Fordítóprogramok beadandó

Simándi Balázs - FNG4AU

A feladat:

A feladat ugye az volt, hogy flex, illetve Bison használatával parse-oljunk egy nyelvet, számoljunk szintetizált attribútumokat és ezekkel kezdjünk valamit.

Az ötletem az volt, hogy készítek egy jsonlint.org-hoz hasonló JSON validatort, ami bemenetként kap egy JSON fájlt, a program leellenőrzi, hogy megfelel-e a json.org-on található specifikációnak, majd egy output fájlba kiírja a beolvasott objektum struktúráját, illetve az objektumról és az objektumban attribútum értékeként megadott objektumokról és tömbökről gyűjt össze különböző információkat, végül ezeket kiírja egy formázott output fájlba, hogy kicsit olvashatóbb legyen.

A fenti ötletemet sikerült is megvalósítani, bár a használt nyelvtant nem teljesen én találtam ki(remélem ez nem gond), hanem a json.org oldalon megtalálhatót alakítottam át olyan formára, hogy a flex is csináljon valamit.

A beadott fájlok:

./examples/*.json
./src/Parser.lex
./src/Parser.y
./src/compile.sh
./src/format.pl
./run.sh

Példa bemeneti fájlok Flex fájl, ebben vannak leírva a tokenek Bison fájl, ebben van a nyelvtan Bash script, összefordítja a flex/bison fájlt Perl script, ez írja ki a formázott kimenetet Bash script, ez kezeli a futást

Futtatás:

A run.sh script elvileg lekezel mindent, megkérdezi, hogy lefordítsa-e a parsert, bekéri a bemeneti fájl nevét, majd a kimeneti és a formázott fájl nevét is, ezután parse-olja a bemenetet, majd kiírja az eredményt az adott fájlokba.

Megjegyzés:

A dokumentációt leszámítva az egész beadandót egy néha elég érdekes dolgokat produkáló 64-bites Ubuntu virtuális gépen írtam, azon működött, remélem nálad sem lesz gond vele.

Példa a futtatásra:

```
× - D balazs@SANIC-PC: ~/Documents/dev/json-parser
balazs@SANIC-PC:~/Documents/dev/json-parser$ ./run.sh
Compile parser? [y/n]
Compiling...
AllSets.json sample2.json sample3.json sample4.json sample5.json sample.json
Name of the input file:
AllSets.json
Name of the output file(defaults to result.txt if left empty):
Name of the formatted output file(defaults to formatted_result.txt if left empty):
Parsing...
Writing formatted output...
Formatted output was written into formatted_result.txt
balazs@SANIC-PC:~/Documents/dev/json-parser$
 Formatted_result.txt (~/Documents/dev/json-parser) -
                                                       nts dev
                                                                json-parser
 Gen ▼ Save
                                 Undo A
formatted_result.txt x
      "common",
                                                                 docs
                                                                                 examples
      "common",
      "COMMON",
"COMMON",
"COMMON",
"COMMON",
"COMMON",
"COMMON",
                                                                                    "LE
                                                                              formatted result.
                                                                 SEC
                                                                                    txt
                                                                 # jso
      "common"
                                                                                   "LEA"
    ----- ARRAY END -----
    # of elements: 15
                                                              README.md
                                                                                 result.txt
   Array contains: 0 objects,0 arrays,15
strings,0 numbers,0 booleans,0 nulls
                                                                run.sh
    "cards" :
    ----- ARRAY BEGIN -----
      ----- OBJECT BEGIN -----
        "layout" : "normal",
        "type" : "Creature — Elemental", "types" :
    Plain Text ▼ Tab Width: 8 ▼ Ln 4, Col 35
                                     Connect to Server
                                                                        "docs" selected (containing 0 items)
```