

2. WYRAŻENIA

2.1. Rodzaje wyrażeń

W ALGOLu używamy trzech rodzajów wyrażeń: arytmetycznych, booleowskich i desygnujących.

Wyrażenie arytmetyczne (arithmetic expression; czytają: aritmetik ekspreszn) służy do obliczania wartości liczbowej.

Wyrażenie booleowskie (Boolean expression; czytają: bułan ekspreszn) służy do obliczania wartości logicznej.

Wyrażenie desygnujące* (designational expression; czytają: designejszyni ekspreszn) służy do wyznaczenia miejsca, do którego należy skoczyć w programie. Wyrażenia te będą omówione w paragrafie 3.6.

Ponadto – ale tylko w GIER-ALGOLu – używamy jeszcze wyrażenia łańcuchowego (string expression; czytają: stryng ekspreszn). Służą one do wprowadzania danych i wyprowadzania wyników. Będą omówione w paragrafie 5.5.

2.2. Warunek "jeśli"

W wyrażeniach algolowskich zarówno arytmetycznych, booleowskich i desygnujących, jak też łańcuchowych, dopuszcza się alternatywy, tzn. rozdawanie wyrażenia na dwa przypadki. Wybór jednego z tych przypadków następuje w zależności od tego, czy jest czy nie jest spełniony tzw. warunek "jeśli". Warunek "jeśli" (if clause; czytają: yf kloż) ma następującą budowę

if

wyrażenie booleowskie

then

(if znaczy "jeśli", then znaczy "to"; czytają: yf, den, gdzie δ jest głoską pośrednią między d i dz).

Warunek "jeśli" sam przez się nie tworzy żadnej całości, po wyrażeniu then musi coś nastąpić. Stanowi on jednak element składniowy dla wyrażeń.

Warunek "jeśli" uważamy za spełniony, jeżeli zawarte w nim wyrażenie booleowskie ma wartość true. W przeciwnym razie, gdy ma ono wartość false, uważamy warunek za niespełniony.

*S. Paszkowski a za nim J. Madey w swoich książkach używają tu określenia "wyrażenie mianujące", które nie wydaje mi się trafne.

Przykład warunku "jeśli":

if $x \geq a+b-2$ then ...

co po polsku znaczyłoby:

jeśli $x \geq a+b-2$, to ...

2.3. Wyrażenia proste

W y r a ż e n i e nazywamy p r o s t y m, jeśli:

nie zawiera warunku "jeśli"

albo

jest objęte nawiasami okrągłymi ()

albo

każde wyrażenie w nim zawarte a nie będące prostym

jest objęte nawiasami okrągłymi ().

Będziemy zatem mieli:

proste wyrażenie arytmetyczne (simple arithmetic expression; czytaj: sympl aritmetik ekspreszn),
proste wyrażenie booleowskie (simple Boolean; czytaj: sympl bułan),
proste wyrażenie desygnujące (simple designational expression; czytaj: sympl designejszynl ekspreszn),
proste wyrażenie łańcuchowe (general string; dzeneral stryng).

Wyrażenie nie będące prostym będziemy nazywać a l t e r n a t y w n y m.

2.4. Wyrażenia alternatywne

Wyrażenie alternatywne w ALGOLu musi mieć następującą budowę:

warunek "jeśli"
wyrażenie proste

else

wyrażenie (tego samego rodzaju co powyższe wyrażenie proste)

Wyraz else (czytaj: els) znaczy "w przeciwnym razie".

Znaczenie tak zbudowanego wyrażenia jest następujące:

- jeśli warunek "jeśli" jest spełniony, tzn. zawarte w nim wyrażenie booleowskie ma wartość true, to całe wyrażenie

alternatywne staje się równoznaczne z wyrażeniem prostym, wymienionym po warunku "jeśli",

- jeżeli warunek "jeśli" nie jest spełniony, tzn. zawarte w nim wyrażenie booleowskie ma wartość false, to całe wyrażenie alternatywne staje się równoznaczne z wyrażeniem wymienionym po else (wyrażenie to może być z kolei też wyrażeniem alternatywnym).

To, że po warunku "jeśli" musi być wyrażenie proste, nie jest istotnym ograniczeniem, ponieważ - jak to wynika z poprzedniego paragrafu - każde wyrażenie przez wzięcie go w nawiasy okrągłe staje się prostym.

To, że wyrażenie po else musi być tego samego rodzaju co wyrażenie proste wymienione po warunku "jeśli", oznacza, że oba muszą być arytmetyczne, albo oba booleowskie, albo oba desygujące, albo oba łańcuchowe. Jest to zrozumiałe ze względu na alternatywną budowę całości.

Z powyższych określeń i ograniczeń wynika, że: 1^o wyrażenie może nie być proste tylko wtedy, gdy jako całość ma budowę wymaganą od wyrażenia alternatywnego, 2^o wyrażenie alternatywne może być częścią innego wyrażenia tylko wtedy, gdy następuje po else albo przez wzięcie w nawiasy okrągłe stało się prostym.

Przykład wyrażenia alternatywnego (arytmetycznego):

if $x > 0$ then $a + 3$ else $a - 1$

Jeśli np. $a = 5$, to dla każdego $x > 0$ wyrażenie powyższe ma wartość 8, a dla każdego $x \leq 0$ wartość 4.

Należy dobrze utrwalić w pamięci, że wyrażenie po warunku "jeśli" musi być proste. Niezachowanie tej reguły jest istotnym błędem w programie.

2.5. Wyrażenia arytmetyczne

Wyrażenie arytmetyczne może być albo prostym wyrażeniem arytmetycznym albo alternatywnym wyrażeniem arytmetycznym. Po wyjaśnieniach podanych w poprzednim paragrafie wystarczy opisać budowę prostego wyrażenia arytmetycznego. Jest ona analogiczna do budowy wyrażen w tradycyjnym języku matematycznym z pewnymi różnicami wynikającymi z ogólnych zasad autokodów. (Patrz paragraf 0.1).

Jako całość proste wyrażenie arytmetyczne jest sumą iloczynów. Dla zapisu tej sumy używamy operatorów arytmetycznych $+$ albo $-$, jak w zapisie tradycyjnym. Proste wyrażenie arytmetyczne może redukować się do jednego tylko iloczynu.

Każdy iloczyn jest utworzony z czynników połączonych operatorami arytmetycznymi \times albo $/$ albo $:$. Operator \times jest operatorem mnożenia, operator $/$ jest operatorem dzielenia, a operator $:$ dzielenia algolowskiego, które będzie określone później w niniejszym paragrafie. Każdy z iloczynów może również redukować się do jednego tylko czynnika.

Każdy czynnik jest albo tzw. elementem podstawowym, albo potęgą o postaci

element podstawowy \wedge element podstawowy \wedge ... \wedge element podstawowy

gdzie \wedge jest operatorem arytmetycznym potęgowania, które wykonujemy kolejno od lewej strony.

Wreszcie elementem podstawowym może być liczba, zmienna, funkcja (por. paragraf 1.15), albo dowolne wyrażenie arytmetyczne ujęte w nawiasy okrągłe.

Jak wynika z powyższego, operatory arytmetyczne łączą tylko proste wyrażenia arytmetyczne, a nigdy alternatywne, co nie jest istotnym ograniczeniem, ponieważ - jak już wiemy - wyrażenie alternatywne przez wzięcie go w nawiasy okrągłe staje się prostym. A oto przykłady elementów podstawowych:

- 1) $+1.63_{\text{p}}-4$
- 2) alfa
- 3) $A[k-2,1]$
- 4) $\sin(x-2xy)$
- 5) $(x1\wedge 2/x2-\text{sqrt}(z))$

W przykładzie 1 elementem podstawowym jest liczba.

W przykładzie 2 elementem podstawowym jest zmienna oznaczona wyrazem alfa.

W przykładzie 3 elementem podstawowym jest zmienna z dwoma indeksami. Pierwszym indeksem jest $k-2$, a drugim 1.

Uwaga: nie mylić z zapisem $k-2.1$ (w przykładzie 3 przecinek oddziela indeksy, a nie oznacza rozdzielenia części całkowitej liczby od jej części ułamkowej, gdyż - jak to już było podkreślane - tylko kropka służy do tego celu).

W przykładzie 4 elementem podstawowym jest funkcja, której argument jest wyrażeniem arytmetycznym.

W przykładzie 5 elementem podstawowym jest całe wyrażenie arytmetyczne na skutek ujęcia go w nawiasy okrągłe.

Przykłady czynników:

- 1) lambda
- 2) $\text{alfa}\wedge\sin(x-2xy)$
- 3) $a\wedge b\wedge c$
- 4) $+1.63_{\text{p}}-4\wedge A[k-2,1]\wedge(x1\wedge 2/x2-\text{sqrt}(z))$
- 5) $a\wedge(b\wedge c)$

W przykładzie 1 czynnikiem jest element podstawowy, mianowicie zmienna o nazwie lambda.

W przykładzie 2 czynnik jest potęgą, której podstawa jest elementem podstawowym (zmienna o nazwie alfa), Wykładnikiem jest element podstawowy omówiony w poprzednim przykładzie 4.

W przykładzie 3 operatory \wedge rozpatrujemy kolejno od lewej strony. Najpierw mamy czynnik o postaci $a\wedge b$,

będący potęgą o podstawie a i wykładniku b . Ten czynnik $a \uparrow b$ jest z kolei podstawą potęgi o wykładniku c . W tradycyjnym zapisie matematycznym czynnik z przykładu 3 napisalibyśmy w postaci

$$(a^b)^c$$

W przykładzie 4 czynnik ma tę samą budowę, co w przykładzie poprzednim, z tym, że ma miejsce a , b , c wstawiono odpowiednio elementy podstawowe 1, 3, 5 z poprzednich przykładów.

W przykładzie 5 czynnik $b \uparrow c$, będący szczególnym przypadkiem wyrażenia arytmetycznego, przez wzięcie w nawiasy okrągłe stał się elementem podstawowym, tworzącym wykładnik potęgi o podstawie a . W tradycyjnym zapisie mielibyśmy tu

$$a^{(b^c)}$$

W iloczynach napotykamy na działanie zwane dzieleniem algorytmicznym. Jest to dzielenie wykonywane tylko na czynnikach całkowitych (integer), dające w wyniku liczbę całkowitą bliską ilorazowi według następującego wzoru

$$a \div b = \text{sign}(a/b) \times \text{int}(\text{abs}(a/b))$$

Na przykład:

$$13 \div 2 = 6$$

$$17 \div (-3) = -5$$

$$3 \div (-4) = -0 = 0$$

Mnożenie dwu czynników wymaga zawsze umieszczenia między nimi operatora mnożenia \times . Opuszczenie tego operatora stanowi jeden z najczęstszych błędów u początkujących programistów, zwłaszcza, gdy mamy do czynienia z iloczynem liczbowego współczynnika przez zmienną, jak np.:

$$2 \times a$$

Należy również pamiętać, że - podobnie jak w tradycyjnym zapisie - nie wolno pisać dwu operatorów arytmetycznych bezpośrednio po sobie, natomiast mogą one być oddzielone nawiasem okrągłym, jak np.:

$$a \times (-2)$$

$$a \uparrow (-3)$$

$$a + (-b)$$

Jak o tym była mowa w paragrafie 1.1. zamiast symbolu \times można używać również $*$, zamiast symbolu \uparrow symbolu \uparrow a zamiast \div symbolu \div .

Operatorem dzielenia jest zawsze $/$, a nigdy dwukropek. Należy pamiętać, że operator ten nie jest równoznaczny z tradycyjną kreską ułamkową, gdyż powodowałoby to wieloznaczności. Aby ich uniknąć, stosujemy nawiasy okrągłe. Na przykład tradycyjne wyrażenia

$$\frac{a}{b+c}$$

$$\frac{a}{b+c}$$

$$\frac{a}{b^c}$$

$$\frac{a}{bc}$$

zapisujemy w ALGOLu odpowiednio w sposób następujący:

$a/(b+c)$ $a/b+c$ $a/b*c$ $a/(b*c)$ albo $a/b/c$.

W ALGOLu używamy nawiasów kwadratowych tylko do indeksów dla zmiennych. W wyrażeniach stosujemy ponadto tylko nawiasy okrągłe. Natomiast wolno użyć nawiasów nadmiernych, jeśli czynimy to zgodnie z regułami. Na przykład wolno nam zamiast $a \uparrow b \uparrow c$ napisać $(a \uparrow b) \uparrow c$, lub zamiast $a \times b+c$ napisać $(a \times b) + c$, czy nawet $((a \times b)) + c$. Zauważmy, że w ALGOLu nawiasy okrągłe będą zawsze tej samej wielkości, gdyż dysponujemy tylko jedną parą symboli $()$.

W ALGOLu obowiązuje ta sama kolejność wykonywania działań, co w tradycyjnym języku matematycznym, a mianowicie operator \uparrow ma pierwszeństwo przed pozostałymi, operatory \times / $:$ są pomiędzy sobą jednakowo ważne, ale mają pierwszeństwo przed operatorami $+$ i $-$, które znów pomiędzy sobą są jednakowo ważne. Działania jednakowo ważne wykonujemy kolejno od lewej strony do prawej. Na przykład w wyrażeniu

$$a/b+c \times d \uparrow f-g$$

najpierw będzie wykonane potęgowanie $d \uparrow f$, potem dzielenie a/b , potem mnożenie $c \times (d \uparrow f)$, następnie dodawanie $(a/b) + (c \times d \uparrow f)$ i na koniec odjęcie g .

W przypadku konieczności zmiany kolejności działań stosujemy nawiasy okrągłe na zasadach analogicznych do istniejących w tradycyjnym języku matematycznym, z tym że nawiasy okrągłe mogą zawierać się w nawiasach okrągłych.

Działania algolowskie komplikuje w porównaniu do tradycyjnych fakt podziału liczb, zmiennych, wartości funkcji i wartości wszelkich wyrażeń arytmetycznych na typy integer i real. Obowiązują tu następujące zasady:

- w działaniach mogą występować zarówno wartości integer jak real, z wyjątkiem dzielenia algolowskiego dopuszczalnego tylko dla wartości integer,

- jeśli co najmniej jedna z wartości w prostym wyrażeniu arytmetycznym jest real, to wartość tego wyrażenia jest real,

- wynik dzielenia z operatorem / jest zawsze real,

- wynik dzielenia algolowskiego jest zawsze integer,

- potęga liczby całkowitej integer z wykładnikiem również integer dodatnim ma wartość integer, w pozostałych przypadkach potęga ma wartość real (o ile nie jest nieokreślona, jak np. potęga o wykładniku real z liczby ujemnej, co - jak już wiemy z paragrafu 1.15 - będzie powodowało sygnał alarmowy).

Ponadto działania algolowskie komplikuje fakt wykonywania ich przez maszynę z ograniczoną dokładnością. I tak np. na skutek zaokrągleń wyników możemy nieraz zamiast wartości dodatniej bardzo bliskiej zera lub równej zera otrzymać dla danego wyrażenia arytmetycznego wartość ujemną, która - aczkolwiek bardzo bliska zera - uniemożliwi wyciągnięcie z niej pierwiastka kwadratowego czy logarytmu. Do sprawy tej powrócimy później.

A oto przykłady wyrażeń arytmetycznych:

- 1) $x2-\text{sqrt}(x2 \uparrow 2+1)$
- 2) $\text{Amplitudaxsin}(\text{omegax}t+fi)$
- 3) $A[a+b \times c]+B(a+bc) \uparrow 2$

- 4) $\text{Matr}[1+\text{Matr}[1,1],1]+\arctan(2x)\sqrt{1+\ln(x)}$
- 5) $(\text{if } x < 0 \text{ then } a+1 \text{ else } a+2)+2x(\text{if } x < 2 \text{ then } a \text{ else } a-3)-3$
- 6) $\text{if } mx(a+b) \leq c \text{ then } A[1,3,7] \text{ else } A[1,7,3]$
- 7) $\text{if } A \text{ then } B \text{ else } C+D$
- 8) $\text{if } c < 0 \text{ then } -1 \text{ else if } c=0 \text{ then } 0 \text{ else } 1$
- 9) $\text{if } a < 0 \text{ then } -x-1/x+1/x^2 \text{ else if } b < 0 \text{ then } \text{abs}(x)+1$
 $\text{else } \sqrt{x^2-1}$
- 10) $\text{if } a < b \text{ then } \arctan(a/b) \text{ else if } a=0 \text{ then } 0 \text{ else}$
 $\text{if } b=0 \text{ then } \pi/2 \text{ else } \pi/2-\arctan(b/a)$

W pierwszych pięciu przykładach mamy do czynienia z prostymi wyrażeniami arytmetycznymi (w przykładzie 5 występują co prawda wyrażenia alternatywne, ale przez wzięcie w nawiasy okrągłe stały się one elementami podstawowymi). W następnych pięciu mamy do czynienia z alternatywnymi wyrażeniami arytmetycznymi.

W przykładzie 1 mamy wyrażenie arytmetyczne, które w tradycyjnym języku miałooby postać

$$x_2 - \sqrt{x_2^2 + 1}$$

Wyrażenie z przykładu 2 miałooby w języku tradycyjnym postać

$$A \sin(\omega t + \varphi),$$

gdzie A oznaczałoby amplitudę.

Wyrażenie z przykładu 3 jest sumą dwu składników: zmiennej A o jednym indeksie, określonym przez wyrażenie $a+b \times c$, oraz kwadratu wartości jakiejś funkcji B w punkcie $a+bc$, gdzie bc jest nazwą zmiennej, a nie iloczynem zmiennych b i c , ponieważ między b i c nie ma operatora mnożenia. To że B jest nazwą funkcji, a nie zmiennej indeksowanej, rozpoznajemy po nawiasach okrągłych, a to że $B(a+bc)$ nie oznacza iloczynu zmiennej B przez wyrażenie $a+bc$ po braku operatora mnożenia między nimi. W tradycyjnym języku matematycznym wyrażenie z przykładu 3 mogłoby mieć postać

$$A_{a+bc} + [B(a+bc)]^2$$

Funkcja B jest jakąś procedurą funkcyjną, ponieważ nie ma jej na liście procedur standardowych.

Wyrażenie z przykładu 4 jest również sumą dwu składników. Pierwszym składnikiem jest zmienna indeksowana o nazwie Matr i dwu indeksach, z których pierwszy jest sumą jedynki i zmiennej indeksowanej $\text{Matr}[1,1]$ a drugi jedynką. To, że dla określenia indeksu zmiennej o nazwie Matr używamy zmiennej z tej samej tablicy o nazwie Matr , nie stanowi żadnej przeszkody, nawet gdyby obie zmienne: określona i określająca były identyczne. Gdyby na przykład zmienna Matr

[1,1] miała wartość 0, to zmienna, której indeks określamy, miałyby postać

$$\text{Matr}[1+0,1],$$

czyli Matr [1,1] i jako identyczna z poprzednią miałyby wartość 0. Drugi składnik jest potęgą o podstawie $\arctan(2x)$ i wykładniku $1+\ln(x)$. W tradycyjnym języku matematycznym wyrażenie z przykładu 4 mogłoby mieć postać

$$M_{1+M_{1,1},1} + (\arctg 2x)^{1+\ln x}$$

gdybyśmy nazwę Matr skrócili do jednej litery M.

W celu rozszyfrowania przykładu 5 rozpatrzmy trzy przypadki:

$$\text{I. } x < 0, \quad \text{II. } 0 \leq x < 2, \quad \text{III. } x \geq 2.$$

W przypadku I warunek "jeśli" w obu alternatywnych wyrażeniach arytmetycznych, ujętych w nawiasy okrągłe, ma wartość true. Wobec tego pierwsze z tych wyrażeń jest wtedy równoznaczne z $a+1$, a drugie z a , skąd całość

$$a+1 + 2xa - 3$$

czyli $3xa - 2$. W przypadku II warunek "jeśli" w pierwszym z dwu wyrażeń alternatywnych ma wartość false, a w drugim true. Wobec tego pierwsze z tych wyrażeń jest wtedy równoznaczne z $a+2$, a drugie z a , natomiast całość

$$a+2 + 2xa - 3$$

czyli $3xa - 1$. W przypadku III warunek "jeśli" w obu wyrażeniach alternatywnych ma wartość false i wobec tego pierwsze z tych wyrażeń jest równoznaczne z $a+2$, a drugie z $a-3$, skąd całość

$$a+2 + 2x(a-3) - 3$$

czyli

$$3xa - 7.$$

Jak widać z powyższego, wyrażenie z przykładu 5 można by równie dobrze napisać w postaci

$$3xa - (\text{if } x < 0 \text{ then } 2 \text{ else if } x < 2 \text{ then } 1 \text{ else } 7)$$

W tradycyjnym języku matematycznym można by napisać wyrażenie z przykładu 5 następująco:

$$\text{dla } x < 0: \quad 3a-2,$$

$$\text{dla } 0 \leq x < 2: \quad 3a-1,$$

$$\text{dla } x \geq 2: \quad 3a-7.$$

W przykładzie 6 mamy do czynienia z alternatywnym wyrażeniem arytmetycznym. Jeśli nierówność

$$m \times (a+b) \leq c$$

jest spełniona, tzn. wyrażenie booleowskie przez tę nierówność utworzone ma wartość true, całość redukuje się do zmiennej

indeksowanej $A[1,3,7]$ o trzech indeksach. Jeśli wspomniana nierówność nie jest spełniona, czyli utworzone przez nią wyrażenie boolowskie ma wartość false, całość jest równoznaczna ze zmienną indeksowaną $A[1,7,3]$. Należy jeszcze raz podkreślić, że to, czy nierówność $m \times (a+b) \leq c$ jest spełniona, czy nie, zależy od aktualnych wartości zmiennych liczbowych m, a, b i c . Wobec tego w różnych momentach wykonywania programu jeden i ten sam warunek "jeśli" może mieć czasem wartość true, a czasem false. W omawianym przykładzie warunek $m \times (a+b) \leq c$ w momencie, gdy na przykład $m=1, a=b=2, c=5$, ma wartość true, a w momencie, gdy na przykład $m=2, a=3, b=4, c=0$, wartość false.

W przykładzie 7 mamy do czynienia znowu z alternatywnym wyrażeniem arytmetycznym. Z budowy tego wyrażenia wynika, że A musi być zmienną boolowską, a B, C i D zmiennymi liczbowymi. Pewne wątpliwości może budzić operator dodawania, a mianowicie, czy łączy on zmienne C i D w sensie wyrażenia

if A then B else (C+D)

czy też całość ma budowę

(if A then B else C)+D

Otóż ze względu na przyjętą umowę, że operatory arytmetyczne łączą tylko proste wyrażenia arytmetyczne, jest możliwa tylko pierwsza z powyższych dwu interpretacji.

Wyrażenie w przykładzie 8 jest - jak łatwo zauważyć - równoznaczne z $\text{sign}(c)$. Wystarczy w tym celu porównać to wyrażenie z określeniem funkcji $\text{sign}(W)$ w paragrafie 1.15.

W przykładzie 9 mamy do czynienia z wyrażeniem, którego budowę można następująco podkreślić używając nawiasów okrągłych:

if a < 0 then (-x-1/x+1/x²) else (if b < 0 then abs(x)+1
else sqrt(x²-1))

W tradycyjnym języku matematycznym można to zapisać następująco:

| | |
|-------------------------------|--------------------------------------|
| Gdy $a < 0$: | $-x - \frac{1}{x} + \frac{1}{x^2}$, |
| gdy $a \geq 0$ i $b < 0$: | $ x + 1$, |
| gdy $a \geq 0$ i $b \geq 0$: | $\sqrt{x^2 - 1}$, |

W przykładzie 10 mamy do czynienia z wyrażeniem, którego budowę można uwypuklić następująco:

if a < b then arctan(a/b) else
if a=0 then 0 else
if b=0 then pi/2 else
 $pi/2 - \arctan(b/a)$

W tradycyjnym języku matematycznym można to zapisać następująco:

$$\begin{aligned} \text{Dla } a < b : & \quad \operatorname{arctg} \frac{a}{b} \\ \text{Dla } a \geq b \text{ i } a = 0 : & \quad 0 \\ \text{Dla } a \geq b, \quad a \neq 0 \text{ i } b = 0 : & \quad \frac{\pi}{2} \\ \text{Dla } a \geq b, \quad a \neq 0 \text{ i } b \neq 0 : & \quad \frac{\pi}{2} - \operatorname{arctg} \frac{b}{a} \end{aligned}$$

2.6. Zadanie

Następujące wyrażenia algolowskie napisać w tradycyjnym języku matematycznym:

- 1) $(2x_a - 3x_b) / ((5x_a + b)^{-5})$
- 2) $\sqrt{z^2 + z + 1}^3 - \sqrt{z^2 - z + 1}^3 + 2 \exp(kxz)$
- 3) $a^{\sqrt{(3x_k - 2x_n - 1)}} / (b^{\sqrt{(2x_k + 3)}} + (c - d)^{\sqrt{(k - n)}} / (4x_b - c)^{\sqrt{4}})$
- 4) $(a[1,1]x_{a[2,2]} - a[1,2]x_{a[2,1]}) / (\sqrt{a[1,1]}^2 + a[2,2]^2)$
- 5) $\sin(2x_{\alpha} - 3x_{\beta}) \times \cos(\sqrt{\operatorname{abs}(x)}) / \sin(\sqrt{1 - \pi}/4)^3 + 1$
- 6) $(2x_a + 3x_b + 4x_c)^{\sqrt{(3x_a + 4x_b + 5x_c)}} (a - b) + (a + b)^{\sqrt{c}} / (a + 1)$
- 7) $(((((x - 2xy) - (z + 3xy) \times (x - z)) / ((2xx + 1)xy - 1) + 1/y) / (x + y + z)^2 + 5) \times x / (1 + 1 / (1 + 1 / (1 + 1/x))))$
- 8) $(x_a - (x_b + x_c / x_a - 1 / ((x_b + x_c)^{\sqrt{(-k+1)} - 1}))) / (((x_c - x_b) / x_a + 1) - x_c)$
- 9) $\ln(\operatorname{abs}(\cos(a[\alpha]) / \cos(a[\beta]))) \times \sin(\lambda x (\pi/2) - \gamma)$
- 10) if $x^2 \leq 1$ then $2xx^3 - x^2 + 4xx - 5$ else $(z^5 - 5xx^3 + x) \exp(x)$
- 11) if $x \leq y$ then $\sqrt{y^2 - x^2}$ else if $x > 0$ then 0 else $x - 2xy$
- 12) if $x_1 \leq x_2$ then 1 else $-1 + (\text{if } x_2 < 5xx_1 \text{ then } 2 \text{ else } 5) \times (x_1 - x_2)$

2.7. Zadanie

Które z podanych niżej wyrażen są algolowskimi, a które nie:

- 1) $ABC \setminus CBA \setminus BAC$
- 2) $1/2/3/4/5$
- 3) $x/y(z[t])+1$
- 4) $-(-(-(-x)))$
- 5) $(a+b+c) \times_{10} 2/1,2$
- 6) $\alpha \setminus -3 + \beta$
- 7) $A[x+y] - 2 \times A[1,2]$
- 8) $ALFA[((x)),1,(((x)))]$
- 9) 1
- 10) $\text{Gamma} / -2 \times X1$
- 11) $ZZZ \times \sin[\alpha + \beta]$
- 12) ${}_6 6/2,3_{10} - 3 \times \alpha \setminus 2.2$
- 13) $(a+b)/(a-2b)$
- 14) $A[1,3,5] - A[2+x, A(0,2,4), 5-x]$
- 15) $(a+2 \times b + 3 \times c) / [(x-y) + 2 \times (x+y \setminus 2)] + 1$
- 16) if A then B else A+1
- 17) if x>0 then x \setminus 2+1 else x=0
- 18) if X \setminus 0 then X+sqrt(X) else X \times X
- 19) if a<b then a+b+c else a-b+c
- 20) if delta>0 then (if delta=0 then x1 else x2)
else if x1<x2 then x2+x1 else x1-x2+3
- 21) if x>0 then if y=0 then A else B else if z>0 then C else D
- 22) if a+b+c/2 then a+b else b+c
- 23) $\sin(\sqrt{x}) + (\text{if } x > 5 \text{ then } 2 \text{ else } 7)$
- 24) $(\text{if } Y23 > Y11 \text{ then } xu-1 \text{ else } 3) \times (4 + \text{if } Y9 = 0 \text{ then } 6 \text{ else } xu)$

2.8. Zadanie

Następujące wyrażenia napisać w języku ALGOL:

$$1) (2x^2 - 3x + 5) \left(y - \frac{x^3}{x^2 - y^2} + 1 \right) [1 + 2x(\sqrt{x} - \sqrt[4]{y})]^2$$

$$2) (a+b)^{1-j} + \left(\frac{\sqrt{a}+1}{\sqrt{a+1}} \right)^5$$

$$3) \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{x}}}}}}$$

$$4) (a^{b+c})^{d+e(f+g)}$$

$$5) \frac{x_1 - \frac{x_2}{x_3(x_4 - x_1^2 - x_2^3)}}{\frac{x_1}{x_2} + 1 - \frac{x_3(x_1 - \sqrt{x_4})}{x_3 - x_2}}$$

$$6) \sqrt{A^2 + B^2} \cdot \sin(\alpha + \beta) \cdot \operatorname{tg} \varphi \cdot e^{\alpha \varphi}$$

$$7) [x(x+1)+2]x-3\sqrt[5]{\ln|1-x^2|+3}$$

$$8) \frac{a_{12}b_{22}-a_{21}b_{11}}{a_{22}b_{12}-a_{11}b_{21}} [(a_{33}^2 + a_{44}^2)^2 + 1]^{\frac{1}{2}}$$

$$9) \text{ dla } X < Y : \quad X/Y$$

$$\text{ dla } X = Y : \quad 1$$

$$\text{ dla } X > Y : \quad X^3 - Y^3$$

$$10) \text{ Gdy } a^2 = b^2, \text{ to } \sqrt[3]{(x^2+1)^2}, \text{ w przeciwnym razie } \frac{1}{x^2}$$

- 11) $\frac{1}{x-y}$, jeżeli $x \neq y$; natomiast $\frac{1}{x}$ w przypadku, gdy $x=y \neq 0$.

Gdy $x=y=0$ ma być 0.

- 12) Jeżeli $XY \leq 0$, to X .

Jeżeli $XY > 0$ i $Z > 0$, to $X+Z$.

Jeżeli $XY > 0$, $Z \leq 0$ i $\alpha < \frac{\pi}{2}$, to $X \sin \alpha$.

Jeżeli $XY > 0$, $Z \leq 0$, $\alpha \geq \frac{\pi}{2}$ i $\delta < \varepsilon$, to $\varepsilon^2 e^{-\alpha X}$.

Jeżeli $XY > 0$, $Z \leq 0$, $\alpha \geq \frac{\pi}{2}$, $\delta \geq \varepsilon$ i $Z \leq 10^{-8}$, to $1 + \varepsilon XY$.

Jeżeli $XY > 0$, $Z \leq 0$, $\alpha \geq \frac{\pi}{2}$, $\delta \geq \varepsilon$ i $Z \geq 10^{-8}$, to $X + YZ + \alpha \delta \varepsilon$.

2.9. Zadanie

Zakładając, że $a=1$, $B1c=-2$, $\text{gamma}=4$, $XX=-5$, $SM[1]=3$, $SM[2]=0$, $SM[3]=-6$, obliczyć wartości następujących wyrażeń:

- $SM[\text{gamma}-2xa]+B1c \times SM[SM[-B1c]+a]]$
- $(XX \wedge (\text{gamma}+B1c)-a)/SM[3xa]+\text{sqrt}(\text{gamma})$
- $\text{abs}(B1c) \wedge (a+1) \wedge (-XX-\text{gamma}) \wedge (3xa-1) \wedge (1/\text{gamma}) \wedge SM[-a-B1c]$
- $2 \times \text{gamma}; SM[1]$
- $((a-B1c) \times SM[3]+a); XX$
- $((3 \times \text{gamma}+1)/(1-B1c)-(3 \times \text{gamma}+1); (1-B1c)) \times \text{abs}(SM[3])+5$
- $((\cos((2xa+B1c) \times (\text{gamma}+XX \wedge 3)+SM[7xa+XX]))+B1c) \wedge \text{gamma}-XX)/(-B1c)$
- $\text{sqrt}(\text{sqrt}(\text{sqrt}(XX \wedge SM[a]; B1c \times \text{gamma}-(SM[3]+B1c))))$
- $SM[\text{entier}(5 \times SM[1]/\text{gamma})] \wedge SM[\sin(2 \times SM[1]+SM[3])-B1c]$
- $(((((\text{gamma}-3xa)/B1c+SM[1])/XX \times B1c-XX)/SM[3]+a) \wedge 2-B1c) \wedge (a-1)-1$