Rozwiązania zadań testowych autorstwa studentów: Jakub Czapiga, Piotr Lubczyński, Marcin Czerenko, Karol Bieleń, Rafał Zemła, Wojciech Musiał, Andrzej Kępka, Oleksandr Masliukivskyi, Jan Malina

1.1. Dopełnieniem języka **a*b*** jest

[NIE] **(A)** **b*a***
pojedynczy symbol *a* zawiera się w obu językach

[NIE] **(B)** $\mathbf{b^+a^+}$
brak słowa *aba*, które nie występuje w języku **a*b***

[TAK] **(C)** $\mathbf{a^*b^+a(a|b)^*}$
jest to dopełnienie języka **a*b***

[NIE] **(D)** $\mathbf{(a^+b^*a^+|b^+a^+)(a|b)^*}$
istnieje możliwość utworzenia ciągu samych symboli *a*

[TAK] **(E)** $\mathbf{a^*b^+a^+(a|b)^*}$
jest to dopełnienie języka **a*b***

[NIE] **(F)** $\mathbf{a^*b^+a(a|b)^*|\varepsilon}$
słowo puste może występować w obu językach

[NIE] **(G)** język regularny, ale żaden z powyższych
języki z podpunktu (C) oraz (E) spełniają warunek

[NIE] **(H)** język nie będący językiem regularnym
języki regularne są zamknięte ze względu na dopełnienie

1.2. Język nad alfabetem $\Sigma = \{0, 1\}$ będący zbiorem wszystkich niepustych łańcuchów zerojedynkowych, w których każde dwa zera przedzielone są co najmniej jedną jedynką, może być opisany następującym wyrażeniem regularnym:

[NIE] **(A)** **1*(011*)*01***
brakuje np. słowa *1*

[NIE] **(B)** **1*|1*(011*)*01***
zawiera słowo puste

[NIE] **(C)** **(1|01*0)(1|01*0)***
pierwszy nawias dopuszcza dwa zera stojące obok siebie

[TAK] **(D)** **11*|1*(011*)*01***

11* z całą pewnością spełnia warunki, natomiast zawartość nawiasu w **1*(011*)*01*** gwarantuje rozdzielenie zer co najmniej jedną jedynką

[NIE] **(E)** **(1|01*0)***
dopuszcza dwa zera stojące obok siebie

[NIE] **(F)** **1*01*(101*)***
brakuje np. słowa *1*

[NIE] **(G)** **1*01*(101*)*|1***
zawiera słowo puste

[TAK] **(H)** **1*01*(101*)*|1*1**
dopuszcza słowa złożone z samych jedynek, gwarantuje rozdzielanie zer co najmniej jedną jedynką

1.3. Jeżeli **r**, **s**, **t** są wyrażeniami regularnymi, to zachodzą następujące tożsamości:

[TAK] **(A)** **r∅ = ∅r = ∅**
zbiór pusty jest wynikiem konkatenacji zbioru pustego z dowolnym zbiorem

[TAK] **(B)** **r|∅ = ∅|r = r**
zbiór pusty jest elementem neutralnym sumy teoriomnogościowej

[TAK] **(C)** **r|r = r**
wynikiem sumy teoriomnogościowej dwóch identycznych zbiorów jest ten właśnie zbiór

[NIE] **(D)** **r|ε = ε|r = r**
Jeżeli wyrażenie **r** nie zawiera słowa pustego, to równość nie zachodzi

[TAK] **(E)** **r(s|t) = rs|rt**
Zarówno lewa i prawa strona reprezentuje łańcuchy rozpoczynające się wyrażeniem **r**, a kończące wyrażeniem **s** lub **t**

[TAK] **(F)** **∅* = ε**
domknięciem Kleene'ego zbioru pustego jest zbiór zawierający słowo puste

[TAK] **(G)** **ε* = ε**
domknięciem Kleene'ego zbioru zawierającego słowo puste jest ten zbiór

[NIE] **(H)** **r(rs|s)*r = rr*s(rr*s)*r**
kontrprzykład: tylko prawa strona może wygenerować łańcuchy rozpoczynające i kończące się literą *r*, w którym przynajmniej dwa razy litera *s* występuje obok siebie: **rsss*r**

1.4. Dopełnieniem języka **ab*** jest

[NIE] **(A)** **b*a***
pojedynczy symbol *a* zawiera się w obu językach

[NIE] **(B)** $\mathbf{b^+a^+}$
słowo puste nie występuje w obu językach

[NIE] **(C)** **ab*a(a|b)***
słowo puste nie występuje w obu językach

[NIE] **(D)** **b(a|b)***
słowo puste nie występuje w obu językach

[NIE] **(E)** **(b|ab*a)(a|b)***
słowo puste nie występuje w obu językach

[TAK] **(F)** **(b|ab*a)(a|b)*|ε**
dopełnienie zawiera wszystkie słowa nienależące do języka **ab***

[NIE] **(G)** język regularny, ale żaden z powyższych
Język z podpunktu (F) spełnia warunek

[NIE] **(H)** język nie będący językiem regularnym
Języki regularne są zamknięte ze względu na dopełnienie

1.5. Język nad alfabetem $\Sigma = \{0, 1\}$ będący zbiorem wszystkich niepustych łańcuchów zerojedynkowych, w których liczba zer jest parzysta, może być opisany następującym wyrażeniem regularnym:

[NIE] **(A)** **11|(0|1)1(0|1)|(0|1)1(0|1)*1(0|1)**
ponieważ **(0|1)*** generuje dowolną ilość zer

[NIE] **(B)** **1*|1*(011*)*01***
ponieważ **(011*)*** generuje dowolną ilość zer

[TAK] **(C)** **(1|01*0)(1|01*0)***
ponieważ najpierw występuje jedynka lub dwa zera z dowolnym ciągiem jedynek pomiędzy nimi, a następnie taki sam ciąg dowolną ilość razy

[NIE] **(D)** **11*|1*(011*)*01***
ponieważ **(011*)*** generuje dowolną ilość zer

[NIE] **(E)** **(1|01*0)***
ponieważ zawiera słowo puste

[NIE] **(F)** **(1*|1*01*01*)*|11***
ponieważ zawiera słowo puste

[NIE] **(G)** **(1*|1*01*01*)*|11*|1*01*01***
ponieważ zawiera słowo puste

[TAK] **(H)** **(1*|1*01*01*)*(11*|1*01*01*)**
ponieważ lewy nawias tworzy ciąg jedynek lub par zer z dowolną ilością jedynek z każdej strony, prawy nawias to prawie to samo, ale wyklucza możliwość powstania słowa pustego

1.6. Jeżeli **r***,* **s***,* **t** są wyrażeniami regularnymi, to zachodzą następujące tożsamości:

[TAK] **(A)** **(ε|r)* = r***
ponieważ nawet jeśli po lewej stronie **r** nie zawierało w sobie *ε*, to jest domknięte, a więc *ε* już należy, natomiast po prawej stronie jest domknięcie **r**, a więc *ε* lub dowolne złożenia słów z **r**

[TAK] **(B)** **rε = εr = r**
ponieważ konkatenacja ze słowem pustym z dowolnej strony daje to samo

[TAK] **(C)** **r*s|s = r*s**
ponieważ **s** zawiera się w **r*s** (**r*** może być słowem pustym)

[TAK] **(D)** **(r|s)* = (r*s)*r* = (s*r)*s***
ponieważ wszystkie trzy wyrażenia dają dowolne ciągi wyrażeń **s** i **r**

[TAK] **(E)** **(r|s)* = (r*s*)***
ponieważ po obu stronach są dowolne ciągi wyrażeń **s** i **r**

[NIE] **(F)** **r*s|rs* = rs(r*|s*)**
ponieważ po lewej stronie możemy stworzyć **s**, a po prawej mamy narzucone **rs** na początku

[NIE] **(G)** **∅* = ∅**
ponieważ domknięcie zawsze zawiera *ε*, a więc nie jest zbiorem pustym

[NIE] **(H)** **r(rs|s)*r = r(sr|r)***
ponieważ po lewej stronie możemy stworzyć **rssr**, a po prawej nie

1.7. Rozważamy języki regularne nad alfabetem *{a, b, c}*, takie że liczba ich słów o długości n wynosi dokładnie n^2 dla każdego $n > 1$. Przykładem takiego języka może być:

[TAK] **(A)** **a*ba*ba* | a*ca*ca* | a*ba***

Słowa nie powtarzają się pomiędzy składnikami sumy teoriomnogościowej. Pierwszy i drugi składnik pozwalają na wygenerowanie takiej samej liczby słów. W przypadku pierwszego składnika słowo o długości *n* zawiera dwie litery *b* które mogą pojawić się na każdej z *n* pozycji, więc możliwych jest $\binom{n}{2}$ słów. Trzeci składnik pozwala na utworzenie *n* różnych słów, ponieważ *b* może stać na każdym z *n* miejsc. Ostatecznie daje to $2\frac{n!}{2!(n-2)!} + n = n(n-1) + n = n^2$.

[NIE] **(B)** **a*ca*ca* | a*ba***

Język regularny z punktu (A) miał odpowiednią liczbę słów, a ten (B) ma mniej bo w wyrażeniu regularnym (B) brakuje jednego składnika sumy teoriomnogościowej z wyrażenia regularnego z (A).

[TAK] **(C)** **$a^*b^+a^+$ | $b^*a^+b^+$ | a^*c^+**

Jest to przykład analogiczny do (A), bo:

$a^*b^+a^+ = a^*bb^*aa^*$

$b^*a^+b^+ = b^*aa^*bb^*$

$a^*c^+ = a^*cc^*$

Słowa nie powtarzają się pomiędzy składnikami sumy teoriomnogościowej. Zacznijmy od ostatniego, najprostszego składnika.

c można ustawić w każdym z *n* miejsc, pozostałe wypełniane są odpowiednio *a* przed i *c* po. To oznacza, że ten składnik pozwala na wygenerowanie *n* słów o długości *n*.

Pierwszy i drugi składnik sumy teoriomnogościowej pozwalają na utworzenie takiej samej ilości słów. W tym przypadku *a* można ustawić na każdym z *n* miejsc a *b* na każdym z *n-1* miejsc, jednak trzeba ten iloczyn podzielić przez 2 aby nie uwzględniać przypadków kiedy *a* i *b* są w odwrotnej niż prawidłowa kolejności.

$$\frac{n(n-1)}{2} + \frac{n(n-1)}{2} + n = n^2$$

[NIE] **(D)** **$b^*a^+b^+$ | a^*c^+**

Język regularny z punktu (C) miał odpowiednią liczbę słów, a ten ma mniej bo w wyrażeniu regularnym (D) brakuje jednego składnika sumy teoriomnogościowej z wyrażenia regularnego z (C).

[NIE] **(E)** **(a|b)*c(a|b)***

Dla słowa o długości *n*, *c* można ustawić w każdym z *n* miejsc, na pozostałych miejscach (*n-1*) może wystąpić *a* lub *b*, więc ilość możliwych słów to
$n2^{n-1} \neq n^2$ dla $n \geq 3$

[NIE] **(F)** **(a|b)*ca***

To wyrażenie regularne pozwala utworzyć następującą liczbę słów o długości *n*: $\sum_{i=0}^{n-1} 2^i$ Jest tak, ponieważ ilość słów zależy wyłącznie od tego ile razy wystąpi część **(a|b)**, a może się to wydarzyć maksymalnie *n-1* razy. Resztę słowa wypełnia *c* i odpowiednia liczba *a*. Podany wzór dla $n=2$ daje wynik $3 \neq 4$ k a $n^2=4$.

1.8. Które z następujących wyrażeń regularnych nie reprezentuje języka złożonego ze wszystkich słów nad alfabetem *{a,b}* zawierających co najmniej dwa wystąpienia symbolu *a*?

[TAK] **(A) (ab)*a**

To wyrażenie nie reprezentuje takiego języka, ponieważ
1) nie zawiera słowa *aab*, które należy do języka
2) zawiera słowo *a* które nie należy do języka

[TAK] **(B) a(ba)***

To wyrażenie nie reprezentuje takiego języka, ponieważ
1) nie zawiera słowa *aab*, które należy do języka
2) zawiera słowo *a*, które nie należy do języka

[NIE] **(C) (a|b)*ab*a(a|b)***

To wyrażenie reprezentuje język, ponieważ oprócz dwóch ustawionych „na sztywno” *a*, dookoła może występować cokolwiek (jeżeli chcielibyśmy mieć więcej *a* pomiędzy tymi dwoma ustawionymi, to można to zrealizować inaczej, czyli po lewej lub prawej stronie od tych ustawionych).

[NIE] **(D) b*ab*a(a|b)***

To wyrażenie reprezentuje język, bo część **b*ab*a** gwarantuje wystąpienie dwóch *a* i dowolnej ilości *b* dookoła, a końcówka **(a|b)*** pozwala na utworzenie dowolnego słowa nad alfabetem *{a,b}*.

[TAK] **(E)** $\mathbf{(a|b)^{+}a(a|b)^{+}a(a|b)^{+}}$

To wyrażenie regularne nie reprezentuje tego języka, ponieważ występują w nim trzy składniki $\mathbf{(a|b)^{+}}$, które wartościują się do co najmniej jednego znaku, więc niemożliwe jest utworzenie słowa *aa*, musi być przynajmniej 5 znaków, np. *babab*.

[TAK] **(F)** $\mathbf{(a|b)^{*}a^{+}(a|b)^{*}a^{+}}$

To wyrażenie regularne nie reprezentuje tego języka, ponieważ nie można wygenerować za jego pomocą słowa kończącego się na *b*.

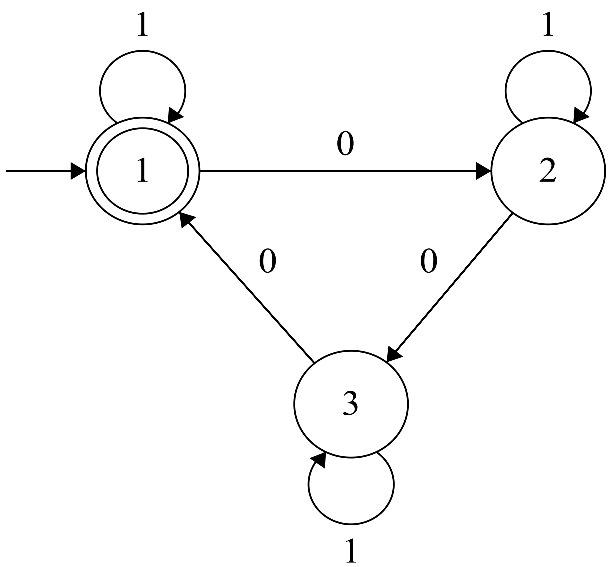
[NIE] **(G)** $\mathbf{(a|b)^{*}a(a|b)^{*}a(a|b)^{*}}$

To wyrażenie regularne reprezentuje ten język, ponieważ oprócz dwóch *a* ustawionych „na sztywno” można pomiędzy nimi umieścić dowolne słowa nad alfabetem *{a,b}*. Chyba najbardziej oczywiste i czytelne wyrażenie regularne reprezentujące ten język.

[TAK] **(H)** **a(a|b)*a**

To wyrażenie regularne nie reprezentuje tego języka, ponieważ:
1) nie da się uzyskać słów zaczynających się na literę *b*, np. *baa*
2) nie da się uzyskać słów kończących się na literę *b*, np. *aab*

1.9. Wyrażenie regularne odpowiadające wszystkim słowom bitowym, w których liczba zer jest podzielna przez 3 to:



[NIE] **(A)** **(01*01*01)*1***

– nie możemy wygenerować słowa, w którym 1 poprzedza pierwsze 0

[NIE] **(B)** **(0*01*01*01*)*1***

– nie możemy wygenerować słowa, w którym 1 poprzedza pierwsze 0

[NIE] **(C)** **(1*01*01*01*)***

– nie możemy wygenerować słowa składającego się z samych jedynek

[TAK] **(D)** **(1*01*01*01*)*1***

– odpowiada narysowanemu automatowi

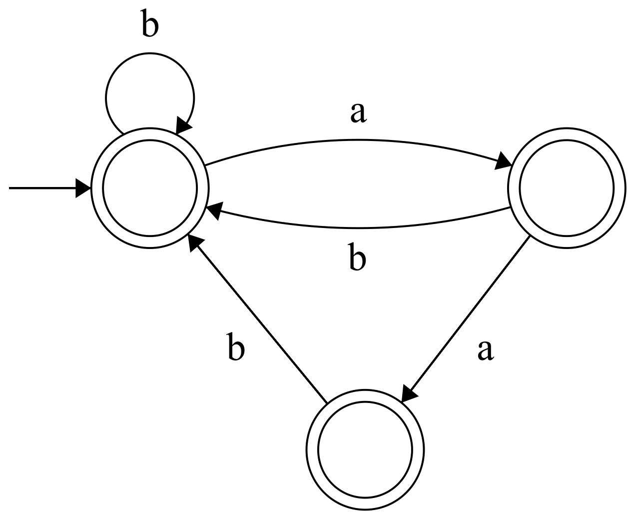
[TAK] **(E)** **(1*01*01*0)*1***

– odpowiada narysowanemu automatowi i jest równoważny powyższemu

[TAK] **(F) 1*(01*0(01*01*0|1)*01*| ε)**

– **1*** i **ε** pozwalają na akceptację w pierwszym stanie, **(01*0** pozwala na dojście do trzeciego stanu, **(01*01*0|1)*** zawsze kończy się w trzecim stanie, a **01*** gwarantuje nam następne przejście do stanu akceptującego

1.10. Niech *Σ = {a,b}* oraz niech *L* będzie językiem nad alfabetem *Σ* złożonym ze wszystkich słów nie zawierających podłańcucha *aaa*. Następujące wyrażenie regularne odpowiada językowi *L*:



[NIE] **(A) (a|b)*(a|aa)*(a|b)***

– zarówno **(a|b)*** jak i **(a|aa)*** pozwalają wygenerować podłańcuch *aaa*

[NIE] **(B) (a|b)*(ε|a|aa)*(a|b)***

– zarówno **(a|b)*** jak i **(ε|a|aa)*** pozwalają wygenerować podłańcuch *aaa*

[TAK] **(C) (b|ab|aab)*(ε|a|aa)**

– pozwala na wygenerowanie wszystkich słów niezawierających podłańcucha *aaa*

[NIE] **(D) (b|ab|aab)***

– nie można wygenerować słowa kończącego się na *a* lub *aa*, lub zawierającego tylko *a* lub *aa*

[TAK] **(E) (ε|a|aa)(b|ba|baa)***

– pozwala na wygenerowanie wszystkich słów niezawierających podłańcucha *aaa*

[NIE] **(F) (b|ba|baa)***

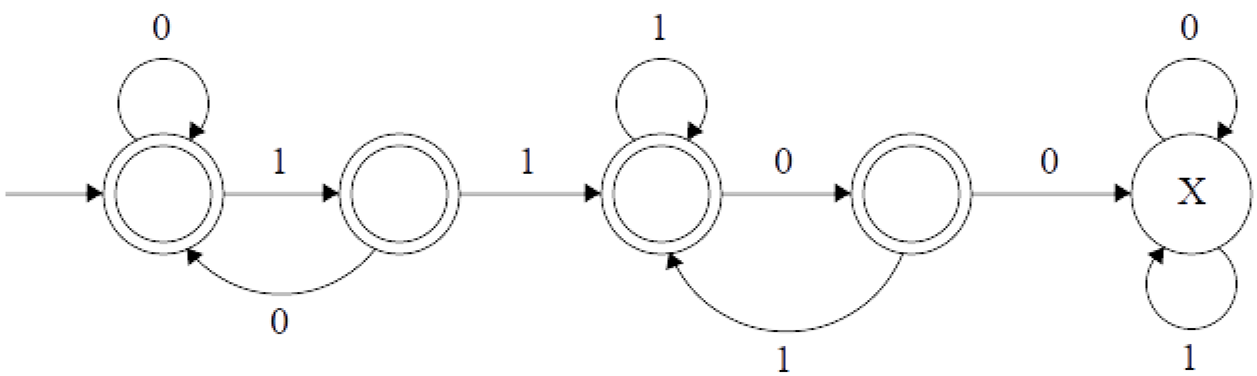
– nie można wygenerowanie słowa zaczynającego się na *a* lub *aa*, lub zawierającego tylko *a* lub *aa*

[TAK] **(G) (b|ab|aab)*|(b*a)(bb*a|abb*a)*|(b*a(bb*a)*a)(bb*a(bb*a)*a)***

– pierwszy składnik sumy teoriomnogościowej pozwala wygenerować wszystkie słowa niezawierające podłańcucha *aaa*, z wyjątkiem słów *a*, *aa* i tych kończących się na *a* lub *aa*. Drugi składnik sumy teoriomnogościowej pozwala wygenerować słowo *a* i wszystkie kończące się na *a*. Trzeci składnik sumy teoriomnogościowej pozwala wygenerować słowo *aa* i wszystkie kończące się na *aa*

1.11. Niech $\Sigma = \{0, 1\}$ oraz niech L będzie językiem nad alfabetem Σ będącym zbiorem wszystkich łańcuchów, w których każdy podłańcuch zawierający dwa lub więcej kolejne zera pojawia się przed jakimkolwiek podłańcuchem zawierającym dwie lub więcej kolejne jedynki. Następujące wyrażenie regularne odpowiada językowi L:

Automat akceptujący język z zadania:



[NIE] **(A) (1|01)*(ε|0|00(0|10)*(ε|1))**

Wyrażenie regularne nie spełnia warunków zadania. Po wystąpieniu dwóch kolejnych jedynek mogą wystąpić dwa kolejne zera.

[TAK] **(B) 0*|0*1(00*1)*|0*1(00*1)*1(1|01)*(ε|0)**

Wyrażenie regularne spełnia warunki zadania i odwzorowuje automat skończony przedstawiony powyżej.

[NIE] **(C) 1*|1*0(11*0)*|1*0(11*0)*0(0|10)*(ε|1)**

Odpowiednik wyrażenia z punktu (B) po zamienieniu symboli *0* na *1* i *1* na *0*. Wyrażenie to nie generuje słów, które są akceptowane w czwartym stanie (od lewej) powyższego automatu skończonego, gdyż po wystąpieniu dwóch kolejnych jedynek mogą wystąpić dwa kolejne zera.

[TAK] **(D)** **(0|10)*(ε|1|11(1|01)*(ε|0))**

Odpowiednik wyrażenia z punktu (A) po zamienieniu symboli *0* na *1* i *1* na *0*. Generowany język odpowiada językowi akceptowanemu przez automat skończony na rysunku powyżej.

[NIE] **(E)** **0*|0*1(00*1)*|0*1(00*1)*1(1|01)***

Jest to wyrażenie (B), ale bez słów zawierających znak *1* i jednocześnie kończących się znakiem *0*, przez co wyrażenie to nie generuje wszystkich słów języka.

[TAK] **(F)** **(0*(10)*)*(ε|1|11(1|01)*(ε|0))**

Wyrażenie jest odpowiednikiem tego z punktu (D). Część **(0*(10)*)*** można przekształcić w **(0|10)***.

[NIE] **(G)** **0*|0*1(00*1)*|0*1(00*1)*1(1|01)*0**

Jest to wyrażenie (B), ale bez słów zawierających znak *0* i jednocześnie kończących się znakiem *1*, przez co wyrażenie to nie generuje wszystkich słów języka.

1.12. Rozważa się następujące języki regularne: $L_1 = \boldsymbol{\varepsilon}$ oraz $L_2 = \varnothing$. Dla tych języków prawdziwe jest:

[TAK] **(A)** $L_1 \cap L_2 = \varnothing$

Przecięcie języka pustego z jakimkolwiek językiem zawsze daje język pusty.

[TAK] **(B)** $L_1^* = L_2^*$

Z tożsamości $\boldsymbol{\varepsilon}^* = \boldsymbol{\varepsilon}$ wynika $L_1^* = \boldsymbol{\varepsilon}^* = \boldsymbol{\varepsilon} = L_1$.

Z definicji $\varnothing^* = \boldsymbol{\varepsilon}$ wynika $L_2^* = \varnothing^* = \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^* = L_1^*$.

[NIE] **(C)** $L_1^* \not\subset L_2^*$

Z uzasadnienia punktu (B).

[NIE] **(D)** $L_2^* \not\subset L_1^*$

Z uzasadnienia punktu (B).

[TAK] **(E)** $L_1^* \cup L_2^* = (L_1 \cup L_2)^*$

$L_1^* \cup L_2^* = \boldsymbol{\varepsilon}^* \cup \varnothing^* = \boldsymbol{\varepsilon} \cup \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}$

$(L_1 \cup L_2)^* = (\boldsymbol{\varepsilon} \cup \varnothing)^* = \varnothing^* = \boldsymbol{\varepsilon}$

Z powyższych wyprowadzeń wynika, że równość jest prawdziwa.

[TAK] **(F)** $L_1 L_2 = L_2 L_1$

$\boldsymbol{\varepsilon}\varnothing = \varnothing\boldsymbol{\varepsilon} \Rightarrow \varnothing = \varnothing$

Konkatenacja **ε** i ∅ w dowolnej kolejności zawsze da ∅**,** więc równość jest spełniona.

[TAK] **(G)** $L_1 \not\subset L_1L_2$

$\boldsymbol{\varepsilon} \not\subset \boldsymbol{\varepsilon}\varnothing \Rightarrow \boldsymbol{\varepsilon} \not\subset \varnothing$

[NIE] **(H)** $L_2 \not\subset L_1L_2$

$\varnothing \not\subset \varnothing\boldsymbol{\varepsilon} \Rightarrow \varnothing \not\subset \varnothing$

Sprzeczność.

1.13. Rozważa się następujące języki regularne: L_1 = **(aa)*** oraz L_2 = **a(aa)***. Dla tych języków prawdziwe jest:

[TAK] **(A)** $L_1 \cap L_2 = \varnothing$

L_1, L_2 są rozłączne – nie mają żadnego wspólnego słowa.

[NIE] **(B)** $L_1^* = L_2^*$

L_1^* = **((aa)*)* = (aa)***

L_2^* = **(a(aa)*)* = a***

[TAK] **(C)** $L_1 \not\subset L_2$

L_1, L_2 są rozłączne – żaden z nich nie jest podzbiorem drugiego.

[TAK] **(D)** $L_2 \not\subset L_1$

L_1, L_2 są rozłączne – żaden z nich nie jest podzbiorem drugiego.

[TAK] **(E)** $L_1^* \cup L_2^* = (L_1 \cup L_2)^*$

Obie strony równości sprowadzają się do **a***.

[TAK] **(F)** $L_1L_2 = L_2L_1$

Słowa należące do L_1 oraz L_2 składają się wyłącznie z symboli a, więc kolejność nie gra roli.

[TAK] **(G)** $L_1 \not\subset L_1L_2$

L_1 składa się z słów długości parzystej, a L_1L_2 składa się z słów długości nieparzystej.

[NIE] **(H)** $L_2 \not\subset L_1L_2$

Język L_1 zawiera słowo puste i w sposób oczywisty L_2 jest podzbiorem L_2, więc L_2 będzie podzbiorem L_1L_2.

1.14. Rozważamy języki regularne nad alfabetem $\{a, b, c\}$, takie że liczba ich słów o długości n wynosi dokładnie $n \cdot 2^{n-1}$ dla każdego $n > 0$. Przykładem takiego języka może być:

[NIE] **(A)** **a*ba*ba* | a*ca*ca* | a*ba***

Liczba słów o długości *n* dla powyższego języka wynosi dokładnie n^2. W przypadku pierwszego oraz drugiego członu wybierane są 2 z *n* miejsc na symbole inne niż *a*. W przypadku trzeciego członu wybierane jest jedno z *n* miejsc na symbol *b*. Stąd: $2 * \binom{n}{2} + n = n^2$

[NIE] **(B)** **a*ca*ca* | a*ba***

Język z punktu B to okrojony język z punktu A, więc jeśli w punkcie A liczba słów o długości *n* wynosi n^2 to dla języka B słów tych będzie jeszcze mniej.

[NIE] **(C)** **$a^*b^+a^+ \mid b^*a^+b^+ \mid a^*c^+$**

Liczba słów o długości *n* dla powyższego języka wynosi dokładnie n^2. W przypadku pierwszego członu wybierana jest pozycja pierwszego symbolu *b* (z *n-1* możliwych). Uwzględnione muszą zostać wszystkie warianty liczb symboli *b*. Przykładowo jeśli pierwszy symbol *b* wystąpi na przedostatniej pozycji w słowie to jest to jedyna możliwość. Jeżeli pierwszy symbol *b* wystąpi na trzeciej od końca pozycji to warianty są dwa (*...baa*, *...bba*). Należy zauważyć, że problem sprowadza się do sumy ciągu arytmetycznego: $\frac{1+(n-1)}{2} * (n-1)$. Drugi człon języka jest analogiczny do pierwszego. W trzecim należy wybrać jedną z *n* pozycji wystąpienia pierwszego symbolu *c*. Ostatecznie liczba możliwych do uzyskania słów redukuje się do: $2 * \frac{1+(n-1)}{2} * (n-1) + n = n^2$.

[NIE] **(D)** **$b^*a^+b^+ \mid a^*c^+$**

Język z punktu D to okrojony język z punktu C, więc jeśli w punkcie C liczba słów o długości *n* wynosi n^2 to dla języka D słów tych będzie jeszcze mniej.

[TAK] **(E)** **(a|b)*c(a|b)***

Liczba słów o długości *n* dla powyższego języka wynosi dokładnie $n \cdot 2^{n-1}$. W pierwszej kolejności wybierana jest jedna z *n* pozycji dla symbolu *c*. Następnie na pozostałych *n-1* pozycjach wybierany jest jeden z dwóch symboli co redukuje się do $n \cdot 2^{n-1}$.

[NIE] **(F)** **(a|b)*ca***

Język z punktu F to okrojony język z punktu E (brak drugiej sumy teoriomnogościowej). Wynika z tego, że język F posiada mniej słów długości *n* niż język E, który spełnia warunki zadania.

1.15. Rozważa się następujące języki regularne: L_1 = **aa** oraz L_2 = **a**. Dla tych języków prawdziwe jest:

[TAK] **(A)** $L_1 \cap L_2 = \varnothing$

tak, bo język L_1 zawiera tylko słowo aa, natomiast język L_2 zawiera tylko słowo a. Część wspólna (iloczyn) rozłącznych względem siebie zbiorów daje nam zbiór pusty ($\varnothing$).

[NIE] **(B)** $L_1^* = L_2^*$

nie, bo L_1^* zawiera słowa $\{\varepsilon, aa, aaaa, aaaaaa, \ldots\}$, czyli słowo puste i słowa składające się z parzystej liczby liter a, natomiast L_2^* zawiera słowa $\{\varepsilon, a, aa, aaa, \ldots\}$. Stąd wynika, że równość nie zachodzi (zbiory są względem siebie różne).

[NIE] **(C)** $L_1^* \not\subset L_2^*$

nie, bo L_1^* jest podzbiorem L_2^*. L_1^* zawiera słowo puste i słowa z parzystą liczbą liter a, natomiast L_2^* zawiera słowo puste i dowolne słowo składające się z liter a. Stąd wynika, że każde słowo z L_1^* jest jednocześnie słowem należącym do L_2^*.

[TAK] **(D)** $L_2^* \not\subset L_1^*$

tak, bo L_1^* zawiera słowo puste i słowa z parzystą liczbą liter a, natomiast L_2^* zawiera słowo puste i każde dowolne słowo składające się z liter a. Zbiór wszystkich słów składających się z liter a nie może byc podzbiorem zbioru wszystkich słów o parzystej liczbie liter a.

[TAK] **(E)** $L_1^* \cup L_2^* = (L_1 \cup L_2)^*$

tak, bo po lewej stronie mamy zbiór wszystkich słów składających się z liter a (włącznie ze słowem pustym), natomiast po prawej stronie również jesteśmy w stanie otrzymać dowolne słowo składające się z liter a (włącznie ze słowem pustym).

[TAK] **(F)** $L_1L_2 = L_2L_1$

tak, bo L_1L_2 to **aaa**, natomiast L_2L_1 to **aaa**. Złożenia te dają nam te same słowa.

[TAK] **(G)** $L_1 \not\subset L_1L_2$

tak, bo $L_1 =$ **aa**, natomiast $L_1L_2 =$ **aaa**. $\{aa\}$ nie jest podzbiorem $\{aaa\}$.

[TAK] **(H)** $L_2 \not\subset L_1L_2$

tak, bo $L_2 =$ **a**, natomiast $L_1L_2 =$ **aaa**. $\{a\}$ nie jest podzbiorem $\{aaa\}$.

1.16. Rozważamy języki regularne nad alfabetem *{a, b, c}*, takie że liczba ich słów o długości *n* wynosi dokładnie $2^n – 1$ dla każdego $n > 0$. Przykładem takiego języka może być:

[NIE] **(A)** **a*ba*ba* | a*ca*ca* | a*ba***

Wyrażenie to 3 składniki sumy teoriomnogościowej. Rozważamy słowo o długości *n*. Pierwszy składnik odpowiada $\binom{n}{2}$ takim słowom, gdyż dwie litery *b* mogą zajmować dwie dowolne pozycje w słowie. W przypadku drugiego składnika jest analogicznie do składnika pierwszego. W trzecim składniku rozważamy na ilu miejscach może się znaleźć jedna litera *b* w słowie o długości *n*, co daje nam *n* możliwości. Ostatecznie otrzymujemy: $\binom{n}{2} + \binom{n}{2} + n = n^2$

[NIE] **(B)** **a*ca*ca* | a*ba***

Wyrażenie to 2 składniki sumy teoriomnogościowej. Rozważamy słowo o długości *n*. Pierwszy składnik odpowiada $\binom{n}{2}$ takim słowom, gdyż dwie litery *c* mogą zajmować dwie dowolne pozycje w słowie. W przypadku składnika drugiego jedna litera *b* może znaleźć się na dowolnym miejscu w słowie, co daje nam *n* możliwości. Ostatecznie otrzymujemy: $\binom{n}{2} + n = \frac{n^2}{2} + \frac{n}{2}$

[NIE] **(C)** **a*b$^+$a$^+$ | b*a$^+$b$^+$ | a*c$^+$**

Wyrażenie to 3 składniki sumy teoriomnogościowej. W przypadku pierwszego członu wybierana jest pozycja pierwszego symbolu *b* (z *n-1* możliwych). Uwzględnione muszą zostać wszystkie warianty położenia symboli *b*. Przykładowo jeśli pierwszy symbol *b* wystąpi na przedostatniej pozycji w słowie to jest to jedyna możliwość. Jeżeli pierwszy symbol *b* wystąpi na trzeciej od końca pozycji to warianty są dwa (*...baa*, *...bba*). Należy zauważyć, że problem sprowadza się do sumy ciągu arytmetycznego *1+2+...+(n-1)*. W trzecim przypadku literę *c* możemy umieścić na dowolnym z *n* miejsc. Po położeniu na danej pozycji literze *c*, wszystkie litery na prawo od niej to *c*, natomiast wszystkie litery na lewo od niej to *a*. Daje nam to po 1 możliwości dla każdej z pozycji, na której położono literę *c*. Sumarycznie dla słowa o długości *n* otrzymujemy *n* takich ułożeń. Ostatecznie otrzymujemy: $2 \cdot \frac{1+(n-1)}{2} \cdot (n-1) + n = n^2$

[NIE] **(D)** **b*a$^+$b$^+$ | a*c$^+$**

Wyrażenie to 2 składniki sumy teoriomnogościowej. Składniki są takie same jak drugi i trzeci składnik z podpunktu (C). Liczba możliwości to $\frac{1+(n-1)}{2} \cdot (n-1) + n = \frac{n^2}{2} + \frac{n}{2}$.

[NIE] **(E)** **(a|b)*c(a|b)***

Rozważamy wszystkie możliwe ułożenia litery *c* w słowie o długości *n*. Litera *c* w słowie o długości *n* może się znaleźć na dowolnej z *n* pozycji, co daje nam *n* możliwości. Będąc na danej pozycji na pozostałych z miejsc może się znaleźć litera *a* lub *b* co daje 2 możliwości na każdej z pozostałych *n-1* pozycji, co daje 2^{n-1} możliwości.Ostatecznie liczba możliwych do utworzenia słów, to: $n \cdot 2^{n-1}$.

[TAK] **(F)** **(a|b)*ca***

Rozważamy wszystkie możliwe ułożenia litery *c* w słowie o długości *n*. Układamy literę *c* na każdym z pól słowa o długości *n* idąc od lewej. Liczba możliwości dla każdego z ułożenia litery *c*, to 2^{i-1}, gdzie i to pozycja litery *c* (startujemy od pozycji 1 a kończymy na pozycji *n*). Sumujemy wszystkie możliwości co prowadzi nas do ogólnego wzoru: $1 + 2^1 + 2^2 + \cdots + 2^{n-1} = 2^n - 1$.

1.17. Dopełnieniem języka **ba*** jest

[NIE] **(A)** **a*b***

Bo takie wyrażenie zawiera łańcuch *b*

[NIE] **(B)** $\mathbf{a^+b^+}$

Bo takie wyrażenie nie zawiera łańcucha pustego

[NIE] **(C)** **ba*b(a|b)***

Jak wyżej, nie zawiera łańcucha pustego

[NIE] **(D)** **b(a|b)***

Bo takie wyrażenie zawiera łańcuchy, wśród których będą łańcuchy z **ba*** i nie zawiera pustego

[TAK] **(E)** **b(a|ba*b)(a|b)*|ε**

To wyrażenie nie wygeneruje żadnego łańcucha który zawierałby się w tym z polecenia, jest też łańcuch pusty

[NIE] **(F)** **(a|ba*b)(a|b)***

Brak łańcucha pustego

[NIE] **(G)** **(ε|a|ba*b)(a|b)***

Bo to jest **(a|b)***

[NIE] **(H)** język regularny, ale żaden z powyższych

Gdyż podpunkt (E) jest prawdziwy

[NIE] **(I)** język nie będący językiem regularnym

Bo język nie będący regularnym nie może być dopełnieniem języka regularnego

1.18. Jeżeli **r**, **s** są wyrażeniami regularnymi, to zachodzą następujące tożsamości:

[TAK] **(A)** $(\varepsilon|r)^* = r^*$

Bo po lewej stronie dostaniemy 0 lub więcej **r** albo łańcuch pusty, po prawej stronie – też 0 lub więcej **r** i też jest możliwość dostania łańcucha pustego.

[TAK] **(B)** $r\varepsilon = \varepsilon r = r$

Łańcuch pusty jest elementem neutralnym konkatenacji

[TAK] **(C)** $r^*s|s = r^*s$

Po lewej dostaniemy elementy zawierające dowolną (w tym zero) ilość łańcuchów z **r** i kończące się łańcuchem z **s**, po prawej – to samo

[TAK] **(D)** $(r|s)^* = (r^*s)^*r^* = (s^*r)^*s^*$

Bo wszędzie dostaniemy łańcuchy z **r** i **s**, gdzie **r** i **s** mogą być na różnych miejscach (nie mamy jakiegoś wyrażenia gdzie widać, że **r** lub **s** zawsze będzie na początku lub końcu)

[NIE] **(E)** $(r|s)^* = r^*s^*|s^*r^*$

Bo po lewej stronie da się dostać **rsrs**, po prawej - nie

[NIE] **(F)** $r^*s|rs^* = rs(r^*|s^*)$

Bo wyrażenie po prawej stronie zawiera łańcuchy zaczynające się zawsze od łańcuchów opisanych przez **rs**, a po lewej – niekoniecznie, gdyż mogą wystąpić łańcuchy zaczynające się od łańcuchów opisanych przez kilka kolejnych **r**

[NIE] **(G)** $\varnothing^* = \varnothing$

Bo lewa część to **ε**, zaś prawa – zbiór pusty

[TAK] **(H)** $(rs|r)^*r = r(sr|r)^*$

Po obu stronach są łańcuchy zaczynające i kończące się na **r** w których nigdy nie wystąpi dwa **s** obok siebie.