WPI - rozwiązania zadań

18 listopada 2019

1 Gramatyki bezkontekstowe

Zadanie 1. Podaj gramatykę języka $L = \{w \in \{a, b\}^* : \#(a, w) = \#(b, w)\}$.
Rozwiązanie zadania 1.

```
S ::= <eps> | aSbS | bSaS
```

Zadanie 2. Podaj gramatykę języka $L = \{w \in \{a, b\}^* : \#(a, w) \neq \#(b, w)\}$.
Rozwiązanie zadania 2.

```
S ::= A | B
A ::= RaA | RaR
B ::= RbB | RbR
R ::= <eps> | aRbR | bRaR
```

Zadanie 3. Podaj gramatykę języka $L = \{w \in \{a, b\}^* : \#(a, w) = 2\#(b, w)\}$.

Rozwiązanie zadania 3.

```
S ::= <eps> | SbSaSaS | SaSbSaS | SaSaSbS
```

Zadanie 4. Podaj gramatykę języka L poprawnych wyrażeń nawiasowych `()`.

Rozwiązanie zadania 4.

```
S ::= (S)S | <eps>
```

lub niejednoznaczana wersja:

```
S ::= (S) | SS | <eps>
```

Zadanie 5. Podaj gramatykę języka L poprawnych wyrażeń nawiasowych `()[]`, przy czym nawiasy kwadratowe nie mogą być zawarte w okrągłych.
Rozwiązanie zadania 5.

```
S ::= [S]S | (P)S | <eps>
P ::= (P)P | <eps>
```

Zadanie 6. Podaj gramatykę języka $L = \{w = a^n b^k c^m : n + k = m\}$.
Rozwiązanie zadania 6.

```
S ::= aSc | X
X ::= bXc | <eps>
```

Zadanie 7. Podaj gramatykę języka $L = \{w = a^n b^k c^m : n + m = k\}$.
Rozwiązanie zadania 7.

```
S ::= XY
X ::= aXb | <eps>
Y ::= bYc | <eps>
```

Zadanie 8. Podaj gramatykę języka $L = \{w = a^n b^k c^m : n + m = k + 1\}$.
Rozwiązanie zadania 8.

```
S ::= aXY | XYc
X ::= aXb | <eps>
Y ::= bYc | <eps>
```

Zadanie 9. Podaj gramatykę języka $L = \{w = a^n b^k c^m : n + m \neq k\}$.
Rozwiązanie zadania 9.

```
S ::= XBY | AXY | XYC | AXYC
X ::= aXb | <eps>
Y ::= bYc | <eps>
A ::= aA | a
B ::= bB | b
C ::= cC | c
```

Zadanie 10. Podaj gramatykę języka $L = \{w = a^n b^k c^m : n + 2m = k + 2\}$.
Rozwiązanie zadania 10. S ::= aaXY | XYc X ::= aXb | <eps> Y ::= bbYc | <eps>

Zadanie 11. Wyrażenia arytmetyczne z prawostronnie łącznym potęgowaniem, czyli że np. $x * y^{z^v}$ parsuje się jako $x * (y^{(z^v)})$
Rozwiązanie zadania 11.

```
W ::= S | -S | W+S | W-S
S ::= C | S*C | S div C | S mod C
C ::= P | P^C
P ::= L | (W)
L ::= [1-9][0-9]* | 0
```

Zadanie 12. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : \#(a,w) = 2k,\ k \geq 0\}$.

Rozwiązanie zadania 12.

```
S ::= aBaS | bS | <eps>
B ::= bB | <eps>
```

Zadanie 13. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : w \neq \texttt{ababab}\}$.

Rozwiązanie zadania 13.

```
S ::= aA | bX | <eps>
A ::= bB | aX | <eps>
B ::= aC | bX | <eps>
C ::= bD | aX | <eps>
D ::= aE | bX | <eps>
E ::= bF | aX | <eps>
F ::= bG | aX | <eps>
G ::= aX | bX
X ::= aX | bX | <eps>
```

Zadanie 14. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : w \text{ nie zawiera } aa\}$.

Rozwiązanie zadania 14.

```
S ::= aB | B
B ::= bS | <eps>
```

Zadanie 15. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : w \text{ nie zawiera } aaa\}$.

Rozwiązanie zadania 15.

```
S ::= aaB | aB | B
B ::= bS | <eps>
```

Zadanie 16. Podaj gramatykę języka $L = \{w \in \{a,b\}^* : w \neq a^n b^n a^n\}$.

Rozwiązanie zadania 16.

```
S ::= A | AB | APQ | PBQ | PQB | ABABX | BX
P ::= aPb | ab
Q ::= bQa | ba
A ::= aA | a
B ::= bB | b
X ::= aX | bX | <eps>
Y ::= aX | bX
```

Zadanie 17. Przez segment słowa $v \in A^*$ składający się z tych samych liter x nazwiemy takie słowo $w = x^k$ dla pewnego $x \in A$ oraz $k \in N+$, że $v = w'ww''$ dla pewnych $w', w'' \in A^*$. Segment taki jest maksymalny, jeśli ani w' nie kończy się na x, ani w'' nie zaczyna się od x. Napisz gramatykę generującą wszystkie słowa v nad alfabetem $\{a, b, c\}$ o tej właściwości, że każdy maksymalny segment złożony z samych liter b w słowie v musi być bezpośrednio poprzedzony przez dłuższy segment złożony z samych liter a.

Rozwiązanie zadania 17.

```
S ::= <eps> | C S | T S
T ::= A | a T b
A ::= a | aA
C ::= c | cC
```

Zadanie 18. Podaj gramatykę, która wygeneruje słowa nad alfabetem $0, 1$, które mają tyle samo jedynek na pozycjach parzystych co na pozycjach nieparzystych.

Rozwiązanie zadania 18.

```
S ::= X | X 0 | X 01 X 1
X ::= <eps> | N X | A X B X | B X A X
N ::= 00 | 10
A ::= 01
B ::= 10
```

Zadanie 19. Napisz funkcję:
`function Iloczyny6(const T:array of integer; k:integer):Boolean;` która przyjmie wartość `true`, wtedy i tylko wtedy gdy iloczyn liczb zapisanych w każdym segmencie tablicy T długości k jest podzielny przez 6 dla $k > 0$.

Rozwiązanie zadania 19. Dla każdego segmentu długości k należy zliczyć ile liczb jest podzielnych przez 2 (l_2), oraz ile liczb jest podzielnych przez 3 (l_3). Funkcja przyjmuje wartość `true` wtw gdy dla każdego segmentu obie wartości są dodatnie.

```
function Iloczyny6(const T:array of integer; k:integer):Boolean;
var n,i,l2,l3:integer;
begin
  n:=Length(T);
  Iloczyny6 := (n>=k);
  l2 := 0; l3 := 0;
  for i:=0 to n-1 do begin
     if i>=k then if T[i-k] mod 2=0 then dec(l2);
     if i>=k then if T[i-k] mod 3=0 then dec(l3);
     if T[i] mod 2=0 then inc(l2);
     if T[i] mod 3=0 then inc(l3);
     if (i>=k-1) and ((l2=0) or (l3=0)) then Iloczyny6 := false;
  end;
end;

begin
    assert(iloczyny6([2,3,6], 2)=true);
    assert(iloczyny6([2,3,3], 2)=false);
end.
```

Zadanie 20. Dana jest tablica int $A[n]$, gdzie $n \geq 4$, zawierająca zera i jedynki. Oznaczmy przez $a(x, y)$ słowo złożone po kolei z liter $A[x], A[x+1], \ldots, A[y]$ (zakładamy, że $0 \leq x \leq y < n$). Oznaczmy przez $\#(b, w)$ liczbę wystąpień cyfry b w słowie w. Napisz fragment programu, który dla danej liczby int n i tablicy int A[n] wyznaczy takie indeksy p i d, że:

- $0 < p < d < n - 1$ i
- $\#(0, a(0, p-1)) > \#(1, a(0, p-1))$ i
- $\#(0, a(p, d-1)) = \#(1, a(p, d-1))$ i
- $\#(0, a(d, n-1)) < \#(1, a(d, n-1))$.

Program powinien nadać zmiennym p i d dowolne wartości spełniające powyższe warunki, jeśli takie wartości istnieją. W przeciwnym razie zmienne p i d powinny po wykonaniu algorytmu mieć wartość 0.

Rozwiązanie zadania 20. Niech $f(i) = \#(0, a(0, i-1)) - \#(1, a(0, i-1))$ dla $i = 0..n$ (dla $x > y$ definiujemy $a(x, y)$ jako słowo puste).
Niech $S = \max(f(0) = 0, f(n)) + 1$.
Szukamy takich indeksów p i d, że:

- $0 < p < d < n - 1$ (uwaga! tutaj $d < n - 1$ jest dosyć nienaturalne!)
- $f(p) = S$
- $f(d) = S$.

Najłatwiej odszukać pierwszy indeksu p ($1 \leq p$) taki, że $f(p) = S$. Następnie szukamy pierwszego indeksu d ($p < d < n - 1$), takiego, że $f(d) = S$. Jeśli oba indeksy istnieją to są rozwiązaniem zadania, wpp. zwracamy $p = d = 0$.

Zadanie 21. Dana jest liczba całkowita int $n > 0$ oraz tablica int $A[n]$ zawierająca ciąg liczb posortowany rosnąco. Napisz fragment programu, który dla danej liczby int k, nada zmiennej int *wynik* liczbę przedstawień k jako sumy dwóch różnych liczb z tablicy A.

Rozwiązanie zadania 21.

```
int liczbaPrzedstawien(int A[], int n, int k) {
    int l=0,p=n-1,wynik=0;
    while (l<p) {
        while(l<p && A[l]+A[p]>k) p--;
        if (l<p && A[l]+A[p]==k) wynik++;
        l++;
    }
    return wynik;
}
```

Zadanie 22. Tablica `A:tab` zawiera albo liczby dodatnie, albo wartości -1, przy czym wszystkie wartości -1 występują w jednym spójnym segmencie. Ponadto dla dowolnych i, j takich że $1 \leq i \leq j \leq n$ zachodzi $(A[i] \neq -1$ oraz $A[j] \neq -1) \Rightarrow (A[i] <= A[j])$. Napisz funkcję:
`znajdz(const A:tab; x:Integer):1..n`, która znajdzie w tablicy A indeks zadanej liczby dodatniej x, o ile ona się tam znajduje. Jeśli liczby x nie ma w A, to wartością funkcji powinno być -1.

Rozwiązanie zadania 22. Ponieważ wartości -1 występują w spójnym fragmencie to nadal możemy użyć nieznacznie zmodyfikowanego wyszukiwania binarnego. Redukujemy problem do wersji w której wartości -1 występują jedynie w prefiksie (lub sufiksie) tablicy.

```
function znajdz(const A:tab; x:Integer):1..n;
var l,m,p:integer;
begin
  TODO
end;
```

Zadanie 23. Napisz funkcję:
`takiesame(var A:array[1..2*N]of integer):Boolean;` która sprawdzi, czy z zapisanego w tablicy ciągu $2N$ liczb całkowitych można wybrać N-elementowy podciąg tak, by ciągi elementów wybranych i niewybranych były takie same.

Rozwiązanie zadania 23. Niestety nie znam rozwiązania wielomianowego. Algorytmy zachłanne nie działają.

Zadanie 24. Dany jest typ tab=array[1..n]of Integer, przy czym n>0. Dla tablicy A:tab oraz liczby całkowitej k powiemy, że segment $A[d..g]$ jest k-płaski, jeśli dla każdych i, j takich, że $d \leq i, j \leq g$ zachodzi nierówność $A[i] - A[j] \leq k$. Napisz funkcję:
`kaplaski(const A:tab, k:Integer):Integer`,
która wyznaczy długość najdłuższego k-płaskiego segmentu w tablicy A.

Rozwiązanie zadania 24. Zadanie Piloci z 17 OI. `http://oi.edu.pl/static/attachment/20110704/oi17.pdf`

Zadanie 25. Niech $T = \{1, \ldots, n\}$. Funkcja $f : T \to T$ jest niemalejąca, a jej wartości są zapisane w tablicy `F: array[1..n] of Integer` tak, że $f(k) = F[k]$ dla $k = 1, \ldots, n$. Definiujemy funkcję $gf : T \to T$ w następujący sposób: $gf(k) = |a \in T : f(a) = k|$, gdzie przez $|X|$ oznaczamy liczność (moc) zbioru X. Napisz fragment programu, który zmiennej rekord nada wartość równą maksimum spośród wartości przeciwobrazów funkcji gf z jednoelementowych podzbiorów zbioru T, czyli

$$\max(|gf - 1(k)| : k = 1, ..., n).$$

Rozwiązanie zadania 25. TBD

Zadanie 26. Dane są dwie funkcje $f, g : \{0,1\}^* \to Z$ określone w następujący sposób: $f(\epsilon) = 0$, $f(0w) = g(w) + 1$, $f(1w) = g(w) - 1$, $g(\epsilon) = 0$, $g(0w) = f(w) - 1$, $g(1w) = f(w) + 1$. dla $w \in \{0,1\}^*$. Zdefiniujmy język $L = w \in \{0,1\}^* : f(w) = 0$. Napisz funkcję która sprawdzi, czy słowo zapisane w tablicy T należy do języka L.

Rozwiązanie zadania 26. TBD

Zadanie 27. Dana jest następująca funkcja:

```
function coto(n:Integer):Integer;
var i,x,y:Integer;
begin
    x:=0; y:=1;i:=2;
    while i<=n do begin
        x := x+y; y := x-y; i:=i+2
    end;
    if odd(n) then coto := y
    else coto := x
end;
```

Określ dla każdego n, co jest wartością funkcji `coto(n)` i udowodnij to.

Rozwiązanie zadania 27. TBD

Zadanie 28. Dana jest następująca funkcja:

```
function coto(n:Integer):Integer;
var x,y:Integer;
begin
    x:=0; y:=1;
    while y<=n do begin
        x := x-y; y := y-x
    end;
    if odd(n) then coto := y
    else coto := -x
end;
```

Określ dla każdego n, co jest wartością funkcji `coto(n)` i udowodnij to.

Rozwiązanie zadania 28. TBD

Zadanie 29. Dany jest typ tab=array[1..n] of Real, n>0 oraz ciąg prostokątów P1,...,Pn. Tablice A,B:tab reprezentują długości boków tych prostokątów tak, że dla każdego i=1,...,n para (A[i],B[i]) określa długości boków prostokąta Pi. Ciąg tych prostokątów ma tę szczególną własność, że tablica A jest posortowana niemalejąco, a tablica B nierosnąco. Napisz funkcję minprzek(const A,B:tab):Real, która obliczy długość najkrótszej przekątnej prostokątów P1,...,Pn. Uwaga: funkcja pierwiastka kwadratowego ma w Pascalu nazwę sqrt.

Rozwiązanie zadania 29. Wszyskiwanie w tym zadaniu nie jest możliwe (TODO: jakiś dowód). Pozostaje wyszukanie minimalnej przekątnej w czasie liniowym.

Zadanie 30. Dany jest typ tab=array[1..n] of Real, n>0. Napisz funkcję mindist(const A,B:tab):Real, która obliczy wartość min(|A[k]-B[k]| : 1<=k<=n), przy założeniu, że tablica A jest posortowana rosnąco, a B malejąco.

Rozwiązanie zadania 30. TBD

Zadanie 31. W tablicy A typu tab2=array[1..N] of Real jest zapisana wysokość pokrywy śnieżnej na dachu pewnego budynku w dniach 1..N i wiemy, że jest to ciąg słabo bitoniczny (najpierw niemalejący, potem nierosnący). Każdy

centymetr świeżego śniegu (z danego dnia) waży nw, zaś centymetr śniegu z dnia poprzedniego jest c razy cięższy niż w dniu poprzednim. Kolejne warstwy śniegu układają się na poprzednich; jeśli śnieg topnieje to dzieje sie to w kolejności odwrotnej, czyli najpierw topnieje śnieg z wczoraj, potem z przedwczoraj itd. Napisz funkcję
`zaciezko(const A:tab2; c,r,nw:Real):Boolean`,
która sprawdzi, czy był taki dzień, w którym ciężar śniegu na dachu przekroczył wartość krytyczną r.

Rozwiązanie zadania 31. TBD

Zadanie 32. Niech A będzie tablica typu `tab3=array[0..n-1] of Integer`, gdzie n jest parzyste i większe od zera. Napisz funkcję
`k_antybiton(var A:tab3):Boolean`,
sprawdzającą, czy istnieje takie k z przedziału 0..n-2, że tablica B, równa tablicy A przesuniętej cyklicznie o k elementów w prawo, jest ściśle antybitoniczna, czyli od początku do pewnego miejsca malejąca i od tego samego miejsca do końca rosnąca.

Rozwiązanie zadania 32. TBD

Zadanie 33. Rozważamy funkcję $f : \{1, .., n\} \to \{1, .., n\}$. Dla każdego naturalnego k określamy fk(x) = f(x) dla k=0 oraz fk(x)=fk-1(f(x)) dla k>0. Powiemy, że y jest zależne od x jeśli istnieje k > 0 takie, że y=fk(x). O funkcji f wiemy tyle, że wyznaczona przez nią relacja zależności jest antysymetryczna. Dwa elementy x i y są niezależne, jeśli ani x nie jest zależne od y ani y od x. Zbiór X nazwiemy niezależnym, jeśli każde dwa jego elementy są niezależne. Funkcję f definiujemy za pomocą tablicy typu tab4=array[1..n] of Integer. Napisz funkcję maxniezalezne(var f:tab4):Integer, która wyznaczy liczność najliczniejszego niezależnego podzbioru zbioru $\{1, ..., n\}$ wyznaczonego przez funkcję f.

Rozwiązanie zadania 33. TBD

Zadanie 34. (*) Dla danego typu tab01: array[1..n] of 0..1 napisz funkcję maxrot(Const A:tab01):Integer, która wyznaczy takie k, żeby jedynki i zera w tablicy powstałej przez przesunięcie A o k pozycji w prawo ułożyły się w możliwie dużą liczbę w zapisie dwójkowym. Zrobienie tego w czasie liniowym jest możliwe, ale wykracza poza zakres materiału tego semestru

Rozwiązanie zadania 34. Algorytm liniowy: K. S. Booth. Lexicographically least circular substrings. Inf. Process. Lett., 10(4/5):240–242, 1980.

Zadanie 35. Dane są dwa typy:
`cecha=array[0..n-1] of 0..1` oraz `mantysa=array[0..m-1] of 0..1`.
Napisz funkcję:
`wieksza(const c1,c2:cecha; const m1,m2:mantysa):Boolean`
która sprawdzi, czy liczba zapisana w systemie zmiennopozycyjnym o cesze `c1` i mantysie `m1` jest większa niż liczba o cesze `c2` i mantysie `m2`. Zakładamy, że minimalna cecha reprezentuje 0 i że nie ma ukrytego bitu jednej drugiej.

Rozwiązanie zadania 35. TBD

Zadanie 36. Dla typów, jak w zadaniu poprzednim, wygeneruj kolejną co do wielkości liczbę rzeczywistą do danej. Procedure kolejna(var c:cecha;var m:mantysa); Jeśli mieliśmy największą możliwą liczbę w tablicach c:cecha, m:mantysa, to zostaw tablice w spokoju i niczego nie zmieniaj.

```
Procedure kolejna(var c:cecha;var m:mantysa);
```

Rozwiązanie zadania 36. TBD

Zadanie 37. Oblicz błąd względny wykonania działania 2/7+3/11 w systemie zmiennopozycyjnym z wykładu.

Rozwiązanie zadania 37. TBD