

# Limbaje Formale si Translatoare

## Laborator 6

Martie 2021

### 1 Scopul Lucrarii

1. Finalizarea gramaticii pentru afirmatii if
2. Implementarea unui interpretor pentru o parte din limbajul LISP

### 2 Desfasurarea Laboratorului

#### 2.1 If parser

Finalizati gramatica pentru recunoasterea afirmatiilor if din laboratorul anterior. Folositi fisierul if.l6.y inclus in fisierle laboratorului. Adaugati functiile din if\_functions.c descrise anterior, ca actiuni.

- (a) Mai intai, compilati si analizati functiile definite in if\_functions.c

```
gcc if_functions.c
```

- (b) Verificati codul inclus in comentariile din prima parte a fisierului, si comentati sectiunile necesare
- (c) Verificati definitia tipului pentru simbolurile nonterminale
- (d) Odata ce codul este functional, eliminati regula

```
%prec IFX frm if stmt first
```

si verificati arborele pentru intrarea

```
if(a>2) then if (a >5) then a=1 else a=2
```

#### 2.2 Interpretor LISP

Scrieti un interpretor pentru o parte din limbajul LISP, incepand de la fisierele lisp.l si lisp.y

- (a) Fisierul test\_cons defineste structura unei celule cons si functiile. Compilati fisierul si rulati-l. Veti folosi functiile si structurile de aici in lisp.y.
- (b) Folositi lisp.l si lisp.y si inlocuiti

```
/* your code here */
```

cu sectiunile necesare de cod. Interpretorul lisp trebuie sa trateze urmatoarele functii: CONS, CAR, CDR, APPEND si plusul binar.

Exemple de functionare:

```
>2
2
>'(1 2 3)
(1 2 3)
>(CAR '(1 2 3))
1
>(CDR '(1 2 3))
(2 3)
>(+ 1 2)
3
>(CONS 1 '(2))
(1 2)
```

(c) Adaugati la limbaj lista vida "()"

Hints:

- Incercati sa completati mai intai codul pentru obtinerea unui parser pentru subsetul de functii LISP date (fara actiuni corespondente productiilor). Mai apoi, adaugati actiuni pentru a obtine un interpretor.
- $'(1\ 2\ 3)$  e recunoscut ca  $form \Rightarrow Lform \Rightarrow '(enum)$