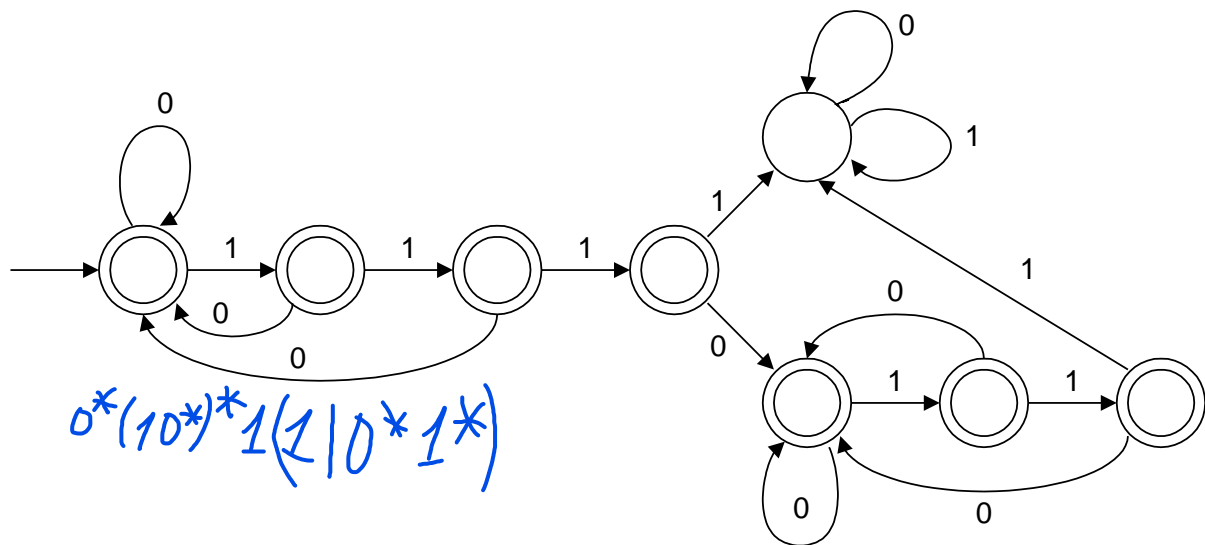


**Zad. 1.**

Dany jest poniższy automat skończony:

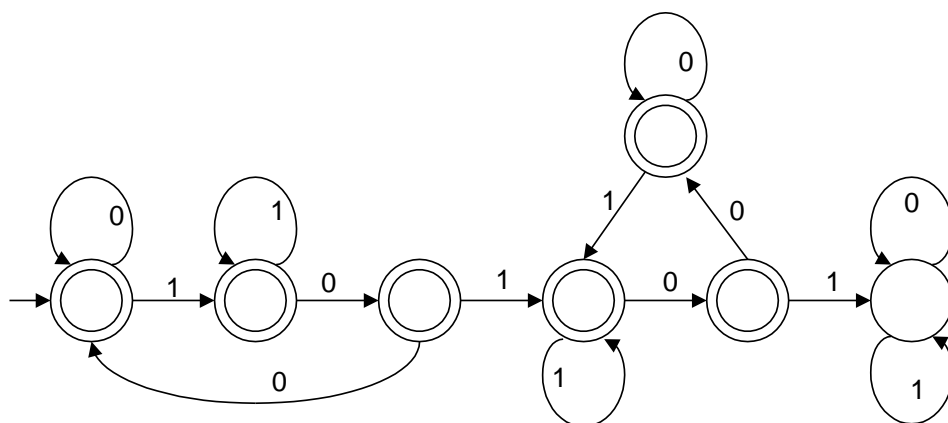


Język akceptowany przez ten automat to:

- ☒ (A) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych z wyjątkiem łańcucha  $1111$  ✓
- ☒ (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden podłańcuch zbudowany z trzech kolejnych jedynek
- ☒ (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden raz podłańcuch  $101$
- ☒ (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych nie zawierających podłańcucha  $1110$
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów, w których każdy podłańcuch zawierający dwa lub więcej kolejne zera pojawia się przed jakimkolwiek łańcuchem zawierającym dwie lub więcej kolejne jedynki

**Zad. 2.**

Dany jest poniższy automat skończony:

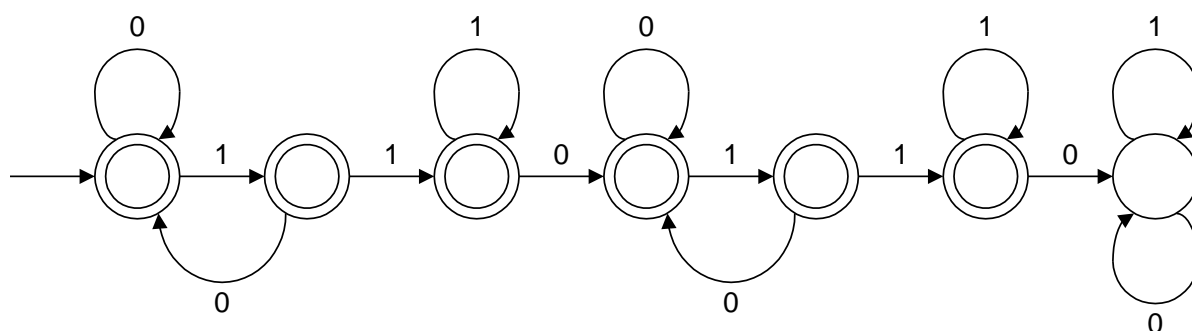


Język akceptowany przez ten automat to:

- ~~(A)~~ język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych z wyjątkiem łańcucha  $1111$
- ~~(B)~~ język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden raz podłańcuch  $101$
- ~~(C)~~ język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych nie zawierających podłańcucha  $1110$
- ~~(D)~~ język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden podłańcuch zbudowany z trzech kolejnych jedynek
- ~~(E)~~ język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów, w których każdy podłańcuch zawierający dwa lub więcej kolejne zera pojawia się przed jakimkolwiek łańcuchem zawierającym dwie lub więcej kolejne jedynek

### Zad. 3.

Dany jest poniższy automat skończony:

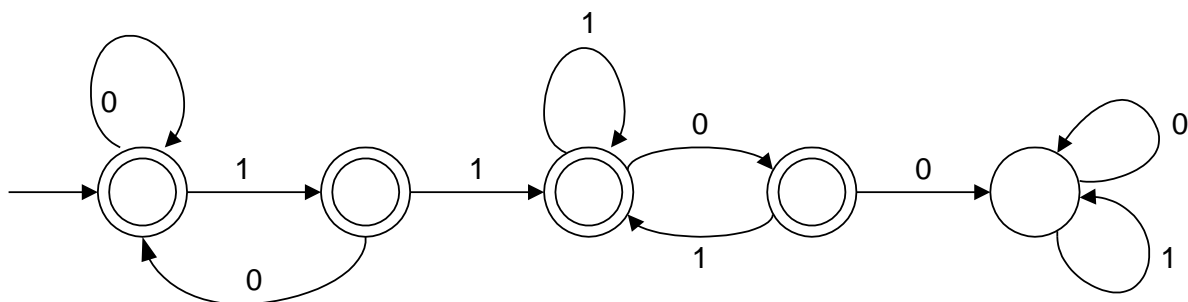


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych z wyjątkiem łańcucha  $1111$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden podłańcuch zbudowany z trzech kolejnych jedynek
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden raz podłańcuch  $110$
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych nie zawierających podłańcucha  $110110$
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów, w których każdy podłańcuch zawierający dwa lub więcej kolejne zera pojawia się przed jakimkolwiek łańcuchem zawierającym dwie lub więcej kolejne jedynek

### Zad. 4.

Dany jest poniższy automat skończony:

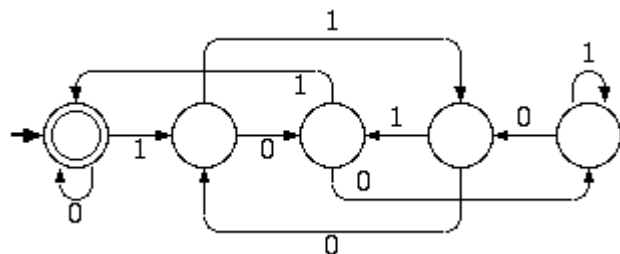


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych z wyjątkiem łańcucha  $1100$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden raz podłańcuch  $101$
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych nie zawierających podłańcucha  $1100$
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających co najwyżej jeden podłańcuch zbudowany z trzech kolejnych jedynek
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów, w których każdy podłańcuch zawierający dwa lub więcej kolejne zera pojawia się przed jakimkolwiek łańcuchem zawierającym dwie lub więcej kolejne jedynek

#### Zad. 5.

Dany jest poniższy automat skończony:

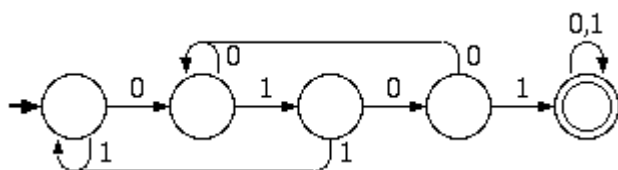


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

#### Zad. 6.

Dany jest poniższy automat skończony:

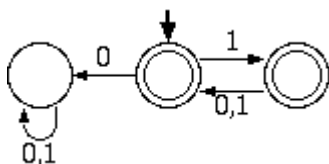


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

### Zad. 7.

Dany jest poniższy automat skończony:

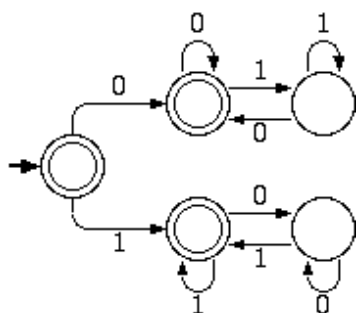


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

**Zad. 8.**

Dany jest poniższy automat skończony:

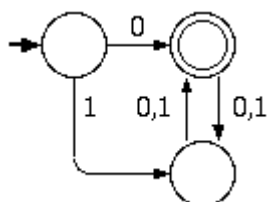


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

**Zad. 9.**

Dany jest poniższy automat skończony:

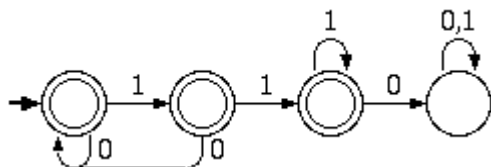


Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

**Zad. 10.**

Dany jest poniższy automat skończony:



Język akceptowany przez ten automat to:

- (A) język nad alfabetem  $T = \{0, 1\}$ , będący zbiorem wszystkich słów zerojedynkowych zawierających tyle samo podłańcuchów  $01$  co podłańcuchów  $10$
- (B) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych rozpoczynających się od zera i mających nieparzystą długość lub rozpoczynających się od jedynki i mających parzystą długość
- (C) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, które stanowią binarną reprezentację liczb podzielnych przez 5
- (D) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których na każdej nieparzystej pozycji stoi jedynka
- (E) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych zawierających podłańcuch  $0101$
- (F) język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych niezawierających podłańcucha  $110$

**Zad 11.**

Następujące wyrażenie regularne:

$(10^*1|0)^*$

opisuje język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których:

- ~~(A)~~ drugim od początku i przedostatnim symbolem jest jedynka,
- ~~(B)~~ występują co najmniej dwie jedynki,
- (C) liczba jedynek jest parzysta,
- (D) każde dwie jedynki są przedzielone co najmniej jednym zerem,
- (E) każda jedynka jest poprzedzona co najmniej jednym zerem i po każdej jedynce występuje co najmniej jedno zero.

**Zad 12.**

Następujące wyrażenie regularne:

$0^*|00^*10(0|10)^*$

opisuje język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynkowych, w których:

- (A) drugim od początku i przedostatnim symbolem jest jedynka,
- (B) każda jedynka jest poprzedzona co najmniej jednym zerem i po każdej jedynce występuje co najmniej jedno zero.
- (C) występują co najmniej dwie jedynki,
- (D) liczba jedynek jest parzysta,

(E) każde dwie jedyńki są przedzielone co najmniej jednym zerem,

**Zad 13.**

Następujące wyrażenie regularne:

$11(0|1)1(0|1)|(0|1)1(0|1)^*1(0|1)$

opisuje język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynekowych, w których:

- (A) drugim od początku i przedostatnim symbolem jest jedynka,
- (B) występują co najmniej dwie jedyńki,
- (C) liczba jedynek jest parzysta,
- (D) każde dwie jedyńki są przedzielone co najmniej jednym zerem,
- (E) każda jedynka jest poprzedzona co najmniej jednym zerem i po każdej jedynce występuje co najmniej jedno zero.

**Zad 14.**

Następujące wyrażenie regularne:

$1^*1^*(011^*)^*01^*$

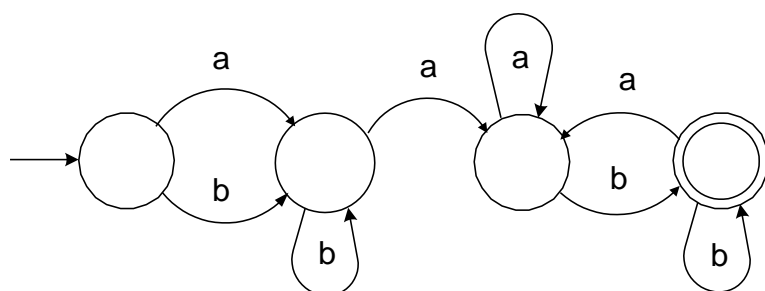
opisuje język nad alfabetem  $T = \{0, 1\}$  będący zbiorem wszystkich łańcuchów zerojedynekowych, w których:

- (A) drugim od początku i przedostatnim symbolem jest jedynka,
- (B) każda jedynka jest poprzedzona co najmniej jednym zerem i po każdej jedynce występuje co najmniej jedno zero.
- (C) występują co najmniej dwie jedyńki,
- (D) liczba jedynek jest parzysta,
- (E) każde dwa zera są przedzielone co najmniej jedną jedyneką,

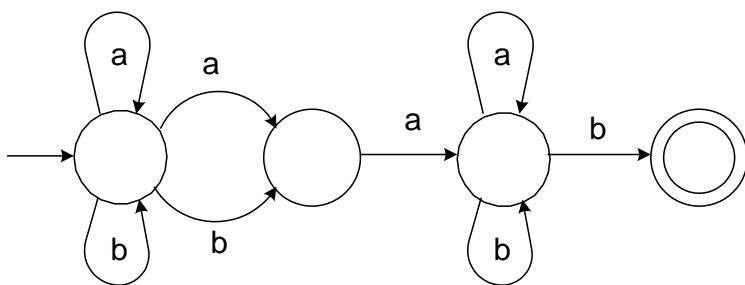
**Zad. 15.**

Dane są dwa automaty skończone:

Automat I:



Automat II:



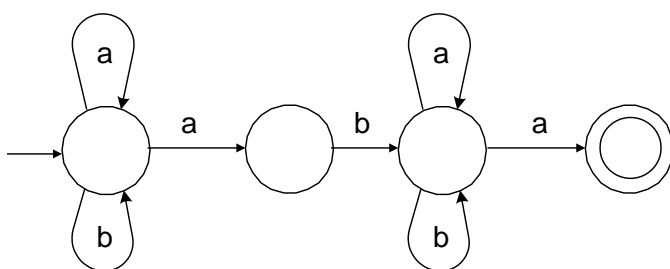
Które z poniższych stwierdzeń są prawdziwe?

- (A) Automat I jest deterministyczny, ale nie jest zupełny, zaś automat II jest zupełny, ale nie jest deterministyczny.
- (B) Automat I jest deterministyczny i zupełny, zaś automat II nie jest ani deterministyczny ani zupełny.
- (C) Automat II jest deterministyczny i zupełny, zaś automat I nie jest ani deterministyczny ani zupełny.
- (D) Automat II jest deterministyczny, ale nie jest zupełny, zaś automat I jest zupełny, ale nie jest deterministyczny.
- (E) Obydwa automaty nie są ani deterministyczne, ani zupełne.

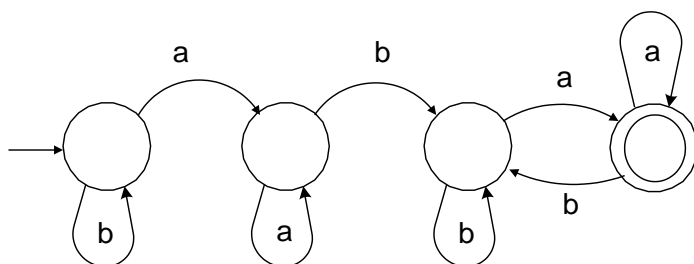
#### Zad. 16.

Dane są dwa automaty skończone:

Automat I:



Automat II:



Które z poniższych stwierdzeń są prawdziwe?

- (A) Automat I jest deterministyczny, ale nie jest zupełny, zaś automat II jest zupełny, ale nie jest deterministyczny.
- (B) Automat I jest deterministyczny i zupełny, zaś automat II nie jest ani deterministyczny ani zupełny.
- (C) Automat II jest deterministyczny, ale nie jest zupełny, zaś automat I jest zupełny, ale nie jest deterministyczny.

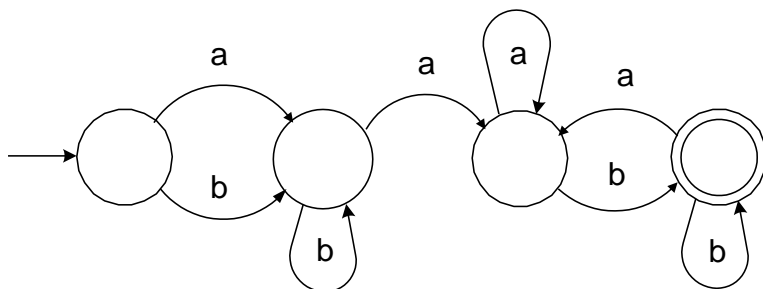


- (D) Automat II jest deterministyczny i zupełny, zaś automat I nie jest ani deterministyczny ani zupełny.
- (E) Obydwa automaty nie są ani deterministyczne, ani zupełne.

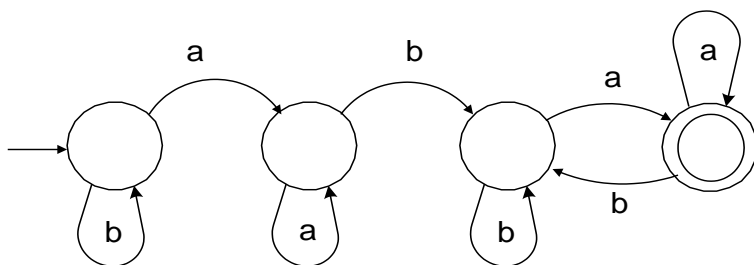
**Zad. 17.**

Dane są dwa automaty skończone:

Automat I:



Automat II:



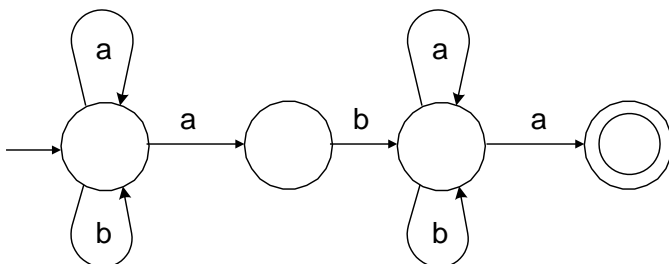
Które z poniższych stwierdzeń są prawdziwe?

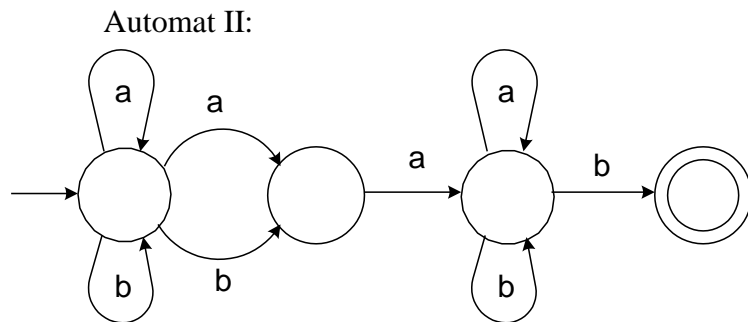
- (A) Automat I jest deterministyczny, ale nie jest zupełny, zaś automat II jest zupełny, ale nie jest deterministyczny.
- (B) Automat I jest deterministyczny i zupełny, zaś automat II nie jest ani deterministyczny ani zupełny.
- (C) Oba automaty są deterministyczne i zupełne.
- (D) Automat II jest deterministyczny, ale nie jest zupełny, zaś automat I jest zupełny, ale nie jest deterministyczny.
- (E) Obydwa automaty nie są ani deterministyczne, ani zupełne.

**Zad. 18.**

Dane są dwa automaty skończone:

Automat I:





Które z poniższych stwierdzeń są prawdziwe?

- (A) Automat I jest deterministyczny, ale nie jest zupełny, zaś automat II jest zupełny, ale nie jest deterministyczny.
- (B) Automat I jest deterministyczny i zupełny, zaś automat II nie jest ani deterministyczny ani zupełny.
- (C) Automat II jest deterministyczny, ale nie jest zupełny, zaś automat I jest zupełny, ale nie jest deterministyczny.
- (D) Automat II jest deterministyczny i zupełny, zaś automat I nie jest ani deterministyczny ani zupełny.
- (E) Obydwa automaty nie są ani deterministyczne, ani zupełne.

#### Zad. 19.

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

- (A)  $11|(0|1)1(0|1)|(0|1)1(0|1)^*1(0|1)$
- (B)  $1^*|1^*(011^*)^*01^*$
- ☒ (C)  $(1|01^*0)(1|01^*0)^*$
- (D)  $11^*|1^*(011^*)^*01^*$
- (E)  $(1|01^*0)^*$

#### Zad. 20.

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

- (A)  $11|(0|1)1(0|1)|(0|1)1(0|1)^*1(0|1)$
- (B)  $1^*|1^*(011^*)^*01^*$
- (C)  $11^*|1^*(011^*)^*01^*$
- (D)  $(1|01^*0)(1|01^*0)^*$
- (E)  $(1|01^*0)^*$

#### Zad. 21.

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

- (A)  $11|(0|1)1(0|1)|(0|1)1(0|1)^*1(0|1)$
- (B)  $1^*|1^*(011^*)^*01^*$
- (C)  $(1|01^*0)(1|01^*0)^*$
- (D)  $11^*|1^*(011^*)^*01^*$
- (E)  $(1|01^*0)^*$

**Zad. 22.**

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

- (A)  $11|(0|1)1(0|1)|(0|1)1(0|1)^*1(0|1)$
- (B)  $1^*|1^*(011^*)^*01^*$
- (C)  $11^*|1^*(011^*)^*01^*$
- (D)  $(1|01^*0)(1|01^*0)^*$
- (E)  $(1|01^*0)^*$

**Zad. 23.**

Który z poniższych języków jest generowany przez następującą gramatykę?

$$\begin{aligned} S &\rightarrow ABC \\ A &\rightarrow aAb \mid ab \\ B &\rightarrow bBa \mid ba \\ C &\rightarrow aCb \mid ab \end{aligned}$$

- (A)  $\{ a^n b^{n+m} a^{m+k} b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (B)  $\{ a^n b^m a^m b^n a^k b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (C)  $\{ a^n b^m a^k b^k a^m b^n \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (D)  $\{ a^n b^m a^k b^n a^m b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (E)  $\{ a^n b^{m+k} a^{m+n} b^{n+k} a^k b^{k+m+n} \mid n \geq 1, m \geq 1, k \geq 1 \}$

**Zad. 24.**

Który z poniższych języków jest generowany przez następującą gramatykę?

$$\begin{aligned} S &\rightarrow aSb \mid aAb \\ A &\rightarrow bAa \mid bBa \\ B &\rightarrow aBb \mid ab \end{aligned}$$

- (A)  $\{ a^n b^{n+m} a^{m+k} b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (B)  $\{ a^n b^m a^m b^n a^k b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (C)  $\{ a^n b^m a^k b^n a^m b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (D)  $\{ a^n b^m a^k b^k a^m b^n \mid n \geq 1, m \geq 1, k \geq 1 \}$
- (E)  $\{ a^n b^{m+k} a^{m+n} b^{n+k} a^k b^{k+m+n} \mid n \geq 1, m \geq 1, k \geq 1 \}$

**Zad. 25.**

Który z poniższych języków jest generowany przez następującą gramatykę?

$$\begin{aligned} S &\rightarrow AC \\ A &\rightarrow aAb \mid aBb \\ B &\rightarrow bBa \mid ba \end{aligned}$$

$$C \rightarrow aCb \mid ab$$

- (A)  $\{ a^n b^{n+m} a^{m+k} b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (B)  $\{ a^n b^m a^m b^n a^k b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (C)  $\{ a^n b^m a^k b^k a^m b^n \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (D)  $\{ a^n b^m a^k b^n a^m b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (E)  $\{ a^n b^{m+k} a^{m+n} b^{n+k} a^k b^{k+m+n} \mid n \geq 1, m \geq 1, k \geq 1 \}$

**Zad. 26.**

Który z poniższych języków jest generowany przez następującą gramatykę?

$$S \rightarrow aSb \mid aBb$$

$$A \rightarrow bAa \mid ba$$

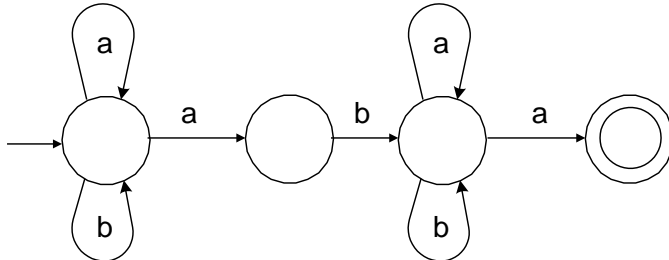
$$B \rightarrow aBb \mid aAb$$

- (A)  $\{ a^n b^{n+m} a^{m+k} b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (B)  $\{ a^n b^m a^m b^n a^k b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (C)  $\{ a^n b^m a^k b^n a^m b^k \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (D)  $\{ a^n b^m a^k b^k a^m b^n \mid n \geq 1, m \geq 1, k \geq 1 \}$   
 (E)  $\{ a^n b^m a^m b^n \mid n \geq 2, m \geq 1 \}$

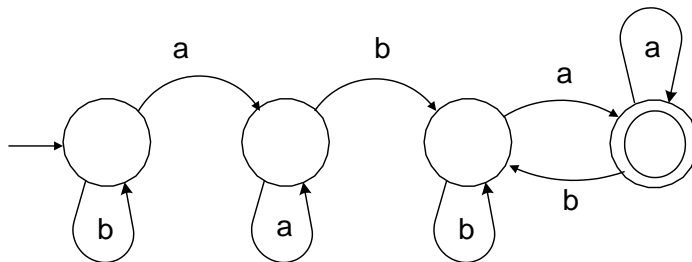
**Zad. 27.**

Dane są dwa automaty skończone:

Automat I:



Automat II:



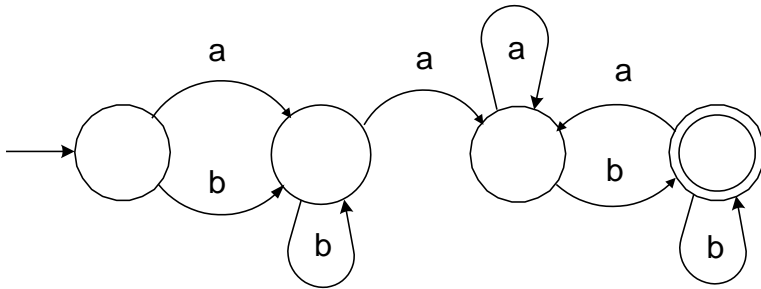
Jaki język jest akceptowany przez powyższe automaty skończone?

- (A) Automat I –  $(a|b)^*(a|b)a(a|b)^*b$ , automat II –  $(a|b)^*ab(a|b)^*a$   
 (B) Automat I –  $(a|b)b^*aa^*(a|b)b^*$ , automat II –  $(a|b)^*(a|b)a(a|b)^*b$   
 (C) Obydwa automaty –  $(a|b)^*ab(a|b)^*a$   
 (D) Obydwa automaty –  $(a|b)^*(a|b)a(a|b)^*b$   
 (E) Automat I –  $(a|b)^*ab(a|b)^*a$ , automat II –  $b^*aa^*bb^*(a|b)a^*$

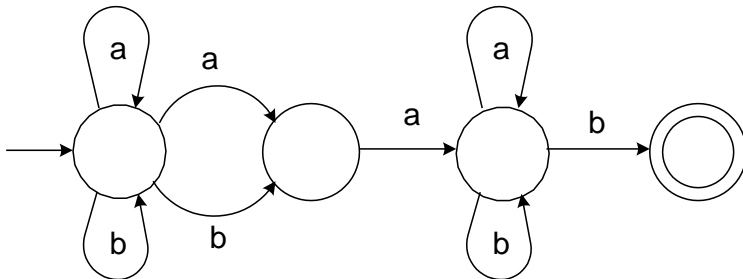
**Zad. 28.**

Dane są dwa automaty skończone:

Automat I:



Automat II:



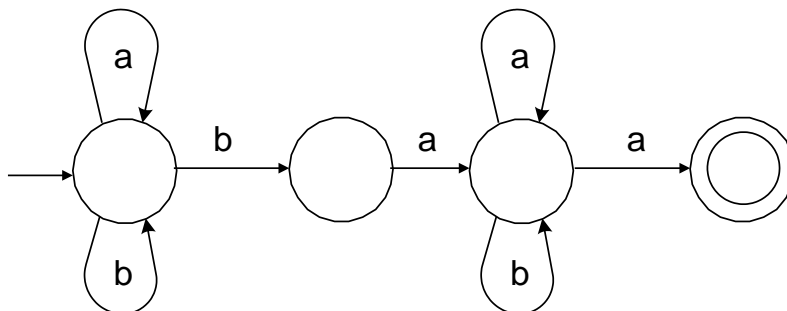
Jaki język jest akceptowany przez powyższe automaty skończone?

- (A) Obydwa automaty –  $(a|b)^*(a|b)a(a|b)^*b$
- (B) Automat I –  $(a|b)^*(a|b)a(a|b)^*b$ , automat II –  $(a|b)^*ab(a|b)^*a$
- (C) Automat I –  $(a|b)b^*aa^*(a|b)b^*$ , automat II –  $(a|b)^*(a|b)a(a|b)^*b$
- (D) Obydwa automaty –  $(a|b)^*ab(a|b)^*a$
- (E) Automat I –  $(a|b)^*ab(a|b)^*a$ , automat II –  $b^*aa^*bb^*(a|b)a^*$

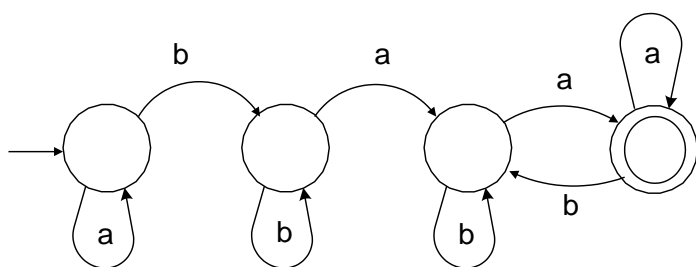
**Zad. 29.**

Dane są dwa automaty skończone:

Automat I:



Automat II:



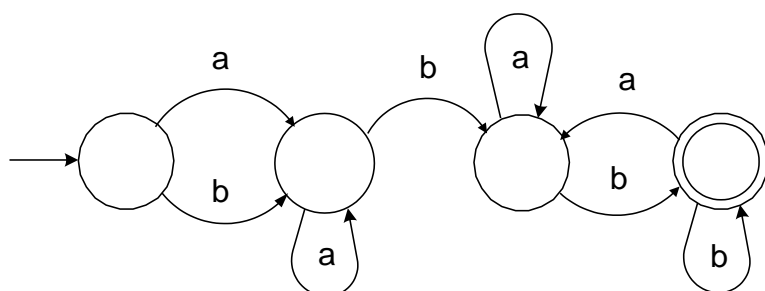
Jaki język jest akceptowany przez powyższe automaty skończone?

- (A) Automat I –  $(a|b)^*(a|b)a(a|b)^*b$ , automat II –  $(a|b)^*ab(a|b)^*a$   
 (B) Automat I –  $(a|b)b^*aa^*(a|b)b^*$ , automat II –  $(a|b)^*(a|b)a(a|b)^*b$   
 (C) Obydwa automaty –  $(a|b)^*ab(a|b)^*a$   
 (D) Obydwa automaty –  $(a|b)^*ba(a|b)^*a$   
 (E) Automat I –  $(a|b)^*ab(a|b)^*a$ , automat II –  $b^*aa^*bb^*(a|b)a^*$

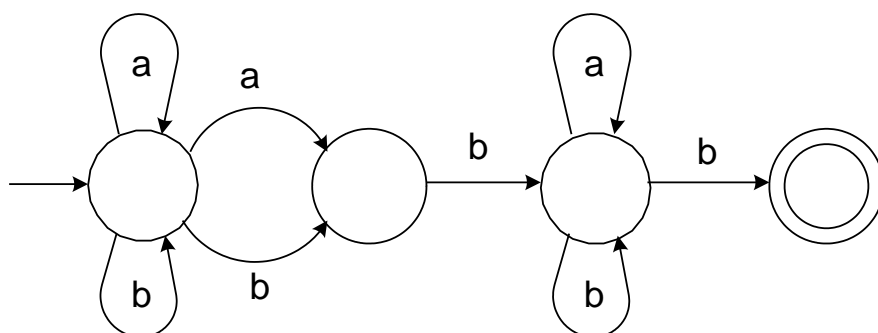
### Zad. 30.

Dane są dwa automaty skończone:

Automat I:



Automat II:



Jaki język jest akceptowany przez powyższe automaty skończone?

- (A) Obydwa automaty –  $(a|b)^*(a|b)a(a|b)^*b$   
 (B) Automat I –  $(a|b)^*(a|b)a(a|b)^*b$ , automat II –  $(a|b)^*ab(a|b)^*a$   
 (C) Automat I –  $(a|b)b^*aa^*(a|b)b^*$ , automat II –  $(a|b)^*(a|b)a(a|b)^*b$   
 (D) Obydwa automaty –  $(a|b)^*(a|b)b(a|b)^*b$   
 (E) Automat I –  $(a|b)^*ab(a|b)^*a$ , automat II –  $b^*aa^*bb^*(a|b)a^*$

### Zad. 31.

Dana jest następująca gramatyka atrybutywna typu S:

|                      |  |
|----------------------|--|
| $S \rightarrow L$    | $S.s \leftarrow L.p2$  |
| $L \rightarrow aL_1$ | $L.p1 \leftarrow 'a' \parallel L_1.p1$<br>$L.p2 \leftarrow 'a' \parallel L_1.p1 \parallel L_1.p1 \parallel L_1.p2$ |
| $L \rightarrow a$    | $L.p1 \leftarrow 'a'$<br>$L.p2 \leftarrow 'a'$   |

Uwaga: symbol  $\parallel$  oznacza konkatenaację łańcuchów znakowych, 'a' oznacza literę  $a$  rozumianą jako łańcuch znakowy (o długości jeden).

Które z poniższych stwierdzeń jest prawdziwe?

- (A) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^2, m > 0 \}$
- (B) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^3, m > 0 \}$
- (C) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m > 0 \}$
- (D) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 3^m, m > 0 \}$
- (E) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m = 2^k, k > 0 \}$

### Zad. 32.

Dana jest następująca gramatyka atrybutywna typu S:

|                      |  |
|----------------------|--|
| $S \rightarrow L$    | $S.s \leftarrow L.p$                   |
| $L \rightarrow aL_1$ | $L.p \leftarrow L_1.p \parallel L_1.p$ |
| $L \rightarrow a$    | $L.p \leftarrow 'aa'$                  |

Uwaga: symbol  $\parallel$  oznacza konkatenaację łańcuchów znakowych, 'a' oznacza literę  $a$  rozumianą jako łańcuch znakowy (o długości jeden).

Które z poniższych stwierdzeń jest prawdziwe?

- (A) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^2, m > 0 \}$
- (B) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^3, m > 0 \}$
- (C) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m > 0 \}$
- (D) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 3^m, m > 0 \}$

- (E) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m = 2^k, k > 0 \}$

### Zad. 33.

Dana jest następująca gramatyka atrybutywna typu S:

|                      |  |
|----------------------|--|
| $S \rightarrow L$    | $S.s \leftarrow L.p3$  |
| $L \rightarrow aL_1$ | $L.p1 \leftarrow 'a' \parallel L_1.p1$<br>$L.p2 \leftarrow 'a' \parallel L_1.p1 \parallel L_1.p1 \parallel L_1.p2$<br>$L.p3 \leftarrow 'a' \parallel L_1.p1 \parallel L_1.p1 \parallel L_1.p1 \parallel L_1.p2 \parallel L_1.p2 \parallel L_1.p2 \parallel L_1.p3$ |
| $L \rightarrow a$    | $L.p1 \leftarrow 'a'$<br>$L.p2 \leftarrow 'a'$<br>$L.p3 \leftarrow 'a'$  |

Uwaga: symbol  $\parallel$  oznacza konkatenację łańcuchów znakowych, 'a' oznacza literę *a* rozumianą jako łańcuch znakowy (o długości jeden).

Które z poniższych stwierdzeń jest prawdziwe?

- (A) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^2, m > 0 \}$
- (B) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^3, m > 0 \}$
- (C) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m > 0 \}$
- (D) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 3^m, m > 0 \}$
- (E) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m = 2^k, k > 0 \}$

### Zad. 34.

Dana jest następująca gramatyka atrybutywna typu S:

|                      |  |
|----------------------|--|
| $S \rightarrow L$    | $S.s \leftarrow L.p$                                   |
| $L \rightarrow aL_1$ | $L.p \leftarrow L_1.p \parallel L_1.p \parallel L_1.p$ |
| $L \rightarrow a$    | $L.p \leftarrow 'aaa'$                                 |

Uwaga: symbol  $\parallel$  oznacza konkatenację łańcuchów znakowych, 'aaa' oznacza łańcuch trzech liter *a*.

Które z poniższych stwierdzeń jest prawdziwe?

- (A) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^2, m > 0 \}$



- (B) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = m^3, m > 0 \}$
- (C) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m > 0 \}$
- (D) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 3^m, m > 0 \}$
- (E) Słowa języka generowanego przez gramatykę bezkontekstową, na której oparta jest powyższa gramatyka atrybutywna mają postać  $\{ a^n \mid n > 0 \}$ , zaś słowa języka będącego wynikiem tłumaczenia mają postać  $\{ a^n \mid n = 2^m, m = 2^k, k > 0 \}$

### Zad. 35.

Głównym zadaniem parsera jest:

- ~~(A)~~ Optymalizacja tzw. kodu pośredniego.
- ☒ (B) Wykonywanie analizy syntaktycznej – sprawdzanie poprawności składniowej i tworzenie drzewa rozbioru dla analizowanego ciągu tokenów. ?v
- (C) Wykonywanie analizy semantycznej – np. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.
- ~~(D)~~ Generacja kodu docelowego i optymalizacja przedziału zasobów, głównie związanych z procesorem i pamięcią wewnętrzną.
- (E) Wykonywanie analizy leksykalnej – sprawdzanie poprawności elementarnych składników tłumaczonego tekstu i przekształcanie ich na tokeny.

### Zad. 36.

Głównym zadaniem skanera jest:

- (A) Optymalizacja tzw. kodu pośredniego. X
- (B) Wykonywanie analizy semantycznej – np. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.
- (C) Wykonywanie analizy syntaktycznej – sprawdzanie poprawności składniowej i tworzenie drzewa rozbioru dla analizowanego ciągu tokenów.
- (D) Generacja kodu docelowego i optymalizacja przedziału zasobów, głównie związanych z procesorem i pamięcią wewnętrzną.
- (E) Wykonywanie analizy leksykalnej – sprawdzanie poprawności „słownikowej” elementarnych składników tłumaczonego tekstu i przekształcanie ich na tokeny.

### Zad. 37.

Które z poniższych stwierdzeń dotyczących parserów jest **nieprawdziwe**?

- (A) Parser jest rodzajem deterministycznego automatu ze stosem.
- (B) Zadaniem analizatora syntaktycznego jest sprawdzanie poprawności składniowej i tworzenie drzewa rozbioru dla analizowanego ciągu tokenów.
- (C) Celem projektanta parsera jest zbudowanie deterministycznego automatu ze stosem w oparciu o daną gramatykę bezkontekstową.

- (D) Istnieją gramatyki bezkontekstowe, w oparciu o które nie można zbudować parsera top-down przy równoczesnej możliwości zbudowania w oparciu o nie parsera bottom-up.
- (E) W oparciu o gramatykę z lewostronną rekursją nie można zbudować parsera typu bottom-up.

**Zad. 38.**

Które z poniższych stwierdzeń dotyczących parserów jest **nieprawdziwe**?

- (A) Parser jest rodzajem deterministycznego automatu ze stosem.
- (B) Zadaniem analizatora syntaktycznego jest sprawdzanie poprawności składniowej i tworzenie drzewa rozbioru dla analizowanego ciągu tokenów.
- (C) Celem projektanta parsera jest zbudowanie deterministycznego automatu ze stosem w oparciu o daną gramatykę bezkontekstową.
- (D) Istnieją gramatyki bezkontekstowe, w oparciu o które nie można zbudować parsera bottom-up przy równoczesnej możliwości zbudowania w oparciu o nie parsera top-down.
- (E) W oparciu o gramatykę z lewostronną rekursją nie można zbudować parsera typu top-down.

**Zad. 39.**

Które z poniższych stwierdzeń jest **nieprawdziwe**?

- (A) Projektant skanera wykorzystuje głównie mechanizmy teoretyczne związane z automatami skończonymi oraz gramatykami i wyrażeniami regularnymi.
- (B) Parser dokonuje rozbioru wejściowego ciągu tokenów w oparciu o gramatykę bezkontekstową opisującą składnię języka źródłowego, budując drzewo rozbioru syntaktycznego.
- (C) Dwoma zasadniczymi metodami parsingu są: metoda top-down polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia lewostronnego poczynając od korzenia drzewa (symbolu początkowego gramatyki) oraz metoda bottom-up polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia prawostronnego poczynając od liści drzewa (analizowanego ciągu tokenów).
- (D) Podstawowymi zadaniami analizy semantycznej jest m.in. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.
- (E) Generowanie kodu ostatecznego (docelowego) jest niezależne od platformy systemowo-sprzętowej, na której ma działać program będący wynikiem tłumaczenia.

**Zad. 40.**

Które z poniższych stwierdzeń jest **nieprawdziwe**?

- (A) Projektant parsera wykorzystuje głównie mechanizmy teoretyczne związane z automatami skończonymi oraz gramatykami i wyrażeniami regularnymi.
- (B) Parser dokonuje rozbioru wejściowego ciągu tokenów w oparciu o gramatykę bezkontekstową opisującą składnię języka źródłowego, budując drzewo rozbioru syntaktycznego.

- (C) Dwoma zasadniczymi metodami parsingu są: metoda top-down polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia lewostronnego poczynając od korzenia drzewa (symbolu początkowego gramatyki) oraz metoda bottom-up polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia prawostronnego poczynając od liści drzewa (analizowanego ciągu tokenów).
- (D) Podstawowymi zadaniami analizy semantycznej jest m.in. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.
- (E) Generowanie kodu ostatecznego (docelowego) jest silnie zależne od platformy systemowo-sprzętowej, na której ma działać program będący wynikiem tłumaczenia.

#### Zad. 41.

Które z poniższych stwierdzeń jest **nieprawdziwe**?

- (A) Projektant skanera wykorzystuje głównie mechanizmy teoretyczne związane z automatami skończonymi oraz gramatykami i wyrażeniami regularnymi.
- (B) Parser dokonuje rozbioru wejściowego ciągu tokenów w oparciu o gramatykę bezkontekstową opisującą składnię języka źródłowego, budując drzewo rozbioru syntaktycznego.
- (C) Dwoma zasadniczymi metodami parsingu są: metoda top-down polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia lewostronnego poczynając od korzenia drzewa (symbolu początkowego gramatyki) oraz metoda bottom-up polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia prawostronnego poczynając od liści drzewa (analizowanego ciągu tokenów).
- (D) Podstawowymi zadaniami analizy semantycznej jest m.in. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.
- (E) Analiza syntaktyczna bazuje na metodach opartych o gramatyki atrybutywne i definicje kierowane składnią.

#### Zad. 42.

Które z poniższych stwierdzeń jest **nieprawdziwe**?

- (A) Projektant parsera wykorzystuje głównie mechanizmy teoretyczne związane z automatami skończonymi oraz gramatykami i wyrażeniami regularnymi.
- (B) Parser dokonuje rozbioru wejściowego ciągu tokenów w oparciu o gramatykę bezkontekstową opisującą składnię języka źródłowego, budując drzewo rozbioru syntaktycznego.
- (C) Dwoma zasadniczymi metodami parsingu są: metoda top-down polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia lewostronnego poczynając od korzenia drzewa (symbolu początkowego gramatyki) oraz metoda bottom-up polegająca na budowaniu drzewa rozbioru syntaktycznego poprzez odtwarzanie wyprowadzenia prawostronnego poczynając od liści drzewa (analizowanego ciągu tokenów).
- (D) Podstawowymi zadaniami analizy syntaktycznej jest m.in. sprawdzanie zgodności typów składników poszczególnych konstrukcji językowych, sprawdzanie unikalności

etykiet i innych elementów języka, kontrola poprawności przekazywania sterowania w programie. itd.

- (E) Analiza semantyczna bazuje na metodach opartych o gramatyki atrybutywne i definicje kierowane składnią.

**Zad. 43.**

Które z poniższych trzech formuł dotyczących wyrażeń regularnych **a** i **b** można uznać za tożsamości?

Formuła I:

$$\mathbf{a(a|b)^*|b(a|b)^* = (a|b)^*(a|b)}$$

Formuła II:

$$\mathbf{a(a|b)^*a = a(ba|a)^*}$$

Formuła III:

$$\mathbf{a(a|b)^*b = a(a^*b^*)^*b}$$

- (A) Tylko formuła III nie jest tożsamością, pozostałe są tożsamościami.  
(B) Tylko formuła II nie jest tożsamością, pozostałe są tożsamościami.  
(C) Żadna formuła nie jest tożsamością.  
(D) Tylko formuła I nie jest tożsamością, pozostałe są tożsamościami.  
(E) Wszystkie formuły są tożsamościami.

**Zad. 44.**

Które z poniższych trzech formuł dotyczących wyrażeń regularnych **a** i **b** można uznać za tożsamości?

Formuła I:

$$\mathbf{a(a|b)^*|b(a|b)^* = (a|b)^*(a|b)}$$

Formuła II:

$$\mathbf{a(a|b)^*b = a(a^*b^*)^*b}$$

Formuła III:

$$\mathbf{a(ab|b)^*b = aa^*b(aa^*b)^*}$$

- (A) Tylko formuła III nie jest tożsamością, pozostałe są tożsamościami.  
(B) Żadna formuła nie jest tożsamością.  
(C) Tylko formuła I nie jest tożsamością, pozostałe są tożsamościami.  
(D) Wszystkie formuły są tożsamościami.  
(E) Tylko formuła II nie jest tożsamością, pozostałe są tożsamościami.

**Zad. 45.**

Które z poniższych trzech formuł dotyczących wyrażeń regularnych **a** i **b** można uznać za tożsamości?

Formuła I:

$$\mathbf{b(ab|b)^*a = b(ba|a)^*a}$$

Formuła II:

$$\mathbf{a(a|b)^*a = a(ba|a)^*a}$$

Formuła III:

$$a(ab|b)^*a = aa^*b(aa^*b)^*a$$

- (A) Tylko formuła III nie jest tożsamością, pozostałe są tożsamościami.
- (B) Tylko formuła II nie jest tożsamością, pozostałe są tożsamościami.
- (C) Żadna formuła nie jest tożsamością.
- (D) Tylko formuła I nie jest tożsamością, pozostałe są tożsamościami.
- (E) Wszystkie formuły są tożsamościami.

**Zad. 46.**

Które z poniższych trzech formuł dotyczących wyrażeń regularnych **a** i **b** można uznać za tożsamości?

Formuła I:

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

Formuła II:

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

Formuła III:

$$a(ab|b)^*b|\emptyset = \varepsilon|aa(b|ba)^*b$$

- (A) Tylko formuła III nie jest tożsamością, pozostałe są tożsamościami.
- (B) Żadna formuła nie jest tożsamością.
- (C) Tylko formuła I nie jest tożsamością, pozostałe są tożsamościami.
- (D) Wszystkie formuły są tożsamościami.
- (E) Tylko formuła II nie jest tożsamością, pozostałe są tożsamościami.

**Zad. 47.**

Poniższy język:

$$\{ ab, ba, aba, bab, abab, baba, ababa, babab, \dots \}$$

jest generowany przez gramatykę:

- (A)  $S \rightarrow A \mid B$   
 $A \rightarrow ab \mid aB$   
 $B \rightarrow ba \mid bA$
- (B)  $S \rightarrow A \mid B$   
 $A \rightarrow a \mid ab \mid abA$   
 $B \rightarrow b \mid ba \mid baB$
- (C)  $S \rightarrow A \mid B$   
 $A \rightarrow \varepsilon \mid a \mid ab \mid abA$   
 $B \rightarrow \varepsilon \mid b \mid ba \mid baB$
- (D)  $S \rightarrow a \mid b \mid Sa \mid Sb$
- (E)  $S \rightarrow a \mid b \mid aA \mid bB$   
 $A \rightarrow bS \mid b$   
 $B \rightarrow aS \mid a$

**Zad. 48.**

Poniższy język:

**{ a, b, ab, ba, aba, bab, abab, baba, ababa, babab, ... }**

jest generowany przez gramatykę:

(A)  $S \rightarrow A \mid B$

$A \rightarrow ab \mid aB$

$B \rightarrow ba \mid bA$

(B)  $S \rightarrow A \mid B$

$A \rightarrow a \mid ab \mid abA$

$B \rightarrow b \mid ba \mid baB$

(C)  $S \rightarrow A \mid B$

$A \rightarrow \varepsilon \mid a \mid ab \mid abA$

$B \rightarrow \varepsilon \mid b \mid ba \mid baB$

(D)  $S \rightarrow a \mid b \mid Sa \mid Sb$

(E)  $S \rightarrow a \mid b \mid aA \mid bB$

$A \rightarrow bS \mid b$

$B \rightarrow aS \mid a$

#### Zad. 49.

Poniższy język:

**{  $\varepsilon$ , a, b, ab, ba, aba, bab, abab, baba, ababa, babab, ... }**

jest generowany przez gramatykę:

(A)  $S \rightarrow A \mid B$

$A \rightarrow ab \mid aB$

$B \rightarrow ba \mid bA$

(B)  $S \rightarrow A \mid B$

$A \rightarrow a \mid ab \mid abA$

$B \rightarrow b \mid ba \mid baB$

(C)  $S \rightarrow A \mid B$

$A \rightarrow \varepsilon \mid a \mid ab \mid abA$

$B \rightarrow \varepsilon \mid b \mid ba \mid baB$

(D)  $S \rightarrow a \mid b \mid Sa \mid Sb$

(E)  $S \rightarrow a \mid b \mid aA \mid bB$

$A \rightarrow bS \mid b$

$B \rightarrow aS \mid a$

#### Zad. 50.

Poniższy język:

**{ ab, ba, abab, baba, ababab, bababa, ... }**

jest generowany przez gramatykę:

(A)  $S \rightarrow A \mid B$

$A \rightarrow ab \mid abA$

$B \rightarrow ba \mid baB$

- (B)  $S \rightarrow A \mid B$   
 $A \rightarrow a \mid ab \mid abA$   
 $B \rightarrow b \mid ba \mid baB$
- (C)  $S \rightarrow A \mid B$   
 $A \rightarrow \varepsilon \mid a \mid ab \mid abA$   
 $B \rightarrow \varepsilon \mid b \mid ba \mid baB$
- (D)  $S \rightarrow ab \mid ba \mid Sab \mid Sba$
- (E)  $S \rightarrow ab \mid ba \mid abA \mid baB$   
 $A \rightarrow bS \mid b$   
 $B \rightarrow aS \mid a$

### Zad. 51.

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Przykładem błędu leksykalnego w programie komputerowym jest użycie niedozwolonego znaku w identyfikatorze.
- (B) Błędem syntaktycznym w programie komputerowym jest opuszczenie nawiasu zamykającego, jeśli wcześniej użyto nawiasu otwierającego.
- (C) Błędem semantycznym w programie komputerowym jest użycie w wyrażeniu niezadeklarowanego wcześniej identyfikatora.
- (D) Niedozwolony skok do wnętrza ciała pętli programowej należy do grupy błędów syntaktycznych.
- (E) Użycie tego samego identyfikatora kategorii w dwóch różnych definicjach typu wyliczeniowego to przykład błędu semantycznego.

### Zad. 52.

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Przykładem błędu leksykalnego w programie komputerowym jest użycie niedozwolonego znaku w identyfikatorze.
- (B) Błędem syntaktycznym w programie komputerowym jest opuszczenie nawiasu zamykającego, jeśli wcześniej użyto nawiasu otwierającego.
- (C) Błędem semantycznym w programie komputerowym jest użycie w wyrażeniu niezadeklarowanego wcześniej identyfikatora.
- (D) Niedozwolony skok do wnętrza ciała pętli programowej należy do grupy błędów semantycznych.
- (E) Użycie tego samego identyfikatora kategorii w dwóch różnych definicjach typu wyliczeniowego to przykład błędu syntaktycznego.

### Zad. 53.

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Przykładem błędu syntaktycznego w programie komputerowym jest użycie niedozwolonego znaku w identyfikatorze.
- (B) Błędem syntaktycznym w programie komputerowym jest opuszczenie nawiasu zamykającego, jeśli wcześniej użyto nawiasu otwierającego.
- (C) Błędem semantycznym w programie komputerowym jest użycie w wyrażeniu niezadeklarowanego wcześniej identyfikatora.

- (D) Niedozwolony skok do wnętrza ciała pętli programowej należy do grupy błędów semantycznych.
- (E) Użycie tego samego identyfikatora kategorii w dwóch różnych definicjach typu wyliczeniowego to przykład błędu semantycznego.

**Zad. 54.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Przykładem błędu leksykalnego w programie komputerowym jest użycie niedozwolonego znaku w identyfikatorze.
- (B) Błędem leksykalnym w programie komputerowym jest opuszczenie nawiasu zamykającego, jeśli wcześniej użyto nawiasu otwierającego.
- (C) Błędem semantycznym w programie komputerowym jest użycie w wyrażeniu niezadeklarowanego wcześniej identyfikatora.
- (D) Niedozwolony skok do wnętrza ciała pętli programowej należy do grupy błędów semantycznych.
- (E) Użycie tego samego identyfikatora kategorii w dwóch różnych definicjach typu wyliczeniowego to przykład błędu semantycznego.

**Zad. 55.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Kompilacja to całościowe tłumaczenie tekstu źródłowego programu lub pewnego jego modułu.
- (B) Interpretacja to cykliczne tłumaczenie niewielkich partii tekstu źródłowego programu i natychmiastowe ich wykonywanie.
- (C) Interpretacja jest wolniejsza od kompilacji, gdyż w czasie wykonywania programu następuje także jego tłumaczenie.
- (D) Języki skryptowe są powszechnie używane w zastosowaniach sieciowych, gdyż umożliwiają przesyłanie programów w postaci plików tekstowych i interpretowanie ich u odbiorcy.
- (E) Do wykonywania skompilowanego programu potrzebna jest obecność kompilatora w pamięci wewnętrznej komputera.

**Zad. 56.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Kompilacja to całościowe tłumaczenie tekstu źródłowego programu lub pewnego jego modułu.
- (B) Interpretacja to cykliczne tłumaczenie niewielkich partii tekstu źródłowego programu i natychmiastowe ich wykonywanie.
- (C) Interpretacja jest szybsza od kompilacji, gdyż nie wymaga zapisywania wyniku tłumaczenia w pamięci zewnętrznej.
- (D) Języki skryptowe są powszechnie używane w zastosowaniach sieciowych, gdyż umożliwiają przesyłanie programów w postaci plików tekstowych i interpretowanie ich u odbiorcy.
- (E) Do wykonywania przetłumaczonego programu nie jest potrzebna obecność kompilatora w pamięci wewnętrznej komputera.



**Zad. 57.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Interpretacja umożliwia interakcyjne ingerowanie programisty w tekst źródłowy programu podczas tłumaczenia.
- (B) Kompilacja to cykliczne tłumaczenie niewielkich partii tekstu źródłowego programu i natychmiastowe ich wykonywanie.
- (C) Interpretacja jest wolniejsza od kompilacji, gdyż interpretacja wiąże się z wielokrotnym tłumaczeniem np. fragmentów kodu wchodzących w skład ciał pętli, podczas gdy ciała pętli podczas kompilacji tłumaczone są tylko raz.
- (D) Języki skryptowe są powszechnie używane w zastosowaniach sieciowych, gdyż umożliwiają przesyłanie programów w postaci plików tekstowych i interpretowanie ich u odbiorcy.
- (E) Do wykonywania przetłumaczonego programu nie jest potrzebna obecność kompilatora w pamięci wewnętrznej komputera.

**Zad. 58.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Kompilacja umożliwia interakcyjne ingerowanie programisty w tekst źródłowy programu podczas tłumaczenia.
- (B) Interpretacja to cykliczne tłumaczenie niewielkich partii tekstu źródłowego programu i natychmiastowe ich wykonywanie.
- (C) Interpretacja jest wolniejsza od kompilacji, gdyż interpretacja wiąże się z wielokrotnym tłumaczeniem np. fragmentów kodu wchodzących w skład ciał pętli, podczas gdy ciała pętli podczas kompilacji tłumaczone są tylko raz.
- (D) Języki skryptowe są powszechnie używane w zastosowaniach sieciowych, gdyż umożliwiają przesyłanie programów w postaci plików tekstowych i interpretowanie ich u odbiorcy.
- (E) Do wykonywania przetłumaczonego programu nie jest potrzebna obecność kompilatora w pamięci wewnętrznej komputera.

**Zad. 59.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Automat ze stosem realizujący parsing w technologii top-down (LL) akceptuje przez pusty stos i nie używa mechanizmu stanów.
- (B) Automat ze stosem realizujący parsing w technologii bottom-up (LR) wykorzystuje mechanizm stanów (także do akceptacji wejścia).
- (C) Przy budowaniu tablicy parsera top-down (LL(1)) wykorzystywane są zbiory  $FIRST_1$  i  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.
- (D) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory sytuacji dopuszczalnych dla przedrostków żywotnych.
- (E) Przy budowaniu tablicy parsera bottom-up (SLR(1)) nie możemy skorzystać z gramatyki wymagającej lewostronnej faktoryzacji.

**Zad. 60.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Automat ze stosem realizujący parsing w technologii top-down (LL) akceptuje przez pusty stos i nie używa mechanizmu stanów.
- (B) Automat ze stosem realizujący parsing w technologii bottom-up (LR) wykorzystuje mechanizm stanów (także do akceptacji wejścia).
- (C) Przy budowaniu tablicy parsera top-down (LL(1)) wykorzystywane są zbiory  $FIRST_1$  i  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.
- (D) Przy budowaniu tablicy parsera top-down (LL(1)) wykorzystywane są zbiory sytuacji dopuszczalnych dla przedrostków żywotnych.
- (E) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.

#### Zad. 61.

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Automat ze stosem realizujący parsing w technologii top-down (LL) akceptuje przez pusty stos i nie używa mechanizmu stanów.
- (B) Automat ze stosem realizujący parsing w technologii bottom-up (LR) nie może być oparty o gramatykę wymagającą lewostronnej faktoryzacji.
- (C) Przy budowaniu tablicy parsera top-down (LL(1)) wykorzystywane są zbiory  $FIRST_1$  i  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.
- (D) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory sytuacji dopuszczalnych dla przedrostków żywotnych.
- (E) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.

#### Zad. 62.

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Automat ze stosem realizujący parsing w technologii bottom-up (LR) nie może być oparty o gramatykę wymagającą lewostronnej faktoryzacji.
- (B) Automat ze stosem realizujący parsing w technologii bottom-up (LR) wykorzystuje mechanizm stanów (także do akceptacji wejścia).
- (C) Przy budowaniu tablicy parsera top-down (LL(1)) wykorzystywane są zbiory  $FIRST_1$  i  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.
- (D) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory sytuacji dopuszczalnych dla przedrostków żywotnych.
- (E) Przy budowaniu tablicy parsera bottom-up (SLR(1)) wykorzystywane są zbiory  $FOLLOW_1$  wyznaczone dla symboli nieterminalnych gramatyki.

#### Zad. 63.

Które z poniższych stwierdzeń dotyczących gramatyk atrybutywnych **nie jest prawdziwe**?

- (A) Gramatyka atrybutywna to system bazujący na gramatyce bezkontekstowej, w którym symbolom gramatyki przypisuje się atrybuty, a produkcjom reguły wyliczania tych atrybutów.
- (B) Gramatyki atrybutywne wykorzystywane są do przeprowadzania analizy leksykalnej.
- (C) Jeśli wszystkie atrybuty w gramatyce atrybutywnej zawsze posiadają reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, to potrafimy wyliczyć wartości tych atrybutów równocześnie z parsingiem typu bottom-

up wykorzystując mechanizm drugiego stosu obsługiwanego synchronicznie z zasadniczym stosem parsera.

- (D) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to pod pewnymi warunkami dotyczącymi reguł obliczania tych ostatnich atrybutów można wyliczyć wartości wszystkich atrybutów równocześnie z parsingiem typu top-down.
- (E) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to może zajść konieczność dwufazowej obsługi takich atrybutów: w pierwszej fazie parser buduje drzewo rozbioru, a w drugiej fazie jest ono „dekorowane” wartościami atrybutów podczas często wielokrotnego przeglądania jego wierzchołków.

#### **Zad. 64.**

Które z poniższych stwierdzeń dotyczących gramatyk atrybutywnych **nie jest prawdziwe**?

- (A) Gramatyka atrybutywna to system bazujący na gramatyce bezkontekstowej, w którym symbolom gramatyki przypisuje się atrybuty, a produkcjom reguły wyliczania tych atrybutów.
- (B) Gramatyki atrybutywne wykorzystywane są do przeprowadzania analizy semantycznej i generacji kodu pośredniego.
- (C) Jeśli wszystkie atrybuty w gramatyce atrybutywnej zawsze posiadają reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, to potrafimy wyliczyć wartości tych atrybutów równocześnie z parsingiem typu top-down wykorzystując mechanizm drugiego stosu obsługiwanego synchronicznie z zasadniczym stosem parsera.
- (D) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to pod pewnymi warunkami dotyczącymi reguł obliczania tych ostatnich atrybutów można wyliczyć wartości wszystkich atrybutów równocześnie z parsingiem typu bottom-up, ale kosztem pewnej modyfikacji bazowej gramatyki bezkontekstowej (wprowadzenie produkcji pustych).
- (E) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to może zajść konieczność dwufazowej obsługi takich atrybutów: w pierwszej fazie parser buduje drzewo rozbioru, a w drugiej fazie jest ono „dekorowane” wartościami atrybutów podczas często wielokrotnego przeglądania jego wierzchołków.

#### **Zad. 65.**

Które z poniższych stwierdzeń dotyczących gramatyk atrybutywnych **nie jest prawdziwe**?

- (A) Gramatyka atrybutywna to system bazujący na gramatyce bezkontekstowej, w którym symbolom gramatyki przypisuje się atrybuty, a produkcjom reguły wyliczania tych atrybutów.
- (B) Gramatyki atrybutywne wykorzystywane są do przeprowadzania analizy semantycznej i generacji kodu pośredniego.
- (C) Jeśli wszystkie atrybuty w gramatyce atrybutywnej zawsze posiadają reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, to potrafimy wyliczyć wartości tych atrybutów równocześnie z parsingiem typu bottom-up wykorzystując mechanizm drugiego stosu obsługiwane synchronicznie z zasadniczym stosem parsera.
- (D) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to pod pewnymi warunkami dotyczącymi reguł obliczania tych ostatnich atrybutów (tzw. gramatyki L-atrybutywne) można wyliczyć wartości wszystkich atrybutów równocześnie z parsingiem typu top-down, ale kosztem pewnej modyfikacji bazowej gramatyki bezkontekstowej (wprowadzenie produkcji pustych).
- (E) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to może zajść konieczność dwufazowej obsługi takich atrybutów: w pierwszej fazie parser buduje drzewo rozbioru, a w drugiej fazie jest ono „dekorowane” wartościami atrybutów podczas często wielokrotnego przeglądania jego wierzchołków.

#### **Zad. 66.**

Które z poniższych stwierdzeń dotyczących gramatyk atrybutywnych **nie jest prawdziwe**?

- (A) Gramatyka atrybutywna to system bazujący na gramatyce regularnej używanej do analizy leksykalnej, w którym symbolom gramatyki przypisuje się atrybuty, a produkcjom reguły wyliczania tych atrybutów.
- (B) Jeśli wszystkie atrybuty w gramatyce atrybutywnej zawsze posiadają reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, to potrafimy wyliczyć wartości tych atrybutów równocześnie z parsingiem typu bottom-up wykorzystując mechanizm drugiego stosu obsługiwane synchronicznie z zasadniczym stosem parsera.
- (C) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to pod pewnymi warunkami dotyczącymi reguł obliczania tych ostatnich atrybutów (tzw. gramatyki L-atrybutywne) można wyliczyć wartości wszystkich atrybutów równocześnie z parsingiem typu top-down.
- (D) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to pod pewnymi warunkami dotyczącymi reguł obliczania tych ostatnich atrybutów (tzw. gramatyki L-atrybutywne) można wyliczyć wartości wszystkich atrybutów

równocześnie z parsyngiem typu bottom-up, ale kosztem pewnej modyfikacji bazowej gramatyki bezkontekstowej (wprowadzenie produkcji pustych).

- (E) Jeśli w gramatyce atrybutywnej istnieją nie tylko atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” stanowią lewe strony produkcji, ale także atrybuty posiadające reguły przypisane do tych produkcji, w których „właściciele atrybutów” pojawiają się w prawych stronach produkcji, to może zajść konieczność dwufazowej obsługi takich atrybutów: w pierwszej fazie parser buduje drzewo rozbioru, a w drugiej fazie jest ono „dekorowane” wartościami atrybutów podczas często wielokrotnego przeglądania jego wierzchołków.

#### **Zad. 67.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Bloki pośredniego kodu trójadresowego złożone z sekwencji skoków warunkowych i bezwarunkowych są wynikiem tłumaczenia wyrażeń logicznych wg zasady „short-circuit”
- (B) Optymalizacja „przez szparkę” pośredniego kodu trójadresowego pozwala na usprawnianie fragmentów kodu będących sekwencjami skoków warunkowych i bezwarunkowych.
- (C) Optymalizacja kodu pośredniego metodą analizy przepływu pozwala na optymalizację sekwencji skoków, mimo że rozkazy skoków są podstawą konstrukcji grafu obliczeń.
- (D) Usuwanie wspólnych podwyrażeń podczas optymalizacji pośredniego kodu trójadresowego może być dokonane metodą analizy przepływu.
- (E) Wykrywanie zmiennych indukowanych, związana z nimi redukcja mocy kodu oraz eliminacja zmiennych indukowanych jest wykonywana z wykorzystaniem metod analizy przepływu we fragmentach grafu obliczeń odpowiadających pętlom.

#### **Zad. 68.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Bloki pośredniego kodu trójadresowego złożone wyłącznie z sekwencji skoków warunkowych i bezwarunkowych są wynikiem tłumaczenia wyrażeń logicznych uwzględniającego każdorazowo konieczność obliczenia całego wyrażenia logicznego.
- (B) Optymalizacja „przez szparkę” pośredniego kodu trójadresowego pozwala na usprawnianie fragmentów kodu będących sekwencjami skoków warunkowych i bezwarunkowych.
- (C) Optymalizacja kodu pośredniego metodą analizy przepływu nie pozwala na optymalizację sekwencji skoków, gdyż rozkazy skoków są podstawą konstrukcji grafu obliczeń.
- (D) Warunkiem poprawnego przeprowadzenia transformacji usprawniających pośredni kod trójadresowy, takich jak usuwanie wspólnych podwyrażeń, eliminacja propagacji kopiowania, eliminacja kodu martwego, itd., jest wcześniejsze zebranie odpowiednich informacji o całym grafie obliczeń metodą analizy przepływu.
- (E) Wykrywanie zmiennych indukowanych, związana z nimi redukcja mocy kodu oraz eliminacja zmiennych indukowanych jest wykonywana z wykorzystaniem metod analizy przepływu we fragmentach grafu obliczeń odpowiadających pętlom.

#### **Zad. 69.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Bloki pośredniego kodu trójadresowego złożone z sekwencji skoków warunkowych i bezwarunkowych są wynikiem tłumaczenia wyrażeń logicznych wg zasady „short-circuit”
- (B) Optymalizacja „przez szparkę” pośredniego kodu trójadresowego pozwala na usprawnianie fragmentów kodu będących sekwencjami skoków warunkowych i bezwarunkowych.
- (C) Optymalizacja kodu pośredniego metodą analizy przepływu nie pozwala na optymalizację sekwencji skoków, gdyż rozkazy skoków są podstawą konstrukcji grafu obliczeń.
- (D) Warunkiem poprawnego przeprowadzenia transformacji usprawniających pośredni kod trójadresowy, takich jak usuwanie wspólnych podwyrażeń, eliminacja propagacji kopiowania, eliminacja kodu martwego, itd., jest wcześniejsze zebranie odpowiednich informacji o całym grafie obliczeń metodą analizy przepływu.
- (E) Wykrywanie zmiennych indukowanych, związana z nimi redukcja mocy kodu oraz eliminacja zmiennych indukowanych jest wykonywana z wykorzystaniem metod analizy przepływu w całym grafie obliczeń.

#### **Zad. 70.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Bloki pośredniego kodu trójadresowego złożone z sekwencji skoków warunkowych i bezwarunkowych są wynikiem tłumaczenia wyrażeń logicznych wg zasady „short-circuit”
- (B) Optymalizacja „przez szparkę” pośredniego kodu trójadresowego pozwala na usprawnianie fragmentów kodu będących sekwencjami skoków warunkowych i bezwarunkowych.
- (C) Optymalizacja kodu pośredniego metodą analizy przepływu nie pozwala na optymalizację sekwencji skoków, gdyż rozkazy skoków są podstawą konstrukcji grafu obliczeń.
- (D) Usuwanie wspólnych podwyrażeń podczas optymalizacji pośredniego kodu trójadresowego może być dokonane metodą „przez szparkę”.
- (E) Wykrywanie zmiennych indukowanych, związana z nimi redukcja mocy kodu oraz eliminacja zmiennych indukowanych jest wykonywana z wykorzystaniem metod analizy przepływu we fragmentach grafu obliczeń odpowiadających pętlom.

#### **Zad. 71.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Dla każdego automatu skończonego istnieje równoważny mu automat skończony deterministyczny.
- (B) Dla każdego automatu ze stosiem istnieje równoważny mu deterministyczny automat ze stosiem.
- (C) Dla każdej maszyny Turinga istnieje równoważna jej deterministyczna maszyna Turinga.
- (D) Nie dla każdej maszyny Turinga istnieje równoważna jej maszyna Turinga z własnością stopu (rozstrzygająca).

#### **Zad. 72.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Nie dla każdego automatu skończonego istnieje równoważny mu automat skończony deterministyczny.
- (B) Nie dla każdego automatu ze stosem istnieje równoważny mu deterministyczny automat ze stosem.
- (C) Dla każdej maszyny Turinga istnieje równoważna jej deterministyczna maszyna Turinga.
- (D) Nie dla każdej maszyny Turinga istnieje równoważna jej maszyna Turinga z własnością stopu (rozstrzygająca).

**Zad. 73.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Dla każdego automatu skończonego istnieje równoważny mu automat skończony deterministyczny.
- (B) Nie dla każdego automatu ze stosem istnieje równoważny mu deterministyczny automat ze stosem.
- (C) Nie dla każdej maszyny Turinga istnieje równoważna jej deterministyczna maszyna Turinga.
- (D) Nie dla każdej maszyny Turinga istnieje równoważna jej maszyna Turinga z własnością stopu (rozstrzygająca).

**Zad. 74.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Dla każdego automatu skończonego istnieje równoważny mu automat skończony deterministyczny.
- (B) Nie dla każdego automatu ze stosem istnieje równoważny mu deterministyczny automat ze stosem.
- (C) Dla każdej maszyny Turinga istnieje równoważna jej deterministyczna maszyna Turinga.
- (D) Dla każdej maszyny Turinga istnieje równoważna jej maszyna Turinga z własnością stopu (rozstrzygająca).

**Zad. 75.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Zastąpienie niedeterministycznej maszyny Turinga symulującą ją maszyną deterministyczną zostaje okupione wykładniczym wzrostem złożoności obliczeń.
- (B) Zastąpienie niedeterministycznego automatu skończonego równoważnym mu deterministycznym automatem skończonym nie zostaje okupione żadnym wzrostem złożoności obliczeń.
- (C) Deterministyczny automat skończony charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (wykonuje dokładnie jeden krok na jeden symbol wejściowy).

- (D) Deterministyczny automat ze stosem charakteryzuje się co najmniej kwadratowym czasem obliczeń ze względu na długość analizowanego wejścia.

**Zad. 76.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Zastąpienie niedeterministycznej maszyny Turinga symulującą ją maszyną deterministyczną nie zostaje okupione żadnym wzrostem złożoności obliczeń.
- (B) Zastąpienie niedeterministycznego automatu skończonego równoważnym mu deterministycznym automatem skończonym nie zostaje okupione żadnym wzrostem złożoności obliczeń.
- (C) Deterministyczny automat skończony charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (wykonuje dokładnie jeden krok na jeden symbol wejściowy).
- (D) Deterministyczny automat ze stosem charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (choć nie można powiedzieć, że wykonuje dokładnie jeden krok na jeden symbol wejściowy, jak to ma miejsce w przypadku automatu skończonego).

**Zad. 77.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Zastąpienie niedeterministycznej maszyny Turinga symulującą ją maszyną deterministyczną zostaje okupione wykładniczym wzrostem złożoności obliczeń.
- (B) Zastąpienie niedeterministycznego automatu skończonego równoważnym mu deterministycznym automatem skończonym zostaje okupione wielomianowym wzrostem złożoności obliczeń.
- (C) Deterministyczny automat skończony charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (wykonuje dokładnie jeden krok na jeden symbol wejściowy).
- (D) Deterministyczny automat ze stosem charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (choć nie można powiedzieć, że wykonuje dokładnie jeden krok na jeden symbol wejściowy, jak to ma miejsce w przypadku automatu skończonego).

**Zad. 78.**

Które z poniższych stwierdzeń **nie jest prawdziwe**?

- (A) Zastąpienie niedeterministycznej maszyny Turinga symulującą ją maszyną deterministyczną zostaje okupione wykładniczym wzrostem złożoności obliczeń.
- (B) Zastąpienie niedeterministycznego automatu skończonego równoważnym mu deterministycznym automatem skończonym nie zostaje okupione żadnym wzrostem złożoności obliczeń.
- (C) Deterministyczny automat skończony charakteryzuje się co najmniej kwadratowym czasem obliczeń ze względu na długość analizowanego wejścia.



- (D) Deterministyczny automat ze stosem charakteryzuje się liniowym czasem obliczeń ze względu na długość analizowanego wejścia (choć nie można powiedzieć, że wykonuje dokładnie jeden krok na jeden symbol wejściowy, jak to ma miejsce w przypadku automatu skończonego).

**Zad. 79.** Dla optymalizacji kodu pośredniego metodą analizy przepływu **nieprawdziwe** jest stwierdzenie:

- (A) Każdy blok podstawowy jest pojedynczą instrukcją kodu trójadresowego różną od instrukcji skoku, wywołania podprogramu lub powrotu z podprogramu.
- (B) Wierzchołkami grafu obliczeń są bloki podstawowe.
- (C) Każda krawędź w grafie obliczeń jest odzwierciedleniem możliwości przekazania sterowania między wierzchołkami grafu obliczeń zgodnie ze skierowaniem krawędzi.
- (D) Graf obliczeń jest to skierowany graf tworzony na podstawie trójadresowego kodu pośredniego.
- (E) Główną rolę w tworzeniu struktury grafu obliczeń poprzez przekształcanie strumienia instrukcji trójadresowych tworzących kod pośredni pełnią instrukcje skoków, wywołania podprogramu lub powrotu z podprogramu.

**Zad. 80.** Która spośród poniższych transformacji optymalizujących kod pośredni **nie może** być wykonana metodą optymalizacji przez szparkę?

- (A) Eliminacja zbędnych skoków,
- (B) Eliminacja wspólnych podwyrażeń,
- (C) Optymalizacja przebiegu sterowania poprzez zmniejszenie liczby skoków realizowanych w trakcie wykonania programu (wykrywanie i modyfikacja sytuacji typu: „skok do skoku”),
- (D) Reorganizacja kodu zawierającego skoki poprzez zmianę usytuowania instrukcji skoków, w celu zmniejszenia liczby skoków realizowanych w czasie wykonania programu,
- (E) Eliminacja tożsamości algebraicznych.

**Zad. 81.** Dla optymalizacji kodu pośredniego metodą analizy przepływu **nieprawdziwe** jest stwierdzenie:

- (A) Blok podstawowy jest ciągiem instrukcji trójadresowych, takich że jeśli sterowanie zostanie przekazane do pierwszej instrukcji tego bloku, to opuści blok po wykonaniu ostatniej instrukcji w bloku.
- (B) Wierzchołkami grafu obliczeń są pojedyncze instrukcje kodu trójadresowego różne od instrukcji skoku, wywołania podprogramu lub powrotu z podprogramu.
- (C) Każda krawędź w grafie obliczeń jest odzwierciedleniem możliwości przekazania sterowania między wierzchołkami grafu obliczeń zgodnie ze skierowaniem krawędzi.
- (D) Graf obliczeń jest to skierowany graf tworzony na podstawie trójadresowego kodu pośredniego.
- (E) Główną rolę w tworzeniu struktury grafu obliczeń poprzez przekształcanie strumienia instrukcji trójadresowych tworzących kod pośredni pełnią instrukcje skoków, wywołania podprogramu lub powrotu z podprogramu.

**Zad. 82.** Która spośród poniższych transformacji optymalizujących kod pośredni **nie może** być wykonana z wykorzystaniem metody analizy przepływu?

- (A) Reorganizacja kodu zawierającego skoki poprzez zmianę usytuowania instrukcji skoków, w celu zmniejszenia liczby skoków realizowanych w czasie wykonania programu,
- (B) Eliminacja wspólnych podwyrażeń,
- (C) Eliminacja propagacji kopiowania,
- (D) Eliminacja martwego kodu,
- (E) Przemieszczanie kodu niezmienniczego poza ciało pętli.

**Zad. 83.** Dla optymalizacji kodu pośredniego metodą analizy przepływu **nieprawdziwe** jest stwierdzenie:

- (A) Blok podstawowy jest ciągiem instrukcji trójadresowych, takich że jeśli sterowanie zostanie przekazane do pierwszej instrukcji tego bloku, to opuści blok po wykonaniu ostatniej instrukcji w bloku.
- (B) Wierzchołkami grafu obliczeń są bloki podstawowe.
- (C) Każda krawędź w grafie obliczeń jest odzwierciedleniem możliwości przekazania sterowania między wierzchołkami grafu obliczeń zgodnie ze skierowaniem krawędzi.
- (D) Graf obliczeń jest to skierowany graf tworzony na podstawie trójadresowego kodu pośredniego.
- (E) Główną rolę w tworzeniu struktury grafu obliczeń poprzez przekształcanie strumienia instrukcji trójadresowych tworzących kod pośredni pełnią instrukcje przypisania oraz instrukcje operowania na elementach jednowymiarowych tablic.

**Zad. 84.** Która spośród poniższych transformacji optymalizujących kod pośredni **nie może** być wykonana z wykorzystaniem metody analizy przepływu?

- (A) Eliminacja wspólnych podwyrażeń,
- (B) Eliminacja propagacji kopiowania,
- (C) Eliminacja martwego kodu,
- (D) Optymalizacja przebiegu sterowania poprzez zmniejszenie liczby skoków realizowanych w trakcie wykonania programu (wykrywanie i modyfikacja sytuacji typu: „skok do skoku”),
- (E) Redukcja mocy kodu dzięki analizie zmiennych indukowanych dla pętli,

**Zad. 85.** Dla optymalizacji kodu pośredniego metodą analizy przepływu **nieprawdziwe** jest stwierdzenie:

- (A) Blok podstawowy jest ciągiem instrukcji trójadresowych, takich że jeśli sterowanie zostanie przekazane do pierwszej instrukcji tego bloku, to opuści blok po wykonaniu ostatniej instrukcji w bloku.
- (B) Wierzchołkami grafu obliczeń są bloki podstawowe.
- (C) Graf obliczeń jest to skierowany graf acykliczny tworzony na podstawie trójadresowego kodu pośredniego.
- (D) Każda krawędź w grafie obliczeń jest odzwierciedleniem możliwości przekazania sterowania między wierzchołkami grafu obliczeń zgodnie ze skierowaniem krawędzi.

- (E) Główną rolę w tworzeniu struktury grafu obliczeń poprzez przekształcanie strumienia instrukcji trójadresowych tworzących kod pośredni pełnią instrukcje skoków, wywołania podprogramu lub powrotu z podprogramu.

**Zad. 86.** Która spośród poniższych transformacji optymalizujących kod pośredni **nie może** być wykonana z wykorzystaniem metody analizy przepływu?

- (A) Eliminacja wspólnych podwyrażeń,
- (B) Eliminacja zbędnych skoków,
- (C) Eliminacja propagacji kopiowania,
- (D) Eliminacja martwego kodu,
- (E) Eliminacja zmiennych indukowanych wewnątrz ciała pętli.