



Tutoriaal 7 Verdere oefeninge met LEX

Rekenaarwetenskap 324
Teoretiese Rekenaarwetenskap

28 April 2004

1 Inleiding

Gegee twee of meer programme, wil ons somtyds weet hoe die stel programme met mekaar ooreenstem. Verskeie redes bestaan hoekom ons wil weet hoe twee of meer programme met mekaar ooreenstem of anders gestel, van mekaar verskil. Een rede is om tussen verskillende weergawes van 'n program te onderskei. 'n Ander is om moontlike plagiaat op te spoor.

2 Handtekening van 'n program

Gestel daar bestaan tellers, t_i vir $i \in [1 \dots n]$ wat dui om die verskillende unieke dele van 'n program. Tabel 1 toon 'n lys van die belangrikste dele van 'n program, volgens [2]. 'n Handtekening van 'n program word dan definieer as

$$\text{handtekening} = \sum_{i=1}^n \alpha_i \times t_i \mod M \quad (1)$$

waar α_i 'n paslike gewig is vir elke teller en M 'n geskikte waarde is wat "goeie" handtekeninge lewer, [2]. 'n Stel "goeie" handtekeninge sal baie na aan mekaar lê vir programme wat min of meer dieselfde is en vër van mekaar vir programme wat baie verskil. Daar bestaan verskeie ander formules in die literatuur om goeie handtekeninge te verkry.

Verskillende id's / veranderlike name
Alle sleutelwoorde soos if en while
Operatore soos +, -, *, / en %
Heelgetalle
Aantal kommentaarstellings
Lengte van kommentaarstellings
Lengte van program

Tabel 1: Items in 'n program

3 Opdrag

U moet 'n program skryf deur van *Flex* gebruik te maak om die verskillende dele van 'n program, soos in tabel 1 uiteengesit, te tel. Dan deur van vergelyking (1) of enige ander sinvolle vergelyking gebruik te maak, moet u program 'n handtekening vir 'n C program bereken en dit aan die gebruiker toon. Die invoer vir u program is die naam van 'n enkele lêer as 'n bevellynargument tot u program. U kan dit egter uitbrei om meer as een program te aanvaar as toevoer.

Om u tyd te spaar met die tik van die sleutelwoorde, et cetera, kan u `readme.txt` vanaf die web aflaai. Hierin is die nodige sleutelwoorde, sowel as al die verskillende komponente wat u program moet kan herken. U hoef byvoorbeeld nie wisselpuntgetalle en C++ tipe kommentaar te herken nie. In `Tests.tar.gz` sal u 'n stel programme vind om u program mee te toets.

Belangrik

Hierdie is 'n opdrag waarin u *Flex* moet gebruik. Dit is dus nie aanvaarbaar om groot dele C kode te skryf wat as 'n leksikale ontleder sal optree nie. Die naam van u *Flex* spesifikasie, moet `signature.flex` wees. Die uitvoerbare program (executable) se naam moet `signature` wees. Plaas beide `signature` en `signature.flex` in `rw324/tut7` (**belangrik: alles klein letters**). Maak asseblief seker dat u program korrek vertaal. 'n Program wat nie vertaal nie, gaan vir u nul punte beteken. **Datum en tyd vir inhandiging: Woensdag 5 Mei om 12:00.**

4 Bonuspunte

U kan bonuspunte vir die volgende ontvang:

- Die gebruik van 'n formule wat 'n beter handtekening kan berekening as vgl. 1.
- As u enige ander dele van 'n program herken wat nie gespesifieer word nie, byvoorbeeld wisselpuntgetalle.

Verwysings

- [1] Clough P, *Plagiarism in natural and programming languages: an overview of current tools and technologies*, Universiteit van Sheffield, (Julie 2002)
- [2] Dodds, R, *Prakties 2, Rekenaarwetenskap 354*, (7 Augustus 2002)
- [3] Hoad T, Zobel J, *Methods for Identifying Versioned and Plagiarised Documents*, RMIT University