



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Gramatyki atrybutywne - część 3

Teoria kompilacji

Dr inż. Janusz Majewski
Katedra Informatyki



Gramatyka zupełna i dobrze zdefiniowana

Definicja:

Gramatyka atrybutywna jest zupełna, jeśli dla wszystkich symboli $X \in (V \cup \Sigma)$ są spełnione następujące warunki:

$$\forall p = (X \rightarrow \chi) \in P: AS(X) \subseteq AF(p)$$

$$\forall q = (Y \rightarrow \mu X \nu) \in P: AI(X) \subseteq AF(q)$$

$$AS(X) \cup AI(X) = A(X)$$

Ponadto: $AI(S) = \emptyset$

Definicja

Gramatyka atrybutywna jest dobrze zdefiniowana jeśli dla każdego drzewa rozbioru syntaktycznego słowa z $L(G)$ wszystkie atrybuty są efektywnie obliczalne.



Zbiór (graf) bezpośrednich zależności atrybutów

Definicja

Dla każdej produkcji $p = (X_0 \rightarrow X_1 X_2 \dots X_n) \in P$ zbiorem (relacją, grafem) bezpośrednich zależności atrybutów jest zbiór:

$$DDP(p) = \{(X_i.a \rightarrow X_j.b) : X_j.b \leftarrow f(\dots, X_i.a, \dots) \in R(p)\}$$

Gramatyka atrybutywna jest lokalnie acykliczna, jeśli graf $DDP(p)$ jest acykliczny dla każdej produkcji $p \in P$.

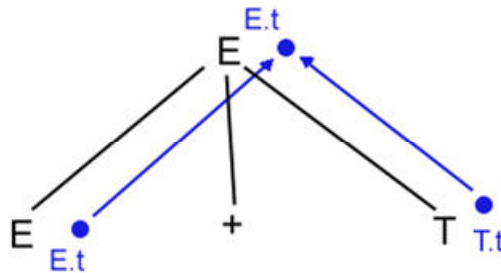


Graf bezpośrednich zależności atrybutów

Przykład

Produkcja $E \rightarrow E_1 + T$ Reguła semantyczna $E.t \leftarrow E_1.t \parallel T.t \parallel '+'$

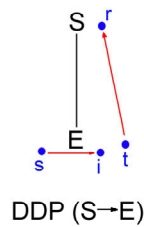
Graf bezpośrednich zależności atrybutów dla produkcji $E \rightarrow E + T$



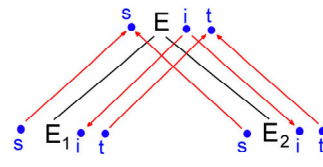


Przykład – grafy bezpośrednich zależności atrybutów

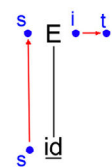
(1) $S \rightarrow E$	$E.i \leftarrow g(E.s)$ $S.r \leftarrow E.t$
(2) $E \rightarrow E_1 E_2$	$E.s \leftarrow f_s(E_1.s, E_2.s)$ $E_1.i \leftarrow f_{i1}(E.i)$ $E_2.i \leftarrow f_{i2}(E.i)$ $E.t \leftarrow f_t(E_1.t, E_2.t)$
(3) $E \rightarrow id$	$E.s \leftarrow id.s$ $E.t \leftarrow h(E.i)$



DDP ($S \rightarrow E$)



DDP ($E \rightarrow EE$)



DDP ($E \rightarrow id$)



Grafy bezpośrednich zależności atrybutów

Zbiory (relacje) bezpośrednich zależności atrybutów $DDP(p)$ dają informacje o własnościach lokalnych gramatyki atrybutywnej. Dla poprawnego obliczenia atrybutów potrzebne są informacje z bezpośredniego „otoczenia” poszczególnych produkcji, czyli informacje o bardziej globalnym charakterze.



Graf zależności

Definicja

Niech S będzie atrybutowanym drzewem rozbioru syntaktycznego słowa z $L(G)$ i niech K_0, \dots, K_n będą wierzchołkami odpowiadającymi zastosowaniu produkcji $p = (X_0 \rightarrow X_1 \dots X_n)$. Piszemy $K_i.a \rightarrow K_j.b$ jeśli $(X_i.a \rightarrow X_j.b) \in DDP(p)$. Zbiór $DT(S) = \{(K_i.a \rightarrow K_j.b)\}$ uwzględniający wszystkie zastosowania produkcji w drzewie S nazwiemy relacją zależności nad drzewem S , a odpowiadający mu graf zorientowany – grafem zależności.

Twierdzenie

Gramatyka atrybutywna jest dobrze zdefiniowana wtedy i tylko wtedy, gdy jest zupełna i graf $DT(S)$ jest acykliczny dla każdego drzewa S rozbioru syntaktycznego odpowiadającego słowu z języka $L(G)$.



Algorytm konstruowania grafu zależności

We: gramatyka atrybutywna $AG = \langle G, A, R \rangle$ oraz drzewo rozbioru syntaktycznego słowa z $L(G)$

Wy: graf zależności atrybutów dla danego drzewa rozbioru syntaktycznego

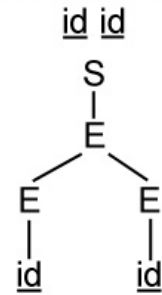
```
for każdy węzeł „n” w drzewie rozbioru syntaktycznego do
  for każdy atrybut „a” symbolu gramatyki znajdującego się
    w węźle „n” do
    zbuduj dla „a” węzeł w grafie zależności;
for każdy węzeł „n” w drzewie rozbioru syntaktycznego do
  for każda reguła semantyczna  $b \leftarrow f(c_1, \dots, c_k)$  związana z
    produkcją stosowaną w węźle „n” do
    for  $i := 1$  to  $k$  do
      zbuduj połączenie z węzła odpowiadającego
        „ $c_i$ ” do węzła odpowiadającego „ $b$ ”;
```



Przykład – konstruowanie grafu zależności (1)

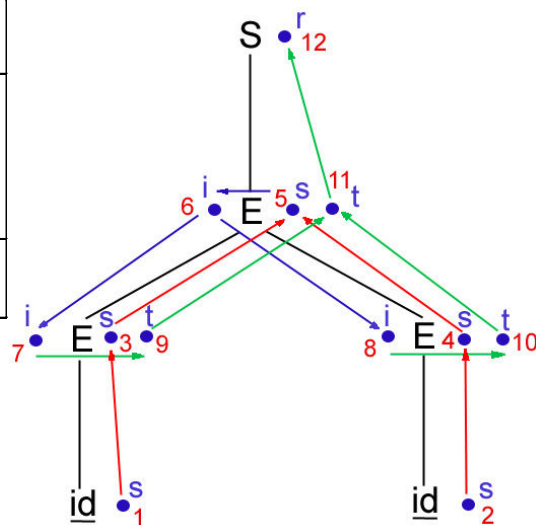
$S \rightarrow E$	$E.i \leftarrow g(E.s)$ $S.r \leftarrow E.t$
$E \rightarrow E_1E_2$	$E.s \leftarrow f_s(E_1.s, E_2.s)$ $E_1.i \leftarrow f_{i1}(E.i)$ $E_2.i \leftarrow f_{i2}(E.i)$ $E.t \leftarrow f_t(E_1.t, E_2.t)$
$E \rightarrow \underline{id}$	$E.s \leftarrow \underline{id}.s$ $E.t \leftarrow h(E.i)$

Analizowane słowo:



Przykład – konstruowanie grafu zależności (2)

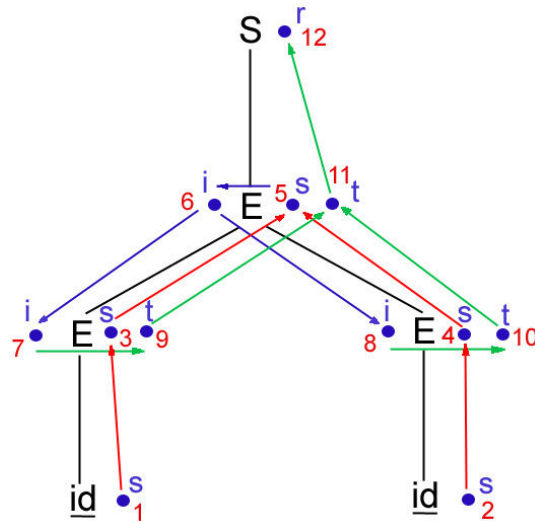
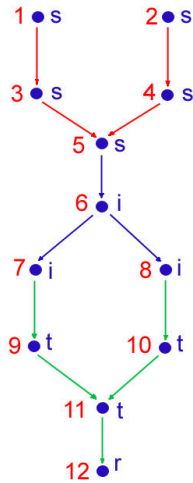
$S \rightarrow E$	$E.i \leftarrow g(E.s)$ $S.r \leftarrow E.t$
$E \rightarrow E_1E_2$	$E.s \leftarrow f_s(E_1.s, E_2.s)$ $E_1.i \leftarrow f_{i1}(E.i)$ $E_2.i \leftarrow f_{i2}(E.i)$ $E.t \leftarrow f_t(E_1.t, E_2.t)$
$E \rightarrow \underline{id}$	$E.s \leftarrow \underline{id}.s$ $E.t \leftarrow h(E.i)$





Przykład – konstruowanie grafu zależności (3)

Obliczanie atrybutów



Uporządkowanie topologiczne skierowanego grafu acyklicznego

Uporządkowanie topologiczne zorientowanego grafu acyklicznego jest to każda permutacja m_1, m_2, \dots, m_k wierzchołków tego grafu taka, że każda krawędź grafu biegnie od wierzchołka wcześniejszego w tym uporządkowaniu do wężła późniejszego. Np. jeśli $m_i \rightarrow m_j$ jest krawędzią biegnącą od wierzchołka m_i do wierzchołka m_j , to m_i pojawia się w uporządkowaniu wcześniej niż m_j .

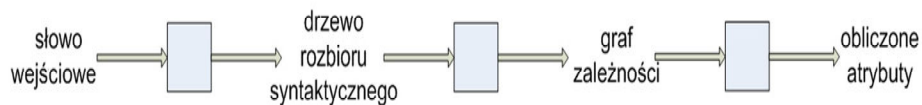
Porządek topologiczny w grafie zależności daje właściwą kolejność obliczania atrybutów w drzewie rozbioru syntaktycznego.



Metody obliczania reguł semantycznych i atrybutów

1. Metody oparte o drzewa rozbioru syntaktycznego

Kolejność obliczania atrybutów jest ustalana w czasie kompilacji na podstawie drzewa rozbioru syntaktycznego poprzez skonstruowanie grafu zależności dla każdego wejściowego słowa.



Metoda ta jest nieskuteczna tylko w przypadku gdy graf zależności dla rozważanego słowa wejściowego posiada cykle.



Metody obliczania reguł semantycznych i atrybutów

2. Metody bazujące na regułach semantycznych

Kolejność obliczania atrybutów jest rozstrzygana na etapie konstrukcji kompilatora. Analizowane są reguły semantyczne związane z poszczególnymi produkcjami (automatycznie lub ręcznie). Dla każdej produkcji kolejność obliczania atrybutów związanych z tą produkcją jest zdeterminowana już podczas konstrukcji kompilatora.



Metody obliczania reguł semantycznych i atrybutów

3. Metody nie uwzględniające bezpośrednio reguł semantycznych

Kolejność obliczania atrybutów jest ustalana bez bezpośredniego rozważania reguł semantycznych. Często tłumaczenie (analiza semantyczna) ma miejsce równoległe z analizą syntaktyczną: kolejność obliczania atrybutów jest wymuszana przez parser. Taka zasada postępowania ogranicza klasę gramatyk atrybutywnych, które mogą opisywać dokonywane tłumaczenie.



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

Problem: dana gramatyka atrybutywna AG i drzewo rozbioru syntaktycznego pewnego słowa z języka $L(G)$. Obliczyć atrybuty symboli w wierzchołkach drzewa rozbioru syntaktycznego. Założenie: gramatyka atrybutywna jest dobrze zdefiniowana.

Przypomnienie:

Relacja \leq określona na zbiorze A spełniająca warunki:

(I) $\forall a \in A : x \leq x$ (zwrotność)

(II) $(x \leq y) \wedge (y \leq x) \Rightarrow x = y$

(III) $(x \leq y) \wedge (y \leq z) \Rightarrow x \leq z$ (przechodniość)

Nazywana jest relacją porządkującą zbiór A .

(Dawniej relację taką nazywano relacją częściowego porządku.)



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

1. Dla każdego $X \in (V \cup \Sigma)$ skonstruować relację \mapsto porządkującą zbiór jego atrybutów $A(X)$ tak, aby $(X.a \mapsto X.b) \Rightarrow$ wartość $X.a$ musi być znana dla obliczenia wartości $X.b$

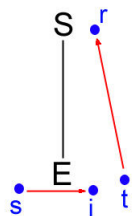
Ogólny algorytm budowania relacji \mapsto nie jest objęty niniejszym wykładem. Dalsze rozważania mają sprawdzić trafność podjętej decyzji dotyczącej relacji \mapsto .

2. Dla każdej produkcji $p \in P$ zbudować zbiór (relację) zależności atrybutów $DP(p)$

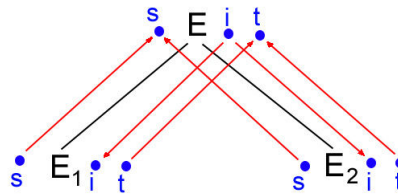


Przykład

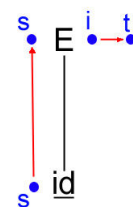
(1) $S \rightarrow E$	$Ei \leftarrow g(E.s)$ $Sr \leftarrow Et$
(2) $E \rightarrow E_1E_2$	$E.s \leftarrow f_s(E_1.s, E_2.s)$ $E_1.i \leftarrow f_{i1}(E.i)$ $E_2.i \leftarrow f_{i2}(E.i)$ $Et \leftarrow f_t(E_1.t, E_2.t)$
(3) $E \rightarrow id$	$E.s \leftarrow id.s$ $Et \leftarrow h(E.i)$



DDP ($S \rightarrow E$)



DDP ($E \rightarrow EE$)



DDP ($E \rightarrow id$)



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

Zbiory (relacje) bezpośrednich zależności atrybutów $DDP(p)$ dają informacje o własnościach lokalnych gramatyki atrybutywnej. Dla poprawnego obliczenia atrybutów potrzebne są informacje z bezpośredniego „otoczenia” poszczególnych produkcji, czyli informacje o bardziej globalnym charakterze.

Definicja

Dla każdej produkcji $p = (X_0 \rightarrow X_1 \dots X_n) \in P$

$$NDDP(p) = DDP(p)^+ \setminus \{(X_i.a \rightarrow X_j.b) : X_i.a, X_j.b \in AF(p)\}$$

Jest znormalizowanym domknięciem przechodnim grafu $DDP(p)$.



Przykład c. d.

$$NDDP(S \rightarrow E) = \{(E.s \rightarrow E.i), (E.t \rightarrow S.r)\} \quad (= DDP(S \rightarrow E))$$

$$NDDP(E \rightarrow E_1 E_2) = \{(E_1.s \rightarrow E.s), (E_2.s \rightarrow E.s), (E.i \rightarrow E_1.i), (E.i \rightarrow E_2.i), (E_1.t \rightarrow E.t), (E_2.t \rightarrow E.t)\} \quad (= DDP(E \rightarrow E_1 E_2))$$

$$NDDP(E \rightarrow \underline{id}) = \{(\underline{id}.s \rightarrow E.s), (E.i \rightarrow E.t)\} \quad (= DDP(E \rightarrow \underline{id}))$$

(W naszym przykładzie obliczenie $NDDP$ nie zmieniło poprzedniej zawartości DDP .)



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

Definicja

Indukowane zależności atrybutów gramatyki atrybutywnej AG definiuje się następująco:

$$(1) \forall p \in P : IDP(p) := NDDP(p)$$

(2)

$$\forall X \in (N \cup T) : IDS(X) := \{ (X.a \rightarrow X.b) : \exists q : (X.a \rightarrow X.b) \in IDP(q)^+ \}$$

$$(3) \forall p = (X_0 \rightarrow X_1 \dots X_n) \in P$$

$$IDP(p) := IDP(p) \cup IDS(X_0) \cup \dots \cup IDS(X_n)$$

(4) Powtarzać kroki (2) i (3) dopóty, dopóki zbiory IDS i IDP ulegają zmianie



Przykład c. d.

$$IDP(S \rightarrow E) = NDDP(S \rightarrow E) \cup \{ (E.i \rightarrow E.t) \}$$

$$IDP(E \rightarrow E_1 E_2) = NDDP(S \rightarrow E) \cup$$

$$\{ (E.s \rightarrow E.i), (E_1.s \rightarrow E_1.i), (E_2.s \rightarrow E_2.i), (E.i \rightarrow E.t), (E_1.i \rightarrow E_1.t), (E_2.i \rightarrow E_2.t) \}$$

$$IDP(E \rightarrow \underline{id}) = NDDP(E \rightarrow \underline{id}) \cup \{ (E.s \rightarrow E.i) \}$$

$$IDS(S) = IDS(\underline{id}) = \emptyset$$

$$IDS(E) = \{ (E.s \rightarrow E.i), (E.i \rightarrow E.t) \}$$

Intuicyjnie na podstawie $IDS(E)$ można określić relację porządkującą zbiór $A(E)$:

$$\left\{ (E.s \mapsto E.i), (E.i \mapsto E.t), (E.s \mapsto E.t), \begin{pmatrix} E.s \mapsto E.s \\ E.i \mapsto E.i \\ E.t \mapsto E.t \end{pmatrix} \right\}$$

(czyli $E.s \mapsto E.i \mapsto E.t$)



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

Definicja

Dla każdej produkcji $p = (X_0 \rightarrow X_1 \dots X_n) \in P$

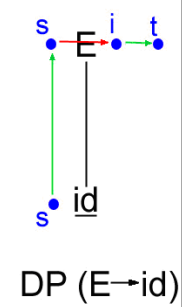
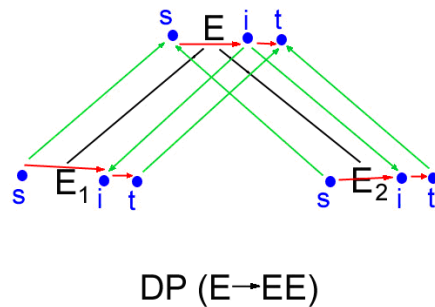
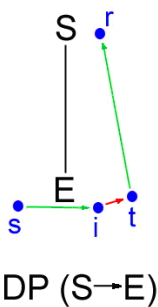
$DP(p) = IDP(p) \cup \{(X_i.a \rightarrow X_i.b) : X_i.a \mapsto X_i.b\}$
nazywamy zbiorem (relacją) zależności atrybutów.
 \mapsto - relacja porządkująca zbiór $A(X_i)$



Przykład c. d.

$$\left. \begin{aligned} DP(S \rightarrow E) &= IDP(S \rightarrow E) \\ DP(E \rightarrow EE) &= IDP(E \rightarrow EE) \\ DP(E \rightarrow \underline{id}) &= IDP(E \rightarrow \underline{id}) \end{aligned} \right\} (*)$$

(*) – dla uporządkowania $E.s \mapsto E.i \mapsto E.t$





Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

3. Badamy acykliczność grafów $DP(p)$ dla każdej $p \in P$

Definicja

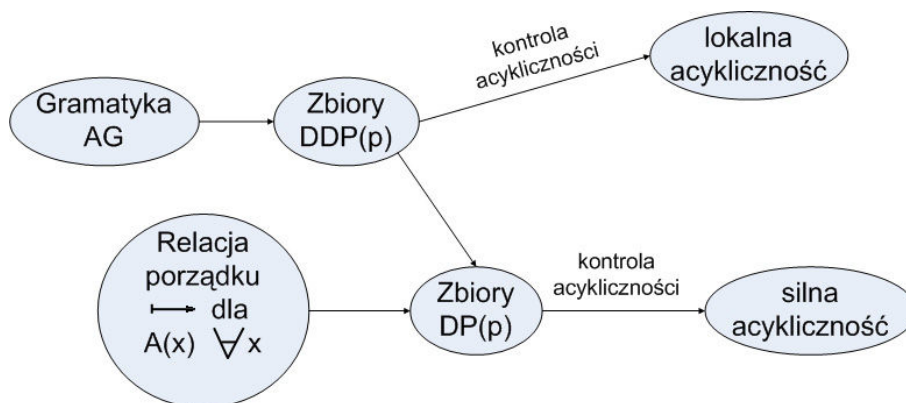
Gramatyka atrybutywna jest silnie acykliczna, gdy graf $DP(p)$ jest acykliczny dla każdej produkcji $p \in P$.

Twierdzenie

Atrybuty w gramatyce atrybutywnej mogą być wyliczane w kolejności wynikającej z relacji porządkującej (\mapsto) zbiory $A(X) (\forall X \in (V \cup \Sigma))$ wtedy i tylko wtedy, gdy gramatyka ta jest silnie acykliczna.



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego





Przykład c. d.

Można sprawdzić, że zbiory $DP(p) \forall p \in P$ w naszej gramatyce są acykliczne. Ponieważ zbudowano je dla uporządkowania atrybutów symbolu E takiego, że $E.s \mapsto E.i \mapsto E.t$, więc atrybuty symbolu E powinny być obliczane w tej kolejności.



Obliczanie atrybutów w atrybutowanym drzewie rozbioru syntaktycznego

4. Konstruujemy funkcje rekurencyjne do obliczania atrybutów syntetyzowanych.

Funkcja zwraca wartość atrybutu syntetyzowanego symbolu A . Parametrem wejściowym jest wskaźnik wierzchołka odpowiadającego symbolowi A .

Jeśli do obliczenia atrybutu syntetyzowanego potrzebna jest wartość atrybutu dziedziczony symbolu A , co wynika z przyjętego uporządkowania \mapsto zbioru $A(A)$, to wartość tego atrybutu dziedziczony jest parametrem wejściowym funkcji.



Przykład c.d.

$$E \rightarrow E_1 E_2 \quad \{E.s \leftarrow f_s(E_1.s, E_2.s)\}$$
$$E \rightarrow id \quad \{E.s \leftarrow id.s\}$$

```
function Es(n:node):Es_type;
var
  s1,s2:Es_type;
begin
  case produkcja zastosowana w węzle n of
    'E → E1E2': begin
      s1:=Es(child(n,1));
      s2:=Es(child(n,2));
      Es:=fs(s1,s2);
      end;
    'E → id': begin
      Es:=get_id_s;
      end;
  else error;
  end;
end;
```



Przykład c.d.

$$E \rightarrow E_1 E_2 \quad \{E_i.i \leftarrow f_n(E.i)$$
$$E_2.i \leftarrow f_{i2}(E.i)$$
$$E.t \leftarrow f_t(E_1.t, E_2.t) \}$$
$$E \rightarrow id \quad \{E.t \leftarrow h(E.i)\}$$

```
function Et(n:node; i: Ei_type):Et_type;
var
  i1, i2 : Ei_type;
  t1, t2 : Et_type;
begin
  case produkcja zastosowana w węzle n of
    'E → E1E2': begin
      i1 := fi1(i);
      t1 := Et(child(n,1), i1);
      i2 := fi2(i);
      t2 := Et(child(n,2), i2);
      Et := ft(t1, t2);
      end;
    'E → id': begin
      Et:=h(i);
      end;
  else error;
  end;
end;
```



Przykład c.d.

$$S \rightarrow E \quad \left\{ \begin{array}{l} E_1.i \leftarrow g(E.s) \\ S.r \leftarrow E.t \end{array} \right\}$$

```
function Sr(n:node):Sr_type;  
var  
    s : Es_type;  
    i : Ei_type;  
    t : Et_type;  
begin  
    s := Es(child(n,1));  
    i := g(s);  
    t := Et(child(n,1), i);  
    Sr := t;  
end;
```



Przykład c.d.

$$S \rightarrow E \quad \left\{ \begin{array}{l} E_1.i \leftarrow g(E.s) \\ S.r \leftarrow E.t \end{array} \right\}$$

```
function Sr(n:node):Sr_type;  
var  
    s : Es_type;  
    i : Ei_type;  
    t : Et_type;  
begin  
    s := Es(child(n,1));  
    i := g(s);  
    t := Et(child(n,1));  
    Sr := t;  
end;
```




Przykład c.d.

Wywołanie dla rozpoczęcia obliczania atrybutów:

$value := Sr(root);$

gdzie:

$root$ - korzeń drzewa rozbioru syntaktycznego (wskaźnik)