

u program sursă C

preprocesare

program sursă C

compilare

program asamblare

asamblare

cod mașină relocabil

linker /

cod mașină absolut

program sursă

an. lexical

an. sintactice

an. semantică

gen. de cod intermediar

optimizare cod

generare de cod

p. de analiză

Tratament
simbol

p. sinteză

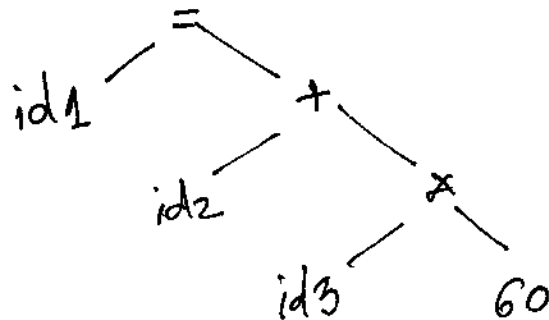
Tabela
de
simboluri

pozitie = initial + rata * 60;

↓
[an. lexicală]

↓
 $id_1 = id_2 + id_3 * 60;$

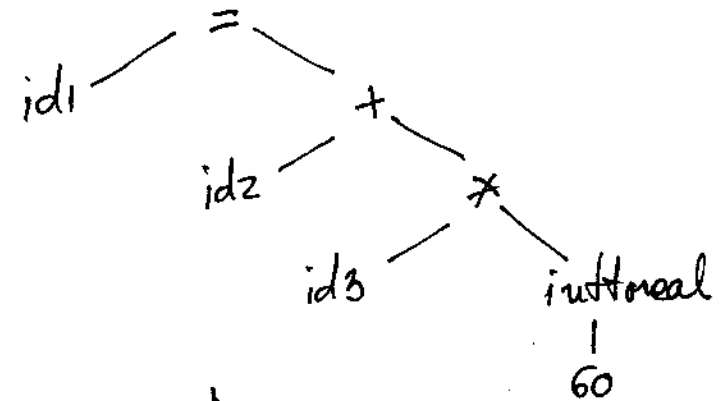
↓
[an. sintactică]



↓
[an. semantică]



pozitie, initial, rata → real



↓
[generare cod intermediar]

temp1 = initial(60)

temp2 = id3 * temp1

temp3 = id2 + temp2

id1 = temp3

↓
[optimizare de cod]

↓
temp1 = id3 * 60.0

id1 = id2 + temp1

↓
[generare de cod]

MOVF id3, R2
MULF #60.0, R2
MOVF id2, R1
ADDF R2, R1
MOVF R1, id1

T. calculabilității - studiul matematic al Maximilor de Calcul (M.C.) și al capab. în

Model $\begin{cases} \text{M.C.} \\ \text{datele} \end{cases}$ - șiruri de simboluri

Def.

Un alfabet este o mulțime finită de simboluri. (Notat Σ).

ex: alfabet roman $\{a, \dots, z\}$

alfabet binar $\{0, 1\}$

Def.

Un șir peste un alfabet este o secvență finită de simboluri din alfabet.

Notatie

- ϵ → șirul vid
- Σ^* → mulțimea tuturor șirurilor peste Σ .

Def
Orice mulțime de șiruri peste un alfabet Σ , adică orice submulțime din Σ^* , este un limbaj (Σ^* , \emptyset , Σ sunt limbaie).

Specificarea limbajelor

• Limbaj \rightarrow mulțime \rightarrow specificare mulțimii

Limbaj \rightarrow finit \rightarrow enumerarea elementelor

\rightarrow infinit \rightarrow prin proprietăți, $L = \{w \in \Sigma^* \mid w \text{ are proprietatea } P\}$.

Operații cu limbaie

• opțiuni specifice mulțimilor \cup, \cap, \setminus

• opțiuni specifice:

Def - concatenare

L_1, L_2 - limbaie peste alf. Σ , concatenarea lor

$L = L_1 \circ L_2$ sau $L = L_1 L_2$, unde

$L = \{w \mid w = x \cdot y, x \in L_1, y \in L_2\}$

Def - închiderea tranzitivă (Kleene Star)

L^* este mulțimea tuturor șirurilor obținute prin concatenarea a 0 sau mai multe șiruri din L . (concatenarea a 0 șiruri este ϵ , concatenarea unui șir este acel șir).

$$L^* = \{ w \in \Sigma^* \mid w = w_1 \circ w_2 \circ \dots \circ w_k, k \geq 0, w_1, \dots, w_k \in L \}.$$

ex:

$$\Sigma = \{0, 1\}$$

$$L_1 = \{ w \in \Sigma^* \mid w \text{ are un număr par de } 0 \}$$

$$L_2 = \{ w \in \Sigma^* \mid w \text{ începe cu } 0 \text{ și restul simbolurilor sunt } 1 \}$$

$$L = L_1 \circ L_2 = \{ w \in \Sigma^* \mid w \text{ are număr impar de } 0 \}$$

Obs:

Utilizarea notăției Σ^* pt a reprezenta mulțimea tuturor șirurilor peste Σ este consistentă cu notăția Kleene Star asupra Σ privit ca un alfabet finit.

Notăție $L^+ = LL^*$

$$L^+ = \{ w \mid w = w_1 \circ w_2 \circ \dots \circ w_k, k \geq 1, w_1, \dots, w_k \in L \}.$$

Reprezentarea finită a limbajelor

Limbaj finit \rightarrow enumerarea elem.

\hookrightarrow infinit $\rightarrow ?$

Reprezentare finită - sir de simboluri peste un alf. Σ .

$\Sigma^* \rightarrow$ numărabilă $\Rightarrow f: \mathbb{N} \rightarrow \Sigma^*$

- 1) $\forall k \geq 0$, numărăm sirurile de lg. k înaintea celor de lg. $k+1$
- 2) n^k enumerate lexicografic

Multiplu de limbajelor $\Rightarrow \mathcal{P}(\Sigma^*) \stackrel{\text{Mod}}{=} 2^{\Sigma^*}$

Def - Multiplu putere \rightarrow multiplu de submultiplu care se pot forma pe o multiplu dată.

ex: $A = \{a, b\}$

$2^A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$.

Unul rezultat al T. Calculabilității

Oricât de puternice ar fi metodele de reprezentare, numai o mulțime numărabilă de limbaje poate fi reprezentate, atât timp cât reprezentarea este finită.