

# Pràctica Compiladors: Compilador LANS

Ismael El Habri

Lluís Trilla

21 de gener de 2019

## 1 Modificacions Part Léxica i Sintàctica

### 1.1 TK\_STRING\_LITERAL

La nostra regla TK\_STRING\_LITERAL que ens serveix per identificar els literals de string és

```
TK_STRING_LITERAL: TK_OP_QUOTE ( ~( ' ' | '\n' | '\\ ' ) | ( '\\ ' ' ' ) ) * TK_OP_QUOTE;
```

I anteriorment, TK\_OP\_QUOTE no era un fragment, de manera que el lexer ens acceptava coses del estil "abc; . Ho vam corregir per tal que fós un fragment, i es va corregir l'error.

### 1.2 TK\_CONST\_INT

A la regla TK\_CONST\_INT anteriorment reconeixíem 0000 però no 0001, és a dir, només acceptavem 0s a l'esquerra en cas que el valor fós 0. Vam corregir la regla de manera que va quedar així:

```
TK_CONST_INT: (( '0' | DIGIT ) + );
```

I es va resoldre l'error esmentat. Això s'ha fet també amb els reals, per temes de consistència.

```
TK_CONST_REAL: ((( '0' | DIGIT ) + ' . ' ) ( '0' | DIGIT ) +  
( 'E' ' - ' ? ( DIGIT | '0' ) ( '0' | DIGIT ) * ) ? );
```

### 1.3 TK\_MULTILINE\_COMMENTS

A la regla TK\_MULTILINE\_COMMENTS continuem usant l'operador "\*\*?". Si féssim servir  
(. \*)?

enlloc de

(. ) \* ?

aleshores considerariem cadenes com "/\*comentari\*/codiVàlid()/\*comentari2\*/" com a un sòl comentari, el qual inclouria també codiValid().

## 2 Estructura Taula de Símbols

La taula de Símbols consisteix en un Mapa de Registres.

Els registres consisteixen en una classe amb els següents camps:

Camp	Tipus	Significat
lexema	String	nom de l'entrada.
tipus	String	tipus del lexema.
tipusLexema	String	tipus de lexema que conté l'entrada.
campsAddicionals	Vector de Pairs de dos Strings	Camps addicionals, com ara els camps de les tuples o els paràmetres de les funcions i accions.
adreca	Long	Adreça on es situa l'entrada en la màquina.

Aquí una taula amb els possibles continguts dels camps:

Camp	Constant	Variable	Tupla
tipus	tipus	tipus	null
tipusLexema	const"	"var"	"tupla"
campsAddicionals	null	null	tipus i camps
adreca	Adreça a la ConstantPool	Adreça variable	0

Camp	Alias	Acció/Funció	Vector
tipus	tipus real	tipus retorn, void si és acció	tipus
tipusLexema	"alias"	"funcio"	"vector"
campsAddicionals	null	tipus paràmetres i entr o entrsor	null
adreca	0	adreça funció/acció	adreça vector

A part, amb les tuples, s'han afegit registres com a variables per cada camp de la tupla, usant com a nom de lexema el següent format: "nomTupla.nomCamp". Com que sabem que no hi poden haver variables amb punts al mig, sabem que no hi haurà conflictes.

## 3 Parts no Obligatòries

Hem fet el semàntic de les següents parts no obligatòries:

- Alias
- Tupla
- Vector

Pel que fa la generació de codi:

- Alies
- Tupla

## 4 Decisions

La única decisió que hem fet ha estat que es puguin ficar zeros a l'esquerra en els enters i els reals.