

Коротка інструкція щодо користування програмами пакету phonetic-languages-simplified-examples-array

Автор та розробник ПЗ: Олександр Сергійович Жабенко

Ліцензія: MIT

Copyright (c) 2020-2021 Oleksandr Zhabenko

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Вступ

Існують різні мови. У них є різна структура та правила. При цьому існує можливість створити та використовувати (на основі однієї з існуючих широко вживаних та добре розповсюджених мов, зокрема української у цій роботі) “фонетичну” мову, яка краще підходить для поезії та музики. Можливо навіть створити різні варіанти фонетичної мови. Ця робота пропонує створити кілька різних фонетичних мов на основі української.

Уявіть, що ви можете розуміти інформацію в тексті незалежно від порядку слів і при збереженні лише найбільш необхідної граматики (наприклад, правило не відокремлювати прийменник та наступне слово збережене). Розуміти точно так само, як читати текст (після деякого навчання та тренування, можливо), в якому у словах збережені на своїх позиціях лише перші та останні літери, а всі решта – взаємно перемішані одна з одною. Отже, уявіть, що ви можете розуміти (і виражати ваші думки, почуття, мотиви тощо) повідомлення тексту без дотримання строгого порядку слів.

У такому випадку ви можете впорядкувати слова (зберігши найбільш необхідну граматику для зменшення чи повного усунення можливої двозначності, зумовленої граматиною, точніше зменшенням її обсягу), розмістивши їх таким чином, щоб вони забезпечили собою більш цікаве фонетичне звучання. Ви можете спробувати створити поетичний (чи принаймні дещо більш ритмічний та виразний) текст чи музику. Це також може бути саме по собі розвивальною справою, яка надихає. Але як би ви могли швидко знайти, які комбінації більш чи менш підходять? Крім того, чи може складність алгоритмів бути зменшена?

Це лише деякі з цікавих питань. Ця праця на даний момент не дає повної відповіді на них, але є експериментальною та дослідницькою, при цьому, звичайно, будь-який результат її є цінним.

Українська є мовою без строгих вимог до порядку слів у реченні (хоча є певні усталені переважні варіанти) і має приємне звучання. Отже, вона може бути гарним прикладом та зразком. Крім того, для автора програм це рідна мова.

Навіть якщо ви не бажаєте створити та використовувати “фонетичні” мови, де фонетика є більш важливою, ніж граматика, і тоді ви можете оцінити фонетичний потенціал слів, використаних у тексті, для продукування спеціальним чином озвучених текстів. Це також може бути цінним та помічним у написанні поезії та можливих інших пов’язаних областях.[\[2\]](#)

Передумови користування пакетом програм

Поки програми працюють для робочих станцій (десктопів, desktop, working station тощо), і немає мобільних версій.

Потрібно, щоб були встановлені та налаштовані програми мовою Haskell:

- 1 [GHC](#) (версії не раніше 7.10)
- 2 [Cabal](#)

Виконувані файли цих програм мають бути доступні для пошуку через змінну робочого середовища PATH (це типова їх установка).

Якщо є змога, встановіть за допомогою системного менеджера пакетів (програм) також важливі пакети Haskell [bytestring](#), [vector](#), [heaps](#), [parallel](#). Якщо Ви плануєте користуватися також [r-glpk-phonetic-languages-ukrainian-durations](#), то встановіть також мову програмування (а краще середовище розробки) [R](#).

Якщо потрібні згадані пакети мовою Haskell не встановлено за допомогою системного менеджера, при встановленні пакетів вони будуть завантажені та встановлені автоматично, при цьому додатковий час піде також на їх компіляцію.

Ремарка щодо термінології

У раніших версіях пакетів використовувалися назви “норми” та “метрики” щодо властивостей текстів. Оскільки у значенні типовому для математики (зокрема функціонального аналізу) усі вказані властивості не є власне метриками і нормами (зокрема не виконується нерівність трикутника), то далі всюди буде використано замість слова “метрика” відповідно недвозначне “властивість”, маючи на увазі функціональне представлення останньої.

Встановлення пакету

Відкрийте командний рядок чи термінал і введіть як команди:

```
cabal update
```

```
cabal --reinstall install phonetic-languages-plus
```

```
cabal --reinstall --force-reinstalls install phonetic-languages-simplified-examples-array
```

Також додатково рекомендується встановити наступні пакети:

```
cabal --reinstall --force-reinstalls install r-glpk-phonetic-languages-ukrainian-durations
```

```
cabal --reinstall --force-reinstalls install mmsynbukr-array
```

 (ця остання є опціональною, але корисною для озвучування і не займає багато місця)

Якщо є повідомлення про застарілі варіанти команд, введіть замість update – v1-update, install – v1-install.

Робота з програмою **lineVariantsG3**

Перевірте, щоб папка (каталог), куди cabal встановив виконувані файли програм, була доступна для пошуку в змінній середовища PATH.

Програма зараз підтримує два режими роботи:

- з однією властивістю (звичайний режим);
- [з кількома \(не більше п'яти різних\) властивостями](#).

Останній використовується, якщо серед аргументів командного рядка є група, введена розділювачами +M <тип властивості1> <числові аргументи1> <тип властивості2> <числові аргументи2> <тип властивості3> <числові аргументи3> <тип властивості4> <числові аргументи4> <тип властивості5> <числові аргументи5> -M. Більше про нього у відповідному розділі (дивіться посилання вище).

Робота програми у такому режимі описується далі в окремому розділі.

Для роботи у режимі однієї властивості введіть у командному рядку (чи терміналі) команду:

```
lineVariantsG3 <перший аргумент> [<чи друкувати значення властивості\(ей\)> <чи останнє слово має залишатися на своєму місці>] <числові аргументи> <тип властивості> <український текст>
```

або:

```
lineVariantsG3 <перший аргумент> [<чи друкувати значення властивості\(ей\)> <чи останнє слово має залишатися на своєму місці>] <числові аргументи> <тип властивості> <український текст> <десь серед аргументів як єдина група: обмеження>
```

та натисніть Enter. Додатково можна задавати інтерактивний режим, про що [детальніше дивіться далі](#).

Якщо не задавати групи у квадратних дужках, Ви побачите щось на зразок наступного:

lineVariantsG3 10.0_1.2 yu2 садок вишневий коло хати хрущі над вишнями гудуть (введений український текст в кінці команди – уривок з відомого вірша Тараса Григоровича Шевченка; загалом цей рядок – це введена команда)

садок колохати хрущі гудуть надвишнями вишневий (варіант (загалом може бути кілька таких варіантів, які утворюють одну групу, а також кілька таких груп; усі групи йдуть одна за одною зверху донизу у порядку зменшення кінцевого значення властивості), який максимізує обрану властивість для заданих інтервалів)

[3.6562] (значення обраної властивості до застосування перетворення інтервалів)

[3.6562] (значення обраної властивості після перетворення інтервалів, кінцеве значення властивості для цього рядка)

Зверніть увагу, що текст може (і переважно буде) писатися не так, як він пишеться згідно правил орфографії та пунктуації, але ви можете його прочитати та спробувати зрозуміти. Змінюючи введені перші аргументи, ви (скоріше всього) отримуватимете інші вихідні дані, те ж, вочевидь, стосується й українського тексту. Занадто довгий текст буде скорочений до об'єму, який ви змогли б зрозуміти (можливо, після згаданого раніше тренування) без надзусиль.

Спробуйте оцінити, прочитавши варіант, наскільки він підходить.

УВАГА: Також потрібно пам'ятати, що в режимі однієї властивості числові аргументи передують позначенню властивості, а в режимі кількох властивостей (див. далі) навпаки – позначення властивості починає набір числових аргументів, які стосують її, якщо такі є (інакше використовуються типові значення, які аналогічні пошуку максимального елемента).

Українські інформаційні повідомлення

Щоб програма при роботі виводила інформаційні повідомлення українською мовою (за замовчуванням виводить англійською), потрібно вказати як один з аргументів командного рядка “+U” деś не всередині груп опцій, наприклад, напочатку. Можна також налаштувати аліас для такого варіанту роботи програми, які і для інших варіантів, якщо Вам подобається. За детальною інформацією зверніться до документації командних оболонок.

Більш комплексне використання

Числові аргументи, якщо задані, мають наступне значення.

Перший числовий аргумент – кількість груп з однаковими максимальними значенням властивості (в порядку зменшення), які будуть виведені на екран як результат. Якщо задано більшу кількість, ніж їх є взагалі, то виводяться всі можливі результати, які задовольняють усі інші умови. Якщо не задано числові аргументи, то вважається рівним 1.

Другий числовий аргумент – кількість інтервалів, на які поділяється проміжок між мінімальним та максимальним значенням властивості для даного рядка. Якщо не задано, вважається рівним 1. Значення 0 не дає змоги іншим числовим аргументам далі змінювати результат роботи програми.

Усі наступні числові аргументи (якщо задано, інакше ніяких перестановок не відбувається) – номери інтервалів, які будуть поміняні місцями з максимальним за номером. Це дозволяє змінити структуру даних, які відображаються як результат роботи програми і побачити внутрішні (не максимальні) елементи. Наприклад, числові аргументи 2 6 1 4 (у такому порядку) означатимуть, що в ході виконання програма поверне 2 групи елементів з максимальними значеннями властивості (найбільшим і найбільшим наступним за попереднім), отриманим після перестановки інтервалів; відрізок між максимальним та мінімальним значенням властивості буде поділено на 6 рівних інтервалів, при цьому елементи, які знаходяться у першому та 4, рахуючи від мінімального (інтервал з номером 1) виведе максимальні 2 групи елементів.

Значення, які були в максимальному інтервалі, будуть переміщені в інтервал з найменшим номером серед тих, які переміщені в максимальний. Таким чином, при виводі ці значення будуть виведені найпізніше.

Параметр ++L (++BL) та його використання

УВАГА: Якщо не було серед аргументів командного рядка символів ++L, ++BL, ++I, +IF ... -IF, то після кожного рядка буде відображено 2 числа у квадратних дужках – первинне значення властивості (без переміщення інтервалів) і значення після переміщення. Якщо був (хоча б) один з цих (груп) символів – значення властивостей друкуватися не буде.

Також слід пам'ятати, що:

++BL == ++V ++L (це просто скорочення використання відразу обох параметрів, замість 7 символів потрібно ввести лише 4).

Якщо Ви вказуєте також +IF ... -IF або ++I, тоді цей параметр можна не вказувати (його буде застосовано автоматично), натомість, якщо бажаєте, можете замість нього задати додатково ++V.

Параметр ++V (++BL) та його використання

Якщо де-небудь серед аргументів командного рядка вказати аргумент у вигляді ++V (або ++BL), то програма збереже при виведенні та аналізі останнє слово у рядку на своєму місці – це дуже зручно, коли потрібно, маючи риму, підібрати інші слова. Якщо не вказувати, то усі слова будуть переміщуватися (при необхідності). Робота параметра реалізована фактично як додаткове [обмеження \(constraint\)](#), дивіться нижче. Ви можете також додатково задавати інші обмеження.

++BL == ++V ++L (це просто скорочення використання відразу обох параметрів, замість 7 символів потрібно ввести лише 4).

Про використання інших параметрів трохи згодом.

Режим кількох властивостей (+M . . . -M)

Якщо серед аргументів командного рядка задати групу аргументів, виділену розділювачами +M та -M, так щоб група аргументів виділена розділювачами +A та -A не була всередині цієї, і навпаки (щоб вони не перетиналися), тоді програма працюватиме в режимі кількох властивостей. Значення [властивостей](#) виводитися на екран не будуть, натомість є можливість задати не більше чотирьох різних властивостей та до кожної з них вказати аргументи (дивіться: [Більш комплексне використання](#)). Програма тоді знайде варіанти, які задовольняють кожну з вказаних умов, а потім виведе на екран лише ті варіанти, які зустрічаються у всіх обраних і заданих властивостях з параметрами. Числові аргументи, які стоять після позначення властивості і передують наступному позначенню властивості, відносяться до цієї властивості. Якщо числові аргументи опущено, то використовуються значення за замовчуванням (фактично це еквівалентно простому пошуку максимальних значень властивості).

Загалом, це найбільш комплексне використання даної програми.

Спробуйте, наприклад, задати:

```
lineVariantsG3 +M 02y 3 03y 3 y0 10 -M ++BL <український текст>.
```

Інтерактивний режим (++I) та його використання

Інтерактивний режим (додаткова розширена взаємодія з користувачем, крім необхідної) вмикається і відповідно задається аргументом командного рядка “++I”, який можна ставити будь-де у рядку команди. У такому випадку програма виводить на екран не просто рядки, які задовольняють усі умови, але для кожного рядка виводить також його порядковий номер (починаючи з 1) в

порядку посилення “слабкості” виконання усіх умов (чим більший номер, тим, у загальному випадку, більш імовірним є слабший прояв заданих умов, хоча це не завжди так – зокрема коли потрібно вивести лише одну групу). Після того запитує, яким є вибір користувача і очікує на номер варіанту, введений користувачем. Після чого повертає той варіант без номеру.

Виглядає приблизно так:

```
lineVariantsG3 ++I +M 02y 10 0y 10 y0 50 y2 40 -M садок вишневий коло хати хрущі над вишнями гудуть
Please, check whether the line below corresponds and is consistent with the constraints you have
specified between the +A and -A options. Check also whether you have specified the "++B" or "++BL"
option(s). If it is inconsistent then enter further "n", press Enter and then run the program again
with better arguments. If the line is consistent with your input between +A and -A then just press
Enter to proceed further.
```

садок вишневий колохати хрущі надвишнями гудуть

- 1 вишневий колохати хрущі надвишнями садок гудуть
- 2 вишневий колохати садок хрущі надвишнями гудуть
- 3 вишневий колохати хрущі садок надвишнями гудуть
- 4 вишневий садок колохати хрущі надвишнями гудуть
- 5 колохати вишневий надвишнями садок хрущі гудуть
- 6 хрущі садок вишневий колохати надвишнями гудуть
- 7 колохати хрущі садок вишневий надвишнями гудуть
- 8 хрущі колохати садок вишневий надвишнями гудуть
- 9 хрущі садок колохати вишневий надвишнями гудуть
- 10 хрущі надвишнями садок вишневий колохати гудуть
- 11 колохати хрущі надвишнями садок вишневий гудуть
- 12 надвишнями колохати хрущі садок вишневий гудуть
- 13 надвишнями садок колохати хрущі вишневий гудуть

Please, specify the variant which you would like to become the resulting string by its number.

4

вишневий садок колохати хрущі надвишнями гудуть

Кольори не екран не виводяться, тут відображено для більшої наочності.

Інтерактивний режим запису рядка у файл (+IF ... -IF)

Якщо задати серед аргументів групу з трьох у вигляді +IF <шлях до файлу запису> IF, то у вказаний шлях, якщо є змога, буде дописано фінальний результат роботи програми, окрім того, що він як і в звичайному інтерактивному режимі буде виведений на екран. Ця група може стояти будь-де серед аргументів командного рядка виклику програми, але не повинна міститися всередині інших групу виду +A ... -A, +M ... -M тощо.

У результаті може вийти щось подібне (кольори додано для більшої наочності):

```
lineVariantsG3 +IF hello.txt -IF ++BL +M 02y 10 0y 10 y0 50 y2 40 -M садок вишневий коло хати хрущі  
над вишнями гудуть
```

Please, check whether the line below corresponds and is consistent with the constraints you have specified between the +A and -A options. Check also whether you have specified the "++B" or "++BL" option(s). If it is inconsistent then enter further "n", press Enter and then run the program again with better arguments. If the line is consistent with your input between +A and -A then just press Enter to proceed further.

садок вишневий колохати хрущі надвишнями гудуть

```
1 вишневий колохати хрущі надвишнями садок гудуть
2 вишневий колохати садок хрущі надвишнями гудуть
3 вишневий колохати хрущі садок надвишнями гудуть
4 вишневий садок колохати хрущі надвишнями гудуть
5 вишневий садок хрущі надвишнями колохати гудуть
6 хрущі садок вишневий колохати надвишнями гудуть
7 садок хрущі вишневий надвишнями колохати гудуть
8 колохати хрущі садок вишневий надвишнями гудуть
9 надвишнями колохати хрущі вишневий садок гудуть
10 хрущі колохати садок вишневий надвишнями гудуть
11 хрущі садок колохати вишневий надвишнями гудуть
12 хрущі надвишнями садок вишневий колохати гудуть
```

13 колохати надвишнями садок хрущі вишневий гудуть
14 колохати хрущі надвишнями садок вишневий гудуть
15 колохати хрущі садок надвишнями вишневий гудуть
16 надвишнями колохати садок хрущі вишневий гудуть
17 надвишнями колохати хрущі садок вишневий гудуть
18 хрущі колохати надвишнями садок вишневий гудуть
19 хрущі колохати садок надвишнями вишневий гудуть
20 надвишнями садок колохати хрущі вишневий гудуть
21 садок надвишнями колохати хрущі вишневий гудуть
22 хрущі надвишнями колохати садок вишневий гудуть
23 надвишнями садок хрущі колохати вишневий гудуть
24 хрущі надвишнями садок колохати вишневий гудуть
25 садок хрущі надвишнями колохати вишневий гудуть

Please, specify the variant which you would like to become the resulting string by its number.

4

вишневий садок колохати хрущі надвишнями гудуть

Цей крайній рядок у виводі програми також буде дописаний (appended) до файлу зі вказаною назвою, якщо є змога для даного користувача.

Якщо бажаєте, можна запустити команду повторно з новим текстом та / або новими аргументами. Якщо буде таким же чином вказаний той же файл, тоді новий результат буде дописаний далі у той же файл. Це дає змогу, послідовно застосовуючи цю програму, писати чи переписувати тексти (наприклад, вірші).

Режим одночасних можливих варіацій тексту

Починаючи з версії 0.3.0.0 додана можливість опрацьовувати відразу кілька варіацій тексту, зокрема такі, що відрізняються синонімами, перефразуваннями тощо.

Для цього використовуйте замість простого тексту як крайні аргументи наступну спеціальну конструкцію:

```
{ <варіант1 українського тексту> / <варіант2 українського тексту> / ... / <варіантN українського тексту> }
```

з хоча б двома варіантами всередині фігурних дужок. Ці варіанти будуть опрацьовуватися по черзі кожен зокрема в ході одного виклику програми, і Вами буде обраний один з варіантів (можливо, порожній). У кінці буде можливість обрати серед цих попередньо підготованих варіантів один-єдиний, який і буде результатом (і відповідно, наприклад, буде виведений на екран та записаний у файл, якщо це забезпечено аргументами командного рядка).

Будь ласка, пам'ятайте, що програма у такому режимі забезпечує обробку кожної з можливих комбінацій варіацій, і отже, якщо Ви вказали багато їх (наприклад, 3 варіації на одне слово та 4 на інше створять $3 \times 4 = 12$ варіацій, які будуть послідовно опрацьовані), поки Ви отримаєте кінцевий результат.

Робота з програмою **propertiesTextG3** (та **distributionTextG**)

I варіант (лише рядки)

Перевірте, щоб папка (каталог), куди sabal встановив виконувані файли програм, була доступна для пошуку в змінній середовища PATH.

Далі введіть у командному рядку (чи терміналі) команду:

```
propertiesTextG3 <перший аргумент> <файл з українським текстом> <контроль кількості інтервалів>  
<контроль друку також рядка тексту> <контроль розбивки тексту на рядки> <тип властивості>
```

та натисніть Enter.

Ви побачите щось схоже на:

```
propertiesTextG3 2.1_3.0 ~/sajok.txt s 1 0 04z
```

```
5  
2    2    11    1.0000    5.6200    5.6200    0.30211480    2    1    Тарас ШЕВЧЕНКО  
Вказематі
```

81	83	227	1.0222	2.7902	2.7297	0.53936831	3	1	Садок вишневий колохати
4	10	10	2.2936	2.2936	1.0000	1.39275766	3	5	Хрущі надвишнями гудуть
0	4	7	16.4444	26.2400	1.5957	1.20737478	3	4	Плугатарі зплугами йдуть
36	81	82	2.2345	2.2621	1.0123	1.36997886	3	5	Співають ідучи дівчата
4	4	16	1.1111	4.0625	3.6562	0.43895748	3	1	Аматері вечерять ждуть
81	144	146	1.7778	1.7942	1.0092	1.27248256	3	5	Сем'я вечеря колохати
5	9	16	1.8000	3.2000	1.7778	0.85714286	3	2	Вечірня зіронька встає
1	9	9	8.7059	8.7059	1.0000	1.79393939	3	5	Дочка вечерять подає
36	146	147	4.0625	4.0711	1.0021	1.60221297	3	5	Амати хоче научати
0	4	14	9.0000	32.0400	3.5600	0.54479419	3	2	Так соловейко недає
81	82	145	1.0014	1.7804	1.7778	0.72037537	3	1	Поклала мати колохати
4	4	10	1.0625	2.3906	2.2500	0.62672811	3	1	Маленьких діточок своїх
4	5	16	1.2162	3.9527	3.2500	0.49113233	3	1	Сама заснула колойіх
36	36	91	1.0038	2.5266	2.5170	0.56929225	4	1	Затихло все тільки дівчата
0	0	4	1.0000	9.0000	9.0000	0.20000000	2	1	Тасоловейко незатих

Міжтравня
С-Петербург

Числові стовпчики мають однакове значення для обох варіантів для рядків. Відмінність полягає у тому, що у другому випадку статистика по всьому тексту має більше значення з погляду дослідника (дослідниці), ніж по кожному рядку зокрема.

I стовпчик – це мінімально можливе значення обраної властивості для заданих даних серед усіх можливих варіантів перестановок слів у рядку;

II стовпчик – це актуальне значення обраної властивості для заданих даних у рядку, те, яке реалізується у конкретно цьому варіанті рядка;

III стовпчик – це максимально можливе значення обраної властивості для заданих даних серед усіх можливих варіантів перестановок слів у рядку;

IV стовпчик – це відношення значення властивості для даного рядка та її мінімального значення для слів, з яких складається рядок; число, яке не менше 1.0;

V стовпчик – це відношення максимального значення властивості для слів даного рядка та її мінімального значення, з яких складається рядок; число, яке не менше 1.0 і не менше числа у IV стовпчику;

VI стовпчик – це відношення максимального значення властивості для даного рядка та її актуального значення; число, яке не менше 1.0 і не більше значення у V стовпчику;

VII стовпчик – це відношення актуального значення властивості до середнього арифметичного (півсуми) максимального та мінімального значення для всіх можливих перестановок слів при заданих даних; число, яке відображається з повною обчисленою кількістю знаків після коми; має важливе значення для подальшої статистики для всього тексту;

VIII стовпчик – кількість слів у рядку, деякі з яких можуть складатися з кількох з'єднаних українських слів;

IX стовпчик – номер інтервалу (починаючи з 1), до якого входить актуальне значення властивості для заданих даних;

Далі направо – якщо вказано <контроль друку також рядка тексту> як “1”, тоді тут виводиться рядок тексту, який аналізується; інакше, ці дані не виводяться.

II варіант – статистика по всьому тексту (+ можливо, рядкова)

Перевірте, щоб папка (каталог), куди сабал встановив виконувати файли програм, була доступна для пошуку у змінній середовища PATH.

Далі введіть у терміналі команду:

```
propertiesTextG3 <перший аргумент> <файл з українським текстом> <контроль кількості інтервалів>  
<контроль друку також рядка тексту> <контроль розбивки тексту на рядки> <тип властивості> |
```

```
distributionTextG <той же аргумент щодо кількості рядків> <чи виводити на екран також рядкові дані>
```

та натисніть Enter. У Unix-подібних ОС вертикальна лінія, виділена червоним, слугує для створення пайплайнів (pipelines) у терміналі в shell; для ОС Windows :

```
PowerShell -Command "propertiesTextG3 <перший аргумент> <файл з українським текстом> <контроль кількості  
інтервалів> <контроль друку також рядка тексту> <контроль розбивки тексту на рядки> <тип властивості> |  
distributionTextG <той же аргумент щодо кількості рядків> <чи виводити на екран також рядкові дані>"
```

Ви побачите щось схоже на:

```
propertiesTextG3 2.1_3.0 ~/sajok.txt s 1 0 02y | distributionTextG s 1 +W
```

```
4  
3 3 5 1.0000 1.7000 1.7000 0.74074074 2 1 Тарас ШЕВЧЕНКО  
Вказематі
```

108	108	142	1.0000	1.3111	1.3111	0.86538462
4	34	34	8.5180	8.5180	1.0000	1.78987107
3	25	25	8.1680	8.1680	1.0000	1.78184991
108	109	110	1.0049	1.0194	1.0145	0.99518569
4	4	34	1.0000	8.5180	8.5180	0.21012893
108	245	245	2.2694	2.2694	1.0000	1.38827528
3	34	34	11.1680	11.1680	1.0000	1.83563445
3	25	89	8.1680	29.6720	3.6327	0.53260303
108	277	277	2.5611	2.5611	1.0000	1.43837754
3	3	13	1.0000	4.3111	4.3111	0.37656904
108	142	245	1.3111	2.2694	1.7309	0.80203908
3	3	25	1.0000	8.1680	8.1680	0.21815009
3	12	25	4.0000	8.1680	2.0420	0.87260035
27	29	242	1.0778	8.9778	8.3299	0.21603563
4	4	12	1.0000	3.0508	3.0508	0.49372385

3	1	Садок вишневий колохати
3	4	Хрущі надвишнями гудуть
3	4	Плугатарі зплугами йдуть
3	2	Співають ідучи дівчата
3	1	Аматері вечерять ждуть
3	4	Сем'я вечеря колохати
3	4	Вечірня зіронька встає
3	1	Дочка вечерять подає
3	4	Амати хоче научати
3	1	Так соловейко недає
3	1	Поклала мати колохати
3	1	Маленьких діточок своїх
3	2	Сама заснула колойіх
4	1	Затихло все тільки дівчата
2	1	Тасоловейко незатих

Міжітравня
С-Петербург

1	2	3	4		
9	2	0	5		
56.25%	12.50%	0.00%	31.25%		
0.9098	-0.5590	0	16		
2	3	4	5	6	7
2	13	1	0	0	0

2	.	.	.
6	2	.	5
1	.	.	.
.	.	.	.
.	.	.	.

.	.	.	.
2	0	0	0
6	2	0	5
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Виділення кольором не буде, тут просто згруповані різні типи даних статистики.

Колір	Тип даних	Значення
Червоний	Десятковий дріб з похибкою	<p>Середнє арифметичне значення (математичне сподівання) всіх чисел у VII стовпчику статистики для рядків, плюс-мінус середнє квадратичне відхилення; у випадку обраної властивості “y0” – часто число близьке до 1.0; може містити похибку округлення викликану особливостями знаходження суми чисел з плаваючою крапкою.</p> <p>У випадку, якщо всі рядки тексту виключаються з аналізу (дивіться: пояснення для оранжевого кольору), то виводиться замість матриць та подальшої інформації лише один рядок тексту – сигнальний напис: “1.000+-0.000\tALL!”, де ‘\t’ означає символ табуляції. Цей напис означає, що вказаний текст не підходить для аналізу програмою, а також, що</p>

Синій

Натуральні числа

будь-які дані, які в ході роботи програми можна було б отримати з цього тексту та супутніх текстів, повинні аналізуватися так, щоб не спростувати зроблені висновки на основі усієї сукупності текстів. Простіше кажучи, у такому випадку можна не враховувати текст, бо при правильному підході до аналізу та інтерпретації даних він не повинен скасувати результати.

Матриця розподілу рядків за кількістю слів та інтервалами; елемент матриці у k -му рядку та j -му стовпчику – кількість рядків, для яких значення обраної властивості з вказаними даними потрапляє в інтервал з номером j (нумерація починається з 1), який рівний номеру стовпчика матриці, і містить у рядку k слів (слів або їх сполучень, які відображаються злитно для дотримання мінімальних правил граматики при аналізі та виведенні даних), число k лежить у діапазоні між 2 і 7 включно (рядки, які аналізуються для матриці, містять від 2 до 7 слів (або написаних злитно сполучень)). Таким чином, матриця завжди має 6 рядків, а число стовпчиків буде залежати від введених та наявних даних.

Фактично ця матриця заміняє графік двовимірного розподілу даних.

На екран виводиться двічі, одна від іншої відмежована тильдами. У першому випадку нульові значення не відображаються, замість них стоять крапки. Це елемент візуалізації даних, який дозволяє краще “побачити”, як виглядає розподіл, де числові значення відповідають “висоті” на графіку розподілу

Оранжевий

Цілі невід’ємні числа

Зелений

Відсотки

(значенню функції дискретного двовимірного розподілу). У другому випадку на місці крапок стоять відповідні значення, які всі рівні 0. Дані для матриці отримуються з VIII та IX стовпчиків статистики за рядками.

Загальні кількості рядків. Перше число зліва – кількість рядків, які виключаються з аналізу для матриці, оскільки в них мало даних (1 або менше слів). Рівність 0 означає, що усі рядки виведені на екран беруть участь в аналізі для утворення матриці розподілу. Число справа – загальна кількість рядків у тексті, який виводиться та аналізується (включно з тими рядками, які пораховані зліва).

Розподіл загальної кількості рядків за інтервалами. Сума відсоткових значень природно рівна 100%. Номери інтервалів надписані зверху над відповідними відсотковими значеннями. Наприклад, напис у цих трьох рядках типу:

1 2

10 15

40% 60%

означає, що з усієї загальної кількості рядків, які піддаються аналізу за допомогою програми (містять достатньо даних), 40% припадає на перший інтервал (з меншим значенням властивості), а 60% – на другий (відповідно з більшим значенням властивості). Тобто таких рядків відповідно 10 і 15.

Усі інтервали рівні за величиною, але можуть мати різні кількості рядків (яка знаходиться в ході роботи програми).

Одновимірний розподіл, за ним можна

Бузковий

Натуральні числа

Фіолетовий

Натуральні числа

побудувати гістограму.

Номери інтервалів. Починається відлік з 1. Під ними відповідні кількості рядків, значення властивості для яких згідно даних потрапляє у відповідний за номером інтервал.

Кількість слів у рядках. Лежить в межах від 2 до 7 включно (якщо слів менше, тоді рядок отримує значення рівні 1.0 і вилучається з аналізу програмою для матриці). Під ними – відповідні значення кількості таких рядків. 0 відповідає випадку відсутності рядків з даною кількістю слів (чи сполучень, які відображаються як одне слово).

Уважне вивчення цих даних дає змогу зробити певні висновки щодо тексту, їх сукупності, самої моделі та мови.

Режим статистики за кількома властивостями (+M . . . -M)

Тепер, як і для програми [lineVariantsG3](#), можна використовувати режим кількох властивостей. Для цього замість однієї властивості можна вказати кілька в блоці виділеному розділювачами +M . . . -M.

У такому разі програма виведе на екран щось на зразок:

```
propertiesTextG3 sadok.txt s 1 0 +M y0 0y 02y 03y y2 y3 ууу -M
```

4								
2	4	2	1	2	4	4	1	Тарас ШЕВЧЕНКО
								Вказематі
3	4	4	1	2	4	2	1	Садок вишневий колохати
3	4	1	4	4	4	4	4	Хрущі надвишнями гудуть
3	4	4	4	1	4	1	1	Плугатарі зплугами йдуть
3	2	4	1	1	2	1	4	Співають ідучи дівчата
3	2	2	1	1	1	1	4	Аматері вечерять ждуть
3	1	3	4	1	3	1	3	Сем'я вечерея колохати
3	3	4	4	4	4	4	2	Вечірня зіронька встає

3	4	2	1	4	2	4	1	Дочка вечерять подає
3	4	4	4	1	4	2	1	Амати хоче научати
3	4	1	1	4	1	4	4	Так соловейко недає
3	4	4	2	1	2	4	2	Поклала мати колохати
3	1	4	1	1	1	1	4	Маленьких діточок своїх
3	1	4	2	4	1	2	4	Сама заснула колойіх
4	3	3	1	1	1	1	1	Затихло все тільки дівчата
2	4	4	1	1	1	1	4	Тасоловейко незатих
Міжітравня								
С-Петербург								

У такому разі не потрібно використовувати програму `distributionTextG`, оскільки вона не визначена для такого випадку. Перший стовпчик (виділений кольором тут) – кількості слів у відповідних рядках; далі йдуть стовпчики згідно того порядку, як вони позначені у блоці кількох властивостей відповідно – номери інтервалів, у які потрапляють значення відповідних властивостей. Найперше число справа (виділене червоним кольором і єдине у своєму рядку) – це кількість інтервалів для кожної властивості (вони всі однакові). Вісім стовпчиків у даному випадку до текстових записів – значить, що було 7 ($= 8 - 1$) заданих властивостей у блоці.

Режим "білих рядків"

Програма також переключена тепер на режим “білих рядків”, що означає, що рядки, які містять менше слів, ніж потрібно, щоб забезпечити існування хоча б двох варіантів рядка, не відображають статистику і вона не включається у загальний результат. Для цього у випадку однієї метрики та використання програми `distributionTextG` потрібно викликати останню з додатковим аргументом `+W` (означає `whitelines`).

Наприклад, у такому випадку Ви побачите:

```
propertiesTextG3 sadok0.txt s 1 0 03y ++B | distributionTextG s 1 +W
```

[illegible]

4	14	14	3.2040	3.2040	1.0000	1.52426261	3	4	Хрущі надвишнями гудуть
1	1	4	1.0000	2.7692	2.7692	0.53061224	3	1	Плугатарі зплугами йдуть
36	36	40	1.0000	1.1111	1.1111	0.94736842	3	1	Співають ідучи дівчата
1	1	1	1.4444	1.4444	1.0000	1.18181818	3	4	Аматері вечерять ждуть
36	36	52	1.0000	1.4444	1.4444	0.81818182	3	1	Сем'я вечеря колохати
14	14	14	1.0000	1.0000	1.0000	1.00000000	3	2	Вечірня зіронька встає
1	14	14	14.2400	14.2400	1.0000	1.86876640	3	4	Дочка вечерять подає
37	37	37	1.0000	1.0000	1.0000	1.00000000	3	2	Амати хоче научати
1	4	4	4.0000	4.0000	1.0000	1.60000000	3	4	Так соловейко недає
36	40	40	1.1111	1.1111	1.0000	1.05263158	3	4	Поклала мати колохати
1	1	1	1.0000	1.0000	1.0000	1.00000000	3	2	Маленьких діточок своїх
1	4	4	4.4444	4.4444	1.0000	1.63265306	3	4	Сама заснула колойіх
10	10	37	1.0000	3.7000	3.7000	0.42553191	4	1	Затихло все тільки дівчата

Тасоловейко незатих

Міжітравня

С-Петербург

1	2	3	4			
5	3	0	6			
35.71%	21.43%	0.00%	42.86%			
1.0974+	-0.4077	0	14			
2	3	4	5	6	7	
0	13	1	0	0	0	

.	.	.	.			
4	3	.	6			
1	.	.	.			
.	.	.	.			
.	.	.	.			
.	.	.	.			
~~~~~						
0	0	0	0			
4	3	0	6			
1	0	0	0			
0	0	0	0			

0	0	0	0
0	0	0	0

---

“Білі” рядки тут показані світлим, але не білим кольором.

## Режим статистики з фіксованим закінченням рядка (++B)

Якщо задати як один з аргументів командного рядка для `propertiesTextG3` символи ++B, то програма буде обчислювати усю статистику, наче останнє слово фіксоване обмеженням і не рухається. Фактично у такому випадку смисл цього символу (аргументу) є [аналогічним як і для програми lineVariantsG3](#).

Потрібно пам'ятати, що це звужує інтервал допустимих значень властивостей і при незмінній кількості рядків змінює розподіл всередині інтервалів.

## Контроль кількості інтервалів

Можливі три випадки:

“s” – кількість інтервалів буде визначена за відомим [правилом Стерджеса](#), де кількість випробувань буде рівною результуючій кількості рядків;

“I” – кількість інтервалів буде визначена за рекомендацією В. П. Левинського (див.: Опря А. Т. Статистика (модульний варіант з програмованою формою контролю знань). – Навч. посіб. – К.: Центр учбової літератури, 2012. – 448 с. ISBN 978-611-01-0266-7. С. 60);

**число** – кількістю інтервалів буде задане натуральне число (має бути більше 1, хоча це не перевіряється);

**щось інше** – буде використано 9.

## Контроль друку також рядка тексту

Якщо цей аргумент рівний 1, то праворуч від числових даних порядкової статистики буде виведено на екран також рядок, який аналізується (в уже перетвореному вигляді для аналізу). Інакше рядок не буде виведений.

## Контроль розбивки тексту на рядки

Якщо задати тут 1, то текст буде спочатку згрупований в один рядок, а потім розбитий на рядки методом ділення навпіл (за кількістю слів чи їх сполучень) доти, поки довжина усіх рядків не буде менша за 8 слів чи їх сполучень. Сторонні символи при цьому будуть відфільтровані. Якщо задати 0, то текст буде аналізуватися після фільтрації сторонніх символів (приблизно) у тих рядках, які були спочатку.

## Той же аргумент щодо кількості рядків

Мається на увазі тут має стояти те ж [значення](#), що і на місці контролю кількості інтервалів.

## Чи виводити на екран також і порядкові дані

Тут потрібно поставити 1, щоб програма надрукувала всю статистику (спочатку рядкову, а потім загальну по тексту), інакше буде надруковано лише загальну по тексту.

## Вибірковий аналіз тексту за рядками

Якщо Ви виконаєте команду

`propertiesTextG3 <шлях до файлу з українським текстом для аналізу> @n`

то на екран буде виведений текст з файлу, який буде аналізований з номерами усіх рядків зліва від самих рядків, що відділені від тексту символом табуляції (відображається у вигляді пробілу з непостійною шириною, яка залежить від налаштувань системи).

Тоді Ви можете виконувати цю ж програму (можна і без того, але можете вказати інші номери, ніж вважатиме програма) для аналізу вибраних рядків. Для цього до команд `propertiesTextG3`, крім останньої згаданої, будь-де у рядку команди до вертикальної риски (до пайплайну) додайте номери першого та останнього рядка, відділені символом двокрапки (без будь-яких інших символів, зокрема без пробілів). Можна задавати декілька таких пар, інформація буде виведена у такому ж порядку. Якщо деякі номери рядків будуть зустрічатися кілька разів, вони будуть виведені (якщо вказана така опція) та проаналізовані так само кілька разів. Якщо задано контроль розбивки тексту на рядки, що рівний 1, тоді програма об'єднає і проаналізує ті рядки, номери яких були вказані і відповідають номерам при виводі команди з `@n`.

Це все дозволяє сфокусовано аналізувати текст чи лише його частини.

# Робота з програмою **rewritePoemG3**

Перевірте, щоб папка (каталог), куди cabal встановив виконувані файли програм, була доступна для пошуку в змінній середовища PATH.

Далі введіть у командному рядку (чи терміналі) команду:

`rewritePoemG3 <перший аргумент> <файл з українським текстом> <тип властивості> <числові аргументи>`  
та натисніть Enter.

Ви побачите щось на зразок наступного:

```
rewritePoemG3 10.0_1.2 "sadok.txt" yyy 5 1 2
```

Після успішного завершення виконання програми (не має бути жодних повідомлень) у тій же папці (каталозі), де і файл з текстом, що переписується, має бути файл з додатковим закінченням .new.txt. Саме там записаний перетворений текст (наприклад, вірш) згідно введених даних.

Введені дані стосуються усього тексту, тобто кожного рядка тексту зокрема (після його попередньої обробки програмою).

## Більш комплексне використання

Числові аргументи мають багато в чому схоже значення, як і для програми [lineVariantsG3](#).

Числові аргументи, якщо задані, мають наступне значення.

Перший числовий аргумент – кількість інтервалів, на які поділяється проміжок між мінімальним та максимальним значенням властивості для даного рядка. Якщо не задано, вважається рівним 1. Значення 0 не дає змоги іншим числовим аргументам далі змінювати результат роботи програми.

Усі наступні числові аргументи (якщо задано, інакше ніяких перестановок не відбувається) – номери інтервалів, які будуть поміняні місцями з максимальним за номером. Це дозволяє змінити структуру даних, які відображаються як результат роботи програми і побачити внутрішні (не максимальні) елементи. Наприклад, числові аргументи 6 1 4 (у такому порядку) означатимуть, що в ході виконання програми відрізок між максимальним та мінімальним значенням властивості буде поділено на 6 рівних інтервалів, при цьому елементи, які знаходяться у першому

та 4, рахуючи від мінімального (інтервал з номером 1) буде переміщено до максимального за номером (і значеннями властивості) інтервалу, а потім записано у файл виводу результатів рядок з максимальним значенням властивості.

Значення, які були в максимальному інтервалі, будуть переміщені в інтервал з найменшим номером серед тих, які переміщені в максимальний.

## Порівняльний режим роботи (+C)

Програму `rewritePoemG3` можна запустити також у так званому “порівняльному” режимі, коли вона пропонує рядки (один за одним) з двох заданих файлів і записує вибраний (або порожній рядок, якщо жодний не обрано) у третій файл. Так з двох файлів шляхом їх порядкового порівняння можна створити новий. Це також дозволяє, запустивши програму двічі чи кілька разів з різними параметрами у звичайному режимі, потім запускати її в порівняльному на отриманих файлах і створювати у досить невимушений спосіб їх комбінації – нові варіанти.

**Ремарка:** Якщо Ви плануєте отримати більше “підказок” та рекомендацій від програми, то імовірно простіше (і краще) застосувати інтерактивний режим програми `lineVariantsG3` з кількома метриками.

Для роботи у порівняльному режимі (`comparative mode`):

`rewritePoemG3 +C <файл1> <файл2> <файл3>` (усі файли мають бути різними, причому перші два мають уже існувати з текстом, інакше програма не дасть значимого результату).

## Типи властивостей

Одним з принципів роботи програми є пошук серед варіантів тексту тих, для яких максимальним є значення певної функції, яка називається “властивістю” тексту (`property`) і становить собою певну властивість для рядків. Користувач може самостійно обрати властивість, яка буде використана при роботі програми (це робиться в командному рядку один раз протягом роботи програми заданням (або відсутністю відповідного) командного аргумента). Аргумент командного рядка виклику програми може бути:

- **“y0”** – найперша у часі властивість, заснована на [“періодах унікальності”](#). Ідея полягає у тому, що оцінюються кількості звуків, або пауз, або фонетичних явищ (палаталізації приголосних), які знаходяться між послідовними появами кожного звука не в одному, а в різних словах, і шукається загальна сума таких відстаней для різних слів. Більшому значенню відповідає текст з більш плавно змінним фонетичним малюнком, (імовірно) у залежності від середньої кількості звуків у “періоді унікальності” його може бути легше чи важче промовляти; меншому значенню – навпаки – текст з більш стрімкими змінами фонетичного малюнку, можливо, з підсиленням окремих груп звуків, що більш характерно для інтонаційно виділених і / чи поетичних текстів з закличками чи



підсиленими емоціями. При використанні цієї властивості (і виключно її) перший аргумент рядка виклику програми не має значення (він ігнорується програмою).

- **“0y”** – перша версія властивості аналізу лише ритмічності. Метрика (напівемпірична), заснована на функції ритмічності, що використовує тривалості звуків, які були синтезовані у пакеті програм [mmsyn6ukr-array](#). Функція ритмічності натхненна античною поезією, де замість наголошених та ненаголошених складів чергувалися ритмічно короткі та довгі; також музичними долями, для яких основними є дводольний ритм та тридольний. Функція реалізована таким чином, щоб можливо простіше вловити значні викиди підритмічностей для двоскладових і трискладових випадків. Використовуючи <перший аргумент> можна змінити співвідношення цих підвластивостей і відповідно – змінити властивість.
- **“02y”** – подібна до “0y” властивість, яка використовує інші тривалості звуків, синтезовані завдяки пакету [r-glpk-phoneticlanguages-ukrainian-durations](#). Можливо, одна з найточніших у цій версії з наявних для задачі написання ритмічного тексту. Можна створити інші варіанти тривалостей звуків, використовуючи можливості пакету `r-glpk-phonetic-languages-ukrainiandurations` або іншим способом.
- **“03y”** – подібна до “02y” властивість, яка використовує інші тривалості звуків, синтезовані завдяки пакету `r-glpk-phoneticlanguages-ukrainian-durations`. Можливо, одна з найточніших у цій версії з наявних для задачі написання ритмічного тексту. Можна створити інші варіанти тривалостей звуків, використовуючи можливості пакету `r-glpk-phonetic-languages-ukrainian-durations` або іншим способом.
- **“04y”** – подібна до “02y” властивість, яка використовує інші тривалості звуків, синтезовані завдяки пакету `r-glpk-phoneticlanguages-ukrainian-durations`. Ці тривалості звуків отримані з інших даних, ніж 0y, 02y та 03y, тому обережно змішуйте їх з ними в режимі кількох метрик.
- **“y”** – властивість, яка обчислює властивості “y0” та “0y” у більш ефективний спосіб, ніж кожен з них поодиночі, а потім перемножує отримані дані. Дає більші значення для рядків з більш плавно змінним фонетичним малюнком та такі, які більше ритмізовані (з точки зору властивості “0y”). Використання <першого коефіцієнта> внутрішньо впливає лише на підвластивість “0y”.
- **“y2”** – властивість, подібна до y, але використовує замість другої підвластивості (ритмічності) варіант з “02y”.
- **“y3”** – властивість, подібна до y, але використовує замість другої підвластивості (ритмічності) варіант з “03y”.
- **“y4”** – властивість, подібна до y, але використовує замість другої підвластивості (ритмічності) варіант з “04y”.
- **“yy”** – властивість, яка використовує властивість “y0” та “0y”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “0y”) та групуванням однакових звуків у групи ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “0y”.

- “**yy2**” – властивість, яка використовує властивість “y0” та “02y”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “02y”) та групуванням однакових звуків у групи ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “02y”.
  - “**yy3**” – властивість, яка використовує властивість “y0” та “03y”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “03y”) та групуванням однакових звуків у групи ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “03y”.
  - будь-які інші варіанти даного аргумента – аналогічна до “yy” з тією відмінністю, що замість “0y” використовується “04y”.
- “z”-лінія
- “**0z**”
  - “**02z**”
  - “**03z**”
  - “**04z**”
  - “**z**”
  - “**z2**”
  - “**z3**”
  - “**z4**”
  - “**zz**”
  - “**zz2**”
  - “**zz3**”
  - “**zz4**”

Ці метрики схожі на відповідні, де z замінено на y. Але у них використовуються складніші функції ритмічності, отримані з модуля [Languages.Rhythmicity.Factor](#) пакету `phonetic-languages-rhythmicity`. Обережно використовуйте змішані властивості у режимі кількох властивостей, оскільки вони фактично являють собою різні підходи всередині загального методу, тому можуть давати попарно менш сумісні результати, але при правильному використанні дають гарний результат. Можливо, потрібно трохи практики, також частіше використовуйте програму `propertiesTextG`.

**Порада:** якщо у виводі програми з використанням кількох метрик немає результатів (порожній вивід), збільшіть кількість груп у властивостях (хоча б для однієї) та / або додайте перетворення інтервалів, щоб змінити внутрішню структуру тих чи інших властивостей.

## Перший аргумент

При ознайомленні програми можна використовувати без цього аргументу, або ввівши на його місці 1_. Надалі Ви, можливо, захочете поглибити аналіз. Тоді можна задати також перший аргумент (він йде першим у списку аргументів командного рядка, не рахуючи групу – якщо така наявна – між аргументами +A та -A) у вигляді число1_число2, де число1 та число2 – десяткові додатні дробі або цілі додатні числа, причому одне з них може бути відсутнім (тоді воно вважається рівним 1). Наприклад,

3.4_2

2.987_0.7865

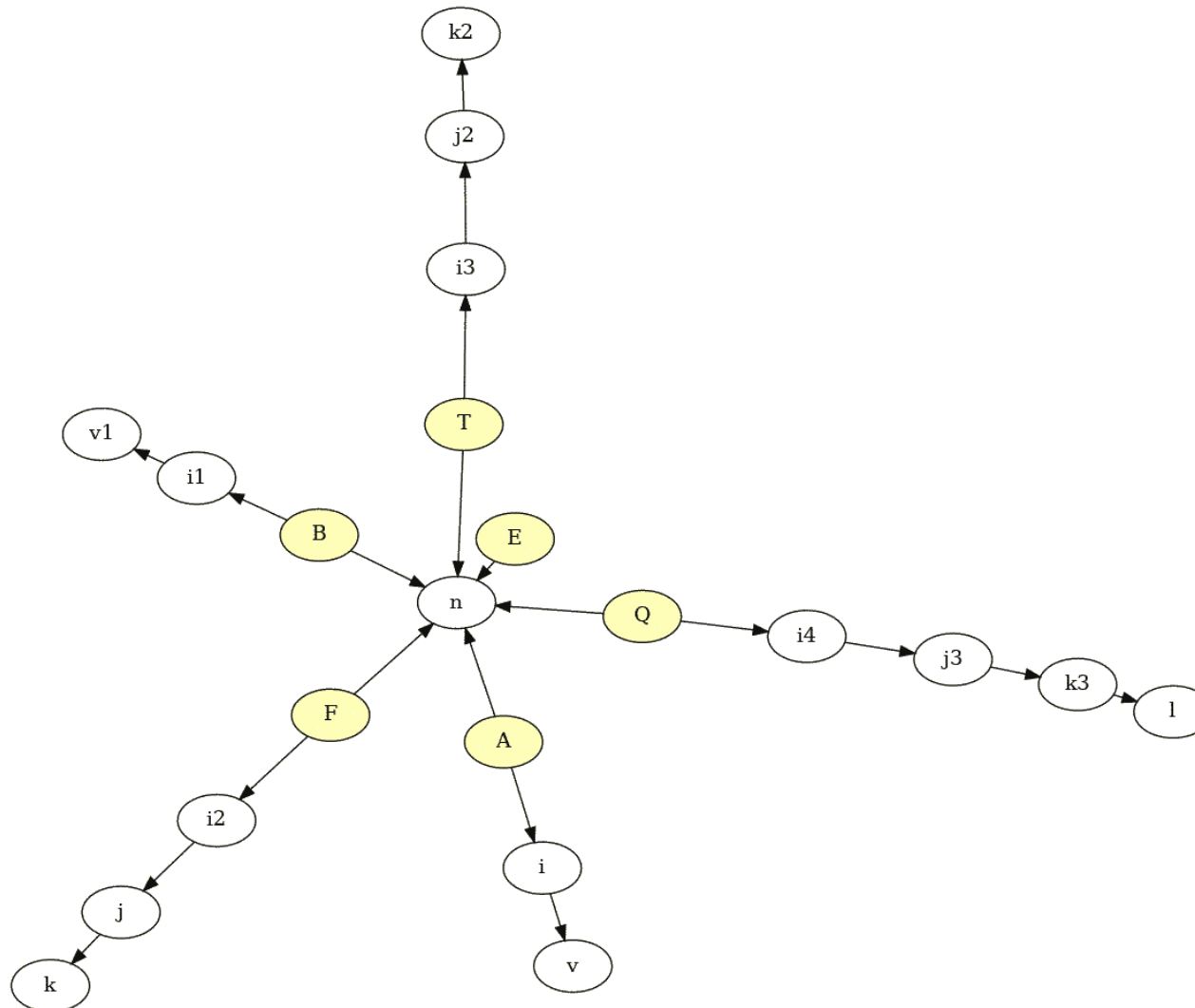
0.0001_

тощо.

Тоді перше число буде коефіцієнтом, який множиться на складову властивості, яка відповідає за ритмічність двоскладових стоп, а другий – за ритмічність трискладових стоп. Відповідно вдалою комбінацією можна підкресли, скомбінувати чи знівелювати вплив ритмічності для двоскладових чи трискладових стоп.

За відсутності цього аргументу програми поведуть себе так, наче він рівний 1_1.

## Обмеження (constraints)



При запуску програми [lineVariantsG3](#) можна вказати як аргументи командного рядка обмеження. Вони дозволяють зменшити кількість обчислень, розглядати лише певні варіанти (наприклад, з певним визначеним порядком деяких слів тощо), що дозволяє фактично розширити можливості програми. Ці обмеження кодуються як аргументи командного рядка між двома спеціальними позначеннями +A та -A. Вони утворюють групу аргументів, які можуть стояти будь-де у рядку вводу даних. У залежності від того, чи вказуються ці аргументи, програма ставить чи не ставить додаткове питання для перевірки та підтвердження (так званий *double check*).

Типів обмежень є 6, їх можна довільно комбінувати, але з дотриманням меж для кожного з них.

На рисунку поряд видно, що усі типи реалізовані з одним аргументом, який

подібний у всіх них – це кількість слів (чи їх сполучень) у рядку. Користувач, запустивши програму, уже не може коригувати в ході її роботи цю кількість, але вона важлива для обмежень загалом. Жодний з цифрових символів у обмеженнях не повинен бути більшим за це число, також це число саме не більше 6 і не менше 0. Також необхідною умовою є те, що жодні цифрові символи в межах одного кодованого обмеження не можуть повторюватися двічі. Наприклад, завідомо не є валідними обмеження: Q2235 (повтор цифр), E2 (цифрові символи там, де їх немає), T247 (7 більша 6), F0 (один символ замість необхідних двох), A37523 (7 більша 6), B5 (один символ, а мають бути ще). Неправильно задані обмеження або не вплинуть на результат (хоча буде очікуватися інше), або викличуть помилку виконання (runtime exception) і зупинку роботи програми. Оскільки результат їх застосування не є простим, тому програма при їх заданні виводить на екран рядок, до якого будуть застосовані введені обмеження з додатковим запитанням, чи всі дані введено правильно.

Типи обмежень та їх значення наведено детальніше у таблиці.

Тип обмеження	Цифрові символи	Значення обмеження та його параметрів
<b>E</b>	Без вводу додаткових цифрових символів.	Відповідає відсутності додаткового обмеження, в ході роботи програми не впливає на кінцевий результат.
<b>Q</b>	4 попарно нерівні цифри в межах від 0 до кількості слів чи їх сполучень мінус 1	Цифри – це індекси 4-х слів чи їх сполучень, взаємний порядок яких при перестановках буде збережений таким. Також якщо ці слова однакові (без врахування великих та малих літер), тоді це зручний спосіб зменшити об’єм даних, який буде аналізуватися.
<b>T</b>	3 попарно нерівні цифри в межах від 0 до кількості слів чи їх сполучень мінус 1	Цифри – це індекси 3-х слів чи їх сполучень, взаємний порядок яких при перестановках буде збережений таким. Також якщо ці слова однакові (без врахування великих та малих літер), тоді це зручний спосіб зменшити об’єм даних, який буде аналізуватися.
<b>F</b>	2 попарно нерівні цифри в межах від 0 до кількості слів чи їх сполучень мінус 1	Цифри – це індекси 2-х слів чи їх сполучень, взаємний порядок яких при перестановках буде збережений таким. Також якщо ці слова однакові (без врахування великих та малих літер), тоді це зручний спосіб зменшити об’єм даних, який буде аналізуватися.
<b>A</b>	1 цифра та ще кілька попарно нерівних цифр (усі між собою нерівні) справа від неї в межах	Перша цифра – індекс елемента, відносно якого визначається розміщення усіх інших елементів (слів чи їх сполучень); усі інші цифри правіше – індекси елементів, які мають стояти в отриманих перестановках ПРАВИШЕ від елемента з

<b>В</b>	від 0 до кількості слів чи їх сполучень мінус 1	індексом, рівним першій цифрі.
	1 цифра та ще кілька попарно нерівних цифр (усі між собою нерівні) справа від неї в межах від 0 до кількості слів чи їх сполучень мінус 1	Перша цифра – індекс елемента, відносно якого визначається розміщення усіх інших елементів (слів чи їх сполучень); усі інші цифри правіше – індекси елементів, які мають стояти в отриманих перестановках ЛІВІШЕ від елемента з індексом, рівним першій цифрі.

## Паралельне виконання програм

Зазвичай усі програми пакету виконуються одним ядром процесора. При цьому для всіх програм, які розглядаються, є можливість включити режим роботи на кількох ядрах – паралельні обчислення. Для цього серед аргументів командного рядка мають бути наступні: **+RTS -N -RTS**

Їх розміщення не впливає на порядок та значення інших аргументів командного рядка, також між записами RTS можуть бути інші параметри. Детальніше про ці параметри дивіться документацію англійською [3].

Можна рекомендувати ці параметри лише для програми **propertiesTextG3**. Для інших програм вони не рекомендуються, хоча Ви можете ними користуватися (вони просто збільшать використання ресурсів).

## Джерела

- 1 Шевченко Т. Г. “Садок вишневий коло хати...”. [Електронний ресурс]. Режим доступу: [Sadok vyshnevyyi kolo khaty](#). Перевірено 09 листопада 2020 р.
- 2 Oleksandr Zhabenko. dobutokO-poetry. [Електронний ресурс]. Режим доступу: [dobutokO-poetry: Helps to order the 7 or less Ukrainian words to obtain somewhat suitable for poetry or music text](#). Перевірено 09 листопада 2020 р.

- 3 Glasgow Haskell Compiler User's Guide. 7.5. Using SMP parallelism. [Електронний ресурс]. Режим доступу: [https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/using-concurrent.html#using-smp-parallelism](https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/using-concurrent.html#using-smp-parallelism) . Перевірено 10 листопада 2020 р.
- 4 Ямбічна строфа // [Енциклопедичний словник класичних мов](#) / Л. Л. Звонська, Н. В. Корольова, О. В. Лазер-Паньків та ін.. — К. : ВПЦ «Київський університет», 2017. — С. 551. — 552 с.
- 5 Смаглій Г. А. Теорія музики : Підруч. для навч. закл. освіти, культури і мистецтв / Г. А. Смаглій. — Х. : Вид-во “Ранок”, 2013. — 392 с. ISBN 978-617-09-1294-7

## Додаткова література

- 6 [Квантитативне віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%BD%D1%82%D0%B8%D1%82%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>. Перевірено 12 листопада 2020 р.
- 7 [Силабічне віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D0%B0%D0%B1%D1%96%D1%87%D0%BD%D0%B5%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>. Перевірено 12 листопада 2020 р.
- 8 [Тонічне віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Перевірено 12 листопада 2020 р.
- 9 [Силабо-тонічне віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D0%B0%D0%B1%D0%BE-%D1%82%D0%BE%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D0%B0%D0%B1%D0%BE-%D1%82%D0%BE%D0%BD%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Перевірено 12 листопада 2020 р.
- 10 [Віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>. Перевірено 12 листопада 2020 р.
- 11 [Античне віршування — Вікіпедія](#) [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%B8%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D1%82%D0%B8%D1%87%D0%BD%D0%B5_%D0%B2%D1%96%D1%80%D1%88%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Перевірено 12 листопада 2020 р.