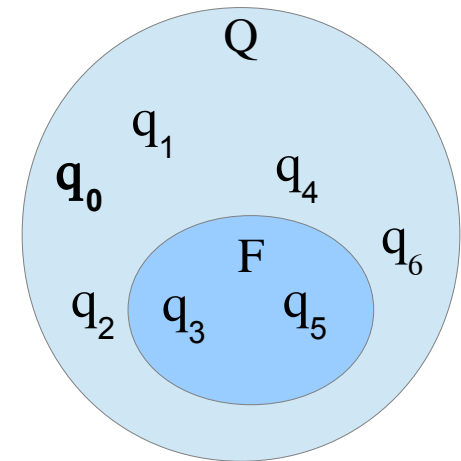
- Formalni sistem
- Pomni le končno količino informacije
- Informacija predstavljena s stanjem
- Stanje se spremeni glede na vhod
- Pravila spreminjanja stanj se imenujejo prehodi





# Definicija DKA

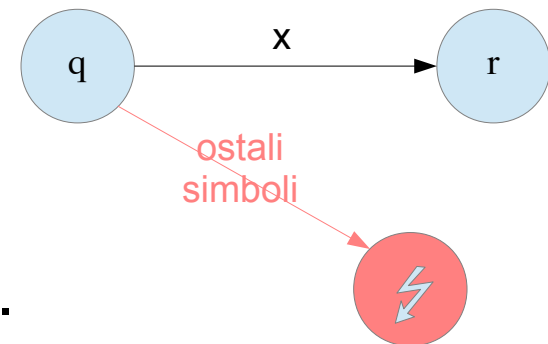
- Peterka  $\langle Q, \Sigma, \delta, q_0, F \rangle$ 
  - $Q$  – končna množica stanj,
    - npr.  $Q = \{ q_0, q_1, q_2, \dots, q_n \}$
  - $\Sigma$  – vhodna abeceda (tudi končna)
  - $\delta$  – funkcija prehodov
  - $q_0$  – začetno stanje,  $q_0 \in Q$
  - $F$  – množica končnih stanj,  $F \subseteq Q$





# Definicija DKA

- Funkcija prehodov  $\delta(q, x)$ 
  - $\delta: Q \times \Sigma \rightarrow Q$
  - $\delta(q, x) = r$ 
    - Če je avtomat v stanju  $q$  in je na vhodu simbol  $x$ , potem gre avtomat v novo stanje  $r$ .
  - Totalna funkcija prehodov
    - Če za nek par  $q$  in  $x$  ni posebej podano  $\delta(q, x)$ , potem gre avtomat v *slepo stanje*.





# Definicija DKA

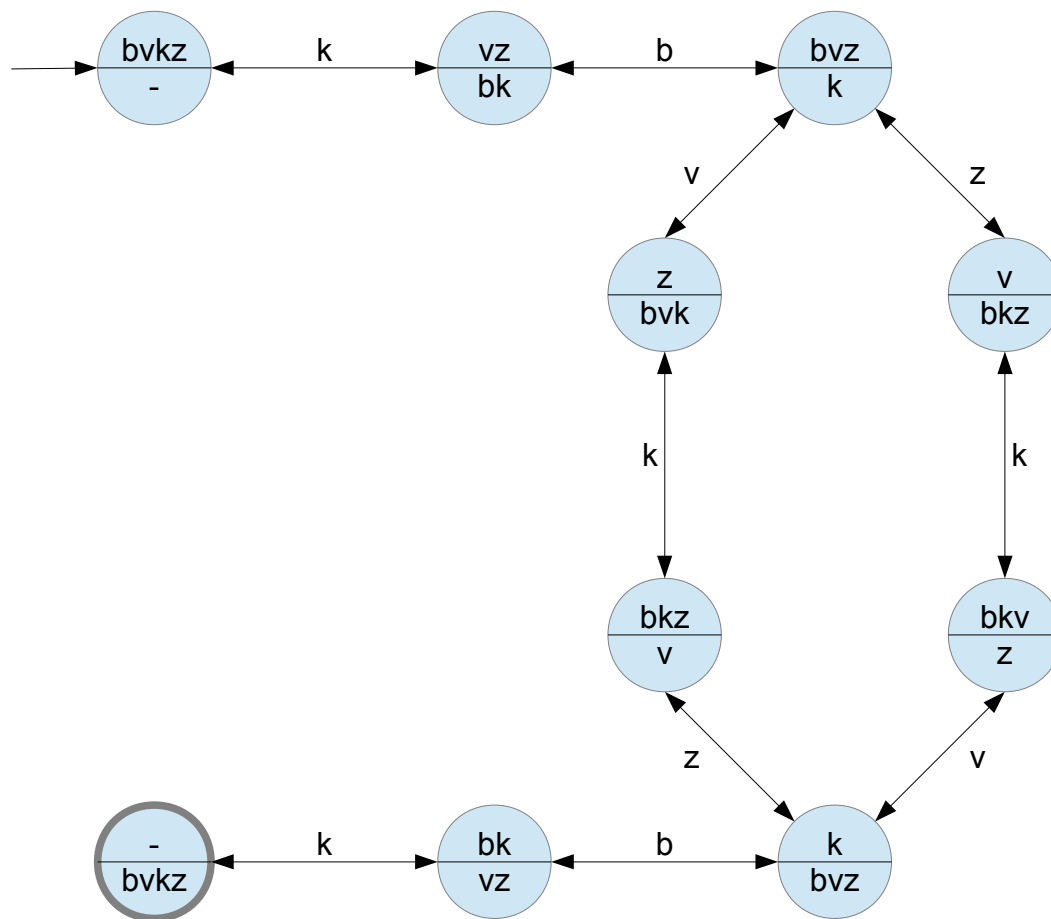
- Funkcija prehodov  $\delta(q, x)$ 
  - Diagram prehodov
  - Tabela prehodov
  - Spisek prehodov





# Funkcija prehodov $\delta(q, x)$

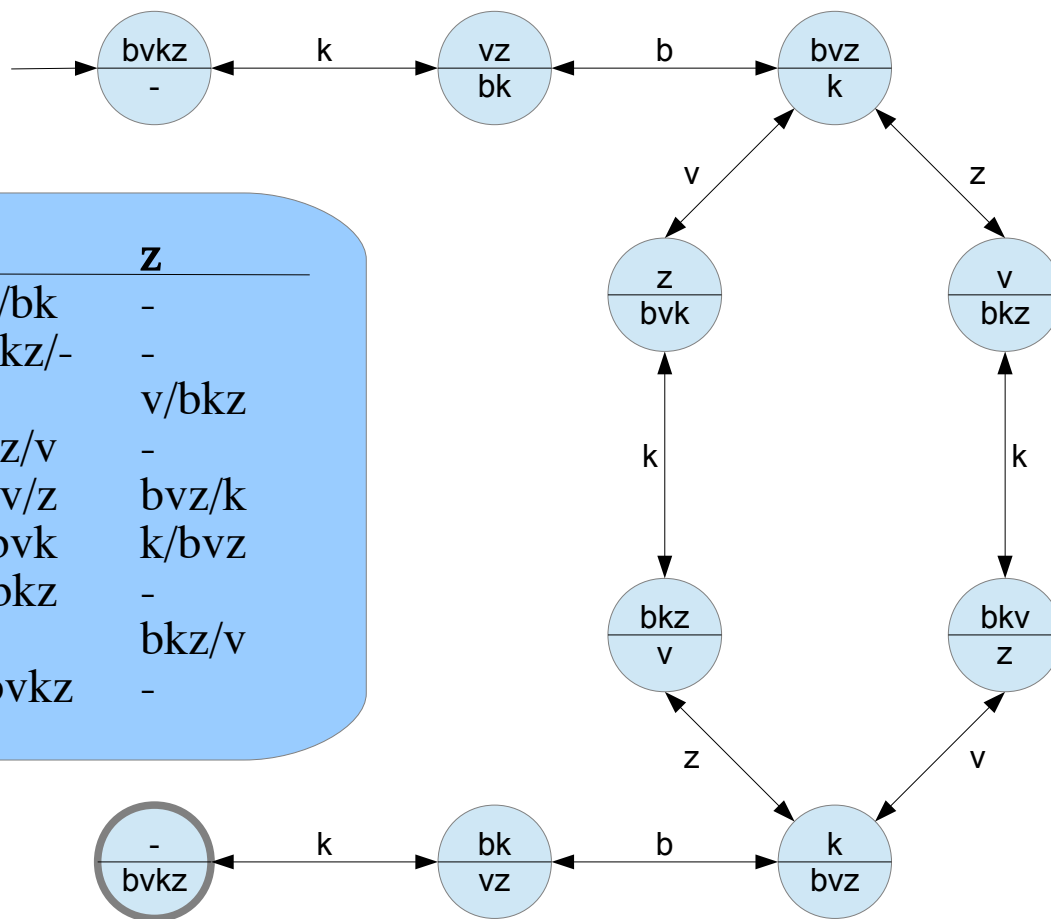
- Diagram prehodov





# Funkcija prehodov $\delta(q, x)$

- Tabela prehodov



	<b>b</b>	<b>v</b>	<b>k</b>	<b>z</b>
<b>bvzk/-</b>	-	-	vz/bk	-
<b>vz/bk</b>	bvz/k	-	bvzk/-	-
<b>bvz/k</b>	vz/bk	z/bvk	-	v/bkz
<b>z/bvk</b>	-	bvz/k	bkz/v	-
<b>v/bkz</b>	-	-	bkv/z	bvz/k
<b>bkz/v</b>	-	-	z/bvk	k/bvz
<b>bkv/z</b>	-	k/bvz	v/bkz	-
<b>k/bvz</b>	bk/vz	bkv/z	-	bkz/v
<b>bk/vz</b>	k/bvz	-	-/bvzk	-

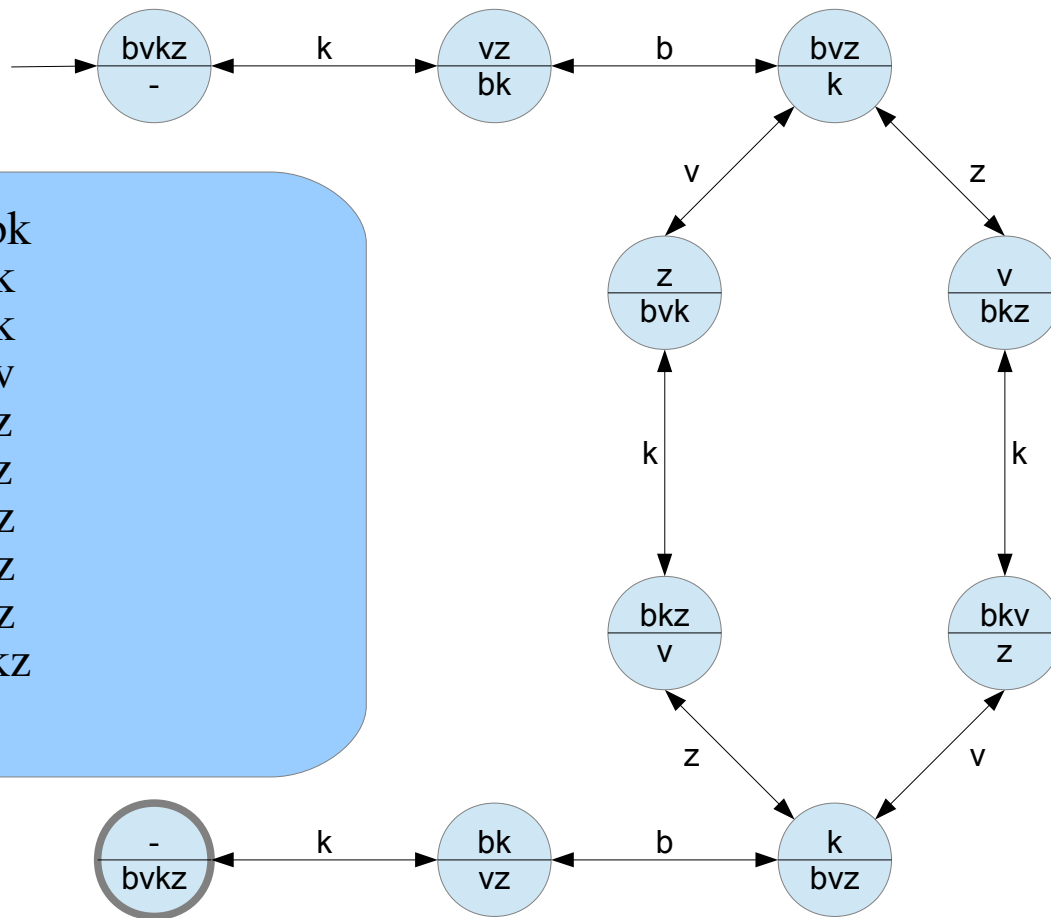




# Funkcija prehodov $\delta(q, x)$

- Spisek prehodov

- $\delta(bvkz/-, k) = vz/bk$
- $\delta(vz/bk, b) = bvz/k$
- $\delta(bvz/k, v) = z/bvk$
- $\delta(z/bvk, k) = bkz/v$
- $\delta(bkz/v, z) = k/bvz$
- $\delta(bvz/k, z) = v/bkz$
- $\delta(v/bkz, k) = bkz/v$
- $\delta(bkv/z, v) = k/bvz$
- $\delta(k/bvz, b) = bk/vz$
- $\delta(bk/vz, k) = -/bvkz$
- ... kaj še manjka?





# Jezik avtomata

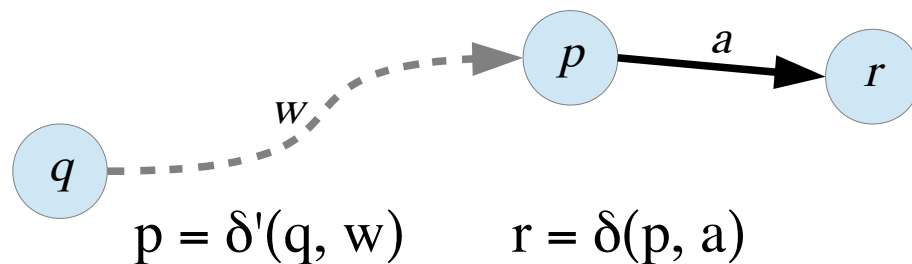
- Avtomat kot razpoznavalnik jezika
  - Bere vhodni niz simbol za simbolom
  - Ko prebere celoten niz se *ustavi*
  - Izvaja prehode glede na *trenutno stanje* in *prebrani simbol*
  - Če se ustavi v stanju  $q \in F$ , potem reče **da**, sicer **ne**





# Jezik avtomata

- Funkcija prehodov
  - Razširimo iz simbolov na nize.
  - $\delta'(q, \varepsilon) = q$
  - $\delta'(q, wa) = \delta(\delta'(q, w), a)$



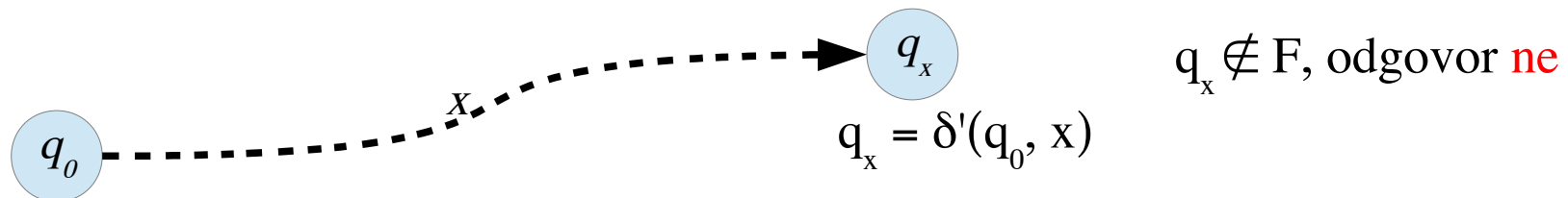
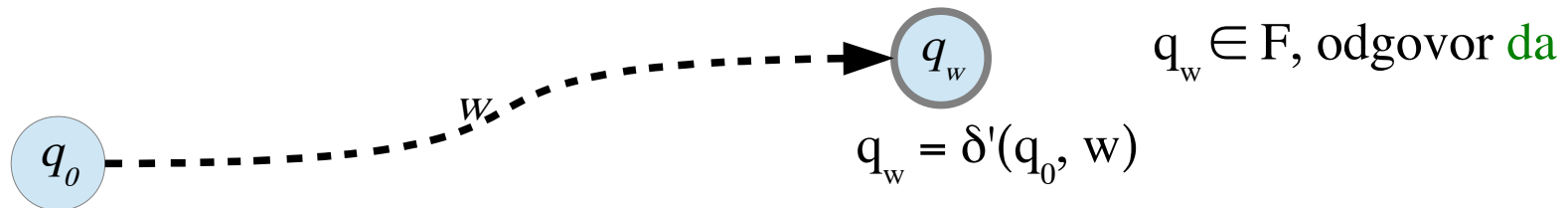


# Jezik avtomata

- Formalna definicija jezika

- Množica vseh besed, ki iz  $q_0$  pripeljejo v poljubno končno stanje

- $L(A) = \{ w \in \Sigma^* \mid \delta'(q_0, w) \in F \}$





# Jezik avtomata

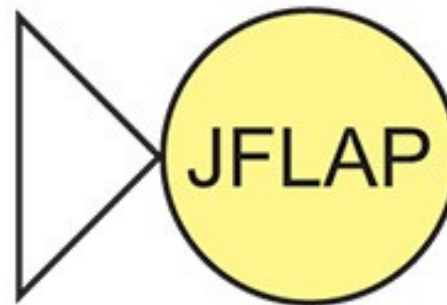
- Avtomat sprejema nize
  - Če  $w \in L(A)$ , potem pravimo, da  $A$  sprejema  $w$
  - Avtomat sprejema le en jezik
  - Lahko sprejme veliko različnih nizov
  - Lahko ne sprejme nobenega
    - prazen jezik





# Gremo v JFlap

- [JFlap.org](http://JFlap.org)



Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko



# Univerzalni avtomat UDKA

8. marec  
2014





# Kako deluje avtomat?

- Avtomat kot razpoznavalnik jezika
  - Bere vhodni niz simbol za simbolom
  - Ko prebere celoten niz se *ustavi*
  - Izvaja prehode glede na *trenutno stanje* in *prebrani simbol*
  - Če se ustavi v stanju  $q \in F$ , potem reče **da**, sicer **ne**





# Univerzalni avtomat

- Je avtomat
- Zna oponašati avtomate
  - Simulator avtomatov
- Katere?
  - **VSE** (univerzalnost)
    - vse končne avtomate





# Simulacija

A

Simulirani avtomat  
*simuliranec*

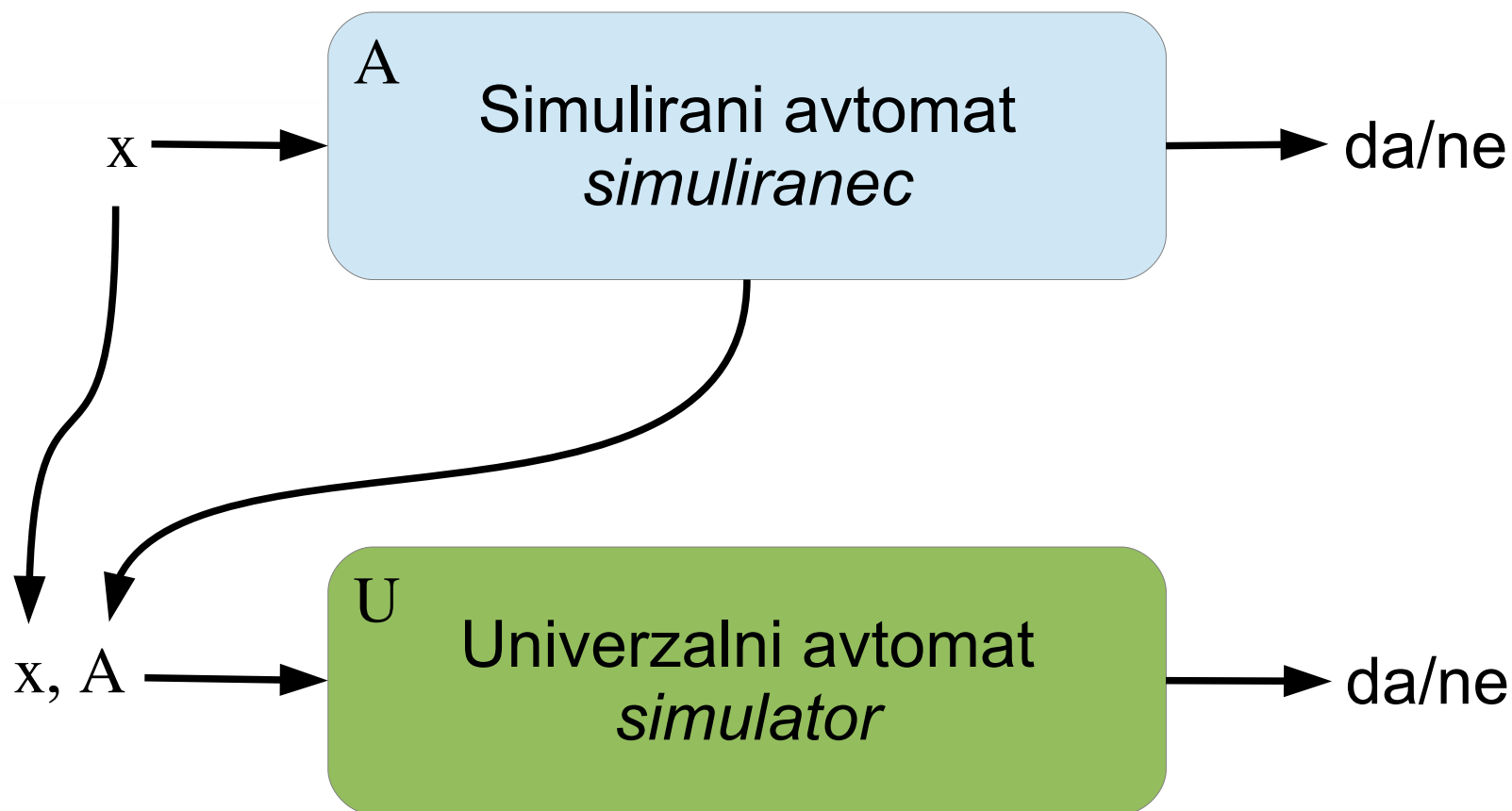
U

Univerzalni avtomat  
*simulator*





# Vhod





# Gremo v python

- Python online interpreter
  - [http://www.compileonline.com/execute\\_python\\_online.php](http://www.compileonline.com/execute_python_online.php)
- Ali lokalno
  - SicTE urejevalnik





# Viri

- Knjiga: Introduction to Automata Theory, Languages, and Computation – Hopcroft, Motwani, Ullman
  - Kasnejše izdaje so manj zahtevne
- Wikipedia
  - Večina slik
- Coursera
  - Tečaj o avtomatih, Ullman

