

# Processadors de Llenguatges

## Pràctica I – Utilització de l'eina de generació d'analitzadors lèxics *lex*

Teresa Alsinet  
teresa.alsinet@udl.cat

### Objectius

La implementació manual de la rutina d'anàlisi lèxica per a qualsevol llenguatge de programació és sempre una tasca costosa. L'objectiu d'aquesta pràctica és utilitzar *lex* com a eina de generació automàtica d'analitzadors lèxics, la qual està especialment dissenyada per ser utilitzada conjuntament amb l'eina de generació automàtica d'analitzadors sintàctics *yacc*.

### Exercicis de programació

#### Exercici 1 (Llenguatge de programació)

Escolliu un llenguatge i, per a aquest llenguatge, implementeu una especificació *lex* que calculi el nombre de paraules reservades, identificadors, constants (enteres, reals, de caràcter, de cadena -string- ....), operadors (aritmètics, lògics, relacionals, d'assignació ...), símbols delimitadors ('(', ')', '[', ']', ',', ';', ':', ...), símbols de separació (blancs, tabuladors, new line ...) i comentaris que apareixen en un programa escrit en el llenguatge. Podeu ignorar qualsevol altre component. Per a cada categoria de components del llenguatge, indiqueu la proporció que representa respecte del total analitzat. El nom del fitxer d'entrada serà proporcionat per l'usuari des de línia de comanda. La sortida s'escriurà directament a la sortida estàndard.

#### Exercici 2 (Llenguatge de la lògica proposicional)

Implementeu una especificació *lex* que reconegui els components del llenguatge fòrmules proposicionals definides sobre les lletres en majúscula, és a dir, sobre el rang [A-Z]

#### Consideracions:

- Els operadors a considerar són: la negació  $!$ , la conjunció  $\wedge$ , la disjunció  $\vee$ , la implicació que representarem amb el string “ $\rightarrow$ ”, i la doble implicació que representarem amb el string “ $\leftrightarrow$ ”.

- Les fòrmules proposicionals podran estar parentitzades, és a dir, els símbols de parèntesis “(” i “)” són elements del llenguatge.
- Les fòrmules són proporcionades des de l’entrada estàndard.
- El caràcter nova línia actuarà com a marca de final fòrmula i la marca de final d’arxiu com a final d’entrada. El processat acaba amb la marca de final d’arxiu.
- Desprecieu els caràcters blancs i tabuladors existents a l’entrada.
- Definir un format per a incloure línies de comentaris.
- Al detectar un error, l’analitzador lèxic emetrà un missatge indicant el caràcter que l’ha produït i el número de línia del programa font on ha estat detectat. La tècnica de recuperació associada és el mode pànic: l’analitzador lèxic despreciarà tots els símbols fins el caràcter nova línia o la marca de final d’arxiu.

**Exemples de fòrmules lèxicament correctes són:**

- $P \wedge Q < - > (!R \vee (Q - > T))$
- $P \wedge < - > (!R \vee (Q - > T))$
- $P \wedge Q < - > (!R \vee (Q - > T$
- $P \quad Q < - > (!R \quad (Q - > T))$

És a dir, a nivell lèxic no identifiquem possibles errors sintàctics. En canvi, **exemples de fòrmules lèxicament incorrectes són:**

- $p \wedge q < - > (!R \vee (Q - > T))$
- $P + Q * (!R \vee (Q - > T))$
- $[P \wedge Q < - > [!R \vee (Q - > T)]]$
- $P \wedge Q <=> (!R \vee (Q => T))$
- $P \wedge Q < - > (!R \vee (Q - > T));$

**Exercici 3** (Lleguatge de les expressions regulars)

Implementeu una especificació *lex* que reconegui els components del llenguatge expressions regulars definides sobre l’alfabet  $\Sigma = \{a, b, c, d\}$ .

**Consideracions:**

- Les expressions regulars són proporcionades des de l’entrada estàndard.

- El caràcter nova línia actuarà com a marca de final d'expressió regular i la marca de final d'arxiu com a final d'entrada. El processat acabarà amb la marca de final d'arxiu.
- Desprecieu els caràcters blancs i tabuladors existents a l'entrada.
- A nivell lèxic, representarem  $\epsilon$  mitjançant la paraula reservada (string) "BUIDA" o "buida".
- Els operadors a considerar són:  $|$   $.$   $*$   $+$   $?$   $($   $)$ , on el símbol  $.$  denota la concatenació.
- Definir un format per a incloure línies de comentaris.
- Al detectar un error, l'analitzador lèxic emetrà un missatge indicant el caràcter que l'ha produït i el número de línia del programa font on ha estat detectat. La tècnica de recuperació associada és el mode pànic: l'analitzador lèxic despreciarà tots els símbols fins el caràcter nova línia o la marca de final d'arxiu.

**Exemples de fòrmules lèxicament correctes són:**

- $(a|BUIDA).c.(d.buida)?$
- $a \ b \ (c|d)+$
- $((a|b|c).(buida|b+.c*))?$
- $(buida|+.c*)?$

És a dir, a nivell lèxic no identifiquem possibles errors sintàctics. En canvi, **exemples de fòrmules lèxicament incorrectes són:**

- $(A|BUIDA).C.(D.buida)?$
- $[a-z]+$
- $zabc+$
- $(a|Empty).c.(d.buida)?$
- $(a|BUIDA).c.(d.buida)?;$

#### **Exercici 4** (Preprocessador)

Estem interessats en dissenyar un preprocessador de llenguatges tipus C, el qual ha de permetre la inclusió d'arxius i l'expansió de macros sense paràmetres (constants simbòliques). Implementa una especificació *lex*, que agafi com a entrada un programa i que generi com a sortida el programa equivalent però sense les sentències

de preprocessador esmentades. És a dir, el programa equivalent contindrà el codi complet dels fitxers capçalera inclosos en el programa mitjançant una sentència de preprocessador, i totes les referències a macros substituïdes per la definició associada en el programa d'entrada. El nom del fitxer d'entrada i de sortida seran proporcionats per l'usuari des de línia de comanda.

### **Exercici 5 Exercici opcional (+ 0,5 punts)**

Amplieu la solució a l'exercici 4 amb el processat de macros amb paràmetres. Podeu fixar les restriccions lèxiques que considereu oportunes.

## **Lliurament**

La documentació a lliurar per a cada exercici de programació és la següent:

1. Especificació *lex*.
2. Mòduls auxiliars emprats en la implementació de la solució global.
3. Joc de proves utilitzat per a la validació de l'exercici.
4. Si ho considereu oportú, un fitxer README amb les particularitats pròpies de la vostra implementació.

Lliurament de la pràctica al `cv.udl.cat` dins d'activitats. Lliurar un arxiu comprimit que agrupi mitjançant la utilitat `tar`, els fitxers fonts (extensions `.l` i `.c`) dels exercicis de programació, els jocs de proves utilitzats per a la validació dels exercicis, el README i un arxiu `makefile` que compili correctament els fitxers font per a cada exercici de programació.

## **Avaluació**

- La pràctica la podeu realitzar de manera individual o en grups de 2 persones.
- El pes d'aquesta pràctica és d'un 15% sobre la nota final de l'assignatura.
- La data límit per lliurar la pràctica és el 22 de març per a l'avaluació contínua. Pels que no la tingueu acabada el 22 de març, la podeu lliurar fins el 20 d'abril (data de l'examen del 1r parcial de l'assignatura).