

Projekt nr 1
Specyfikacja implementacyjna
ATG - Automat Text Generator

Cezary Wolszczak
271100 gr.3

29 marca 2015

Spis treści

1	Informacje ogólne	2
1.1	Podstawowe informacje	2
1.2	Uruchamianie programu	2
2	Opis modułów/pakietów	2
2.1	czytnik.c	2
2.2	atg.c	2
2.3	drzewo.c	3
3	Opis struktur	3
3.1	struct _atg_t	3
3.2	struct _n_gram	3
3.3	struct _drzewo_t	4
4	Opis funkcji	4
4.1	czytnik.c	4
4.2	atg.c	5
4.3	drzewo.c	5
5	Testowanie	6
5.1	Użyte narzędzia	6
6	Github adress	6
7	Diagram	6

1 Informacje ogólne

1.1 Podstawowe informacje

Program będzie uruchamiany za pomocą linii komend w terminalu w systemach operacyjnych z rodziny Unix. Dane wyjściowe będą zaś częściowo zapisywane do plików tekstowych a częściowo wypisywane na wyjście programu również w oknie terminala.

1.2 Uruchamianie programu

Uruchamianie:

```
./ATG -p plik1.txt plik2.txt plik3.txt ...  
./ATG -d plik1.txt plik2.txt plik3.txt ...  
./ATG -m plik.txt -g -s/-a [liczba słów/akapitów] -n [n-gramy] -l [ilość n-gramów]  
./ATG -g -s/-a [liczba słów/akapitów] -n [n-gramy] -l [ilość n-gramów]
```

Legenda:

- p pobiera do programu pliki z tekstem bazowym (w formacie txt) domyślnie czyta z (../dane/cały_tekst.txt)
- d dopisanie danych do pliku pośredniego
- m pobiera plik pośredni (w formacie txt)
- g generuje tekst
- s parametr służący do generacji tekstu o zadanej przez użytkownika liczbie słów
- a parametr służący do generacji tekstu o zadanej przez użytkownika liczbie akapitów
- n rząd n-gramu (domyślnie 2)
- l liczba wypisanych n-gramów (domyślnie 100)
- h pomoc

2 Opis modułów/pakietów

2.1 czytnik.c

Moduł ten ma za zadanie czytanie danych z plików tekstowych, liczenie słów oraz indeksowanie wszystkich słów po czym zapisywanie ich, słowo po słowie w strukturze danych.

2.2 atg.c

Moduł ma za zadanie losować prefiksy oraz dobierać do nich odpowiednie sufiksy zgodnie z wcześniej wyliczonego prawdopodobieństwa. Moduł

ten pobiera dane ze struktury stworzonej uprzednio z czytnika.c po czym zmienione zapisuje je do innej struktury zawierającej prefiksy, sufiksy oraz prawdopodobieństwo wystąpienia.

2.3 drzewo.c

Moduł ma za zadanie w szybki sposób liczyć wystąpienia wszystkich słów, prefiksów oraz sufiksów.

3 Opis struktur

3.1 struct _atg_t

```
typedef struct _atg_t
    char *slovo;
    int index;
    int zliczenia;
    int l_liter;
atg_t;
```

char *slovo	przechowuje słów z wejścia
int index	numer kolejnego słowa
int zliczenia	ilość takich samych słów
int l_liter	liczba liter w slowie

3.2 struct _n_gram

```
typedef struct _n_gram
    char **prefiks;
    char *sufiks;
    int zliczenia_prefiks;
    int zliczenia_sufiks;
    int prawdopodobienstwo;
    int pmi;
n_gram;
```

char **prefiks	tablica tablic prefiksów
char *sufiks	tablica sufiksów
int zliczenia_prefiks	zliczenia prefiksów
int zliczenia_sufiks	zliczenia sufiksów
int prawdopodobienstwo	prawdopodobieństwo wystąpienia n_gramu
int pmi	pointwise mutual information

3.3 struct _drzewo_t

```
typedef struct _drzewo_t
{
    n_gram *gram_drzewo;
    struct _drzewo_t *lewy;
    struct _drzewo_t *prawy;
} wezel_t, *drzewo_t;
```

n_gram *gram_drzewo	wskaźnik na strukture n_gram
struct _drzewo_t *lewy	wskaźnik na lewą gałąź
struct _drzewo_t *prawy	wskaźnika na prawą gałąź

4 Opis funkcji

4.1 czytnik.c

- **atg_t *czytaj_slowa(FILE *in, int *l_slow, int *start_size, atg_t *atg)**

Czyta słowa z pliku, liczy je, oraz zlicza wystąpienia. Zwraca wskaźnik do struktury.

FILE *in	wskaźnik na plik do czytania
int *l_slow	liczba kolejnego przeczytanego słowa
int *start_size	początkowa wielkość tablicy struktury
atg_t *atg	wskaźnik na strukture

- **void zapisz_slowa(atg_t *atg, FILE *in, int *l_slow)**

Zapisuje słowa do pliku.

atg_t *atg	wskaźnik na strukture
FILE *in	wskaźnika na plik do zapisu słów
int *l_slow	liczba słów do zapisu

- **atg_t stworz_slowo(atg_t *atg, int liczba_liter)**

Alokuje komórki pamięci na poszczególne słowa. Zwraca strukture.

atg_t *atg	wskaźnik na strukture
int liczba_liter	liczba liter w słowie

- **atg_t *realloc_tab_slow(atg_t *atg, int size)**

Reallokuje pamięć tablicy struktur. Zwraca wskaźnik na tablice.

atg_t *atg	wskaźnik na strukture
int size	nowa wielkość tablicy struktur

- **void zwolnij_pamiec(atg_t *atg, int l_slow)**

Zwalnia pamięć tablicy struktur.

atg_t *atg	wskaźnik na strukture
int l_slow	liczba wszystkich słów

4.2 atg.c

- **n_gram stworz_n_gram(n_gram *gram, atg_t *atg, int ile_slow, int nr_n_gramu, int los);**

Losuje prefiks i na podstawie prawdopodobieństwa dobiera do niego sufixs.

n_gram *gram	wskaźnik na strukture _n_gram
atg_t *gram	wskaźnik na strukture _atg_t
int ile_slow	liczba wszystkich słów
int nr_n_gramu	indeks tabliny struktur _n_gram
int los	wylosowany prefiks

- **n_gram stworz_prefiks(n_gram *gram, int l_liter_prefiks, int ile_slow)**

Alokuje pamięć dla tablicy tablic prefiksu. Zwraca strukture.

n_gram *gram	wskaźnik na strukture _n_gram
int l_liter_prefiks	liczba liter w prefiksie
int nr_n_gramu	indeks tabliny struktur _n_gram
int los	wylosowany prefiks
int ile_slow	ile słów w danym prefiksie

- **n_gram stworz_sufiks(n_gram *gram, int l_liter_sufiks)**

Alokuje pamięć dla tablicy sufiksów. Zwraca strukture.

n_gram *gram	wskaźnik na strukture _n_gram
int l_liter_prefiks	liczba liter w sufiksie

- **void zwolnij_n_gram(n_gram *gram, int wielkosc_tab_strukt, int rzad_n_gramu)**

Zwalnia pamięć tablicy struktur.

n_gram *gram	wskaźnik na strukture _n_gram
int wielkosc_tab_strukt	wielkość tablicy struktur
int rzad_n_gramu	rząd n_gramu

4.3 drzewo.c

- **drzewo_t wstaw_do_drzewa(drzewo_t t, n_gram *gram, int _rzad_n_gramu, int cmp(const void *, const void *, int rzad_n_gramu));**

Zlicza występowanie słów, prefiksów oraz sufiksów.

drzewo_t t	deklaracja drzewa
int _rzad_n_gramu	rząd n_gramu
n_gram *gram	wskaźnik na strukture _n_gram
int cmp	porównywarka słow n_gramu

5 Testowanie

5.1 Użyte narzędzia

Do testowania posłużą między innymi debuggera "GNU debugger" (GDB), oraz valgrind w celu sprawdzenia wycieków pamięci. Program będzie testowany ręcznie.

6 Github adress

<https://github.com/wolszczc/ATG.git>

7 Diagram

