Universiteit van Amsterdam

Programmeertalen

Technisch rapport Bash

18 februari 2018

Student:
Furkan Simsir
11336900

Tutor:
Willem Vermeulen

Practicumgroep:
A1 ALGOL

Cursus:
PAV

Inhoudsopgave

1 Inleiding

Voor week 2 in het vak programmeertalen is het de bedoeling om een spel te maken in de taal Bash. Het doel van het spel is om een string die bestaat uit a's, b's en c's om te zetten naar een andere string. Om bij deze string te komen moet je stappen zetten waaraan kosten zijn verbonden. Het doel is om de kosten zo min mogelijk te houden om het spel uit te spelen.

De hele opdracht begint eerst met 7 kleine puzzels om aan de taal te wennen. In deze puzzels leer je onder andere:

- Input van de gebruiker vragen en hergebruiken
- Strings filteren en hergebruiken
- Hoe functies werken
- Hoe if-else statements werken

Hierna begin je met het implementeren van de benodigde functies om het uiteindelijke programma te schrijven.

- Je begint met het maken van een zet. Hierbij word een blok van letters geroteerd van plaats in een string en van plaats in het alfabet (a's worden b's, b's worden c's en c's worden weer a's).
- Daarna ga je verder met het selecteren van een blok in een string met een gegeven addressering. Dus de adresstring a kan worden gebruikt om het eerste blok aa in aabbaa te selecteren: **aa**bbaa.
  Adres-string **ba** en **aa** selecteren beide: aabbcc**aa**bbcc.
- Vervolgens bereken je de kosten van elke zet die je maakt. Deze kosten zijn exponentieel met de vorm: 2^x waarbij x de lengte is van het aantal zetten.
- Je gaat verder door het programma te verwijzen naar een bestand waar de missie staat die voltooid moet worden.
- Als laatst moet de gebruiker de huidige stand kunnen saven om op een later moment verder te gaan.

  Dit allemaal komt samen in de uiteindelijke opdracht en een interface die de gebruiker begeleid in het spel. Alle aparte functionaliteit wordt hierbij gekoppeld tot een interface. Maar hoe wordt dit het meest efficient geintegreerd?

2 Methode

2.1 Zet

Zoals beschreven in de opdracht is dit de meest ideale oplossing voor dit pronleem:

aabbaa		(origineel)
aaaa	bb	(knip)
aaaa	cc	(roteer)
aaaacc		(plak)

Wat ik dus eerst heb gedaan is het block selecteren, apart nemen en roteren:

if block is gelijk aan'c' **then**
  return het karakter 'a'
else
  verplaats het karakter met 1 plaats in het alphabet
end if

Vervolgens heb ik dit achter de string geplakt waaruit de block al was gefilterd.

2.2 Addressering

Bij het adresseren ben ik begonnen om eerst een block te selecteren met één addressering. Het plan is om daarna dezelfde functie recursief toe te passen op meerdere adresseringen. Bij elke block heb ik eerst de gegevens verzameld van het stukje string voor en na de block. Zo kon ik bepalen welk stukje daadwerkelijk mijn block was.

if de addressering voorkomt in de string **then**
  doe wat ik hierboven heb beschreven
else
  return de string zonder block
end if

Dit algoritme heb ik dus in een functie gestopt. Deze functie returnt de string voor de block, de block zelf en de string na het blok.
Vervolgens itereer ik over elke character van de adressering om vervolgens dezelfde functie toe te passen met steeds de string na het block als nieuwe input:

for (i=0; i¡(length van de addressering); i++); **do**
Voer de functie uit met als invoer:

- 'i'e plaats in de addressering
- de string na de block.

end for-loop
Print vervolgens het stukje voor de blok, de blok zelf onderstreept en het stukje na de blok aan elkaar.

2.3 Kosten

Dit stuk code bestaat uit een enkele functie dat een berekeing voert met invoer.
De functie neemt als invoer x en voert de formule 2^x uit. Hierbij is x de lengte van de adressering.

3 Metingen

De metingen zijn niet perfect, maar werken grotendeels wel. Zo werkt de adressering niet naar behoren wanneer de block aan het einde van de string voorkomt.

4 Conclusies

De opdracht is niet helemaal af. Tot de kosten is geimplementeerd maar daarna niet. Vandaar dat ik geen complete conclusie kan trekken. Recursief heb ik de adreseringsmethode proberen te implementeren. Dat is op enkele fouten na goed gelukt.