

Atomi lexicali:

- identificatori
- constante (literali)
- cuvinte cheie (*cuvinte rezervate*)
- operatori aritmetici, relationali, ...
delimitatori

spatiile "albe"

cuvinte cheie - simboluri sintactice

cuvinte rezervate - nu pot fi folosite ca identificatori
(semnificatii speciale)

Analiza lexicala

detecteaza



Pentru fiecare *atom* lexical din fisierul sursa

clasifica *atom*

codifica *atom*

Sf.pentru

Analiza lexicala (versiunea 1)

Analiza lexicala

- Date: fisier text
- Rezultate: FIP, TS
 sau: mesaj de eroare

Analiza lexicala

Cat timp (mai exista caractere necitite in prg. sursa) ex
detecteaza atom

Daca e cuvant-cheie sau operator sau separator atunci
adauga in FIP codul corespunzator atomului

Altfel

Daca e identificator sau constanta

Daca nu exista deja in TS atunci adauga-l ; Sf. Daca
adauga in FIP codul coresp. atomului si pozitia in TS

altfel

Eroare

Sf.Daca

Sf. Daca

Sf.cattimp

Analiza lexicala (versiunea 2)

Cat timp (mai exista caractere necitite in prg. sursa) ex
detecteaza (urmatorul) atom

Daca e cuvant-cheie sau operator sau separator atunci
adauga in FIP codul corespunzator atomului

Altfel

Daca e identificator sau constanta

Daca nu exista deja in TS atunci adauga-l ; Sf. Daca
adauga in FIP codul coresp. atomului si pozitia in TS

altfel

Eroare

Sf.Daca

Sf. Daca

Sf.cattimp

Analiza lexicala (versiunea 2)

Cat timp (mai exista caractere necitite in prg. sursa) ex

detecteaza *atom*

Daca *atom* e cuvant-cheie sau operator sau separator atunci

adaugaFIP(cod(*atom*), 0)

Altfel

Daca *atom* e identificator atunci

indice:=poz(*atom*,TS)

adaugaFIP((cod_id, indice)

altfel Daca *atom* e constanta atunci

indice:=poz(*atom*,TS)

adaugaFIP((cod_const, indice)

altfel

MesajEroare, indicatie asupra erorii

sfDaca

Sf.Daca

Sf. Daca

Sf.cattimp

Analiza lexicala (versiunea 2)

Obs.: multe cautari in TS

++ urmatoarele faze !!

- cautare secventiala
- organizare pt. cautare mai rapida
 - ordonare (ordonata lexicografic)
 - tabel ordonat lexicografic
 - arbore binar de cautare echilibrat
 - tabela de dispersie
 - ➔ cautare – fct. de hashing

TS:
comp
x3
forma
b
an
medie
first



Hash, fct. hash H

→ Structuri de date

H: String

c_n, \dots, c_1 → coduri ASCII
→ intregi

- $H = c_n \mod M$
- $H = c_n + c_{n-1} + \dots + c_1 \mod M$
- $H = (\dots(c_n * 256 + c_{n-1}) * 256 + c_{n-2}) \dots * 256 + c_1$
- h_0
 $h_i = h_{i-1} * k + c_i$
 $H = h_n \mod M$



Hash, fct. hash H

→ Structuri de date

H: String

```
// unsigned char *str; int c;  
hash = 5381;  
while (c = *str++)  
    hash = ((hash << 5) + hash) + c;
```

c_n, \dots, c_1 → coduri ASCII

→ intregi

- ...

$h_0=5381$

$h_i=h_{i-1}*k+c_i$

$H=h_n$

$k=33$

mod M



Hash, fct. hash H

→ Structuri de date

H: String

c_n, \dots, c_1 → coduri ASCII

→ intregi

- ...

$$\mathbf{h_0=0}$$

$$h_i = h_{i-1} * k + c_i$$

$$H = h_n$$

$$\mathbf{k=65599}$$

$$\text{mod } M$$



Hash, fct. hash H

→ Structuri de date

H: String

c_n, \dots, c_1 → coduri ASCII
→ intregi

- ...

$$\mathbf{h_0=0}$$

$$h_i = h_{i-1} * \mathbf{k} + c_i$$

$$\mathbf{k=613}$$

$$H = \text{ceiMaiSemnificativi}_{\mathbf{30_}}\text{Biti}(h_n) \mod M$$

$$\mathbf{M=1008}$$



- Detectează atom:

AF pt. fiecare clasă de atomi

