

**Proszę nie przewijać tych slajdów myszką.  
Zamiast tego proszę je pobrać na dysk,  
otworzyć, powiększyć (Ctrl+plus), a potem  
naciskać klawisz ze strzałką w prawo**

---

# Wyrażenia regularne

---

dr inż. Marcin Ciura

`marcin.ciura@uken.krakow.pl`

Wydział Bezpieczeństwa i Informatyki UKEN

# Plan na dziś

- Wyrażenia regularne
- Automaty
- Wyrażenia regularne w języku Python
- ...a ponadto wiadomości o dwóch matematykach, kilka zagadek i kilka ciekawostek

# Wyrażenia regularne

---

## Co to są wyrażenia regularne?

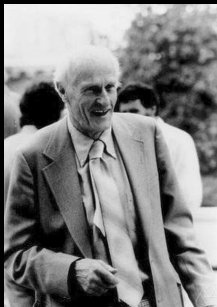
Wyrażenia regularne to programy, które służą do wyszukiwania wzorców w tekstach

Te programy są zapisane w specjalizowanym języku programowania :-)

Programy w tym języku można zagnieżdżać w programach, napisanych w innych językach programowania

Kto wymyślił wyrażenia regularne?

Stephen Cole Kleene (5.1.1909–25.1.1994)



Amerykański matematyk, znany z pewnej gwiazdki. Stworzył teorię obliczalności. Wspinał się po górach, działał na rzecz ochrony przyrody. W 1951 roku, kiedy badał sieci neuronowe, wymyślił wyrażenia regularne

## Przykład

Każdy gen RNA pasuje do tego wyrażenia regularnego:

$(\text{AUG}|\text{CUG}|\text{UUG})(\dots)^*(\text{UAA}|\text{UAG}|\text{UGA})$

Regex i regexp to skróty angielskiego wyrażenia **regular expression**

Programiści często używają tych skrótów zamiast pełnej nazwy



# Definicje

- **Alfabet**: skończony zbiór **symboli** czyli **znaków**,  
na przykład {**A**, **C**, **G**, **T**}, ASCII, Unicode...
- **łańcuch znaków**, krócej **łańcuch**: to samo, co ciąg znaków,  
na przykład **ATGTGA**
- **Długość łańcucha**: liczba znaków w tym łańcuchu. Oznaczamy długość łańcucha, obejmując ten łańcuch kreskami pionowymi,  
na przykład  $|\text{ATGTGA}| = 6$
- **łańcuch pusty**: łańcuch o długości 0 znaków. Oznaczamy łańcuch pusty grecką literą epsilon:  $\epsilon$

- **Wyrażenie regularne** to taki łańcuch, który opisuje pewien zbiór łańcuchów zgodnie z pewnymi **regułami**  
Mówimy o każdym łańcuchu z tego zbioru, że ten łańcuch **pasuje** do danego wyrażenia regularnego  
Możemy też powiedzieć, że **zbiór łańcuchów** opisany przez pewne wyrażenie regularne **pasuje** do tego wyrażenia regularnego

# Elementarne wyrażenia regularne

Każde elementarne wyrażenie regularne składa się z jednego symbolu

- Do symbolu zbioru pustego  $\emptyset$  pasuje pusty zbiór łańcuchów:  $\emptyset$
- Do symbolu pustego łańcucha  $\epsilon$  pasuje zbiór  $\{\epsilon\}$ , który zawiera tylko łańcuch pusty
- Do każdego wyrażenia regularnego złożonego z jednego znaku pasuje jeden łańcuch o długości 1. Ten łańcuch składa się z tego samego znaku, co wyrażenie regularne

Na przykład do wyrażenia regularnego  $a$  pasuje zbiór łańcuchów  $\{a\}$  :-)

## Złożone wyrażenia regularne

- nawiasy
- konkatenacja wyrażeń regularnych
- alternatywa wyrażeń regularnych
- gwiazdka Kleene'a

## Złożone wyrażenia regularne: nawiasy

Jeśli  $R$  jest wyrażeniem regularnym, to:

- wyrażenie regularne  $(R)$  oznacza to samo, co wyrażenie regularne  $R$  :-)

## Złożone wyrażenia regularne: konkatenacja

Jeśli  $R$  i  $S$  są wyrażeniami regularnymi, to:

- Do wyrażenia regularnego  $RS$ , czyli do **konkatenacji** wyrażeń regularnych  $R$  i  $S$ , pasują takie łańcuchy, które powstają, gdy łączymy dowolny łańcuch pasujący do  $R$  z dowolnym łańcuchem pasującym do  $S$

Wyraz konkatenacja pochodzi od angielskiego czasownika *to concatenate*, czyli **łączyć w łańcuch**

## Złożone wyrażenia regularne: konkatenacja

Przykłady konkatenacji wyrażeń regularnych:

- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{d\}$ ,  
a do  $S$  pasuje zbiór łańcuchów  $\{o\}$ ,  
to do  $RS$  pasuje zbiór łańcuchów  $\{do\}$
- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{do, od\}$ ,  
a do  $S$  pasuje zbiór łańcuchów  $\{dać, pisać\}$ ,  
to do  $RS$  pasuje zbiór łańcuchów  $\{dodać, dopisać, oddać, odpisać\}$

## Złożone wyrażenia regularne: alternatywa

Jeśli  $R$  i  $S$  są wyrażeniami regularnymi, to:

- Do wyrażenia regularnego  $R|S$ , czyli do **alternatywy** wyrażeń regularnych  $R$  i  $S$ , pasuje suma dwóch zbiorów: zbioru takich łańcuchów, które pasują do  $R$  i zbioru takich łańcuchów, które pasują do  $S$



## Złożone wyrażenia regularne: alternatywa

Przykłady alternatywy wyrażen regularnych:

- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{do\}$ ,  
a do  $S$  pasuje zbiór łańcuchów  $\{od\}$ ,  
to do  $R|S$  pasuje zbiór łańcuchów  $\{do, od\}$
- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{ręka, noga\}$ ,  
a do  $S$  pasuje zbiór łańcuchów  $\{ręka, głowa\}$ ,  
to do  $R|S$  pasuje zbiór łańcuchów  $\{ręka, noga, głowa\}$

## Złożone wyrażenia regularne: gwiazdka Kleene'a

Jeśli  $R$  jest wyrażeniem regularnym, to:

- Do wyrażenia regularnego  $R^*$ , czyli do domknięcia Kleene'a wyrażenia regularnego  $R$  pasuje zbiór takich łańcuchów, które powstają, gdy łączymy zero lub więcej łańcuchów pasujących do  $R$   
Domknięcie Kleene'a nazywa się również gwiazdką Kleene'a

## Złożone wyrażenia regularne: gwiazdka Kleene'a

Przykłady użycia gwiazdki Kleene'a:

- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{A\}$ , to do  $R^*$  pasuje zbiór łańcuchów  $\{\epsilon, A, AA, AAA, AAAA, AAAAA, AAAAAA, AAAAAA, AAAAAA, \dots\}$
- Jeśli do  $R$  pasuje zbiór łańcuchów  $\{fa, sol\}$ , to do  $R^*$  pasuje zbiór łańcuchów  
 $\{\epsilon,$   
     $fa, sol,$   
     $fafa, fasol, solfa, solsol,$   
     $fafafa, fafasol, fasolfa, fasolsol,$   
     $solfafa, solfasol, solsolfa, solsolsol,$   
     $\dots$   
     $\}$

## Złożone wyrażenia regularne: kolejność działań

Wykonujemy działania w takiej kolejności:

- najpierw wykonujemy działania w nawiasach
- potem stosujemy gwiazdkę Kleene'a
- potem konkatenujemy wyrażenia regularne
- potem budujemy alternatywy wyrażen regularnych

# Zagadka

Które z dwóch wyrażeń regularnych po prawej stronie znaku = jest równoważne wyrażeniu regularnemu po lewej stronie znaku =?

1.  $ab^* = a(b^*)$  czy  $(ab)^*$ ?
2.  $a|b^* = a|(b^*)$  czy  $(a|b)^*$ ?
3.  $ab|cd = (ab)|(cd)$  czy  $a(b|c)d$ ?

Kolejność działań: nawiasy, gwiazdka, konkatenacja, alternatywa

## Zagadka: rozwiązanie

Które z dwóch wyrażeń regularnych po prawej stronie znaku = jest równoważne wyrażeniu regularnemu po lewej stronie znaku =?

1.  $ab^* = a(b^*)$

2.  $a|b^* = a|(b^*)$

3.  $ab|cd = (ab)|(cd)$

Kolejność działań: nawiasy, gwiazdka, konkatenacja, alternatywa

**Proszę Państwa o pytania :-)**

---

# Automaty

---



Amerykański matematyk i informatyk. W 1956 roku zaproponował konstrukcję takich maszyn, które same będą się rozmnażały. Przypuszczał, że takie maszyny będzie można produkować łatwiej niż statki kosmiczne. W tym samym roku wymyślił taki rodzaj automatu, o jakim będzie mowa na tym wykładzie

# Pralka automatyczna



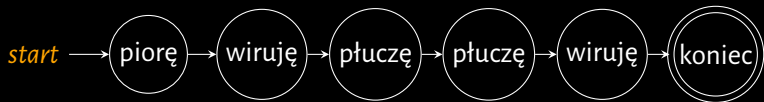
Nie, Edward Forrest Moore nie wynalazł pralki automatycznej :-)

Edward Forrest Moore wymyślił automat skończony

Automat skończony to taki automat, który ma początek i koniec

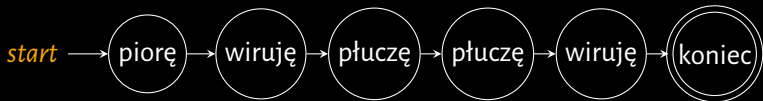
Zrozumiemy automaty skończone dzięki pralce automatycznej

## Jak działa pralka automatyczna?



Pralka automatyczna wie, co robi, ale nie pamięta, jak do tego doszła :-)

## Jak działa pralka automatyczna?



Kółka oznaczają stany pralki automatycznej

Strzałki oznaczają przejścia między stanami pralki automatycznej

Pralka automatyczna wie, w jakim stanie jest, ale nie pamięta, jak do niego doszła :-)

## Automaty skończone

Automaty skończone różnią się od pralki automatycznej tym, że wczytują znaki

Automaty skończone zmieniają swój stan po każdym wczytanym znaku

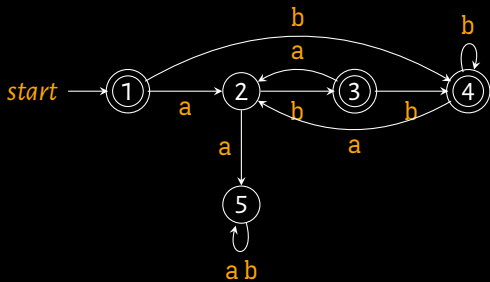
Automat skończony wie, w jakim stanie jest, ale nie pamięta, jak do niego doszedł :-)



Są proste przepisy, czyli algorytmy, które przekształcają wyrażenia regularne na automaty skończone

## Tabela przejść i wyjść automatu skończonego

Tak komputer przechowuje w pamięci automaty skończone:



| stan | końcowy? | a | b |
|------|----------|---|---|
| 1    | true     | 2 | 4 |
| 2    | false    | 5 | 3 |
| 3    | true     | 2 | 4 |
| 4    | true     | 2 | 4 |
| 5    | false    | 5 | 5 |

**Proszę Państwa o pytania :-)**

---



# Wyrażenia regularne w języku Python

---

# Wyrażenia regularne w języku Python

Ta część wykładu jest zapisana w notatniku Jupytera

`Wyrażenia-regularne-w-Pythonie.ipynb`

Tę część wykładu opracowałem na podstawie dokumentacji modułu `re`

Dokumentacja tego pakietu znajduje się pod adresem

`https://docs.python.org/3/library/re.html` i na dysku lokalnym

Widzę tę dokumentację na ekranie, gdy w interpreterze Pythona wydaję polecenia

```
import re
```

```
help(re)
```

**Proszę Państwa o pytania :-)**

---

## Inne zastosowania wyrażeń regularnych

Do tego regexa pasują daty w formacie ISO 8601, na przykład 1791-05-03:

```
^([0-9][0-9][0-9][0-9])-(0[0-9]|1[0-2])-([0-2][0-9]|3[01])$
```

## Inne zastosowania wyrażień regularnych

Zwykle używamy przyimka „we” przed takimi wyrazami, które zaczynają się dwiema spółgłoskami, przy czym pierwsza spółgłoska to „f” lub „w”:

we Francji, we Wrocławiu, we foyer

Przed tymi wyrazami też używamy przyimka „we”:

we dwoje, we troje, we czworo

we mnie

we Lwowie, we łbie, we łbach, we łzach, we mgle, we mgłach, we śnie

## Inne zastosowania wyrażeń regularnych

Używamy przyimka „we” przed wyrazami, które pasują do tego wyrażenia regularnego:

```
^([fvw]([bcdfghjklłmnprstwzż]|o[iy])  
|dwoje$|troje$|czworo$|mnie$  
|lwow|łb|łz|mg[lł]|śn)
```

<https://kobietydokodu.pl/4-wyrazenia-regularne/>

**Dziękuję Państwu za uwagę :-)**

---



## Slajdy dodatkowe

---

# Deterministyczne automaty skończone

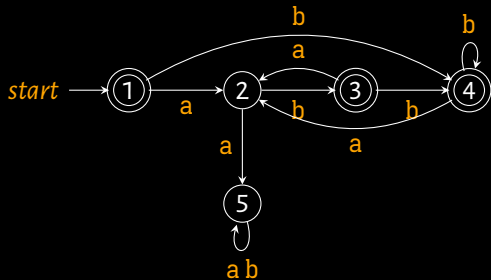
DFA (deterministyczny automat skończony, po angielsku: **Deterministic Finite Automaton**)

Z jednego stanu DFA wychodzi dokładnie jedno przejście oznaczone danym znakiem

Po wczytaniu każdego znaku DFA przechodzi do nowego stanu

## Przykład DFA

$(b^*(a|\varepsilon)b)^*$



# Niedeterministyczne automaty skończone

NFA (niedeterministyczny automat skończony, po angielsku:

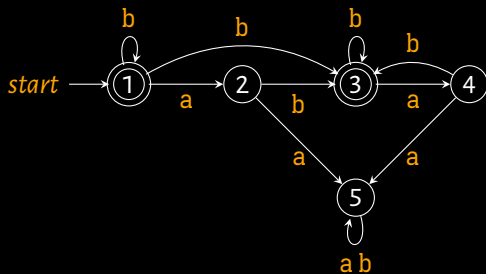
**Nondeterministic Finite Automaton**)

Z jednego stanu NFA może wychodzić więcej niż jedno przejście oznaczone tak samo

Po wczytaniu każdego znaku NFA przechodzi do jednego lub więcej stanów naraz

## Przykład NFA

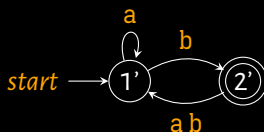
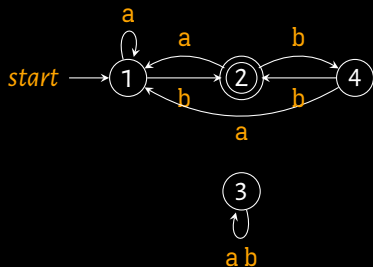
$(b^*(a|\epsilon b)^*)$



# Równoważne automaty skończone

Mówimy, że dwa automaty skończone są równoważne, jeśli rozpoznają te same zbiory łańcuchów

Oto przykład dwóch równoważnych automatów skończonych, NFA i DFA:



[https://en.wikipedia.org/wiki/Stephen\\_Cole\\_Kleene](https://en.wikipedia.org/wiki/Stephen_Cole_Kleene)

<https://my-concept.pl/pl/p/Pralka-automatyczna-SLIM-PP6306s/20820>