

Laboratorul 2.

Analiza Lexicală

Detalii tehnice

Pentru a rula laboratorul din linie de comandă folosim următorii pași:

```
$ export CLASSPATH=".:<path-către-antlr>/antlr-4.7.1-complete.jar:$CLASSPATH"
$ alias antlr4='java -jar <path-către-antlr>/antlr-4.7.1-complete.jar'
$ alias grun='java org.antlr.v4.gui.TestRig'
```

Pentru a testa codul:

```
# Generam codul Java pentru analizorul lexical
$ antlr4 Microcool.g4

# Obținem bytecode-ul Java
$ javac Microcool.java

# Rulăm analizorul lexical pe fișierul "sample-code.ucl" și
# direcționăm output-ul atât către fișierul temp.txt
# cât și pe consolă
$ grun Microcool tokens -tokens sample-code.ucl | tee temp.txt

# Comparăm output-ul dat de program cu referința
$ diff temp.txt reference.txt

# Rulăm analizorul lexical pe input-ul de la consolă.
# Pentru a trimite un EOF: Ctrl+D.
$ grun Microcool tokens -tokens -stdin
```

Exercitiul 1

Implementați analizorul lexical astfel încât să poată clasifica identificatori (nume de variabile și de funcții), numere întregi, operatori (aritmetici, de comparație, de atribuire), simbolul ";" și cuvintele cheie (**if**, **then**, **else**, **fi**, **true**, **false**).

Observații:

- Analizorul lexical va considera toți literalii întregi ca fiind numere pozitive. Semnul de "minus" va fi tratat doar ca operator aritmetic.
- Identificatorii încep cu litera mică, și pot conține apoi litere mici, litere mari și "underscore".

Exercitiul 2

Extindeți analizorul astfel încât să trateze corect numele de tipuri de date, operatorii de comparație, parantezele, acoladele, simbolul ":" și comentariile pe un singur rând.

Observații:

- Numele de tipuri de date încep mereu cu o literă mare. Verificarea validității tipurilor se va face în pasul de analiză semantică.
- Comentariile vor fi ignorate ("skip").

Exercitiul 3

Implementați tratarea literalilor cu virgula mobilă și a comentariilor bloc imbricate împerecheate.

Observații:

- Comentariile care nu au simbol-pereche (ex: "(* (* *)") sunt considerate erori lexice.
- Valoarea `Float` ".25" este echivalentă la nivel semantic cu "0.25". Valoarea "3." este echivalentă cu "3.0". Valoarea "." nu reprezintă un literal `Float` valid.