

LISTY - OKREŚLENIE

Lista – uporządkowany zbiór dowolnej liczby elementów (termów); elementami listy mogą być inne listy.

Definicja rekurencyjna.

Do konstrukcji listy służy predykat $\bullet/2$

a) pusty zbiór elementów jest listą pustą; oznaczenie: $[]$,

b) jeżeli T jest listą, a H – dowolnym termem, to

$$\bullet(H, T)$$

jest listą, gdzie

– H – głowa listy (ang. head)

– T - ogon listy (ang. tail)

Sposoby przedstawiania list:

1. Lista pusta

$[]$

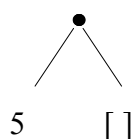
2. Lista jednoelementowa składająca się z elementu:

5

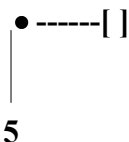
a) struktura (term):

$$\bullet(5, [])$$

b) drzewo:



c) winne grono



d) zapis uproszczony :

$[5]$

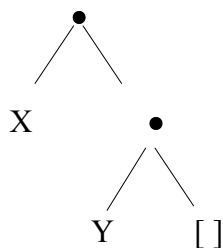
2. Lista dwuelementowa składająca się z elementów:

X, Y

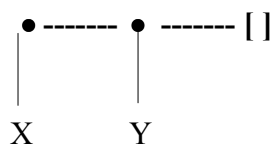
a) struktura:

$\bullet (X, \bullet (Y, []))$

b) drzewo:



c) winne grono



d) zapis uproszczony :

$[X, Y]$

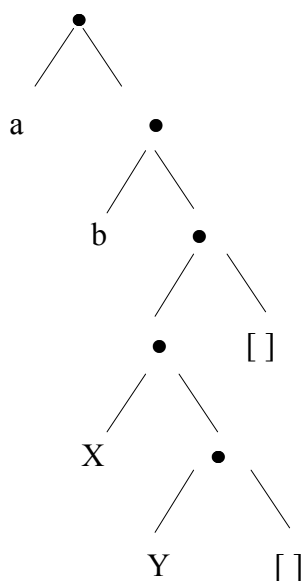
3. Lista trzejelementowa składająca się z elementów:

a, b, [X,Y]

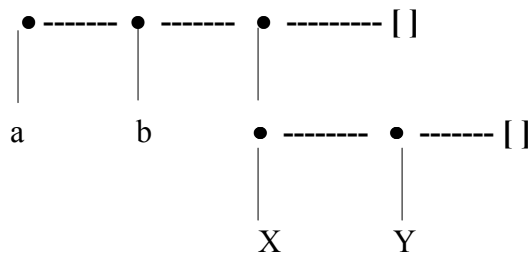
a) struktura:

$\bullet (a, \bullet (b, \bullet (\bullet (X, \bullet (Y, [])), [])))$

b) drzewo:



c) winne grono



d) zapis uproszczony :

[a, b, [X, Y]]

W praktyce stosujemy wyłącznie zapis uproszczony. W związku z tym wprowadza się operację „| „, podziału listy na głowę i ogon.

Zapis

[X | Y]

oznacza listę, której głową jest term X, a ogonem lista Y.

Lista pusta nie posiada głowy ani ogona.

Korzystając z powyższej operacji mamy:

[a, b, c, d] = [a | [b, c, d]] = [a | [b | [c, d]]]

oraz

[a, b | [c, d]] = [a | [b | [c, d]]].

Stąd

[a, b | [c, d]] = [a, b, c, d].

Ogólnie

1) $[s_1, \dots, s_n | [t_1, \dots, t_m]] = [s_1, \dots, s_n, t_1, \dots, t_m], \quad n > 0, m \geq 0$

2) $[s_1, \dots, s_n | [t_1, \dots, t_m | X]] = [s_1, \dots, s_n, t_1, \dots, t_m | X], \quad n, m > 0$

PRZETWARZANIE LIST

Listy są strukturą rekurencyjną, zatem do przetwarzania list stosujemy również procedury (definicje) rekurencyjne.

Ze względu na sposób zakończenia rekurencji w przypadku przetwarzania list, możemy wyróżnić trzy schematy takich definicji:

1. Zakończenie rekurencji, gdy lista jest pusta.

```
%1 warunek kończący rekurencję
reguła([]):-
    pewien_proces([]).

%2 rekurencja
reguła([Głowa|Ogon]):-
    pewien_proces(Głowa),
    reguła(Ogon).
```

2. Zakończenie rekurencji, gdy określony element listy zostanie znaleziony.

```
%1 warunek kończący rekurencję
reguła(Element,[Element|_]):-
    pewien_proces.

%2 rekurencja
reguła(Element,[Głowa|Ogon]):-
    pewien_proces,           %niekoniecznie
    reguła(Element,Ogon).
```

3. Zakończenie rekurencji, gdy określona pozycja na liście zostanie osiągnięta.

%1 warunek kończący rekurencję

```
reguła(1,Element,[Element|_]).
```

%2 rekurencja

```
reguła(Licznik,Element,[_|Ogon]):-
```

```
    Licznik1 is Licznik-1,
```

```
    reguła(Licznik1,Element,Ogon).
```