

# Extraproblem about YACC/BISON-parser tool

Olkoon  $L$  kieli, jonka sanat koostuvat mistä tahansa tekstistä sekä laillisista sulkurakenteista. Ts. kuten ongelmassa 8, mutta nyt joukossa voi esiintyä tekstiä. Esimerkiksi lause ”(suuri) kissaeläin (joko leijona tai tiikeri { jotka harvinaisia } tai sapelihammastiikeri { joka sukupuuttoon kuollut [ks. tarkemmin Kurten] – harmi –} tai leopardi) on viehättävä (mutta vaarallinen!) tuttavuus.” kuuluu kieleen. Voit olettaa yksinkertaisuuden vuoksi, että tekstiosuudet sisältävät vain pieniä kirjaimia a..z sekä välilyöntejä.

Anna kielen kuvaava kontekstiton kielioppi ja laadi sille jäsentäjä UNIX:in `yacc`- tai `bison`-työkalulla!

Ohjeita löytyy esim.

[http://www.cs.princeton.edu/appel/modern/c/software/bison/bison\\_toc.html](http://www.cs.princeton.edu/appel/modern/c/software/bison/bison_toc.html)

Huom! `yacc` ja `bison` tuottavat c-koodisen jäsentäjän, johon sinun on itse lisättävä leksikaalinen analysaattori, pääohjelma ja jokin virheiden käsittelyrutiini.

Let  $L$  be a language, the words of which are constructed from any text and legal parentheses structures. I.e. as in the problem 8, but now there can be text in the middle of parentheses. E.g. sentence ”a (big) cat animal (either lion or tiger {which are rare } or saber-toothed tiger {which extincts [read further Kurten ] – shame –} or leopard) is an attractive (but dangerous!) friend” belongs to the language. You can suppose for simplicity that the text parts consist of only lowercase letters a..z and spaces.

Give a context-free grammar, which describes the language, and make it a recursive parser with UNIX-tool `yacc` or `bison`!

You can find instructions in e.g.

[http://www.cs.princeton.edu/appel/modern/c/software/bison/bison\\_toc.html](http://www.cs.princeton.edu/appel/modern/c/software/bison/bison_toc.html)

N.B.! `yacc` and `bison` produce a C-language parser, into which you should add a lexical analyzer, main program and some error handling routine.