

PII - prvi kolokvijum - januar 2007. - prva grupa

- Definisati:
 - nedeterministički KA sa ε -prelazom, pojam uspešnog izračunavanja u njemu i jezik koji prihvatata
 - skresan KA
- Definisati minimalni automat preko količnika
- Dati definiciju regularnog izraza u postfiksnoj notaciji koristeći operatore $+$, \cdot i $*$, a onda zapisati u postfiksnoj notaciji izraz: $((a+b) \cdot (a \cdot b)) \cdot ((a+b) \cdot *)$
- Da li je proizvod dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Ko inicijalizuje leksički analizador? Opisati sažeto osnovne metode konstrukcije leksičkog analizatora.
- Bez pozivanja na konačne automate pokazati da važi: $(a|b)^* = (a^*b)^*a^*$
- Korišćenjem Thompsonovog algoritma konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem algoritma Glušкова konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{4\}$ $d = \{(1, a, 2), (1, b, 1), (2, a, 1), (2, b, 2), (2, \varepsilon, 3), (3, a, 3), (3, b, 4), (4, a, 4), (4, b, 3), (4, \varepsilon, 1)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.
 - Bez korišćenja konačnih automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne sadrže faktor 111.

PII - prvi kolokvijum - januar 2007. - prva grupa

- Definisati:
 - nedeterministički KA sa ε -prelazom, pojam uspešnog izračunavanja u njemu i jezik koji prihvatata
 - skresan KA
- Definisati minimalni automat preko količnika
- Dati definiciju regularnog izraza u postfiksnoj notaciji koristeći operatore $+$, \cdot i $*$, a onda zapisati u postfiksnoj notaciji izraz: $((a+b) \cdot (a \cdot b)) \cdot ((a+b) \cdot *)$
- Da li je proizvod dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Ko inicijalizuje leksički analizador? Opisati sažeto osnovne metode konstrukcije leksičkog analizatora.
- Bez pozivanja na konačne automate pokazati da važi: $(a|b)^* = (a^*b)^*a^*$
- Korišćenjem Thompsonovog algoritma konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem algoritma Glušкова konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{4\}$ $d = \{(1, a, 2), (1, b, 1), (2, a, 1), (2, b, 2), (2, \varepsilon, 3), (3, a, 3), (3, b, 4), (4, a, 4), (4, b, 3), (4, \varepsilon, 1)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.
 - Bez korišćenja konačnih automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne sadrže faktor 111.

PII - prvi kolokvijum - januar 2007. - prva grupa

- Definisati:
 - nedeterministički KA sa ε -prelazom, pojam uspešnog izračunavanja u njemu i jezik koji prihvatata
 - skresan KA
- Definisati minimalni automat preko količnika
- Dati definiciju regularnog izraza u postfiksnoj notaciji koristeći operatore $+$, \cdot i $*$, a onda zapisati u postfiksnoj notaciji izraz: $((a+b) \cdot (a \cdot b)) \cdot ((a+b) \cdot *)$
- Da li je proizvod dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Ko inicijalizuje leksički analizador? Opisati sažeto osnovne metode konstrukcije leksičkog analizatora.
- Bez pozivanja na konačne automate pokazati da važi: $(a|b)^* = (a^*b)^*a^*$
- Korišćenjem Thompsonovog algoritma konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem algoritma Glušкова konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{4\}$ $d = \{(1, a, 2), (1, b, 1), (2, a, 1), (2, b, 2), (2, \varepsilon, 3), (3, a, 3), (3, b, 4), (4, a, 4), (4, b, 3), (4, \varepsilon, 1)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.
 - Bez korišćenja konačnih automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne sadrže faktor 111.

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1."  ";  
}
```

ako je

\$REGEX='(?m)^(?=.*?a)(?=.*?b)((?!c d).)*\$';	\$STRING="dabar\nbabino\ncaca\ndaca\n";
\$REGEX='x y+';	\$STRING="xyx yyx yyyxyy";
\$REGEX='[^aeiou][aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';
\$REGEX='\B[da-fA-F]+\B';	\$STRING="Heksadekadni broj 3EAF.";
\$REGEX='min max minimax';	\$STRING='minimax';
\$REGEX='<.+>';	\$STRING='<head><title>Naslov</title></head>';
\$REGEX='i.*i';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?-i)ab(?i)cd';	\$STRING='abCdAbcDabcdABCDabcd';
\$REGEX='(t r)+';	\$STRING="trt rrt rrrttrr";
\$REGEX='\w*?\.\w*?';	\$STRING='image.jpg';
\$REGEX='o.*?o';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?m)^(.*(a b).*\$';	\$STRING='qwerty\nasdf\nvbnm\nzabar';
\$REGEX='a+ b+';	\$STRING="aba bba bbbaab";
\$REGEX='(d+)(,\\s*2)+';	\$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';
\$REGEX='(?<[rst])[aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1."  ";  
}
```

ako je

\$REGEX='(?m)^(?=.*?a)(?=.*?b)((?!c d).)*\$';	\$STRING="dabar\nbabino\ncaca\ndaca\n";
\$REGEX='x y+';	\$STRING="xyx yyx yyyxyy";
\$REGEX='[^aeiou][aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';
\$REGEX='\B[da-fA-F]+\B';	\$STRING="Heksadekadni broj 3EAF.";
\$REGEX='min max minimax';	\$STRING='minimax';
\$REGEX='<.+>';	\$STRING='<head><title>Naslov</title></head>';
\$REGEX='i.*i';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?-i)ab(?i)cd';	\$STRING='abCdAbcDabcdABCDabcd';
\$REGEX='(t r)+';	\$STRING="trt rrt rrrttrr";
\$REGEX='\w*?\.\w*?';	\$STRING='image.jpg';
\$REGEX='o.*?o';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?m)^(.*(a b).*\$';	\$STRING='qwerty\nasdf\nvbnm\nzabar';
\$REGEX='a+ b+';	\$STRING="aba bba bbbaab";
\$REGEX='(d+)(,\\s*2)+';	\$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';
\$REGEX='(?<[rst])[aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1."  ";  
}
```

ako je

\$REGEX='(?m)^(?=.*?a)(?=.*?b)((?!c d).)*\$';	\$STRING="dabar\nbabino\ncaca\ndaca\n";
\$REGEX='x y+';	\$STRING="xyx yyx yyyxyy";
\$REGEX='[^aeiou][aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';
\$REGEX='\B[da-fA-F]+\B';	\$STRING="Heksadekadni broj 3EAF.";
\$REGEX='min max minimax';	\$STRING='minimax';
\$REGEX='<.+>';	\$STRING='<head><title>Naslov</title></head>';
\$REGEX='i.*i';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?-i)ab(?i)cd';	\$STRING='abCdAbcDabcdABCDabcd';
\$REGEX='(t r)+';	\$STRING="trt rrt rrrttrr";
\$REGEX='\w*?\.\w*?';	\$STRING='image.jpg';
\$REGEX='o.*?o';	\$STRING='Prevodioci i interpretatori';
\$REGEX='(?m)^(.*(a b).*\$';	\$STRING='qwerty\nasdf\nvbnm\nzabar';
\$REGEX='a+ b+';	\$STRING="aba bba bbbaab";
\$REGEX='(d+)(,\\s*2)+';	\$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';
\$REGEX='(?<[rst])[aeiou][^aeiou]';	\$STRING='Prevodioci i interpretatori';

PII - prvi kolokvijum - januar 2007. - druga grupa

- Definisati:
 - deterministički KA, pojam uspešnog izračunavanja u njemu i jezik koji prihvata
 - standardan KA
- Definisati minimalni automat preko Nerodove ekvivalencije
- Dati definiciju regularnog izraza u prefiksnoj notaciji koristeći operatore $+$, $.$ i $*$, a onda zapisati u prefiksnoj notaciji izraz: $((a+(b.(((a.(b*)) .a)*.b)))*)$
- Da li je unija dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Definisati token i leksemu. Koje se greške otkrivaju u leksičkoj analizi? Kako se vrši oporavak od greške? Kako komuniciraju leksički i sintaksički analizator?
- Koristeći identitet $(a|b)^* = (a^*b)^*a^*$, pokazati da važi $(b^*a)^* = (a^*b)^*a^+|\varepsilon$
- Korišćenjem algoritma Glušкова konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem Tompsonovog algoritma konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b, c\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{2\}$ $d = \{(1, a, 1), (1, c, 4), (1, \varepsilon, 2), (2, a, 3), (2, b, 2), (2, \varepsilon, 1), (3, \varepsilon, 2), (4, b, 3)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje ne sadrže faktor 111.
 - Bez korišćenja automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.

PII - prvi kolokvijum - januar 2007. - druga grupa

- Definisati:
 - deterministički KA, pojam uspešnog izračunavanja u njemu i jezik koji prihvata
 - standardan KA
- Definisati minimalni automat preko Nerodove ekvivalencije
- Dati definiciju regularnog izraza u prefiksnoj notaciji koristeći operatore $+$, $.$ i $*$, a onda zapisati u prefiksnoj notaciji izraz: $((a+(b.(((a.(b*)) .a)*.b)))*)$
- Da li je unija dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Definisati token i leksemu. Koje se greške otkrivaju u leksičkoj analizi? Kako se vrši oporavak od greške? Kako komuniciraju leksički i sintaksički analizator?
- Koristeći identitet $(a|b)^* = (a^*b)^*a^*$, pokazati da važi $(b^*a)^* = (a^*b)^*a^+|\varepsilon$
- Korišćenjem algoritma Glušкова konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem Tompsonovog algoritma konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b, c\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{2\}$ $d = \{(1, a, 1), (1, c, 4), (1, \varepsilon, 2), (2, a, 3), (2, b, 2), (2, \varepsilon, 1), (3, \varepsilon, 2), (4, b, 3)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje ne sadrže faktor 111.
 - Bez korišćenja automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.

PII - prvi kolokvijum - januar 2007. - druga grupa

- Definisati:
 - deterministički KA, pojam uspešnog izračunavanja u njemu i jezik koji prihvata
 - standardan KA
- Definisati minimalni automat preko Nerodove ekvivalencije
- Dati definiciju regularnog izraza u prefiksnoj notaciji koristeći operatore $+$, $.$ i $*$, a onda zapisati u prefiksnoj notaciji izraz: $((a+(b.(((a.(b*)) .a)*.b)))*)$
- Da li je unija dva deterministička konačna automata deterministički konačni automat? Obrazložiti.
- Definisati token i leksemu. Koje se greške otkrivaju u leksičkoj analizi? Kako se vrši oporavak od greške? Kako komuniciraju leksički i sintaksički analizator?
- Koristeći identitet $(a|b)^* = (a^*b)^*a^*$, pokazati da važi $(b^*a)^* = (a^*b)^*a^+|\varepsilon$
- Korišćenjem algoritma Glušкова konstruisati MDKA za $(a|b^+)(a(ab)^*|b)^+$
 - Korišćenjem Tompsonovog algoritma konstruisati MDKA za $(ca)^+b(a|b|c)^*$
- Transformisati sledeći automat u automat bez ε -pokreta: $\Sigma = \{a, b, c\}$, $Q = \{1, 2, 3, 4\}$ $I = \{1\}$ $F = \{2\}$ $d = \{(1, a, 1), (1, c, 4), (1, \varepsilon, 2), (2, a, 3), (2, b, 2), (2, \varepsilon, 1), (3, \varepsilon, 2), (4, b, 3)\}$
- Konstruisati MDKA koji prepoznaje sve niske nad $\Sigma = \{0, 1\}$ koje ne sadrže faktor 111.
 - Bez korišćenja automata, regularnim izrazom opisati sve niske nad $\Sigma = \{0, 1\}$ koje se ne završavaju sa 111.

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1." ";  
}
```

ako je

```
$REGEX='a|b+';  
$REGEX='i.*?i';  
$REGEX='\B[da-fA-F]+\B';  
$REGEX='min|max|minimax';  
$REGEX='<.+>';  
$REGEX='w*?\w*?';  
$REGEX='(?![rst])[aeiou][^aeiou]';  
$REGEX='i.*i';  
$REGEX='(?-i)ab(?i)cd';  
$REGEX='(a|b)+';  
$REGEX='(?m)~.*(a|b).*$';  
$REGEX='a+|b+';  
$REGEX='(?m)^(?=.*a)(?=.*b)((?!c|d).)*$';  
$REGEX='(d+)(,\\s*2)+';  
$REGEX='[^aeiou][aeiou][^aeiou]';
```

```
$STRING="aba bba bbbaab";  
$STRING='Prevodioci i interpretatori';  
$STRING="Heksadekadni broj 3EAF.";  
$STRING='minimax';  
$STRING='<head><title>Naslov</title></head>';  
$STRING='dat.txt';  
$STRING='Prevodioci i interpretatori';  
$STRING='Prevodioci i interpretatori';  
$STRING='abCdAbcDabcdABCDabcd';  
$STRING="aba bba bbbaab";  
$STRING='qwerty\\nasdf\\nvbnm\\nzabar';  
$STRING="aba bba bbbaab";  
$STRING="dabar\\nbabino\\ncaca\\ndaca\\n";  
$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';  
$STRING='Prevodioci i interpretatori';
```

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1." ";  
}
```

ako je

```
$REGEX='a|b+';  
$REGEX='i.*?i';  
$REGEX='\B[da-fA-F]+\B';  
$REGEX='min|max|minimax';  
$REGEX='<.+>';  
$REGEX='w*?\w*?';  
$REGEX='(?![rst])[aeiou][^aeiou]';  
$REGEX='i.*i';  
$REGEX='(?-i)ab(?i)cd';  
$REGEX='(a|b)+';  
$REGEX='(?m)~.*(a|b).*$';  
$REGEX='a+|b+';  
$REGEX='(?m)^(?=.*a)(?=.*b)((?!c|d).)*$';  
$REGEX='(d+)(,\\s*2)+';  
$REGEX='[^aeiou][aeiou][^aeiou]';
```

```
$STRING="aba bba bbbaab";  
$STRING='Prevodioci i interpretatori';  
$STRING="Heksadekadni broj 3EAF.";  
$STRING='minimax';  
$STRING='<head><title>Naslov</title></head>';  
$STRING='dat.txt';  
$STRING='Prevodioci i interpretatori';  
$STRING='Prevodioci i interpretatori';  
$STRING='abCdAbcDabcdABCDabcd';  
$STRING="aba bba bbbaab";  
$STRING='qwerty\\nasdf\\nvbnm\\nzabar';  
$STRING="aba bba bbbaab";  
$STRING="dabar\\nbabino\\ncaca\\ndaca\\n";  
$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';  
$STRING='Prevodioci i interpretatori';
```

10. Šta ispisuje sledeći PERL program

```
while($STRING =~ m/($REGEX)/g) {  
    print $1." ";  
}
```

ako je

```
$REGEX='a|b+';  
$REGEX='i.*?i';  
$REGEX='\B[da-fA-F]+\B';  
$REGEX='min|max|minimax';  
$REGEX='<.+>';  
$REGEX='w*?\w*?';  
$REGEX='(?![rst])[aeiou][^aeiou]';  
$REGEX='i.*i';  
$REGEX='(?-i)ab(?i)cd';  
$REGEX='(a|b)+';  
$REGEX='(?m)~.*(a|b).*$';  
$REGEX='a+|b+';  
$REGEX='(?m)^(?=.*a)(?=.*b)((?!c|d).)*$';  
$REGEX='(d+)(,\\s*2)+';  
$REGEX='[^aeiou][aeiou][^aeiou]';
```

```
$STRING="aba bba bbbaab";  
$STRING='Prevodioci i interpretatori';  
$STRING="Heksadekadni broj 3EAF.";  
$STRING='minimax';  
$STRING='<head><title>Naslov</title></head>';  
$STRING='dat.txt';  
$STRING='Prevodioci i interpretatori';  
$STRING='Prevodioci i interpretatori';  
$STRING='abCdAbcDabcdABCDabcd';  
$STRING="aba bba bbbaab";  
$STRING='qwerty\\nasdf\\nvbnm\\nzabar';  
$STRING="aba bba bbbaab";  
$STRING="dabar\\nbabino\\ncaca\\ndaca\\n";  
$STRING='15, 15, 16, 17, 18, 18, 18, 20, 23';  
$STRING='Prevodioci i interpretatori';
```