Programare Logică

Metoda Algebrei Iniţiale

Aplicaţii

Vom prezenta următoarele aplicaţii ale metodei algebrei iniţiale:

- reprezentarea expresiilor în formă poloneză inversă,
- compilării unei expresii aritmetice folosind o maşina cu stivă şi acumulator.

$G = (S_0, N, T, P)$ pt. expresii

Definim o g.i.c. pentru expresii construite cu variabilele $X = \{x, y, z\}$.

- $G = (S_0, N, T, P)$
  $N = \{\langle prog \rangle, \langle exp \rangle, \langle term \rangle,$
  $\langle fact \rangle, \langle var \rangle\},$
  $S_0 = \langle prog \rangle,$
  $T = \{x, y, z, (,), +, *\}$, $P = \{p0, \ldots, p9\}$,

$G = (S_0, N, T, P)$ pt. expresii

$[p0]\langle prog\rangle \longrightarrow \langle exp\rangle$

$[p1]\langle var\rangle \longrightarrow x$

$[p2]\langle var\rangle \longrightarrow y$

$[p3]\langle var\rangle \longrightarrow z$

$[p4]\langle fact\rangle \longrightarrow \langle var\rangle$

$[p5]\langle fact\rangle \longrightarrow (\langle exp\rangle)$

$[p6]\langle term\rangle \longrightarrow \langle fact\rangle$

$[p7]\langle term\rangle \longrightarrow \langle fact\rangle * \langle term\rangle$

$[p8]\langle exp\rangle \longrightarrow \langle term\rangle$

$[p9]\langle exp\rangle \longrightarrow \langle exp\rangle + \langle term\rangle$

Semantica algebrei iniţiale

$$\begin{tikzcd}
& T_\Sigma \arrow[r, "\mathcal{S}_G"] \arrow[dl, dotted, "\mathcal{P}"] \arrow[d, "A"', "\mathcal{S}"] \arrow[dr, dotted, "\mathcal{C}"] & Lang(G) \\
Pol & \mathcal{A} & Cod
\end{tikzcd}$$

$$\begin{tikzcd}
t_w \arrow[r, shift left, "\mathcal{S}_G"] \arrow[d, "\mathcal{S}_A"] & w \arrow[l, shift left, dashed] \\
Sem(w) = \mathcal{S}_A(t_w)
\end{tikzcd}$$

Algebra Pol - suportul

- Definim forma poloneză postfix a unei expresii din $L(G)$ prin metoda algebrei iniţiale. Pentru aceasta construim $\mathcal{G}$-algebra semantică Pol, unde $\mathcal{G} = (S = N, \Sigma = P)$.
- $Pol_s = T^* = \bigcup_{k \geq 0} T^k$, $s \in N \setminus \{\langle var \rangle\}$
  $Pol_{\langle var \rangle} = X = \{x, y, z\}$

Algebra Pol - operaţiile

- $Pol_p : T^* \to T^*$, $p \in \{p0, p5, p6, p8\}$,
  $Pol_p(w) = w$ oricare $w \in T^*$,
- Pol_{p1}, Pol_{p2}, $Pol_{p3} :\to X$,
  $Pol_{p1} = x$, $Pol_{p2} = y$, $Pol_{p3} = z$,
- $Pol_{p4} : X \to T^*$, $Pol_{p4}(v) = v$, oricare $v \in X$,
- $Pol_{p7} : T^* \times T^* \to T^*$, $Pol_{p7}(\alpha, \beta) = \alpha\beta *$,
- $Pol_{p9} : T^* \times T^* \to T^*$, $Pol_{p7}(\alpha, \beta) = \alpha\beta +$,

Forma poloneză

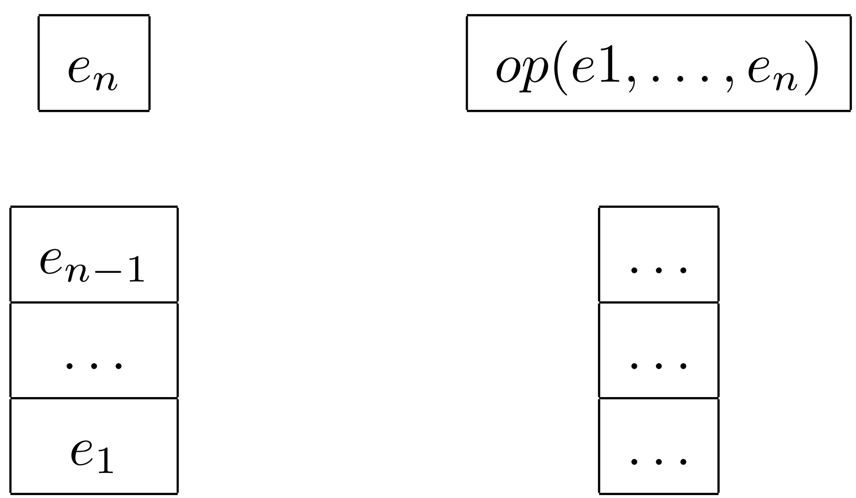
- $\mathcal{S}_G : T_\Sigma \to Lang(G)$ unicul $\mathcal{G}$-morfism
  $\mathcal{P} : T_\Sigma \to Pol$ unicul $\mathcal{G}$-morfism
- Pentru orice $e \in L(G)$, forma poloneză postfix este $\mathcal{P}(t_e)$, unde $t_e \in T_\Sigma$ este unicul termen cu $\mathcal{S}_G(t_e) = e$.
- $e = y * (x + z)$
  $t_e =$
  $p0(p8(p7(p4(p2), p5(p9(p8(p6(p4(p1))), p6(p4(p3)))))))$
  $\mathcal{P}(t_e) = \mathcal{P}(T_{p0}(T_{p8}(T_{p7}(T_{p4}(T_{p2}), T_{p5}(T_{p9}(T_{p8}($
  $T_{p6}(T_{p4}(T_{p1}))), T_{p6}(T_{p4}(T_{p3})))))))) =$
  $Pol_{p0}(Pol_{p8}(Pol_{p7}(Pol_{p4}(Pol_{p2}), Pol_{p5}(Pol_{p9}(Pol_{p8}($
  $Pol_{p6}(Pol_{p4}(Pol_{p1}))), Pol_{p6}(Pol_{p4}(Pol_{p3}))))))) =$
  $Pol_{p7}(y, Pol_{p9}(x, z)) = Pol_{p7}(y, xz+) = yxz + *$

Compilarea unei expresii

- Folosim pentru compilare o maşină cu stivă şi acumulator (MSA)
- Definim limbajul MSA şi algebra semantică Cod. Compilarea expresiei $e \in L(G)$ în limbajul MSA este $\mathcal{C}(t_e)$, unde $t_e \in T_\Sigma$ este unicul termen cu $\mathcal{S}_G(t_e) = e$, iar $\mathcal{C} : T_\Sigma \to Cod$ este unicul $\mathcal{G}$-morfism.

Funcţionarea SMA

- La efectuarea operaţiei $op(e1, \ldots, e_n)$, valorile e_1, ..., e_{n-1} vor fi în stivă, iar valoarea lui e_n în registru.
- Rezultatul evaluării oricărei expresii este depus în registru. Registrul este vârful *real* al stivei. Evaluarea unei expresii lasă stiva neschimbată.



Instrucţiunile SMA

- R este registrul (acumulatorul)
- P este adresa primei poziţii libere din stivă
- Instrucţiunile SMA:

inc P (incrementează P)
dec P (decrementează P)
Ad R (adună valoarea din vârful stivei cu cea din R,
iar rezultatul este depus în R)
Mu R (înmulţeşte valoarea din vârful stivei cu cea din R,
iar rezultatul este depus în R)
ld v (încarcă în R valoarea din v)
st R (mută în stivă valoarea din R)
print (afişează conţinutul lui R)

Algebra Cod - suportul

- $Inst(MSA)$ = instructiunile definite anterior
- codul unei expresii este un şir de instrucţiuni separate prin ";"
- $Cod_s = (Inst(SMA) \cup \{;\})^*$, oricare $s \in S = N$
- Notăm $SI := Cod_s$, $s \in S$

Algebra Cod - operaţiile

- $C_{p0} : SI \to SI$,
  $C_{p0}(\alpha) = \alpha;$print
- $C_p : SI \to SI$, $p \in \{p4, p5, p6, p8\}$
  $C_p(\alpha) = \alpha$
- C_{p1}, C_{p2}, $C_{p3} :\to SI$,
  $C_{p1} =$ld x, $C_{p2} =$ld y, $C_{p3} =$ld z
- $C_{p7} : SI \times SI \to SI$,
  $C_{p7}(\alpha, \beta) = \alpha;$st R;inc P; $\beta;$Mu R;dec P
- $C_{p9} : SI \times SI \to SI$,
  $C_{p9}(\alpha, \beta) = \alpha;$st R;inc P; $\beta;$Ad R;dec P

Compilarea

- Pentru orice $e \in L(G)$, codul evaluării expresiei e este $\mathcal{C}(t_e)$, unde $t_e \in T_\Sigma$ este unicul termen cu $\mathcal{S}_G(t_e) = e$.
- $e = y * (x + z)$
  $t_e =$
  $p0(p8(p7(p4(p2), p5(p9(p8(p6(p4(p1))), p6(p4(p3)))))))$
  $\mathcal{C}(t) = \mathcal{C}(T_{p0}(T_{p8}(T_{p7}(T_{p4}(T_{p2}),$
  $T_{p5}(T_{p9}(T_{p8}(T_{p6}(T_{p4}(T_{p1}))), T_{p6}(T_{p4}(T_{p3})))))))) =$
  $C_{p0}(C_{p8}(C_{p7}(C_{p4}(C_{p2}), C_{p5}($
  $C_{p9}(C_{p8}(C_{p6}(C_{p4}(C_{p1})))), C_{p6}(C_{p4}(C_{p3})))))) =$

Compilarea

$C_{p0}(C_{p7}(C_{p2}, C_{p9}(C_{p1}, C_{p3}))) =$

$C_{p7}(C_{p2}, C_{p9}(C_{p1}, C_{p3}))$;print=

C_{p2};st R;inc P; $C_{p9}(C_{p1}, C_{p3})$;
Mu R;dec P;print=

C_{p2};st R;inc P; C_{p1};st R;inc P; C_{p3};
Ad R;dec P ;Mu R;dec P;print=

Idy;st R;inc P; Id x;st R;inc P; Id z;
Ad R;dec P ;Mu R;dec P;print

$e = y * (x + z)$, $x = 3$, $y = 7$, $z = 2$

Cod	R	Stiva($\leftarrow$)	P
-	-	-	1
ld y;	7	-	1
st R;inc P;	7	7	2
ld x;	3	7	2
st R;inc P;	3	3 7	3
ld z;	2	3 7	3
Ad R;dec P;	5	7	2
Mu R;dec P;	35	-	1
print	35		