Regex-tulkin määrittely

December 21, 2014

Toteutetaan säännöllisten lauseiden tulkkina toimiva ohjelma. Ohjelma ottaa syötteenä säännöllisen lausekkeen ja merkkijonon tai tiedostonimen, ja palauttaa ensimmäisen annetun säännöllisen lausekkeen toteuttavan osamerkkijonon, jos sellainen on olemassa. Käytettävänä ohjelmointikielenä on Haskell.

Tavoitteena on toteuttaa muutama eri säännöllisten lausekkeiden toteutuksessa käytettävä algoritmi ja vertailla näiden toimintaa ja tehokkuutta. Syötteenä saatu säännöllinen lauseke parsitaan ns. monadisella parsimisella[1]. Jokaista säännöllistä lauseketta vastaa epädeterministinen äärellinen tila-automaatti (nondeterministic finite-state automaton, NFA). Sen testaaminen, toteuttaako annettu merkkijono säännöllisen lausekkeen, toteutetaan kolmella tavalla. Ensimmäinen tapa on muodostaa säännöllisen lausekkeen NFA, ja muuttaa se deterministiseksi äärelliseksi tila-automaatiksi (deterministic finite-state automaton, DFA), ja laskea vastaus tuon DFA:n avulla. Toinen tapa on laskea vastaus suoraan NFA:n avulla pitämällä kirjaa jokaisessa vaiheessa mahdollisten tilojen joukosta. Kolmas tapa on hyödyntää säännöllisten lausekkeiden derivaatan käsitettä[2], mikä vastaa erään DFA:n johtamista suoraan säännöllisestä lausekkeesta. Myös muita algoritmeja saatetaan ottaa mukaan.

Käytettyjen algoritmien resurssivaativuudet vaihtelevat, mutta tavoitteena on voida päättää toteuttaako annettu merkkijono säännöllisen lausekkeen vai ei merkkijonon pituuden suhteen lineaarisessa ajassa.

[1] http://www.cs.nott.ac.uk/~gmh/pearl.pdf

[2] http://matt.might.net/articles/implementation-of-regular-expression-matching-in-scheme-with-derivatives