

Teoria Automatów i Języków Formalnych

Ćwiczenia 2: Wyrażenia regularne

dr inż. Marcin Luckner
mluckner@mini.pw.edu.pl

Wydział Matematyki i Nauk Informatycznych

Wersja 1.3
3 marca 2021

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. „Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informatycznych”, realizowane w ramach projektu „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Wyrażenie regularne

Wyrażenie regularne nad alfabetem Σ jest konstruowane w następujący sposób

- \emptyset jest wyrażeniem regularnym,
- ϵ jest wyrażeniem regularnym,
- Każde $a \in \Sigma$ jest wyrażeniem regularnym,
- Jeżeli r i s są wyrażeniami regularnymi to
 - $(r + s)$ suma wyrażen regularnych,
 - (rs) konkatencja wyrażen regularnych,
 - $(r)^*$ domknięcie Kleeneasą wyrażeniami regularnymi.

Języki generowane przez wyrażenia

Wyrażenie regularne nad alfabetem Σ generują następujące języki:

- \emptyset generuje język pusty \emptyset
- ϵ generuje język $\{\epsilon\}$
- każda litera $a \in \Sigma$ generuje język $\{a\}$
- Jeżeli r i s są wyrażeniami regularnymi to
 - $(r + s)$ generuje język $R \cup S$
 - (rs) generuje język RoS
 - $(r)^*$ generuje język R^*

Właściwości wyrażeń regularnych

Zachodzą następujące równości

1. $\emptyset + r = r + \emptyset = r$
2. $\emptyset r = r \emptyset = r$
3. $\epsilon + r = r + \epsilon$
4. $\epsilon r = r \epsilon = r$
5. $r + s = s + r$
6. $(r + s) + t = r + (s + t) = r + s + t$
7. $(rs)t = r(st) = rst$
8. $r(s + t) = rs + rt$
9. $(r + s)t = rt + st$
10. $(r^*)^* = r^*$
11. $(r^*s^*)^* = (r + s)^*$
12. $(r^* + s^*)^* = (r + s)^*$

Konstrukcja wyrażenia regularnego I

Skonstruuj wyrażenie regularne dla następującego języka
Język L nad alfabetem $\Sigma = \{a, A, 9\}$ zawiera słowa, które są niepoprawnymi hasłami. Poprawne hasło składa się z przynajmniej 3 liter i zawiera wszystkie litery z alfabetu.

$A9999aaa \in L$, $aaa9 \notin L$

Konstrukcja wyrażenia regularnego II

- Poprawne hasło składa się z przynajmniej 3 liter

$$(a + A + 9)(a + A + 9)(a + A + 9)(a + A + 9)^*$$

- Poprawne hasło zawiera wszystkie litery z alfabetu

$$\begin{aligned} &(a + A + 9)^* a(a + A + 9)^* A(a + A + 9)^* 9(a + A + 9)^* + \\ &(a + A + 9)^* a(a + A + 9)^* 9(a + A + 9)^* A(a + A + 9)^* + \\ &(a + A + 9)^* A(a + A + 9)^* a(a + A + 9)^* 9(a + A + 9)^* + \\ &(a + A + 9)^* A(a + A + 9)^* 9(a + A + 9)^* a(a + A + 9)^* + \\ &(a + A + 9)^* 9(a + A + 9)^* a(a + A + 9)^* A(a + A + 9)^* + \\ &(a + A + 9)^* 9(a + A + 9)^* A(a + A + 9)^* a(a + A + 9)^* \end{aligned}$$

Konstrukcja wyrażenia regularnego III

- Niepoprawne hasło z mniej niż trzema literami

$$r_1 = (a + A + 9 + \epsilon)(a + A + 9 + \epsilon)$$

- Niepoprawne hasło bez jakiejś litery

$$r_1 = (a + A)^* + (a + 9)^* + (9 + A)^*$$

- Kombinacja

$$(a + A)^* + (a + 9)^* + (9 + A)^*$$

ponieważ $L(r_2) \subset L(r_1)$

Konstrukcja wyrażenia regularnego IV

- Pierwszy z warunków, że hasło ma zawierać wszystkie typy znaków jest stosowany w praktyce.
- Warunek dotyczący długości jest niepraktyczny, bo hasła zazwyczaj muszą być dłuższe.
- Jednakże możemy określić niepoprawne hasła dla dowolnej długości n

$$(a + A)^* + (a + 9)^* + (9 + A)^* + \underbrace{(a + A + 9 + \epsilon)(a + A + 9 + \epsilon) \dots (a + A + 9 + \epsilon)}_{n-1}$$

Zadania I

1. Skonstruuj wyrażenie regularne generujące L i $\Sigma^* \setminus L$
 - 1.1 Język L nad alfabetem $\Sigma = \{a, b\}$ słów bez trzech sąsiadujących identycznych liter. $aabbaa \in L$, $aaab \notin L$
 - 1.2 Język L nad alfabetem $\Sigma = \{0, 1\}$ zawierający binarne reprezentacje liczb nieparzystych bez zer wiodących. $10100101 \in L$, $10100100 \notin L$

Zadania II

2. Skonstruuj wyrażenie regularne dla następującego języka

2.1 Język L nad alfabetem $\Sigma = \{0, 1\}$ zachowanie użytkowników internetowego konta bankowego. 0 to nieudane logowanie, 1 to udane logowanie. System blokuje konto, jeżeli po dwóch kolejnych nieudanych logowaniach nie nastąpiły przynajmniej dwa udane logowania przed wystąpieniem kolejnej pary nieudanych logowań. Słowa języka opisują zachowanie użytkowników, które nie prowadzą do zablokowania konta. Dla zablokowanego konta można nadal podejmować próby logowania $10100101 \in L$, $10100100 \notin L$