

Підхід фонетичних (просодичних) мов до відкриття більш бажаних варіантів текстового рядка (PhLADiPreLiO) з використанням Haskell

Олександр Сергійович Жабенко
ORCID: 0000-0002-4330-2506

9 серпня 2022 р.

Copyright (c) 2020-2022 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.

Анотація

У роботі запропоновано оригінальний підхід до завдання побудови текстів з заданими фонетичними властивостями, який дозволяє ефективно та швидко працювати над задачею, пропонуючи цікаві рішення як у змістовній, так і в інженерній площинах. Побудовано елементи теорії, яка проілюстрована використанням програм мовою Haskell для значного прискорення необхідних обчислень.

Ключові слова

Фонетичні мови; просодичні мови; фонетика; просодія; склади; ритм; метр; віршування; текст; когерентні стани; склад-як-одне-ціле (SaaW); фонетичне-явище-як-одне-ціле (PhoPaaW); Haskell; підхід фонетичних мов до відкриття більш бажаних варіантів рядка (phonetic languages approach to discovering preferred line options – PhLADiPreLiO – phladiprelio – читається "фладіпреліо", з наголосом на перед-останньому складі).

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.

Що мається на увазі?

Якщо хтось буде фразу в мові, порушуючи правила граматики чи семантики, то ця помилка видна майстерному мовцю відразу, вона ідентифікується як така майже миттєво. Натомість, якщо звучання фрази має якісь фонетичні особливості, не рахуючи акценти, наприклад, складність вимови чи навпаки легкість, плавність чи стрибкоподібність тощо, то ідентифкувати це як помилку чи щось суттєве можна не відразу або при спеціальній увазі. Можна уявити це, як надання переваги у мові семантиці (смислу, значенню) та граматиці, але меншої ваги фонетиці. Фонетичною називається мова, яка побудована спеціально для підсилення значення та вагомості саме фонетичної складової.

Фонетична чи просодична мова?

Цікавим є питання, чи називати підхід “фонетичних” чи “просодичних” мов [20]. Але потрібно сказати, що вивчаються власне фонетичні особливості, те, що пов’язане зі звучанням мовлення. Серед них те, що стосується окремих фонетичних явищ у загальному випадку, зокрема фонем чи навіть палаталізації. Ці питання в загальному не є предметом вивчення просодії як науки, як певної складової фонетики, але є предметом ширшого вивчення фонетики. Більше того, немає жодних обмежень і прив’язки запропонованого підходу до власне складів, що більш характерно саме для предмету вивчення просодії. Узагальнення у пакеті `phonetic-languages-simplified-generalized-examples-array` можуть бути зроблені і на більш загальні випадки.

Тим не менше, на даному етапі розвитку переважна більшість інформації тут стосується чи прямо пов’язана зі складами та просодією. Тому залишаю назву “фонетичних мов”, враховуючи, що просодія є більш конкретизованою галуззю фонетики.

Етична складова

Запропонований підхід схожий на підхід теорії музики. Так, в музиці серед усіх звуків виділяються музичні, згодом досліджуються консонанси та дисонанси, ще згодом ноти, інтервали, акорди, мелодії, композиція тощо. Є свої рекомендації, але вони не сковують творців, а допомагають. Подібно і запропонований підхід покликаний до подібної допомоги. Його незвичність на перший погляд не може бути підставою для його заперечення.

Для християн, до яких автор відноситься і сам, важливими є слова Мойсея: "Мойсей вийшов і сказав народу слова Господні, і зібрав сімдесят мужів зі старійшин народу і поставив їх біля скинії. І зійшов Господь у хмарі, і говорив з ним, і взяв від Духа, Який на ньому, і дав сімдесятьом мужам старійшинам. І коли спочив на них Дух, вони стали пророкувати, але потім перестали. Двоє з мужів залишилися в стані, одному ім'я Елдад, а другому ім'я Молад; але і на них спочив Дух [вони були з числа записаних, тільки не входили до скинії], і вони пророкували в стані. І прибіг отрок, і доніс Мойсееві, і сказав: Елдад і Молад пророкують у стані. У відповідь на це Ісус, син Навина, служитель Мойсея, один із обраних його, сказав: господарю мій Мойсею! заборони їм. Але Мойсей сказав йому: чи не ревнуєш ти за мене? о, якби усі в народі Господньому були пророками, коли б Господь послав Духа Свого на них!" (Числа 11:24-29).

Добре, щоб усі вміли добре писати і говорити.

Перша ідея

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 граматикою, 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 граматики, 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.[51]

Тривалості звуків як основа ритмічності

Основою підходу є той факт, що звуки мови мають різну тривалість, яка залежить від багатьох факторів, яка визначається зокрема як способом звукоутворення (різним для кожного з них), так і іншими факторами, які тією чи іншою мірою можна контролювати, але, зазвичай, повний контроль не вимагається і не досягається. Це призводить до того, що чередування звуків, зокрема їх поєднання у складі, утворює певний ритмічний малюнок. Людина має здатність (яка піддається розвитку та тренуванню) розпізнавати риси цього малюнку, порівнювати їх між собою, робити певні фонетично-ритмічні узагальнення.

Питання визначення тривалостей звуків мови є непростим, але точний результат як уже зазначалося не вимагається. У даній реалізації підходу фонетичних мов використано певні статистичні характеристики звуків, зокрема визначено можливі тривалості. Якщо порівняти спосіб визначення тривалостей, який запропоновано та застосовано у програмі пакету `r-glpk-phonetic-languages-ukrainian-durations`, то аналогією буде упаковка об'ємних об'єктів. Для спостерігача упаковка буде уявною моделлю процесу отримання тривалостей звуків. Програма `pldUkr` (її узагальнення `pldPL` з пакету `phonetic-languages-phonetics-basics` також слідує цим шляхом, але у ній немає нормування, оскільки для різних мов може не бути такого явища як палаталізація) методом лінійного програмування шукає мінімальну оболонку (не в строгому математичному значенні), всередині якої можуть «вміститися» звуки мови. Ця оболонка і має аналогію упаковку, у той час як звуки мови мають аналогією об'єкти змінного об'єму всередині упаковки. Один і той же звук може вживатися у різних ситуаціях, у різних словах з різною тривалістю, але програма старається підібрати такі тривалості, які б «охоплювали» (подібно як огинаюча крива «охоплює» те чи інше сімейство кривих) усі ці варіації для усіх звуків, при цьому застосовується додатково певне нормування на тривалість фонетичного явища палаталізації (пом'якшення) приголосного, яке менше всього контролюється людиною, а тому очікується, що ця тривалість є найбільш стійкою до можливих випадкових чи систематичних коливань. Для української мови можлива тривалість, яка не видозмінює сильно звучання визначена експериментально з використанням комп'ютерної програми `tm1`.

Зрештою, нормування не є обов'язковим, важливо, щоб усі тривалості були пропорційні одна одній, тобто важливими є не самі тривалості (що чисельно виражаються як дійсні додатні числа), а їх взаємні співвідношення (допускається множення цих тривалостей одночасно всіх на одне і те ж додатне число, що не впливає на результати підходу).

Поліритм як мультивпорядкована послідовність

Нехай маємо послідовність з наступною структурою. Нехай ми здійснили (взагалі кажучи умовний) поділ послідовності на компактні однозв'язні підгрупи з однаковою кількістю елементів кожна у підгрупі, що фактично означає, що ми розбили послідовність на послідовність підпослідовностей з однаковою кількістю елементів у кожній. Розглянемо внутрішню упорядкованість кожної підпослідовності у розумінні розміщення значень її елементів (їх можна порівнювати за відношенням порядку, тобто вони є типом даних, який має реалізований екземпляр класу `Ord`) та повторюваності елементів. Вважаючи, що елементи підпослідовностей можуть бути попарно різними (чи в окремих випадках і однаковими), будемо порівнювати позиції, на яких розміщуються в підпослідовностях підгрупи елементів, які між собою мають вищий ступінь спорідненості ("близькості", "схожості", "подібності") за величиною та порядком (якщо це тип даних числовий, який має природний порядок, наприклад, `Double`, тоді поняття "спорідненості" за величиною та "спорідненості" за порядком

співпадають). Позначимо такі підгрупи індексами, які в коді модулів мають частіше всього буквенне позначення.

Тоді кожна підпоследовність складатиметься з однакової кількості елементів однієї природи (зокрема, чисел типу Double), у кожній підпоследовності будуть виділені кілька підгруп "подібних" елементів за величиною (і порядком, якщо підпоследовності відсортувати за величиною), кожна з яких матиме свій індекс у вигляді символу (найчастіше у коді – літери). Підгрупи мають мати (насправді приблизно) однакову кількість елементів (у коді це не витримано строго для спрощення останнього, але це так у переважній більшості випадків в силу надлишкової "точності" чисел типу Double, що використовуються). Розглянемо питання, на яких позиціях у підпоследовностях розміщуються елементи з тих же самих підгруп, але які належать до різних підпоследовностей.

Для оцінки цього введемо певні числові функції, які мають регулярну поведінку та дозволяють визначати за їх результатом, чи підпоследовності мають елементи, які належать до відповідних підгруп на тих же самих місцях, чи на різних. Можна показати, що ситуація "на різних" відповідає наявності кількох ритмів – для кожної підгрупи буде свій власний, які між собою не співпадають, водночас ідеальна ситуація "повністю на тих самих місцях" відповідає випадку, коли ці ритми між собою узгоджуються, подібно як це відбувається у випадку когерентності у квантовій фізиці, зокрема просторової та часової когерентності, що є важливим зокрема для розуміння роботи лазерів та мазерів. Поліритми, які між собою когерують, утворюють помітніший загальний ритм, подібно як і наявність когерентності у випромінюванні веде до появи більшої структурованості випромінювання.

Як ілюстрація до ідей розділу наступні дані.

Приклад ритмічної последовності (ідеальний випадок).

```
Prelude Rhythmicity.PolyRhythm Numeric> let f x = putStrLn . showFFloat (Just 4) (sin (2*pi*x)) $ ""
in mapM_ f [0,0.2..4]
0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
0.5878
-0.5878
-0.9511
-0.0000
0.9511
```

```
0.5878
-0.5878
-0.9511
-0.0000
```

```
Prelude Rhythmicity.PolyRhythm Numeric> getPolyChRhData 'a' 5 (PolyCh [True,True,True,False] 5)
(PolyRhythm [1,1,1,1,1]) . map (sin . (*pi) . (*2)) $ [0,0.2..4]
[[RP P c,RP P a,RP P b,RP P e,RP P d],[RP P c,RP P a,RP P b,RP P e,RP P d],[RP P c,RP P a,
RP P b,RP P e,RP P d],[RP P c,RP P a,RP P b,RP P e,RP P d]]
```

А ось приклад послідовності з ритмом, який не є сталим (який менше відчутний).

```
Prelude Rhythmicity.PolyRhythm Numeric> let f x = putStrLn . showFFloat (Just 4)
(sin (27182.81828459045*pi*x)) $ "" in mapM_ f [0,0.01..0.24]
0.0000
-0.5139
-0.8817
-0.9988
-0.8319
-0.4284
0.0969
0.5947
0.9233
0.9894
0.7742
0.3388
-0.1930
-0.6698
-0.9562
-0.9707
-0.7092
-0.2460
0.2872
0.7386
0.9801
0.9428
```

```

0.6375
0.1509
-0.3787
Prelude Rhythmicity.PolyRhythm Numeric> getPolyChRhData 'a' 5 (PolyCh [True,True,True,False] 5)
(PolyRhythm [1,1,1,1,1]) . map (sin . (*27182.81828459045) . (*pi)) $ [0,0.01..0.24]
[[RP P a,RP P b,RP P e,RP P d,RP P c],[RP P d,RP P e,RP P c,RP P b,RP P a],[RP P a,RP P b,
  RP P c,RP P e,RP P d],[RP P d,RP P e,RP P c,RP P b,RP P a],[RP P a,RP P b,RP P c,RP P e,RP P d]]

```

Когерентні стани поліритмічності як одне з суттєвих джерел ритмічності

Описаний паттерн виникнення ритмічності є одним з суттєвих можливих варіантів утворення ритмічності зокрема у текстах чи музиці, але не єдиним. Варто зазначити, що описаний механізм утворення ритмічності, як показують статистичні експерименти з текстами з використанням цього коду (коду бібліотеки та залежних від неї пакетів на сайті Hackage), може бути не єдиним можливим варіантом, але у багатьох випадках має визначальне значення та вплив на перебіг процесу ритмізації (утворення, зміни чи зникнення ритму). Також відомо, що наявність статистичного взаємозв'язку не означає наявності більш глибоких видів зв'язку між явищами, зокрема причинно-наслідкових. "Кореляція не означає причинності". Більш глибокий зв'язок передбачає наявність інших, не статистичних даних, які дозволяють його підтвердити.

Наслідки для реп музики

Код бібліотеки дозволяє на практиці отримати ритмічні зразки, які близькі часто до текстів пісень у стилі реп. Тому це можна віднести до одного з прямих застосувань бібліотеки.[52]

Дитина або хтось новий для мови вчиться читати

Коли дитина тільки починає читати слова мови (або це може бути хтось новий щодо мови). він чи вона починають з вимови фонем для кожного значимого написаного (і, отже, прочитаного) символу. Згодом після деякої практики, дитина починає читати рівно, більш плавно. Тим не менше, якщо текст власне є поетичним твором, зокрема віршованим, ЧАСТО (можливо, зазвичай, чи іноді, чи частіше тощо) є очевидним, що текст, який ось читається у такій манері, має деякі ритмічні властивості, незважаючи на те що фонемі читаються і вимовляються у режимі нерегулярних і до певної міри невідповідних для нормального мовлення тривалостей. Нам буває (часто) достатньо певної організації елементів тексту, щоб відрізнити поетичний текст від непоетичного.

Подібна ж ситуація виникає, коли особа з акцентом (можливо, сильним чи доволі незвичним) читає поетичний текст. Або також в інших ситуаціях. Бібліотека спроектована таким чином, наче Ви маєте справу саме з такими ситуаціями. У ній вважається, що тривалості є фіксованими і відомими (заданими) наперед, і тоді наявність кількох обґрунтованих (більш чи менш) дозволяє оцінити (звичайно, наближено) ритмічні властивості та деякі інші, відповідно до запропонованих тут алгоритмів.

Це все, на думку автора, є підґрунтям для використання бібліотеки та її функціональності в таких випадках.[53]

Функції для збільшення та зменшення

Починаючи з версії пакету `phonetic-languages-rhythmicity 0.5.3.0`, функції для збільшення та зменшення для поліритмічності стали більш подібними до взаємно обернених функцій. Це, очікувано, веде до більш "гладкої" поведінки на початку рядка, в середині та в кінці.

Зауваження: Починаючи з версії `0.6.0.0` пакету `phonetic-languages-rhythmicity` значення властивостей серій "c", "s", "t", "u", "v", а також багатьох інших (починаючи з версії `0.9.0.0`) можуть бути від'ємними, це не впливає на загальну логіку роботи програм. Починаючи з версії `0.8.0.0` пакету `phonetic-languages-simplified-examples-array`, додано також нові властивості, які також можуть бути від'ємними за знаком.

Питання вибору найкращої функції та суміжні питання

Розглянемо наступне питання: припустимо, ми отримали найкращий варіант (на нашу суб'єктивну думку чи на основі якихось критеріїв, це тут несуттєво) рядка тим чи іншим способом (тут спосіб насправді ролі не грає). Чи існує така функція, яка саме цей варіант рядка робить оптимальним, тобто для якої саме такий варіант рядка з поміж усіх можливих перестановок дає максимум? Так, існує. Це нескладно довести. Доведення нагадує принцип роботи еквайзерів.

Нехай $n \in \mathbb{N}$ – це кількість складів у такому рядку. Розташуємо тривалості складів у ряд за зростанням (стандартна процедура для описової статистики). Знайдемо найменшу ненульову різницю між сусідніми значеннями, поділимо її на 5. Позначимо цю отриману величину δ . Тепер розглянемо ряд тривалостей складів саме для нашого найкращого рядка. Пронумеруємо кожний склад від початку, рахуючи від 1. Позначимо $Y = \{y_i, \quad i \in \mathbb{N}, \quad i = 1, 2, \dots, n\}$ множину всіх значень тривалостей в порядку слідування складів у найкращому рядку. Позначимо $X = \{0 = x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{n+1}, \quad i \in \mathbb{N}, \quad i = 1, 2, \dots, n\}$ множину координат точок кінців часових умовних інтервалів, на які поділяє наш найкращий рядок часову пряму (лівий край рівний 0, бо відлік часу починається з 0). Позначимо $M = \{z_i = \frac{x_i + x_{i+1}}{2}, \quad i \in \mathbb{N}, \quad i = 1, 2, \dots, n-1\}$ множину середин відрізків, на які поділяють часову пряму кінці умовних інтервалів. Позначимо $L_1[a, b]$ клас функцій, інтегровних за Лебегом на відрізку $[a, b]$. Позначимо $I(y_i, z_i, \delta)[z_i - \delta, z_i + \delta]$ клас усіх обмежених функцій з $L_1[z_i - \delta, z_i + \delta]$, максимальне та мінімальне значення кожної з яких лежить на відрізку $[y_i - \delta, y_i + \delta]$. Кожну функцію класу I позначимо g . Розглянемо клас функцій F (своєрідних скінченних аналогів відомих дельта-функцій Дірака), визначених наступним чином:

$$f(x, i) = \begin{cases} g \in I(y_i, z_i, \delta)[z_i - \delta, z_i + \delta], & \text{якщо } y_i \text{ є унікальним значенням у множині } Y, x \in [z_i - \delta, z_i + \delta] \\ y_i, & \text{якщо } y_i \text{ має рівне значення з якимось іншим числом з множини } Y, x \in [z_i - \delta, z_i + \delta] \\ 0, & \text{в усіх інших точках} \end{cases}$$

Легко переконатися, що

$$\sum_{i=1}^n \int_{-\infty}^{\infty} f(x, i) dx,$$

де інтегрування здійснюється за Лебегом, і є шуканою функцією (бо лише склади найкращого рядка на своєму місці, враховані зі своїми індексами, дають додатній внесок у її значення, а для всіх інших варіантів функції хоча б деякі зі складів дають 0 внесок), причому вона не єдина в силу того, що у першому рядку визначення $f(x, i)$ функція може мати довільне значення з замкненого непорожнього інтервалу. Отже, існує принаймні один клас функцій, який описується такою формулою, для кожної з яких даний варіант рядка буде оптимальним.

Поставимо наступне питання: якщо розглянути не рядок, а їх сукупність, наприклад, вірш чи поему. Чи буде існувати для усього твору (для усієї цієї сукупності рядків) функція, яка буде робити оптимальним кожний рядок, тобто яка буде описувати увесь твір, кожний рядок у ньому?

У даному випадку, попередній спосіб побудови функції не дає бажаного результату, оскільки вже для двох рядків може бути, що те, що є кращим для одного з них, є не найкращим для іншого. У випадку збільшення кількості рядків ця загальна неоптимальність тільки посилюється. Тим не менше, існування такої функції для різних частинних випадків є принципово можливою ситуацією, що, правда, імовірність її зменшується як зі збільшенням кількості рядків, так і з появою різних особливостей рядків, які посилюють відмінності між ними. Загалом пошук саме такої однієї функції може бути практично недоцільним.

Тим не менше, якщо розглянути увесь твір як один рядок, вважаючи його оптимальним, то описаний щойно спосіб побудови відповідної функції (класу функцій) знову дає результат. Так, кількість складів у ньому (в узагальненому "рядку"-творі) зростає, але сама процедура дає подібні результати (якщо знехтувати можливими однаковими тривалостями та повторами складів у більшому тексті, які призводять до того, що деякі слова можуть бути переставлені місцями, що робить текст лише приблизно оптимальним). Можна запропонувати подальше вдосконалення цієї процедури (наприклад, введення множників, які залежать від значень сусідніх тривалостей складів), що дозволяє зменшити неоптимальність тексту.

Але все одно, якщо перейти від даного тексту до іншого, отримана функція уже, скоріше всього, не буде оптимальною.

Робимо висновок, що для усієї множини доцільно організованих текстів існування універсальної функції бачиться як певна гіпотеза (з майже напевно заперечною відповіддю).

Зв'язок з фракталами

На цей принциповий "злам", принципову неподібність текстів звернемо увагу.

Але спершу зауважимо наступне. Розглянута функція вище робить оптимальним один варіант рядка, водночас можна розглядати не один варіант, а декілька, серед яких обирати найкращий. Тоді задача зводиться до пошуку оптимальної групи варіантів з можливо мінімальною кількістю складових.

Якщо знову розглянути граничний випадок групи з кількістю елементів рівною факторіалу числа слів у рядку, то легко переконатися з комбінаторних міркувань, що така група містить (складається з, є) усі варіанти рядка з даних слів, серед яких має бути той, який буде результатом. Але уже для 5 слів така група містить $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ варіантів по 5 слів, разом це 600 слів, щоб прочитати які потрібно зазвичай кілька хвилин.

Якщо зменшити кількість слів у оптимальній групі, то серед них може не бути потрібного варіанту, але час читання зменшується відповідно. Отже, якщо маємо якийсь спосіб хоча б приблизного упорядкування, то доцільним буде пошук уже оптимального співвідношення "розмір групи – ступінь влучності приблизного упорядкування".

Це призводить до ідеї застосування різних функцій, які легко обчислюються, мають певну характерну поведінку і дозволяють приблизно упорядкувати множини усіх перестановок відповідно до завдання пошуку, і пошук уже не єдиного варіанту, а оптимальної групи варіантів.

При цьому принциповими є етапи пошуку, коли програма не дає повністю бажаного результату. Тоді можна щось змінити в даних (не змінюючи самі слова), або змінити слова, щоб змінити ті варіанти, з якими працює програма, оскільки це змінює структуру множини усіх варіантів і дозволяє отримати інші все ще певною мірою "оптимальні" варіанти з точки зору автора та з точки зору програми. Це дало поштовх так званому рекурсивному режиму роботи, коли зміною даних є злиття слів (і отже, виключення з розгляду варіантів, де вони стоять не підряд), а також режим кількох варіантів.

Цікавим спостереженням також є те, що при рекурсивному режимі можна отримати рядки, в яких більш помітними є "злами" ритмічності, які частіше всього можна включити в ритмічність через введення додаткових пауз. Відомими в теорії є також вірші з паузами (наприклад, так звані цезурами), для яких це може бути корисним.

Також можливість утворити рядки з рядків, а кожен рядок з певних складових, нагадує фрактали. Тим не менше, зв'язок з фракталами потребує більш детального та глибокого вивчення.

Зацікавлений читач може звернутися до літератури.[7, 33, 31, 27, 48, 19, 8, 37, 49, 18, 11, 3, 30, 45, 39, 2, 6, 29, 42, 24, 4, 71, 76]

Аналогія з екстремальним принципом

У механіці та оптиці відомим є принцип екстремальності дії (вводиться фізична величина дії, яка для реальних процесів у цих галузях приймає серед усіх можливих значень за траєкторіями мінімальне (найчастіше), або максимальне значення, або таке, яке не залежить від траєкторії, тобто одним словом екстремальне значення). У термодинаміці є закон зростання ентропії замкнених систем. У випадку механіки та оптики пошук дійсних траєкторій зводиться до пошуку тих з можливих траєкторій, для яких значення функціоналу дії є екстремальним, що дозволяє скористатися апаратом вищої математики для пошуку саме цих екстремальних значень.[47]

Висувалося припущення, що саме аналогія з цим принципом, тобто порівняльний режим роботи у поєднанні з режимом кількох властивостей може дати найкращий результат. Тим не менше, широке коло гарних віршованих рядків (і не тільки їх) не обов'язково

слідують за цим паттерном віршування. Тому загалом принцип слугує своєрідною інтуїцією пошуку, але не закономірністю.

Можливість використовувати власні тривалості репрезентацій звуків чи фонетичних явищ

Програми пропонують за замовчуванням чотири різні множини тривалостей фонетичних представлень, але починаючи з версії 0.13.0.0 є можливість задати власні тривалості. Для цього потрібно прописати їх як числа типу Double у файлі в порядку, який визначається наступним чином:

UZ 'A' D	дз (твердий)	8
UZ 'A' K	дз (м'який)	9
UZ 'B' D	ж (твердий)	10
UZ 'B' K	ж (пом'якшений)	11
UZ 'C' S	й	27
UZ 'D' N	сь	54
UZ 'E' L	ч (твердий)	39
UZ 'E' M	ч (пом'якшений)	40
UZ 'F' L	ш (твердий)	41
UZ 'F' M	ш (пом'якшений)	42
G		55
H	ю	56
I	я	57
J	є	58
K	ї	59
L	'	60
M	'	61
N	нт	62
O	ст	63
P	ть	64
Q	дзь	12
R	зь	13
S	нь	65
T	дь	14
UZ 'a' W	а	1
UZ 'b' D	б (твердий)	15

UZ 'b' K	б (пом'якшений)	16
UZ 'c' D	ц (твердый)	38
UZ 'd' D	д (твердый)	17
UZ 'd' K	д (м'який)	18
UZ 'e' W	е	2
UZ 'f' L	ф (твердый)	43
UZ 'f' M	ф (пом'якшений)	44
UZ 'g' D	г (твердый)	19
UZ 'g' K	г (пом'якшений)	20
UZ 'h' D	г (твердый)	21
UZ 'h' K	г (пом'якшений)	22
UZ 'i' W	і	6
UZ 'j' D	дж (твердый)	23
UZ 'j' K	дж (м'який)	24
UZ 'k' L	к (твердый)	45
UZ 'k' M	к (пом'якшений)	46
UZ 'l' S	л (твердый)	28
UZ 'l' O	л (м'який)	29
UZ 'm' S	м (твердый)	30
UZ 'm' O	м (пом'якшений)	31
UZ 'n' S	н (твердый)	32
UZ 'n' O	н (м'який)	33
UZ 'o' W	о	3
UZ 'p' L	п (твердый)	47
UZ 'p' M	п (пом'якшений)	48
UZ 'q' E	ь	7
UZ 'r' S	р (твердый)	34
UZ 'r' O	р (м'який)	35
UZ 's' L	с (твердый)	49
UZ 't' L	т (твердый)	50
UZ 't' M	т (м'який)	51
UZ 'u' W	у	4
UZ 'v' S	в (твердый)	36
UZ 'v' O	в (пом'якшений)	37
UZ 'w' N	ць	66

UZ 'x' L	х (твердий)	52
UZ 'x' M	х (пом'якшений)	53
UZ 'y' W	и	5
UZ 'z' D	з (твердий)	25
UZ 'z' K	з (м'який)	26

де вказані значення у списку відносяться до фонетичних представлень (з модуля Languages.Phonetic.Ukrainian.Syllable.ArrInt8). Останній стовпець є восьмибітними цілими числами (GHC.Int.Int8), якими представляються ці ж звуки в нових модулях.

При бажанні задати кілька таких множин (до 9 включно), можна з нового рядка вказати '*' чи кілька таких символів, а тоді з наступного рядка буде нова множина значень.

Кожна множина має бути в такому порядку: [1,2,3,4,5,6,7,8,9,10,11,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,66,101]

де число відповідає останньому стовпчику в наведеній вище схемі. 101 (до версії 0.20.0.0 – це було -1, і відповідно стояло не в кінці списку, а на його початку) відповідає паузі між словами (не впливає на результати пошуку рядка). Кожне наступне значення має бути записане у файлі з нового рядка.

Тоді при виклику програми десь серед аргументів командного рядка (немає значення, де саме) вказати «+d» <шлях до файлу зі вказаними даними>. Програми прочитають ці значення і перетворять їх у відповідні значення. Як властивості потрібно тоді використовувати такі, які починаються з літери «Н», а далі злитно з нею відповідне позначення властивості. Наприклад, «Нw04», при цьому остання цифра в записі у такому випадку означатиме порядковий номер множини значень, починаючи від 1 (максимально 9).

Поряд з власними значеннями можна використовувати і задані бібліотечно, як звичайно, в режимі кількох властивостей.

Мінімальна граматики для можливого збереження смислу та зрозумілості

Програми використовують перестановки слів, які нехтують будь-якими (чи принаймні частиною) граматичних зв'язків, порядком слів тощо. Це може призвести (крім необхідності вдумування) до ситуацій коли граматично пов'язані конструкції мови розриваються, їх частини переносяться в інші місця, утворюючи нові зв'язки і змінюючи смисл тексту.

Щоб цього було менше, щоб усунути деякі з таких ефектів, програми застосовують конкатенацію слів, які мають тісний граматичний зв'язок, щоб не розривати їх при аналізі. Це дозволяє зберегти більшу смислову легкість та впізнаваність тексту, а також як побічний ефект збільшити загальну довжину рядка, який може бути проаналізований. В українській мові граматично пов'язані службові чи залежні слова йдуть переважно перед незалежним чи головним, тому застосовано приєднання цих службових чи залежних слів до наступного за ними. Повнота визначення таких випадків не є вичерпною, але розглянуті найбільш часті випадки.

Для загального випадку потрібно мати на увазі, що службові чи залежні слова можуть йти і після незалежного чи головного, тому це потрібно розглядати окремо (і приєднувати такі слова до попереднього, а не до наступного). Наразі в узагальненій версії phonetic-languages-simplified-generalized-examples-array реалізовано обидва варіанти.

Зменшена множина перестановок як варіант універсальної

За замовчуванням програми аналізують універсальну множину перестановок усіх слів та їх конкатенацій, при цьому кількість аналізованих варіантів зростає як факторіал числа таких слів чи сполучень. Текст, організований більш чи менш узгоджено щодо тієї чи іншої властивості, може кардинально відрізнитися від початкового, що ускладнює розуміння та має ефект затримки обчислень.

Для експрес-перевірки можливого покращення тексту з використанням підходу введено множини перестановок лише одного слова відносно тексту, а також двох слів чи їх сполучень як універсальні множини (додатково до повної). При роботі програм вони задаються параметром командного рядка «+р» деє серед аргументів (положення не має значення). У такому випадку використовуються зменшена множина перестановок. Кількість слів та їх сполучень, які можуть розглядатися програмою як один рядок для аналізу збільшується у такому випадку до 10. Щоб використати мінімально можливу множину перестановок, потрібно задати як наступний аргумент командного рядка "1" (лише одне слово), для попарних перестановок – "2" (парна перестановка). Тоді серед аргументів командного рядка буде вираз "+р 1" у першому випадку, "+р 2" – у другому, а для повної множини перестановок, як і раніше, можна не задавати цю групу (повна множина, таким чином, використовується за замовчуванням).

У цьому випадку аналіз відбувається значно швидше (бо значно зменшується кількість розглядуваних випадків), і текст змінюється менше, що дозволяє зберегти його більшу впізнаваність. Можливе послідовне застосування цього випадку, проте слід мати на увазі, що у випадку попарних перестановок так можна отримати не найкращий варіант з точки зору підходу, також це може зайняти у такому випадку навіть більше часу, ніж аналіз усієї множини перестановок (бо деякі варіанти будуть аналізуватися по кілька разів, чого не відбувається у першому випадку). У випадку перестановок лише 1 слова відносно тексту виглядає так, що якщо при послідовному застосуванні вибору максимального елемента двічі підряд за тими ж властивостями отримуємо один і той же варіант, то шанси, що він і буде максимальним для повної множини перестановок, збільшуються в порівнянні з випадком попарних перестановок, але оцінити збільшення (якщо ця гіпотеза дійсно вірна) складно у загальному випадку. Все ж на практиці пошук такого "мінімально покращеного" варіанту є перспективним, оскільки він може добре зберігати смисл, покращуючи ритмічну структуру. При цьому слід враховувати наступне: при пошуку максимального елемента за значенням властивості (тобто без зміни структури) якщо при аналізі зменшеної множини перестановок отриманий текст, який співпадає з початковим, то є хороші шанси, що саме цей варіант є оптимальним з точки зору розглядуваної властивості (хоча це не гарантується). І ще одне: у такому випадку досягається локальний максимум (який може бути глобальним, а може і ні). Якщо ж повторне застосування призводить до утворення іншого (уже третього) варіанту, значить, попередній локальний максимум точно не був глобальним і програма рухається в його напрямку.

Дивіться також: [73, 55, 72, 70, 75, 54, 69, 74, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]

Більше про ритм у музиці та мові

У роботі [12] розглядається музичний ритм, метр, пульс та інші подібні теми.

Під пульсом розуміється один з серій регулярно повторюваних однакових стимулів (звуків). Під метром розуміється міра кількості пульсів між більш чи менш регулярними акцентами. Акцентами названі ті пульси, які для людини видаються виділеними серед інших.

Під ритмом (музичним) у роботі розуміється те, як один чи кілька неакцентованих бітів (пульсів у метричному контексті) групуються навколо одного акцентованого. Акцентованість може виникати по-різному і не співпадає з підсиленнями звуку, а може бути викликана найрізноманітнішими факторами в музиці.

Зазначається, що немає строгих і простих правил визначення ритмічних груп, але є певні тенденції (принаймні у Західній музиці – *Прим. автора*), на які потрібно зважати. Іноді також групування може бути двозначним (багатозначним), поліваріантним, допускати водночас різні варіанти, які співіснують разом.

Групування є архітектонічним, тобто має різні рівні, які взаємопов'язані між собою. На різних рівнях групування є продуктом подібності та відмінності звуків, також близькості та роздільності, які відчуються органами чуттів і організовуються розумово.

Тут зазначу, що програми розглядають структури з однаковою кількістю складів в одній групі, але виділеними можуть бути у групі не один елемент (склад), а кілька, ледве чи не всі. Більше того, розглядається ступінь виділеності, який розглядається як міра значимості тих чи інших складів при обчисленні значень властивостей. Ці групи складів можна співвіднести з метром, тоді як різна кількість значимих складів відповідає тому факту, що метр може існувати безвідносно ритму. У класичному розглянутому підході ритмічною є група з одним акцентованим звуком. У програмах немає чіткої залежності між акцентованим складом і мірою значимості. Це означає, що пропонується інший підхід до цього питання, результати якого можуть бути плідними. Тим не менше, можна часто вважати, що групі значимих складів відповідає ритмічна група на одному з архітектонічних рівнів, і тоді структурам з максимальним значенням відповідних властивостей такого типу відповідатимуть рядки з більш регулярним повторенням ритмічних груп у їх загальних рисах, тобто більша рівномірність на вищому архітектонічному рівні композиції твору, а меншим значенням відповідних властивостей такого типу відповідатимуть нерівномірно розподілені по рядку ритмічні групи, або їх видозміни, тобто більш складний (і тому, можливо, менш впізнаваний та відчутний, менш значимий) ритмічний малюнок на вищому архітектонічному рівні. Також можна орієнтовно вважати, що для максимальних за значеннями відповідних властивостей рядків ритмічне групування пов'язане з метричним, якщо ж розглядати не максимальні елементи, то міра зв'язку між ритмічним групуванням та метричним послаблюється, тобто такі рядки тяжіють (у музичному сенсі) до «вільного» ритму Східної та народної музики та «measured rhythm» григоріанських кантів [12]. Фактично це пропонує доволі широкі можливості для дослідження зв'язку між метричною та ритмічною структурою в мові.

Відмінності між ритмічністю в музиці та мові та їх культурна обумовленість

У роботі [25] показано натомість, що замість групування на основі принципів сприйняття (голосніший звук частіше починає групу, а також подовжений звук чи інтервал частіше завершує групу) для музики можливе групування різне для представників Великобританії та Японії, тобто є значно зумовлене культурою, а також патернами у мовах носіїв. У цій же роботі наведено багато посилань на роботи, у яких розглядалися раніше той чи інший підходи. Також вказані відмінності між англійською (також українська ближча до англійської у такому контексті) та японською мовами, зокрема передумання службових частин мови головним словам в англійській (і українській), тоді як слідує після головних у японській мові.

У роботі [21] згадуються відмінності між мовами у тривалостях складів: можна поділити мови на ті, в яких структура складів більш чи менш варіативна. Це зокрема впливає на неоднаковість тривалостей звуків у різних позиціях, на явища редукції голосних у мовах, де

структура складів більш варіативна. У роботі [5] також зазначається, що, наприклад, у французькій мові є дуже помітним явище подовження тривалостей кінцевих складів, особливо у кінці фрази чи речення. Програми не враховують подібні відмінності в тривалостях складів, але їх врахування може бути задачею подальшого їх вдосконалення.

Про відмінності в ритмічності в музиці та мові йдеться в роботі [28]. У ній застосовано міждисциплінарний підхід. Зокрема аналізом мозкової активності при реакціях на неконгруентності (відступи від структури ритму) в мові показано, що вони значно впливають також на процес розуміння значення мови (подовжують час реакції та розуміння). Також подібні результати щодо впливу ритму на здатність розуміння та мови наводяться в роботі [32].

У роботі [26] розглядаються особливості афро-американської ритмічної системи в музиці, зокрема виділяється така риса, як більш насичена ритмічна розробка менших частин творів та менша варіативність між меншими частинами одного більшого твору, а також ідея виразного ритму (експресивного), в якому на противагу до рівномірного біту Західної музики вводяться приблизний еквівалент біту та значно менший за тривалість «атомічний» ритм, який, можна вважати (умовно), складається з нот приблизно $1/16$ – $1/24$ тривалостей та вносить «виразні деталі». Відзначається також явище суперпозиції різних циклічних ритмів в один загальний (добре простежується, наприклад, в Афро-Кубинській румбі), що схоже на ідею когерентності поліритмів для утворення одного ритму. Явище виразного «атомічного» ритму, його використання та характеристики показують, що є сенс вводити тривалості фонетичних явищ з високою (на перший погляд надмірною) точністю.

У роботі [50] показано, що діти віком 5-24 місяці сильніше реагують на музичні ритми, ніж мовні, також, що вони переважно викликають позитивні емоції у них, також рухову активність.

У роботі [44] розглядається, як ритми в музиці можуть викликати емоційні стани людини чи впливати на них. Зазначається, що це не автоматичний процес, що на нього впливають індивідуальні смаки, обізнаність та тренування.

У роботі [9] робиться спроба дослідити вплив мелодійних та ритмічних змін на сприйняття мелодії. Виявлено, що мелодії розгортаються повільніше (у сприйнятті), якщо у них більше змін по висоті звуків, більше несумісних змін у ритмічній структурі.

У роботі [14] пропонується підхід до визначення початку в звуковому музичному сигналі на основі аналізу також ритму.

У роботі [46] досліджується обробка мозком музичної інформації у досвідчених джаз-музикантів порівняно з некваліфікованими. Показано, що є помітні відмінності у здатності передбачити ритм. Це свідчить про можливість тренування відчуття ритму.

У роботі [10] експериментально показано, що попереднє прослуховування музики з відповідним подібним ритмом покращує здатність виявлення фонем у мовленні, а також цей ефект підсилює попередня аудіо-рухова підготовка до прослуховування мовленнєвих речень. Це може бути застосовано, наприклад, у логопедії. Також робиться припущення про обробку мовних та музичних часових (ритмічних) структур мозком за участю спільних (одних і тих же) ресурсів.

У роботі [15] досліджується зв'язок між часовою та ритмічною структурою музики на рівні ноти, пропонується спосіб, який суттєво враховує ритм, для визначення початку ноти у звуковому сигналі.

У роботі [41] експериментально показано, що якщо йдеться про осмислену фразу, то швидше сприймаються фонем у наголошених складах, а якщо фраза складається з безглузвих слів (без лексичного значення), то суттєвої різниці у часі реакції для наголошених та ненаголошених складів немає. Робиться висновок, що наголошені склади можуть передбачатися (варіант аперцепції), тоді як ненаголошені – ні (так пояснюється різниця у часі реакції). Такий результат нашоує на думку, що наявність динамічних наголосів покращує швидкість осмислення тексту (для тих, хто є носієм культури з мовою, в якій є динамічні словесні наголоси).

У роботі [34] робиться певний підсумок щодо зв'язку ритмічної структури в мові та музиці, зокрема вказується очевидний факт наявності багато чого спільного у ритмічних групуваннях у мові та в музиці, водночас це не робить їх ідентичними. Для дослідження далі пропонується розглядати вплив мовної ритмічної структури на музичну. Згадується, що наявні дані частково підтверджують, що музична ритмічна структура зазнає впливу мовної. Під мовним (лінгвістичним) ритмом розуміється поєднання кількох факторів, які впливають на часову організацію мовлення. По-перше, це чергування слів та пауз, по друге, різні тривалості складів, по-третє, чергування наголошених та ненаголошених складів. Ці фактори можуть призвести до того, що мови можуть бути ритмічно подібними чи різними. Під музичним ритмічним групуванням розуміється також групування у фрази, чи є біт, чи він є періодичним, а також метрична структура. Спільним виступає сам факт групування у фрази, а відмінності переважно у періодичності. Зокрема, автори підкреслюють, що початкова та впливова гіпотеза про розподіл мов на ті, в яких наголоси приблизно рівномірно розподілені, та ті, в яких тривалості складів приблизно однакові, тобто початки складів приблизно рівномірно розподілені, не підтверджується наявними експериментальними даними. Це створює реальні перспективи використання програм для багатьох мов. Тим не менше, лінгвісти зберегли ці поняття та продовжують описувати ними відмінності в мовах. Ведуться дискусії щодо того, чи це справді значимі поняття, чи лише два кінці одного цілого, у якому всі мови займають своє місце. Важливим також є факт відсутності у звичайних умовах періодичності в мовному ритмі, на противагу до поширеного стану в музиці. Також автори висловлюють своє переконання, що вплив мовних ритмів на музичні не є універсальним, а більше характерний для періодів та подій, коли композитори намагаються підкреслити свою національну ідентичність та приналежність.

Поради щодо використання програм

Виходячи з вищесказаного, ритмічність утворених програмами варіантів може в багатьох випадках бути помітнішою, якщо читати слова в рядках без значних пауз між ними (як один фонетичний потік), не намагаючись виділяти наголоси. Це може і не бути так, але якщо отримані варіанти виглядають не дуже ритмічними, спробуйте саме такий варіант, і тоді порівняйте, зробіть висновки, чи підходить саме таке прочитання для Вашої ситуації.

Додаткова інформація

У роботі [40] пропонується метод класифікації музики, який використовує приховані марковські ланцюги, також інформацію щодо ритмічної структури, порівняний за точністю з ручною класифікацією.

У роботі [22] пропонується порівняно ефективний спосіб визначення ритмічної подібності фрагментів музики.

У роботі [16] вивчалось, чи виділяють піаністи-виконавці систематичними варіаціями виконання різні особливості ритмічної структури, метру та мелодії. Найбільш характерними виявилися виділення ритмічного групування.

У роботі [43] автори намагаються застосувати вейвлет-аналіз для отримання характеристик ритмічних структур музичних творів та мовлення, виділено їх математичні особливості, потім застосовано отримані результати для аналізу творів. Показано, що вони мають

розумну перцептивну основу. Пропонується обговорення можливості застосування отриманих результатів до каскадного генерування ритму.

У роботі [38] розглядається статистичний поширений підхід до визначення мовної ритмічної структури, що заснований на вокальних (голосних та їх послідовностей) та інтервокальних (приголосних та їх послідовностей) інтервалах (відстанях, тривалостях) та їх варіативності. Показано, що це дозволяє краще провести типологізацію мов щодо ритмічної структури. Додатково розглядається до вже означених мов з приблизно однаковою періодичністю розподілу наголосів, мов з приблизно однаковою тривалістю складів, також мови з приблизно однаковим розподілом мор (мора – це склад з коротким голосним чи один короткий голосний, який відчувається як найменше ціле в мові замість звуку чи складу). Для оцінок використовуються варіанти індексів попарної варіативності (rPVI, nPVI) та / або середні квадратичні відхилення. Значна частина роботи присвячена описанню точок зору на питання, чи можлива кластеризація та типологізація мов за ритмом таким чином, чи вони усі утворюють певний континуум значень, розподіляючись в межах однієї великої групи. Дослідження продовжуються у цій галузі.

У роботі [17] показано, що замість тривалостей для визначення ритму можна використовувати звучність (sonority), що також дозволяє легше автоматизувати процес сегментації на ритмічні групи. Пізніші дослідження продовжують вивчати це питання, використовуючи, зокрема, приховані марковські ланцюги.

У роботі [35] продовжується вивчення впливу особливостей мовної ритмічної структури на музичну для видатних англійських та французьких композиторів минулого. Ведеться дискусія з цього питання.

У роботі [36] проводиться аналіз цього питання для німецьких та австрійських композиторів за проміжок часу близько 250 років, вказується, що є коливання, які можуть бути пояснені історичним та культурним, а не мовним впливом.

У роботі [13] продовжується ця ж тема у порівнянні також з італійськими композиторами.

У роботі [23] розглядаються теорії мелодійного музичного наголосу. Експериментальний розгляд підтверджує переважно теорію Джозефа Томассена (1982). У цій моделі (теорії) найбільш акцентованими є поворотні точки в мелодії, коли зміна висоти нот відбувається в протилежних напрямках, причому більше це помітно для повороту з переходом від зростання до спадання, ніж для повороту з переходом від спадання до зростання. Але в різних випадках може мати місце мелодійний наголос іншого роду. Показано, що мелодійний наголос може бути доволі слабким фактором у ритмічній структурі музики.

У роботі [21] робиться аналіз, чи можуть слухачі визначати за ритмами музики (мову якої вони добре знають), якою мовою ця пісня. Як виявилось, у багатьох випадках можуть. Слухачі можуть використовувати подібності та відмінності в ритмічних структурах мов для визначення, якою мовою була складена пісня.

Цікава додаткова інформація щодо ритму та суміжних музичних явищ є в роботі:

1. Adam Neely. Solving James Brown's Rhythmic Puzzle. Adam Neely. 2021.

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 --force-reinstalls install phonetic-languages-plus-0.6.0.0
cabal --reinstall --force-reinstalls install phonetic-languages-simplified-examples-array-0.20.0.0
```

Також додатково рекомендується встановити наступні пакети:

```
cabal --reinstall --force-reinstalls install r-glpk-phonetic-languages-ukrainian-durations-0.4.2.0
cabal --reinstall --force-reinstalls install mmsyn6ukr-array
```

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

Якщо є повідомлення про помилки, введіть замість update – v1-update, замість install –

```
--enable-split-sections --enable-split-objs --enable-library-stripping \
--enable-executable-stripping v1-install
```

Якщо це не допомагає, очікуйте на новіші версії пакетів, у яких має бути виправлення багу зі встановленням, або зверніться до розробника на електронну пошту olexandr543@yahoo.com.

У новіших версіях cabal планується використовувати v2-* версії команд за замовчуванням, але поки є додаткові тонкощі при роботі з різними версіями пакетів, бібліотек та програм, тому рекомендовано використовувати v1-* варіанти. Зокрема, якщо після встановлення таким способом (v2*) при спробі завантажити модуль у інтерпретаторі GHCi будуть повідомлення, що відповідні модулі є у прихованому пакеті (hidden package), то запустіть інтерпретатор з прапором -package <назва пакету з відповідним модулем>.

Якщо пакети base, parallel та heaps часто можна встановити з репозиторіїв ОС, то решту пакетів рекомендується встановлювати з сервера Hackage (наведеними вище командами).

Зміни у версії 0.17.0.0

У новій версії зроблено спробу оптимізувати обчислення перетворення українського тексту на дані для аналізу. Також виправлено дублювання значень для ліній 4 та 3 для різних типів властивостей. Якщо Ви бажаєте відтворити попередні результати для лінії 4, то тепер потрібно для цього використовувати лінію 3, натомість 4 лінія потенційно має давати більш рекомендований автором результат. Також змінено правила нумерації версій, щоб з часом дозволити більш стабільні випуски та можливість відтворити результати роботи програм для тієї чи іншої версії у майбутньому.

Зміни у версії 0.18.0.0

У новій версії були виправлені неправильно визначена зміна структури виводу (перетворення інтервалів) так, щоб програми тепер працювали, як описано. Якщо ви використовували раніше зміну структури з аргументами, то всі результати повинні бути повторно розглянуті (заново отримані) з використанням виправленої версії 0.18.0.0. Автор просить вибачення за можливі незручності. Покращена документація. Додано приклади двох нових наборів значень тривалості, доступних як готовий до використання файл за постійним посиланням: відкрити тут. Виправлено помилку з інтерпретатором рядків (див. відповідну інформацію) і додано можливість краще налаштувати поділ сполучення на частини.

Зміни у версії 0.19.0.0

У новій версії додано підтримку додаткових властивостей, а також режиму кількох властивостей для `distributionTextG`. Покращено теоретичну частину даного документа. Виправлено деякі помилки в коді. Переключено в обчисленнях середнього значення та стандартного середнього квадратичного відхилення з популяційних до вибірових версій (тепер стандартне квадратичне відхилення дещо більше). Виправлено помилку в режимі кількох варіацій, якщо Ви не працюєте в інтерактивному режимі.

Зміни у версії 0.20.0.0

У новій версії здійснено суттєву зміну функціональності, додано режим склад-як-одне-ціле (syllable-as-a-whole – SaaW), який дозволяє враховувати не лише тривалості фонем чи фонетичних явищ, але і задавати ваги складів, що дозволяє в першу чергу розрізняти наголошені та ненаголошені склади, а також враховувати широкі коло просодичних факторів, включаючи логічні наголоси, інтонацію, загальний характер вимови тощо, і може бути використане для створення також і силабо-тонічної поезії, музичних творів та ін. Також це узагальнення дозволяє аналізувати ширше коло мов як фонетичні мови. Виправлено помилки у сегментації на склади та слова. Тепер слова з дефісами та апостофом сегментуються кращим, більш відповідним чином (до потреб підходу і загалом). Виправлено помилку для неповної множини перестановок для кодування-декодування 10-го слова, яка могла приводити до неправильної роботи. Здійснено для цього перехід від десяткової до шістнадцяткової системи числення. Загалом усі зміни призводять до того, що вивід програм може частково змінитися, і тому якщо потрібна точність в обчисленнях, то результати потрібно перерахувати заново.

Робота з програмою lineVariantsG3

Перевірте, щоб папка (каталог), куди cabal встановив виконувані файли програм, була доступна для пошуку в змінній середовища PATH. У новій версії є два головних режими роботи:

1. зі складом-як-одним-цілим (SaaW), коли кожний склад розглядається як одне ціле, як базова одиниця, з можливістю задавати числову характеристику у вигляді числа типу Double; він застосовується, якщо:
 - серед позначень властивостей є хоча б одне з літерою 'a' в написанні;
 - серед аргументів командного рядка послідовно (поза межами груп +m ...-m та +a -a) введено "+s" і наступним – натуральне число, не менше від 1, яке позначає, скільки різних множин пар "склад-число" буде задано користувачем для роботи режиму;
2. звичайний (був єдиним можливим до цієї нової версії), режимом фонетичне-явище-як-одне-ціле (PhoPaaW), коли тривалості фонетичних явищ (зазвичай, фонем) задаються завчасно, вони сталі впродовж виклику пограми.

У свою чергу також кожен режим може використовуватися у двох інших режимах роботи (усього $2 \cdot 2 = 4$ варіанти комбінацій режимів):

- 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> [<WX аргумент> <чи друкувати значення властивості(ей)>  
<чи останнє слово має залишатися на своєму місці> <чи використовувати рекурсивний  
інтерактивний режим>] [<чи використовувати режим кількох джерел>]  
<числові аргументи> <тип властивості> <український текст>
```

все в один рядок, або з використанням переносу рядка в терміналі; або:

```
lineVariantsG3 <перший аргумент> [<WX аргумент> <чи друкувати значення властивості(ей)>
<чи останнє слово має залишатися на своєму місці> <чи використовувати рекурсивний
інтерактивний режим>] [<чи використовувати режим кількох джерел>]
<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 уу2 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. Тут і далі вірш цитується за: [77].

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 стосуються 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” десь 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, +f, 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:

$$+bl = +b +l$$

(це просто скорочення використання відразу обох параметрів, замість 5 символів потрібно ввести лише 3).

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

Параметр +b (+bl) та його використання

Якщо де-небудь серед аргументів командного рядка вказати аргумент у вигляді +b (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 = +b +l

(це просто скорочення використання відразу обох параметрів, замість 5 символів потрібно ввести лише 3).

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 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 03y 10 y0 35 -m +bl садок вишневий коло хати хрущі над вишнями гудуть
0
```

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 садок хрущчі надвишнями гудуть
2 вишневий хрущчі колохати садок надвишнями гудуть
3 вишневий хрущчі садок колохати надвишнями гудуть
4 вишневий хрущчі садок надвишнями колохати гудуть
5 садок вишневий колохати хрущчі надвишнями гудуть
6 хрущчі вишневий колохати садок надвишнями гудуть
7 садок вишневий хрущчі колохати надвишнями гудуть
8 хрущчі вишневий садок колохати надвишнями гудуть
9 садок вишневий хрущчі надвишнями колохати гудуть
10 хрущчі вишневий садок надвишнями колохати гудуть
11 хрущчі колохати вишневий садок надвишнями гудуть
12 садок хрущчі вишневий колохати надвишнями гудуть
13 надвишнями садок вишневий хрущчі колохати гудуть
14 садок надвишнями вишневий хрущчі колохати гудуть
15 садок хрущчі вишневий надвишнями колохати гудуть
16 колохати надвишнями садок вишневий хрущчі гудуть
17 надвишнями колохати садок вишневий хрущчі гудуть
18 надвишнями колохати хрущчі вишневий садог гудуть
19 надвишнями хрущчі колохати вишневий садог гудуть
20 садок надвишнями хрущчі вишневий колохати гудуть
21 хрущчі надвишнями садок вишневий колохати гудуть
22 садок хрущчі надвишнями вишневий колохати гудуть
23 хрущчі садок надвишнями вишневий колохати гудуть
```

```

24   колохати надвишньами садок хрушчѣи вишневий гудуть
25   колохати хрушчѣи надвишньами садок вишневий гудуть
26   надвишньами колохати садок хрушчѣи вишневий гудуть
27   надвишньами колохати хрушчѣи садок вишневий гудуть
28   хрушчѣи колохати надвишньами садок вишневий гудуть
29   надвишньами садок колохати хрушчѣи вишневий гудуть
30   надвишньами хрушчѣи колохати садок вишневий гудуть
31   садок надвишньами колохати хрушчѣи вишневий гудуть
32   хрушчѣи надвишньами колохати садок вишневий гудуть
33   надвишньами садок хрушчѣи колохати вишневий гудуть
34   надвишньами хрушчѣи садок колохати вишневий гудуть
35   садок надвишньами хрушчѣи колохати вишневий гудуть
36   хрушчѣи надвишньами садок колохати вишневий гудуть
37   хрушчѣи садок надвишньами колохати вишневий гудуть

```

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

5

садок вишневий колохати хрушчѣи надвишньами гудуть

Інтерактивний режим запису рядка у файл (+f ...)

Якщо задати серед аргументів групу з трьох у вигляді +f <шлях до файлу запису>, 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 гру виду +a ... -a, +m ... -m тощо.

У результаті може вийти щось подібне:

```

lineVariantsG3 +f hello.txt +bl +m 02y 10 0y 10 y0 40 -m садок вишневий коло хати хрущі над вишнями гудуть
0

```

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	хрущчі	колохати	садок	надвишнями	гудуть
2	колохати	вишневий	надвишнями	садок	хрущчі	гудуть
3	садок	вишневий	колохати	хрущчі	надвишнями	гудуть
4	хрущчі	вишневий	колохати	садок	надвишнями	гудуть
5	надвишнями	колохати	вишневий	хрущчі	садог	гудуть
6	хрущчі	колохати	вишневий	садок	надвишнями	гудуть
7	надвишнями	хрущчі	вишневий	колохати	садог	гудуть
8	садок	хрущчі	вишневий	колохати	надвишнями	гудуть
9	садок	надвишнями	вишневий	хрущчі	колохати	гудуть
10	надвишнями	колохати	хрущчі	вишневий	садог	гудуть
11	хрущчі	колохати	садок	вишневий	надвишнями	гудуть
12	надвишнями	хрущчі	колохати	вишневий	садог	гудуть
13	садок	надвишнями	хрущчі	вишневий	колохати	гудуть
14	хрущчі	надвишнями	садок	вишневий	колохати	гудуть
15	колохати	надвишнями	садок	хрущчі	вишневий	гудуть
16	колохати	хрущчі	надвишнями	садок	вишневий	гудуть
17	надвишнями	колохати	садок	хрущчі	вишневий	гудуть
18	надвишнями	колохати	хрущчі	садок	вишневий	гудуть
19	надвишнями	садок	колохати	хрущчі	вишневий	гудуть
20	надвишнями	хрущчі	колохати	садок	вишневий	гудуть
21	садок	надвишнями	хрущчі	колохати	вишневий	гудуть
22	хрущчі	надвишнями	садок	колохати	вишневий	гудуть

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

3

садок вишневий колохати хрущчі 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> }
```

усе в один рядок 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.

УВАГА: починаючи з версії 0.19.0.0:

Якщо Ви не використовуєте інтерактивний режим (або режим запису у файл, або рекурсивний), то варіанти будуть виведені групами, між якими буде порожній рядок.

В інтерактивному режимі (або рекурсивному, або режимі запису у файл) варіанти будуть аналізуватися для кожної комбінації по черзі, а кінцевим буде загальний з-поміж усіх.

Рекурсивний режим роботи (“+r”)

Починаючи з версії 0.9.0.0 можна запускати програму в рекурсивному інтерактивному режимі. Для цього потрібно викликати команду з параметром “+r”, наприклад, на початку після першого аргументу. У такому випадку програма буде виконуватися рекурсивно, пропонує завершити рекурсію на кожному окремому кроці. При цьому її результатом буде останній перед завершенням результат роботи. Цей режим несумісний з обмеженнями (тому що обмеження втрачають своє правильне значення і починають “зміщуватися” з потрібних частин тексту на інші), тому має використовуватися окремо від них, може бути їх альтернативою. При цьому при кожному виклику немає змоги змінити властивості та параметри виклику цих крайніх, тому обирайте їх доречно. Зміни в тексті в цьому режимі задають через так званий рядок інтерпретатора, тобто текстовий ввід, який у арифметичному виразі – числі або частці кодує наступні дії програми.

- Якщо в рядку інтерпретатора введене двоцифрове число, тоді першу цифру програма намагається витлумачити як номер першого слова, до якого застосовується зміна, а другу цифру вона намагається витлумачити як кількість слів, які потрібно конкатенувати (з'єднати разом в одне довше “слово”), включаючи вказане перше. Далі програма (якщо введені дані можна так інтерпретувати

без помилки) працюватиме з новоутвореним текстом. Відлік слів починається з 1. Наприклад, запис “12” означатиме, що програма з’єднає перше слово (цифра “1”) з наступними, щоб кількість з’єднаних була рівна 2 (цифра “2”), тобто з’єднає перші два слова в одне. “34” означатиме, що програма спробує з’єднати 4 послідовних слова, починаючи від 3-го. Якщо це неможливо зробити, програма виконає попередній етап повторно, запропонувавши ввести рядок інтерпретатора ще раз.

- Якщо введено трицифрове чи багатоцифрове число, тоді усі цифри, які не рівні 0, програма намагатиметься приписати номерам слів, які потрібно з’єднати у тому порядку, в якому цифри записані в рядку інтерпретатора.
- Якщо ввести цифру (не рівну 0 і меншу за кількість слів у рядку), потім після неї символ ділення ‘/’ і тоді ціле число зі знаком, то перша цифра до знаку ділення буде означати номер слова (починаючи з 1), яке буде розділятися на дві частини, а друге число буде кількість символів, які програма відрахує від початку цього слова (якщо число додатнє) вліво або з кінця слова вправо (якщо число від’ємне), щоб його поділити (додатній випадок схожий на функцію стандартної бібліотеки Haskell ‘splitAt’). Тоді вона, якщо вказані дані вдалося так інтерпретувати, розділить вказане слово на дві частини (одна з яких може бути порожньою, порожнім рядком) і далі працюватиме з текстом, у якому вказане слово замінено цими двома (або лише першим з них, якщо вони непорожні і кількість слів у рядку уже була рівна 7 до моменту поділу або 10 у випадку неповної можини перестановок). Наприклад, “садоквишневий” рядок інтерпретатора рівний “1/5”, якщо це перше слово у тексті, перетворить на “садок вишневий” (відрахувавши від початку 5 символів) і працюватиме з новим текстом далі. Починаючи з версії 0.18.0.0, додано перевірку програмою для довгих сполучень слів. Якщо після ‘/’ ввести число, модуль якого більший 7 (наприклад, 9 чи -15), то вона спочатку виведе на екран те слово, яке має бути розділене на дві частини згідно введеного значення (як описано щойно), додавши попереду “?: ”; якщо поділ виконаний правильно, тоді потрібно натиснути Enter. Якщо “неточний” поділ, то потрібно вказати слово (цифру 1 чи 2) і через пробіл (“змістити” пробіл вліво або вправо на) N (має бути більшим 0), це все вводиться тут же у новому рядку, і далі знову буде запит “?: ” типу. Програма буде повторювати поділ доки не буде натиснуто просто Enter. Це дозволяє не рахувати більше 8 символів підряд, намагаючись отримати поділ у потрібному місці для довгих сполучень, а швидко визначити “на око”, приблизно, а потім скорегувати. Цю опцію додано для зручності.

WX аргумент

Якщо Ви серед властивостей використовуєте “w” або “x” серію (або обидві), то для них можна задати окремий аргумент, який має починатися з “+x” з наступним записом двох додатніх чисел подвійної точності (типу Double), з’єднаних підкресленням. Наприклад, +x 2.345_0.45676237876. Якщо цей аргумент не задати, то буде використаний аргумент за замовчуванням +x 2.0_0.125.

Зверніть увагу, починаючи з версії 0.16.0.0, тепер між +x та числами є пробіл, раніше його не було.

Перше число подвійної точності буде використане як множник (чи дільник у випадку неспівпадіння) і найбільше впливає на значення властивості, стосується найважливіших складів у стопі (ритмічній групі); натомість другий буде використаний або лише для збільшення значення властивості при співпадінні менш значимих довжин складів у стопі (ритмічній групі) (що відповідає “w” серії), або також і для більш складної форми поведінки (“x” серія).

Більше деталей можна побачити у розділі Типи властивостей.

Режим використання кількох джерел

Якщо серед аргументів командного рядка перед українським текстом (чи замість нього) або після нього задати групу аргументів у обрамленні

```
+t {двозначне число} ... ^t
```

то програма працюватиме з використанням режиму кількох джерел. Замість трьох крапок можна вказати аргументи.

Поведінка програми відрізняється для випадків попарних перестановок (Див. Режим попарних перестановок, задається аргументом командного рядка «+r [1 чи 2]») та повної множини перестановок.

У першому випадку після +t програма очікує одне з чисел 10, 11, 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71, 80, 81, 90, 91, де 0 як друга цифра вказує, що програма рядки з кожного джерела не буде з'єднувати в один рядок перед поділом на частини для аналізу та перетворення, а 1 там же означає, що програма спочатку з'єднає усі рядки кожного джерела в один рядок, а потім застосує до нього поділ на частини. Перша цифра (якщо не рівна 1) означає кількість слів у кожному рядку (крім останнього) у кожному джерелі, які будуть складати нові рядки для аналізу. Якщо перша цифра рівна 1, то це еквівалентно 10 (максимальна кількість слів для цього режиму).

У другому випадку після +t програма очікує одне з чисел 20, 21, 30, 31, 40, 41, 50, 51, 60, 61, 70, 71. Значення їх аналогічне до першого випадку.

На місці трьох крапок можна вказати шляхи до файлів з українським текстом, який буде використаний для аналізу. Якщо залишити порожніми ці три крапки, то програма буде запитувати кожний новий рядок у нескінченному циклі у користувача, а далі опрацьовуватиме його згідно усіх інших аргументів. Можна також у такому випадку лише вказати +t і двоцифрове число.

Спробуйте, наприклад, такі варіанти:

```
lineVariantsG3 +r 3 w02 +t 70 "sadok.txt" "other_poem.txt" "just_text.txt" ^t
lineVariantsG3 +r 3 w02 +t 70
lineVariantsG3 +r 3 w02 +f "file_for_saving.txt" +t 50
```

де очікується, що вказані в рамках

```
+t ... ^t
```

файли існують (програма ігнорує файли, які не вдалося прочитати).

Цей режим також несумісний з обмеженнями (+a ... -a), краще замість них застосовувати рекурсивний режим.

Зверніть увагу, що в новій версії програми (з 0.16.0.0) знак множення змінений на +, між першим аргументом та числом є пробіл.

Додаткові значення

Починаючи з версії 0.19.0.0 пакету `phonetic-languages-simplified-examples-array`, є можливість використати спеціальні додаткові значення для покращення виводу для багатьох властивостей. Усі вони задаються заданням у властивості серед літер також англійські літери 'G' (вона була раніше зарезервована) та спеціальним форматом числового аргумента до неї. Вони мають особливе значення, яке суттєво відрізняється від загальної схеми обчислень.

Поетичний рядок

Розглянемо приклад, який показує, що є доцільність застосування таких додаткових значень.

Якщо Ви збережете текстовий файл з текстовою інформацією зі сторінки `56.csv` під назвою `56.csv` у робочому каталозі (папці) (там, де Ви знаходитесь в терміналі чи консолі) і виконаєте команду:

```
lineVariantsG3 +d 56.csv 11 HI32 весна  уже прийшла і спів пташок лунає звідусіль
0
пташок прийшла уже весна іспів лунає звідусіль
Infinity
Infinity
лунає пташок уже іспів прийшла весна звідусіль
Infinity
Infinity
лунає пташок весна іспів прийшла уже звідусіль
Infinity
Infinity
лунає прийшла уже весна пташок іспів звідусіль
Infinity
Infinity
звідусіль пташок уже іспів прийшла лунає весна
Infinity
Infinity
звідусіль уже прийшла лунає іспів пташок весна
Infinity
Infinity
прийшла лунає звідусіль уже іспів пташок весна
Infinity
```

Infinity

уже прийшла звідусіль іспів весна пташок лунає

Infinity

Infinity

уже весна звідусіль іспів прийшла пташок лунає

Infinity

Infinity

весна пташок звідусіль прийшла іспів уже лунає

Infinity

Infinity

пташок прийшла звідусіль іспів уже весна лунає

Infinity

Infinity

звідусіль весна пташок прийшла уже іспів лунає

Infinity

Infinity

весна пташок звідусіль прийшла уже іспів лунає

Infinity

Infinity

прийшла пташок звідусіль уже іспів весна лунає

Infinity

Infinity

уже пташок звідусіль іспів весна прийшла лунає

Infinity

Infinity

уже весна звідусіль іспів пташок прийшла лунає

Infinity

Infinity

пташок іспів весна звідусіль уже прийшла лунає

Infinity

Infinity

пташок іспів весна звідусіль лунає прийшла уже

Infinity

Infinity

звідусіль пташок весна іспів прийшла лунає уже

Infinity
Infinity
звідусіль весна пташок прийшла лунає іспів уже
Infinity
Infinity
пташок прийшла лунає звідусіль весна іспів уже
Infinity
Infinity
весна звідусіль уже лунає прийшла іспів пташок
Infinity
Infinity
уже лунає весна звідусіль прийшла іспів пташок
Infinity
Infinity
прийшла іспів уже звідусіль весна лунає пташок
Infinity
Infinity
іспів прийшла звідусіль весна уже лунає пташок
Infinity
Infinity
лунає пташок уже весна звідусіль іспів прийшла
Infinity
Infinity
пташок іспів весна звідусіль уже лунає прийшла
Infinity
Infinity
уже весна звідусіль іспів пташок лунає прийшла
Infinity
Infinity
весна іспів уже звідусіль лунає пташок прийшла
Infinity
Infinity
іспів пташок звідусіль весна уже лунає прийшла
Infinity
Infinity

весна звідусіль уже лунає пташок прийшла іспів

Infinity

Infinity

уже лунає весна звідусіль пташок прийшла іспів

Infinity

Infinity

пташок прийшла лунає звідусіль уже весна іспів

Infinity

Infinity

звідусіль прийшла весна уже пташок лунає іспів

Infinity

Infinity

іспів звідусіль весна уже пташок лунає прийшла

[illegible]

0

[illegible]

іспів звідусіль уже весна пташок лунає прийшла

[illegible][illegible]

іспів лунає уже пташок прийшла звідусіль весна

[illegible][illegible]

іспів лунає весна пташок прийшла звідусіль уже

[illegible][illegible]

іспів лунає весна прийшла пташок звідусіль уже

[illegible][illegible]

весна лунає уже прийшла пташок звідусіль іспів

[illegible][illegible]

уже весна звідусіль пташок іспів лунає прийшла

[illegible]

іспів пташок уже прийшла весна звідусіль лунає

[illegible]

іспів уже весна прийшла пташок звідусіль лунає

[illegible][illegible]

іспів звідусіль весна уже пташок лунайє прийшла
 уже лунайє весна звідусіль прийшла іспів пташок
 уже лунайє весна звідусіль пташок прийшла іспів
 пташок іспів весна звідусіль лунайє прийшла уже
 пташок іспів весна звідусіль уже лунайє прийшла
 пташок іспів весна звідусіль уже прийшла лунайє
 іспів лунайє весна прийшла пташоґ звідусіль уже
 іспів лунайє весна пташок прийшла звідусіль уже
 іспів уже весна прийшла пташоґ звідусіль лунайє
 лунайє пташок весна іспів прийшла уже звідусіль
 іспів звідусіль уже весна прийшла лунайє пташок
 іспів звідусіль уже весна пташок лунайє прийшла
 іспів прийшла звідусіль весна уже лунайє пташок
 іспів пташоґ звідусіль весна уже лунайє прийшла
 лунайє пташок уже весна звідусіль іспів прийшла
 лунайє прийшла уже весна пташок іспів звідусіль
 пташок прийшла уже весна іспів лунайє звідусіль
 уже прийшла звідусіль іспів весна пташок лунайє
 уже пташоґ звідусіль іспів весна прийшла лунайє
 пташок прийшла лунайє звідусіль весна іспів уже
 прийшла іспів уже звідусіль весна лунайє пташок
 іспів пташок уже прийшла весна звідусіль лунайє
 прийшла пташоґ звідусіль уже іспів весна лунайє
 пташок прийшла звідусіль іспів уже весна лунайє
 пташок прийшла лунайє звідусіль уже весна іспів
 лунайє пташок уже іспів прийшла весна звідусіль
 звідусіль уже прийшла лунайє іспів пташок весна
 звідусіль пташок уже іспів прийшла лунайє весна
 прийшла лунайє звідусіль уже іспів пташок весна
 іспів лунайє уже прийшла пташоґ звідусіль весна
 іспів лунайє уже пташок прийшла звідусіль весна

При цьому правило, що вище у виводі відображаються ті варіанти рядка, які мають вище значення властивості, не діє.

lineVariantsG3 +r +d 56.csv +u 01 GHI32 весна уже прийшла і спів пташок лунає звідусіль

1 пташок прийшла уже весна іспів лунаїе звідусіль
2 лунаїе пташок уже іспів прийшла весна звідусіль
3 лунаїе пташок весна іспів прийшла уже звідусіль
4 лунаїе прийшла уже весна пташок іспів звідусіль
5 звідусіль пташок уже іспів прийшла лунаїе весна
6 іспів лунаїе уже пташок прийшла звідусіль весна
7 звідусіль уже прийшла лунаїе іспів пташок весна
8 прийшла лунаїе звідусіль уже іспів пташок весна
9 іспів лунаїе уже прийшла пташоґ звідусіль весна
10 уже прийшла звідусіль іспів весна пташок лунаїе
11 іспів уже весна прийшла пташоґ звідусіль лунаїе
12 уже весна звідусіль іспів прийшла пташок лунаїе
13 весна пташоґ звідусіль прийшла іспів уже лунаїе
14 пташок прийшла звідусіль іспів уже весна лунаїе
15 звідусіль весна пташок прийшла уже іспів лунаїе
16 весна пташоґ звідусіль прийшла уже іспів лунаїе
17 прийшла пташоґ звідусіль уже іспів весна лунаїе
18 уже пташоґ звідусіль іспів весна прийшла лунаїе
19 уже весна звідусіль іспів пташок прийшла лунаїе
20 пташок іспів весна звідусіль уже прийшла лунаїе
21 іспів пташок уже прийшла весна звідусіль лунаїе
22 пташок іспів весна звідусіль лунаїе прийшла уже
23 звідусіль пташок весна іспів прийшла лунаїе уже
24 іспів лунаїе весна пташок прийшла звідусіль уже
25 іспів лунаїе весна прийшла пташоґ звідусіль уже
26 звідусіль весна пташок прийшла лунаїе іспів уже
27 пташок прийшла лунаїе звідусіль весна іспів уже
28 весна звідусіль уже лунаїе прийшла іспів пташок
29 уже лунаїе весна звідусіль прийшла іспів пташок
30 весна уже звідусіль прийшла іспів лунаїе пташок
31 прийшла іспів уже звідусіль весна лунаїе пташок
32 іспів звідусіль весна уже прийшла лунаїе пташок
33 іспів звідусіль уже весна прийшла лунаїе пташок
34 іспів прийшла звідусіль весна уже лунаїе пташок
35 весна уже звідусіль пташок іспів лунаїе прийшла

36 уже весна звідусіль пташок іспів лунає прийшла
 37 лунає пташок уже весна звідусіль іспів прийшла
 38 пташок іспів весна звідусіль уже лунає прийшла
 39 уже весна звідусіль іспів пташок лунає прийшла
 40 весна іспів уже звідусіль лунає пташок прийшла
 41 весна іспів звідусіль уже пташок лунає прийшла
 42 іспів звідусіль весна уже пташок лунає прийшла
 43 іспів звідусіль уже весна пташок лунає прийшла
 44 іспів пташоґ звідусіль весна уже лунає прийшла
 45 весна лунає уже пташок прийшла звідусіль іспів
 46 весна звідусіль уже лунає пташок прийшла іспів
 47 уже лунає весна звідусіль пташок прийшла іспів
 48 пташок прийшла лунає звідусіль уже весна іспів
 49 весна лунає уже прийшла пташоґ звідусіль іспів
 50 звідусіль прийшла весна уже пташок лунає іспів

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

23

1 пташок прийшлаужевесна іспів лунає звідусіль
 2 лунає прийшлаужевесна пташок іспів звідусіль

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

021

1 лунає прийшлаужевеснапташок іспів звідусіль

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

лунаїє прийшлаужевеснапташок іспів звідусіль

Так можна пропонувати такий рядок:

Лунає: "Прийшла уже весна пташок", – і спів звідусіль

(потрібно додати, що потрібен певний рівень образного мислення та асоціативності для розуміння такого рядка).

Якщо подивитися на початковий вивід, то це був 4 рядок. Але він міг бути і 50, програма не впорядковує у такому режимі дані, але замість 5040 варіантів на першому етапі пропонує лише 50 (читати усі не потрібно, взагалі кажучи), а далі значно менше.

Вдумливий читач помітить, що рядок вийшов багатозначний, оскільки не можна однозначно сказати, чий спів, і чиєю є пряма мова. Взагалі кажучи, якщо триматися лише фонетичних властивостей, то це збільшує можливість складати такі рядки, але ж можна і не строго триматися фонетичних властивостей, тоді на певному етапі можна обрати той варіант, який кращий саме зі смислової точки зору.

Ще один приклад.

```
lineVariantsG3 +r +d 56.csv +u 01 GHK32 весна { уже / вже } прийшла і спів пташок { лунає / звучить } \
звідусіль
```

```
0
1   пташоґ звучить іспів вжеприйшла весна звідусіль
2   пташок іспів звучить вжеприйшла весна звідусіль
3   пташоґ звучить вжеприйшла іспів весна звідусіль
4   пташоґ звучить вжеприйшла весна іспів звідусіль
5   пташок вжеприйшла весна звучить іспів звідусіль
6   пташок вжеприйшла звучидь звідусіль іспів весна
7   звідусіль весна вжеприйшла іспів пташоґ звучить
8   звідусіль весна пташок вжеприйшла іспів звучить
9   пташок вжеприйшла весна звідусіль іспів звучить
10  звідусіль пташок весна звучить іспів вжеприйшла
11  пташоґ звучидь звідусіль іспів весна вжеприйшла
12  звідусіль весна звучить вжеприйшла іспів пташок
13  звідусіль весна звучить іспів вжеприйшла пташок
14  звучить пташоґ звідусіль вжеприйшла весна іспів
15  пташоґ звідусіль звучить вжеприйшла весна іспів
16  пташок вжеприйшла весна звучидь звідусіль іспів
17  пташоґ звідусіль вжеприйшла звучить весна іспів
```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

012

1 весна звідусіль пташоґзвучить вжеприйшла іспів

2 пташоґзвучить іспів вжеприйшла весна звідусіль

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

021

(/ Ви вказали властивості(ість) та діапазон(и) для них такі, що для даних слів та їх сполучень варіантів немає. Спробуйте змінити параметри виклику програми (бібліотеки) /)

1 пташок весна іспів вжеприйшла лунайє звідусіль

2 пташок іспів весна вжеприйшла лунайє звідусіль

3 вжеприйшла лунайє пташок іспів весна звідусіль

4 лунайє весна вжеприйшла іспів пташоґ звідусіль

5 пташок вжеприйшла весна лунайє іспів звідусіль

6 вжеприйшла лунайє пташок весна іспів звідусіль

7 звідусіль лунайє пташок іспів вжеприйшла весна

8 пташоґ звідусіль вжеприйшла лунайє іспів весна

9 пташок вжеприйшла лунайє звідусіль іспів весна

10 звідусіль весна пташок вжеприйшла іспів лунайє

11 пташок вжеприйшла весна звідусіль іспів лунайє

12 звідусіль лунайє пташок іспів весна вжеприйшла

13 пташоґ звідусіль лунайє іспів весна вжеприйшла

14 лунайє звідусіль вжеприйшла іспів весна пташок

15 вжеприйшла звідусіль лунайє весна іспів пташок

16 звідусіль лунайє пташок весна вжеприйшла іспів

17 пташоґ звідусіль вжеприйшла весна лунайє іспів

18 пташоґ звыідусьіль вжеприйшла лунайе весна іспьів

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результиуючим рядком) за його номером.

4

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

лунайе весна вжеприйшла іспьів пташоґ звыідусьіль

1 пташоґ звучить прийшла весна іспьів уже звыідусьіль
 2 уже пташок прийшла весна звучить іспьів звыідусьіль
 3 пташоґ звучить прийшла весна уже іспьів звыідусьіль
 4 звучить пташок прийшла уже весна іспьів звыідусьіль
 5 весна іспьів уже пташоґ звучить прийшла звыідусьіль
 6 прийшла звыідусьіль пташок іспьів звучить уже весна
 7 пташоґ звыідусьіль прийшла іспьів звучить уже весна
 8 пташок прийшла іспьів звыідусьіль звучить уже весна
 9 пташок іспьів прийшла звыідусьіль звучить уже весна
 10 пташок іспьів звыідусьіль прийшла уже звучить весна
 11 прийшла пташоґ звыідусьіль іспьів уже звучить весна
 12 пташоґ звучидь звыідусьіль іспьів уже прийшла весна
 13 прийшла звучидь звыідусьіль іспьів уже пташок весна
 14 звучить прийшла пташок уже іспьів звыідусьіль весна
 15 пташоґ звыідусьіль звучить уже прийшла іспьів весна
 16 уже звыідусьіль весна прийшла іспьів пташоґ звучить
 17 прийшла весна звыідусьіль уже іспьів пташоґ звучить
 18 прийшла уже звыідусьіль весна іспьів пташоґ звучить
 19 прийшла уже звыідусьіль іспьів весна пташоґ звучить
 20 весна іспьів звыідусьіль прийшла уже пташоґ звучить
 21 пташок іспьів прийшла весна звыідусьіль уже звучить
 22 пташоґ звыідусьіль прийшла іспьів весна уже звучить
 23 прийшла звыідусьіль пташок іспьів весна уже звучить
 24 прийшла пташоґ звыідусьіль іспьів уже весна звучить
 25 весна прийшла пташок уже іспьів звыідусьіль звучить
 26 прийшла пташоґ звыідусьіль уже іспьів весна звучить

27 прийшла весна уже пташоґ звыідусьіль іспьів звучить
28 прийшла уже весна пташоґ звыідусьіль іспьів звучить
29 прийшла уже пташоґ звыідусьіль весна іспьів звучить
30 звыідусьіль пташок уже весна іспьів прийшла звучить
31 весна іспьів звыідусьіль пташок уже прийшла звучить
32 весна звыідусьіль звучить пташок іспьів прийшла уже
33 звыідусьіль пташок весна звучить іспьів прийшла уже
34 пташоґ звучидь звыідусьіль іспьів весна прийшла уже
35 пташок іспьів прийшла звыідусьіль звучить весна уже
36 прийшла звучидь звыідусьіль іспьів весна пташок уже
37 прийшла весна звучить пташоґ звыідусьіль іспьів уже
38 прийшла звучить весна пташоґ звыідусьіль іспьів уже
39 прийшла звучить пташоґ звыідусьіль весна іспьів уже
40 прийшла пташоґ звучидь звыідусьіль іспьів весна уже
41 прийшла пташоґ звыідусьіль весна іспьів звучить уже
42 прийшла пташоґ звыідусьіль іспьів весна звучить уже
43 прийшла звыідусьіль пташок іспьів звучить весна уже
44 пташоґ звыідусьіль звучить весна прийшла іспьів уже
45 пташоґ звучить весна прийшла звыідусьіль іспьів уже
46 пташоґ звыідусьіль прийшла іспьів звучить весна уже
47 пташок прийшла іспьів звыідусьіль звучить весна уже
48 звучить весна уже прийшла звыідусьіль іспьів пташок
49 звыідусьіль звучить уже весна прийшла іспьів пташок
50 звучить уже весна прийшла звыідусьіль іспьів пташок
51 звыідусьіль уже звучить весна прийшла іспьів пташок
52 прийшла весна звучидь звыідусьіль уже іспьів пташок
53 прийшла звучить весна звыідусьіль уже іспьів пташок
54 прийшла звучидь звыідусьіль уже іспьів весна пташок
55 прийшла звучить уже звыідусьіль іспьів весна пташок
56 прийшла весна уже звучидь звыідусьіль іспьів пташок
57 звыідусьіль прийшла уже весна звучить іспьів пташок
58 прийшла уже весна звучидь звыідусьіль іспьів пташок
59 прийшла уже звучидь звыідусьіль весна іспьів пташок
60 прийшла весна звыідусьіль уже іспьів звучить пташок
61 прийшла уже звыідусьіль весна іспьів звучить пташок

62 прийшла весна звучидь звыідусьіль іспів уже пташок
63 прийшла звучить весна звыідусьіль іспів уже пташок
64 прийшла звучидь звыідусьіль іспів уже весна пташок
65 звучить весна звыідусьіль уже іспів прийшла пташок
66 звучить уже звыідусьіль весна іспів прийшла пташок
67 звучить уже звыідусьіль іспів весна прийшла пташок
68 весна уже звыідусьіль звучить іспів прийшла пташок
69 звыідусьіль уже звучить іспів весна прийшла пташок
70 весна іспів уже звучидь звыідусьіль прийшла пташок
71 звучить весна уже пташоґ звыідусьіль іспів прийшла
72 звучить уже весна пташоґ звыідусьіль іспів прийшла
73 звучить уже пташок весна іспів звыідусьіль прийшла
74 звыідусьіль уже звучить весна пташок іспів прийшла
75 пташоґ звучить весна звыідусьіль уже іспів прийшла
76 пташоґ звучидь звыідусьіль уже іспів весна прийшла
77 пташоґ звучить уже звыідусьіль іспів весна прийшла
78 звыідусьіль пташок уже весна іспів звучить прийшла
79 пташоґ звучить весна звыідусьіль іспів уже прийшла
80 пташоґ звучидь звыідусьіль іспів уже весна прийшла
81 звучить весна звыідусьіль уже іспів пташок прийшла
82 звучить уже звыідусьіль весна іспів пташок прийшла
83 звучить уже звыідусьіль іспів весна пташок прийшла
84 весна уже звыідусьіль звучить іспів пташок прийшла
85 звыідусьіль уже звучить іспів весна пташок прийшла
86 весна іспів уже звучидь звыідусьіль пташок прийшла
87 пташоґ звучидь звыідусьіль весна уже прийшла іспів
88 пташоґ звучить весна звыідусьіль уже прийшла іспів
89 пташоґ звыідусьіль звучить уже прийшла весна іспів
90 пташоґ звыідусьіль весна уже прийшла звучить іспів
91 пташоґ звыідусьіль звучить весна прийшла уже іспів
92 пташоґ звучить весна прийшла звыідусьіль уже іспів
93 звыідусьіль звучить весна прийшла уже пташок іспів
94 прийшла весна звыідусьіль звучить уже пташок іспів
95 прийшла весна звучидь звыідусьіль уже пташок іспів
96 прийшла звучидь звыідусьіль весна уже пташок іспів

```

97    прийшла звучить весна звыідусьіль уже пташок іспьів
98    прийшла уже звыідусьіль звучить весна пташок іспьів
99    прийшла звыідусьіль весна уже пташоґ звучить іспьів
100   прийшла весна звучить пташоґ звыідусьіль уже іспьів
101   прийшла звучить весна пташоґ звыідусьіль уже іспьів
102   прийшла звучить пташоґ звыідусьіль весна уже іспьів
103   прийшла весна пташоґ звучить уже звыідусьіль іспьів
104   прийшла пташоґ звучидь звыідусьіль уже весна іспьів
105   прийшла весна звыідусьіль пташок уже звучить іспьів
106   прийшла пташок весна звыідусьіль уже звучить іспьів

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

16

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

567

```

1    уже звыідусьіль весна прийшла іспьівпташоґзвучить
2    прийшла весна звыідусьіль уже іспьівпташоґзвучить
3    прийшла уже звыідусьіль весна іспьівпташоґзвучить

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

012

```

1    прийшлавесна звыідусьіль уже іспьівпташоґзвучить

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

прийшлавесна звѣдусьіль уже іспѣвпташоґзвучить

1 прийшла лунаїе пташок іспѣв уже весна звѣдусьіль
 2 лунаїе прийшла уже пташок іспѣв весна звѣдусьіль
 3 прийшла лунаїе пташок уже іспѣв весна звѣдусьіль
 4 прийшла пташок уже лунаїе іспѣв весна звѣдусьіль
 5 прийшла лунаїе весна іспѣв уже пташоґ звѣдусьіль
 6 пташок іспѣв прийшла весна лунаїе уже звѣдусьіль
 7 лунаїе прийшла весна пташок іспѣв уже звѣдусьіль
 8 прийшла лунаїе пташок весна іспѣв уже звѣдусьіль
 9 прийшла звѣдусьіль пташок іспѣв лунаїе уже весна
 10 пташоґ звѣдусьіль прийшла іспѣв лунаїе уже весна
 11 пташок прийшла іспѣв лунаїе уже звѣдусьіль весна
 12 звѣдусьіль прийшла лунаїе пташок іспѣв уже весна
 13 звѣдусьіль лунаїе прийшла пташок іспѣв уже весна
 14 звѣдусьіль пташок лунаїе прийшла іспѣв уже весна
 15 прийшла пташоґ звѣдусьіль уже іспѣв лунаїе весна
 16 прийшла пташоґ звѣдусьіль іспѣв уже лунаїе весна
 17 лунаїе пташок уже звѣдусьіль іспѣв прийшла весна
 18 звѣдусьіль лунаїе пташок іспѣв уже прийшла весна
 19 пташоґ звѣдусьіль лунаїе уже іспѣв прийшла весна
 20 пташоґ звѣдусьіль лунаїе іспѣв уже прийшла весна
 21 лунаїе прийшла уже звѣдусьіль іспѣв пташок весна
 22 звѣдусьіль лунаїе прийшла іспѣв уже пташок весна
 23 прийшла уже звѣдусьіль лунаїе іспѣв пташок весна
 24 прийшла звѣдусьіль лунаїе уже іспѣв пташок весна
 25 прийшла звѣдусьіль лунаїе іспѣв уже пташок весна
 26 прийшла уже звѣдусьіль лунаїе пташок іспѣв весна
 27 прийшла звѣдусьіль лунаїе уже пташок іспѣв весна
 28 прийшла звѣдусьіль лунаїе пташок уже іспѣв весна
 29 прийшла уже пташоґ звѣдусьіль лунаїе іспѣв весна
 30 пташоґ звѣдусьіль лунаїе прийшла уже іспѣв весна
 31 уже звѣдусьіль весна прийшла іспѣв пташок лунаїе
 32 прийшла весна звѣдусьіль уже іспѣв пташок лунаїе
 33 прийшла уже звѣдусьіль весна іспѣв пташок лунаїе
 34 прийшла уже звѣдусьіль іспѣв весна пташок лунаїе

35 весна іспів звідусіль прийшла уже пташок лунає
36 пташок іспів прийшла весна звідусіль уже лунає
37 пташоґ звідусіль прийшла іспів весна уже лунає
38 прийшла звідусіль пташок іспів весна уже лунає
39 прийшла пташоґ звідусіль іспів уже весна лунає
40 весна прийшла пташок уже іспів звідусіль лунає
41 прийшла пташоґ звідусіль уже іспів весна лунає
42 прийшла весна уже пташоґ звідусіль іспів лунає
43 прийшла уже весна пташоґ звідусіль іспів лунає
44 прийшла уже пташоґ звідусіль весна іспів лунає
45 звідусіль пташок уже весна іспів прийшла лунає
46 весна іспів звідусіль пташок уже прийшла лунає
47 звідусіль лунає пташок іспів весна прийшла уже
48 пташоґ звідусіль лунає іспів весна прийшла уже
49 звідусіль лунає прийшла іспів весна пташок уже
50 прийшла весна звідусіль лунає іспів пташок уже
51 прийшла звідусіль лунає іспів весна пташок уже
52 лунає прийшла весна пташоґ звідусіль іспів уже
53 лунає прийшла весна пташок іспів звідусіль уже
54 звідусіль лунає прийшла пташок іспів весна уже
55 прийшла весна звідусіль лунає пташок іспів уже
56 прийшла лунає пташок весна іспів звідусіль уже
57 звідусіль прийшла лунає пташок іспів весна уже
58 прийшла лунає пташок іспів весна звідусіль уже
59 прийшла весна пташоґ звідусіль лунає іспів уже
60 прийшла пташоґ звідусіль весна іспів лунає уже
61 прийшла пташоґ звідусіль іспів весна лунає уже
62 прийшла звідусіль пташок іспів лунає весна уже
63 лунає пташок весна прийшла звідусіль іспів уже
64 звідусіль пташок лунає прийшла іспів весна уже
65 пташоґ звідусіль прийшла іспів лунає весна уже
66 пташок прийшла іспів лунає весна звідусіль уже
67 весна звідусіль уже прийшла іспів лунає пташок
68 звідусіль лунає прийшла весна уже іспів пташок
69 лунає прийшла уже звідусіль іспів весна пташок

70 прийшла звыідусьіль лунайе уже весна іспьів пташок
71 прийшла звыідусьіль лунайе уже іспьів весна пташок
72 прийшла весна уже звыідусьіль лунайе іспьів пташок
73 прийшла уже звыідусьіль лунайе іспьів весна пташок
74 прийшла весна звыідусьіль уже іспьів лунайе пташок
75 прийшла уже звыідусьіль весна іспьів лунайе пташок
76 прийшла уже весна звыідусьіль іспьів лунайе пташок
77 звыідусьіль лунайе прийшла іспьів весна уже пташок
78 звыідусьіль лунайе прийшла іспьів уже весна пташок
79 прийшла весна звыідусьіль лунайе іспьів уже пташок
80 прийшла лунайе весна іспьів уже звыідусьіль пташок
81 прийшла звыідусьіль лунайе іспьів весна уже пташок
82 прийшла звыідусьіль лунайе іспьів уже весна пташок
83 прийшла весна звыідусьіль іспьів уже лунайе пташок
84 звыідусьіль весна лунайе уже іспьів прийшла пташок
85 лунайе звыідусьіль весна уже іспьів прийшла пташок
86 лунайе весна звыідусьіль уже іспьів прийшла пташок
87 лунайе уже весна іспьів звыідусьіль прийшла пташок
88 звыідусьіль весна уже лунайе іспьів прийшла пташок
89 звыідусьіль уже лунайе весна іспьів прийшла пташок
90 уже лунайе звыідусьіль весна іспьів прийшла пташок
91 звыідусьіль уже лунайе іспьів весна прийшла пташок
92 звыідусьіль весна уже іспьів лунайе прийшла пташок
93 весна уже іспьів лунайе звыідусьіль прийшла пташок
94 уже іспьів лунайе звыідусьіль весна прийшла пташок
95 звыідусьіль весна лунайе іспьів уже прийшла пташок
96 іспьів лунайе звыідусьіль весна уже прийшла пташок
97 звыідусьіль іспьів уже лунайе весна прийшла пташок
98 звыідусьіль весна лунайе уже пташок іспьів прийшла
99 весна звыідусьіль уже пташок іспьів лунайе прийшла
100 лунайе пташок уже звыідусьіль іспьів весна прийшла
101 пташоґ звыідусьіль лунайе уже весна іспьів прийшла
102 пташоґ звыідусьіль лунайе уже іспьів весна прийшла
103 звыідусьіль лунайе пташок іспьів весна уже прийшла
104 звыідусьіль лунайе пташок іспьів уже весна прийшла

105 пташоґ звыідусьіль лунаїе іспьів весна уже прийшла
106 пташоґ звыідусьіль лунаїе іспьів уже весна прийшла
107 звыідусьіль весна лунаїе уже іспьів пташок прийшла
108 лунаїе звыідусьіль весна уже іспьів пташок прийшла
109 лунаїе весна звыідусьіль уже іспьів пташок прийшла
110 лунаїе уже весна іспьів звыідусьіль пташок прийшла
111 звыідусьіль весна уже лунаїе іспьів пташок прийшла
112 звыідусьіль уже лунаїе весна іспьів пташок прийшла
113 уже лунаїе звыідусьіль весна іспьів пташок прийшла
114 звыідусьіль уже лунаїе іспьів весна пташок прийшла
115 звыідусьіль весна уже іспьів лунаїе пташок прийшла
116 весна уже іспьів лунаїе звыідусьіль пташок прийшла
117 уже іспьів лунаїе звыідусьіль весна пташок прийшла
118 звыідусьіль весна лунаїе іспьів уже пташок прийшла
119 іспьів лунаїе звыідусьіль весна уже пташок прийшла
120 звыідусьіль іспьів уже лунаїе весна пташок прийшла
121 звыідусьіль весна лунаїе уже пташок прийшла іспьів
122 уже весна пташок лунаїе прийшла звыідусьіль іспьів
123 пташоґ звыідусьіль лунаїе уже весна прийшла іспьів
124 лунаїе пташок весна прийшла звыідусьіль уже іспьів
125 пташоґ звыідусьіль лунаїе прийшла уже весна іспьів
126 весна пташок прийшла уже лунаїе звыідусьіль іспьів
127 звыідусьіль лунаїе прийшла весна уже пташок іспьів
128 прийшла звыідусьіль лунаїе уже весна пташок іспьів
129 прийшла звыідусьіль лунаїе уже пташок весна іспьів
130 прийшла весна уже звыідусьіль лунаїе пташок іспьів
131 прийшла уже звыідусьіль лунаїе пташок весна іспьів
132 прийшла весна уже пташоґ звыідусьіль лунаїе іспьів
133 прийшла уже весна пташоґ звыідусьіль лунаїе іспьів
134 прийшла уже пташоґ звыідусьіль лунаїе весна іспьів
135 лунаїе прийшла весна пташоґ звыідусьіль уже іспьів
136 прийшла весна звыідусьіль лунаїе пташок уже іспьів
137 прийшла звыідусьіль лунаїе пташок уже весна іспьів
138 прийшла весна пташоґ звыідусьіль лунаїе уже іспьів
139 прийшла пташок весна звыідусьіль уже лунаїе іспьів

140 прийшла пташок уже звідусіль весна лунайе іспів

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

7

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

5461723

1 іспівпташокужелунайеззвідусільпрійшлавесна

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

1 -10

1 іспівпташокужелунайеззвідусільпрійшлавесна

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

1

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

1/-10

?: іспівпташокужелунайеззвідусільпр ійшлавесна

1 2

?: іспівпташокужелунайеззвідусіль прійшлавесна

(/ Ви вказали властивості(ість) та діапазон(и) для них такі, що для даних слів та їх сполучень варіантів немає. Спробуйте змінити параметри виклику програми (бібліотеки) /)

1

2 лунайе весна вжепрійшла іспів пташоґ звідусіль

3 прійшлавесна звідусіль уже іспівпташоґзвучить

4

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

лунає весна вже прийшла іспів пташоґ звідусіль

Також тут можна було б узяти іспівпташокужелунаєзвідусільприйшлавесна.

Отже, на вибір маємо варіанти:

1. іспівпташокужелунаєзвідусільприйшлавесна — І спів пташок уже лунає звідусіль: "Прийшла весна!"
2. лунає весна вже прийшла іспів пташок звідусіль — Лунає: "Весна вже прийшла!", — і спів пташок звідусіль
3. прийшлавесна звідусіль уже іспівпташокзвучить — Прийшла весна, звідусіль уже і спів пташок звучить

Вірш "Щасливе дерево"

Інший приклад.

Розглянемо вірш Олександра Сергійовича Жабенка "Щасливе дерево".

— Яке воно — щасливе дерево?

— Ти можеш бачити — цвіте воно;

Закриєш очі — плодоносить.

Радітимеш, побачивши його

і квітним, і водночас плідним.

— А навесні, коли ще квіту там нема,

Чи є щасливим дерево?

— Напевне. У надії, в тій порі,

що сонце гріє. І вранці

сяють теплі й прохолодні роси.

Приходиш ти до нього навесні

і бачиш дерево щасливим.

Прийдеш улітку, восени
і знову зустрічаєш його милим,
усміхненим, красивим і
дзвінким від співу птахів.

Уранці, ввечері чи вдень
— постійно бачиш його квітним.
Підійдеш ближче — ось воно!
Відходиш далі — хочеш бути знов.

— А взимку чи щасливе дерево?
— Так, бо земля його тримає,
А сніг тепло оберігає та радує людей,
свята Різдвяна радість огортає.
Для дерева затиш той теплий рух,
Та сили додає йому той зріст,
і гнучкість, і міцне здоров'я.

Це пори року й дерево просте. А ми — люди:
Багатші, і рухливі, і повсюди.
У щасті ми активні,
а як ні, — йдемо до нього
поступово,
людяно.

(2021 р.)

propertiesTextG3 Haskell/old/Shchaslyve_derevo.txt s 1 0 +d 56.csv HK02 | distributionTextG s 1 +W
5

32080.0000 281681185542160.0000 40037543026991875000000000.0000 8780585584.2319 1248053086876305500000.000
0 142137796494.7515 0.00000000 4 1 Яке воно щасливе дерево
8080.0000 4012012044040.0000 39087701482596790000000000000000.0000 496536144.0644 48375868171530680000000
000000.0000 97426680312844750000.0000 0.00000000 5 1 Ти можеш бачити цвіте воно
32080.0000 40046154607286250000000000.0000 40046154607286250000000000.0000 1248321527658548800000.0000 124
8321527658548800000.0000 1.0000 2.00000000 3 5 Закриєш очі плодоносить
80080.0000 33026780972305170000000000000000.0000 33026780972305170000000000000000.0000 41242233981400066

[illegible]

2986128.0000 16778002444290623000.0000 40037543026991875000000000.0000 5618648110292.1990 1340784555350335
 7000.0000 2386311.6697 0.00000084 3 1 Авзимку чищасливе дерево
 8080.0000 70305040.0000 33026780972305170000000000000000.0000 8701.1188 4087472892612026000000000000.000
 0 4697640591955452000000000.0000 0.00000000 4 1 Так боземля його тримає
 0.0000 127563422704616470000000000000.0000 Infinity Infinity Infinity Infinity 0.00000000 5 1 Асніг тепло
 оберігає тарадує людей
 186768.0000 85525776.0000 4615475325100448500000000000000000000000.0000 457.9252 2471234539696547600000000
 0000000000.0000 5396589824686827000000000000000000.0000 0.00000000 4 1 свята Різдяна радість огортає
 80080.0000 1124865040.0000 3302678097230517000000000000000000.0000 14046.7662 4124223398140006600000000000.
 0000 293606608774197230000000.0000 0.00000000 5 1 Дядерева затиx той теплий рух
 8080.0000 281217040.0000 3908827887225641000000000000000000.0000 34804.0891 4837658276269358000000000000.0
 000 1389968362950424500000000.0000 0.00000000 5 1 Тасили додає йому той зріст
 2986128.0000 64192192104040.0000 40046154607286250000000000.0000 21496798.5646 13410729415244840000.0000 6
 23847749931.6604 0.00000000 3 1 ігнуцкість іміцне здоров'я
 0.0000 0.0000 Infinity NaN Infinity Infinity 0.00000000 6 1 Цепори року йдерево просте Ами люд
 и
 281217040.0000 1124865040.0000 27799367259008490000000000.0000 4.0000 9885377948295200.0000 247135134175816
 2.5000 0.00000000 3 1 Багатші ірухливі іповсюди
 5473632528.0000 12812904023400.0000 434379570947657200000000.0000 2340.8411 7935855553430.3340 3390172674.0
 039 0.00000000 3 1 Ущасті ми активні
 16048048056040.0000 16048048056040.0000 266394626564040.0000 1.0000 16.5998 16.5998 0.11363756 2 1 айакні
 йдемо донього
 поступово
 людяно

1 2 3 4 5

29 0 0 0 2

93.55% 0.00% 0.00% 0.00% 6.45%

0.1327+-0.4989 0 31

2 3 4 5 6 7

1 14 10 5 1 0

1

12 . . . 2

10

5
1
.
~~~~~					
1	0	0	0	0	
12	0	0	0	0	2
10	0	0	0	0	0
5	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0

---

---

Як бачимо, наявність Infinity часто дозволяє отримати лише обмежену інформацію.  
Подивимося, наприклад, на наступний рядок: "Для дерева затих той теплий рух".

```
lineVariantsG3 +d 56.csv 3 НК02 Для дерева затих той теплий рух
0
  затих теплий той Длядерева рух
3302678097230517000000000000000000.0000
3302678097230517000000000000000000.0000
  затих той теплий Длядерева рух
400375430269918750000000000.0000
400375430269918750000000000.0000
  рух затих теплий той Длядерева
400375430269918750000000000.0000
400375430269918750000000000.0000
  рух затих теплий Длядерева той
400375430269918750000000000.0000
400375430269918750000000000.0000
  рух теплий затих той Длядерева
27799367259008490000000000.0000
27799367259008490000000000.0000
```

Працювати з великими числами незручно, а з Infinity у підході інтералів ще більш незручно.  
Тому скористаємося іншою додатковою властивістю.

```
lineVariantsG3 +d 56.csv +m GHK02 2118 HK02 15 -m Для дерева зати́х той те́плий рух
0
зати́х рух те́плий дльа́дерева той
зати́х те́плий той дльа́дерева рух
зати́х той те́плий дльа́дерева рух
руг зати́х те́плий дльа́дерева той
руг зати́х той дльа́дерева те́плий
рух те́плий той дльа́дерева зати́х
зати́х рух те́плий той дльа́дерева
руг зати́х те́плий той дльа́дерева
руг зати́х той те́плий дльа́дерева
рух те́плий зати́х той дльа́дерева
```

або навіть таким варіантом:

```
lineVariantsG3 +d 56.csv +m GHK02 2118 HK02 15 GHK02 23 -m Для дерева зати́х той те́плий рух
0
зати́х той те́плий дльа́дерева рух
руг зати́х те́плий дльа́дерева той
руг зати́х той дльа́дерева те́плий
руг зати́х те́плий той дльа́дерева
рух те́плий зати́х той дльа́дерева
```

Можна бачити, що з-поміж цих кількох варіантів цікавим у звучанні є "рух зати́х той для́ дерева те́плий", але заміна не вписується в загальний ритмічний малюнок вірша, тому її не можна рекомендувати тут.

## Загальна структура додаткових властивостей

Опишемо загальну структуру додаткових властивостей.

Вона так само задається буквенно-числовими та числовими аргументами, але завжди складається лише з двох компонент.

Перша — позначення типу додаткової властивості — до типу властивості додається в написанні серед літер (зазвичай, зручно з неї починати) латинська літера *G*. Наприклад, щоб застосувати додаткову властивість до властивості *Hw02*, позначаємо її *GHw02*.

Друга — антиінтервальний рядок вибірки (новий термін) — послідовність десяткових цифр, яка розуміється програмою lineVariantsG3 згідно таблиці:

Перша цифра	Тип нерівності	Друга цифра	Вираз	Параметр 1	Параметр 2
0	>				
1	<				
2	>	1	$\lg y >$		
		2	$637.0 \cdot y > base$	$base$	
		3	$100.0 \cdot \sin(ky) > 0.5 \cdot base1$	$k$	$base1$
		4	$100.0 \cdot \cos(ky) > 0.5 \cdot base1$	$k$	$base1$
		5	$1000.0 \cdot \sin(ky) > base2$	$k$	$base2$
		6	$1000.0 \cdot \cos(ky) > base2$	$k$	$base2$
		7	$100.0 \cdot \sin(ky) > -0.5 \cdot base1$	$k$	$base1$
		8	$100.0 \cdot \cos(ky) > -0.5 \cdot base1$	$k$	$base1$
		–	$y^k > base1$	$k$	$base1$
–	<	1	$\lg y <$		
		2	$637.0 \cdot y < base$	$base$	
		3	$100.0 \cdot \sin(ky) < 0.5 \cdot base1$	$k$	$base1$
		4	$100.0 \cdot \cos(ky) < 0.5 \cdot base1$	$k$	$base1$
		5	$1000.0 \cdot \sin(ky) < base2$	$k$	$base2$
		6	$1000.0 \cdot \cos(ky) < base2$	$k$	$base2$
		7	$100.0 \cdot \sin(ky) < -0.5 \cdot base1$	$k$	$base1$
		8	$100.0 \cdot \cos(ky) < -0.5 \cdot base1$	$k$	$base1$
		–	$y^k < base1$	$k$	$base1$

де  $base > 0, base \in N$ ,  
 $base1 \in [1..100], base1 \in N$ ,  
 $base2 \in [1..1000], base2 \in N$ ,  
 $k \in [0..9]$  — цифра,  $base1 > 0, base1 \in N$ .

Підкреслення в таблиці означає будь-яку іншу цифру, крім тих, які стоять вище у тому ж стовпчику.

Таким чином, наприклад, GНw02 2118 означає, що буде використана для обчислення властивість Нw02, але з неї будуть узяті лише ті значення, які без перестановки інтервалів задовольняють нерівність:  $\lg y > 18$ , тобто числа, порядок яких більше 18 (вони більші за 1 000 000 000 000 000 000) (у тому числі Infinity). Gt32 01 означає, щодля властивості t32 будуть узяті лише ті, значення для яких задовольняють нерівність  $y > 1$ . Gw02 0100000 та Gw02 215 фактично еквівалентні, але перша з них не обчислює десятковий логарифм (можна очікувати,

що вона працює дещо швидше).

## Режим склад-як-одне-ціле (SaaW)

Цей режим викликається через задання серед властивостей хоча б однієї з додатковою літерою 'а' в позначенні та додатковим заданням як аргумента командного рядка послідовно поза групами +m ...-m та +a ...-а аргументів "+s" натуральне число більше 0 — кількість множин груп пар "склад-число". Дозволяє задавати як тривалості складів, так і просодичні властивості, наприклад, логічні наголоси, динамічні наголоси, мелодичні наголоси (хоча методика останнього поки у процесі дослідження), інтонаційні особливості, загальні характеристики вимови тощо. При цьому у багатьох випадках зручною є система чисел "2-3-4-6-8-11-15", кожне з яких більше за попереднє не менше, ніж у 1,3 рази. Якщо пропустити ввід числа, то воно приймається програмою рівним 4. Можна задавати і числа з плаваючою крапкою, формат – Double (але враховуючи спосіб обчислення більшості властивостей, це скоріше всього буде надлишкова точність, і тому можна обмежитися лише цими числами).

Наприклад, якщо задавати силу наголошення складів, де склад біля наголошеного у невиділеному логічним наголосом слові відповідає значенню 4.0 за замовчуванням (натиснено просто Enter на місці порожнього числа, яке тоді приймається програмою рівним 4; 6 – наголошений склад у логічно ненаголошеному слові, 8 – наголошений склад у логічно наголошеному слові з невеликим підсиленням, 3 – склад вимовляється трохи "побіжно", він віддалений від наголошеного):

```
lineVariantsG3 3 +i a01y +s 1  десь тут була подольаночка
```

```
0
```

```
?  десь    6
```

```
?  туд
```

```
?  бу      6
```

```
?  ла
```

```
?  по      3
```

```
?  до
```

```
?  льа     8
```

```
?  но      6
```

```
?  чка
```

```
1      подольаночка дезь була тут
```

```
2      десь туд була подольаночка
```

```
3      туд була десь подольаночка
```

```
4      дезь була тут подольаночка
```

```
5      десь подольаночка туд була
```

```
6      тут подольаночка дезь була
```

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

2

десь туд була подольаночка

Звернемо увагу, що слова пишуться "у фонетичній транскрипції", тобто не за правилами орфографії, а за особливостями вимови.

**N.B.:** Це хороший спосіб побачити, як приблизно власне насправді звучить текст українською (з певними зауваженнями, про які далі по тексту).

```
lineVariantsG3 0.001_100.0 3 +i a01y +s 1 +u десь тут була подольаночка
```

0

? десь 6

? туд

? бу

? ла 6

? по 3

? до

? льа 8

? но 6

? чка

1 подольаночка дезь була тут

2 десь туд була подольаночка

3 туд була десь подольаночка

4 дезь була тут подольаночка

5 десь подольаночка туд була

6 тут подольаночка дезь була

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

десь туд була подольаночка

Цікаво, що з точки зору цієї властивості, наголоси більше ритмізуються за трискладовим паттерном, тобто більше відповідають трискладовій стопі (наразі дактилю).

Інший приклад подібного підходу:

```
lineVariantsG3 3 aw01 +s 1 садок вишневий колохати
```

0

```

? са
? док    6
? ви
? шне    6
? вий
? ко     6
? ло
? ха     6
? ти
  садок вишневий колохати
4.0000
4.0000
  вишневий колохати садок
0.8125
0.8125
  вишневий садок колохати
0.4375
0.4375

```

Розглянемо детальніше значення параметрів, з якими викликається режим. Число після "+s" має бути натуральним, не менше 1 — позначає кількість множин пар "склад-число", які кодують обчислення. Якщо вказати 1, як у прикладах вище, то така множина лише одна — отже, при виконанні буде запропоновано ввести числові значення для складів лише один раз підряд. Якщо сказати 2 — то буде запропоновано двічі, 3 — тричі і т. д. При цьому остання цифра у позначенні властивостей з літерою 'а' має бути не більшою за це число. Немає сенсу і вказувати більше число, ніж найбільша остання цифра у записі таких властивостей, бо це призведе до зайвого вводу даних, які ніде не будуть використані, що просто забере час.

Інший приклад, у даному разі з 2:

```

lineVariantsG3 100.0_-100.0 3 a02y +s 2 хрущі над вишнями гудуть
0
? хру
? шчыі    6
? на
? дви     6
? шньа
? ми     6
? гу

```

64

```
? дуть 6
? хру
? шчыі 6
? на
? дви 6
? шняа
? ми 6
? гу 6
? дуть 8
гудуть хруці надвишнями
924.0000
924.0000
хруці гудуть надвишнями
924.0000
924.0000
хруці надвишнями гудуть
624.0000
624.0000
надвишнями гудуть хруці
624.0000
624.0000
надвишнями хруці гудуть
-355.5556
-355.5556
гудуть надвишнями хруці
-355.5556
-355.5556
```

Якщо зрозуміти, що перед нами очікується двоскладова стопа, то доцільно спробувати викликати її з першим аргументом 10.0_0.0, щоб нівелювати вплив трискладовості:

```
lineVariantsG3 100.0_0.0 3 a02y +s 2 хруці над вишнями гудуть
0
? хру
? шчыі 6
? на
```



```

?  дви    6
?  шнѣа
?  ми     3
?  гу
?  дуть   6
?  хру
?  шчѣі   6
?  на
?  дви    6
?  шнѣа
?  ми     6
?  гу
?  дуть   6
хрущі надвишнями гудуть
1024.0000
1024.0000
надвишнями гудуть хрущі
1024.0000
1024.0000
гудуть хрущі надвишнями
1024.0000
1024.0000
хрущі гудуть надвишнями
1024.0000
1024.0000
надвишнями хрущі гудуть
44.4444
44.4444
гудуть надвишнями хрущі
44.4444
44.4444

```

Але бачимо також, що в максимумі знаходиться більша частина варіантів, що зменшує в загальному випадку можливості використання.

Зверніть увагу, пропонувалося двічі ввести дані для тих же складів, і використовувалися дані другого вводу.

Також схоже, що Тарас Шевченко повторює семантичну структуру першого рядка, щоб створити ідилічний гармонійний настрій. Відповідність між: хрущі – садок, вишневий – над вишнями, (є) колохати – гудуть підсилює перевагу саме обраного ним варіанта рядка.

З іншого боку, комбінована властивість з врахуванням як двоскладових, так і трискладових стоп дає наступне:

lineVariantsG3 3 a02y +s 2 хрущі над вишнями гудуть

0

? хру

? шчыі 6

? на

? дви 6

? шньа

? ми 3

? гу

? дуть 6

? хру

? шчыі 6

? на

? дви 6

? шньа

? ми 3

? гу 6

? дуть 8

хрущі надвишнями гудуть

14.2400

14.2400

надвишнями гудуть хрущі

14.2400

14.2400

гудуть хрущі надвишнями

11.2400

11.2400

хрущі гудуть надвишнями

11.2400

11.2400

надвишнями хрущі гудуть

4.4444

4.4444

гудуть надвишнями хрущі

4.4444

4.4444

## SaaW у режимі кількох властивостей

Все вищесказане стосується і режиму кількох властивостей. При цьому літера 'Н' у позначенні властивості, якщо у ній же використовується 'а', ігнорується, тобто SaaW має пріоритет над режимом власних тривалостей PhoPaaW ('Н').

Наприклад:

lineVariantsG3 100.0_0.001 +m a01x 5 02y 20 -m +s 1 +v 3 якісь цікаві ми отримуємо результати

3

? йа

? кьісь 6

? цьі

? ка 6

? вьі

? ми 6

? о

? три 6

? му

? йе 5

? мо

? ре

? зульт 6

? та 8

? ти

ми цікаві отримуємо якісь результати ->144.25->144.25

отримуємо якісь цікаві ми результати ->144.25->144.25

ми результати якісь цікаві отримуємо ->144.25->144.25

якісь отримуємо цікаві ми результати ->144.02777777777777->144.02777777777777

результати цікаві ми якісь отримуємо ->144.02777777777777->144.02777777777777

якісь результати ми цікаві отримуємо ->144.02777777777777->144.02777777777777

результати ми якісь цікаві отримуємо ->144.02777777777777->144.02777777777777

ми цікаві результати якісь отримуємо ->144.02777777777777->144.02777777777777  
цікаві якісь отримуємо ми результати ->82.36111111111111->82.36111111111111  
ми результати отримуємо цікаві якісь ->82.36111111111111->82.36111111111111  
ми отримуємо цікаві результати якісь ->82.36111111111111->82.36111111111111  
цікаві результати отримуємо ми якісь ->82.36111111111111->82.36111111111111  
ми результати якісь отримуємо цікаві ->82.36111111111111->82.36111111111111  
ми якісь результати отримуємо цікаві ->82.36111111111111->82.36111111111111  
якісь цікаві отримуємо ми результати ->81.69444444444444->81.69444444444444  
цікаві отримуємо якісь ми результати ->81.69444444444444->81.69444444444444  
отримуємо якісь ми цікаві результати ->81.69444444444444->81.69444444444444  
якісь ми отримуємо цікаві результати ->81.69444444444444->81.69444444444444  
отримуємо результати ми цікаві якісь ->81.69444444444444->81.69444444444444  
результати ми отримуємо цікаві якісь ->81.69444444444444->81.69444444444444  
цікаві ми отримуємо результати якісь ->81.69444444444444->81.69444444444444  
результати цікаві отримуємо ми якісь ->81.69444444444444->81.69444444444444  
цікаві результати якісь ми отримуємо ->81.69444444444444->81.69444444444444  
отримуємо якісь результати ми цікаві ->81.69444444444444->81.69444444444444  
результати ми якісь отримуємо цікаві ->81.69444444444444->81.69444444444444  
ми результати отримуємо якісь цікаві ->81.69444444444444->81.69444444444444  
якісь цікаві результати отримуємо ми ->81.69444444444444->81.69444444444444  
якісь цікаві ми отримуємо результати ->81.25->81.25  
отримуємо цікаві ми якісь результати ->81.25->81.25  
якісь отримуємо ми цікаві результати ->81.25->81.25  
отримуємо ми якісь цікаві результати ->81.25->81.25  
результати цікаві ми отримуємо якісь ->81.25->81.25  
цікаві ми результати отримуємо якісь ->81.25->81.25  
ми цікаві отримуємо результати якісь ->81.25->81.25  
отримуємо цікаві ми результати якісь ->81.25->81.25  
цікаві результати ми якісь отримуємо ->81.25->81.25  
ми якісь результати цікаві отримуємо ->81.25->81.25  
отримуємо якісь ми результати цікаві ->81.25->81.25  
результати якісь ми отримуємо цікаві ->81.25->81.25  
ми отримуємо результати якісь цікаві ->81.25->81.25  
отримуємо результати якісь цікаві ми ->81.25->81.25  
отримуємо якісь результати цікаві ми ->81.25->81.25

результати цікаві отримуємо якісь ми ->81.25->81.25  
цікаві отримуємо якісь результати ми ->81.25->81.25

отримуємо цікаві якісь ми результати ->10000.036->10000.036  
результати ми отримуємо цікаві якісь ->3600.144->3600.144  
отримуємо цікаві якісь результати ми ->3600.144->3600.144  
цікаві результати якісь отримуємо ми ->3600.144->3600.144  
ми отримуємо цікаві результати якісь ->3600.081->3600.081  
ми цікаві результати отримуємо якісь ->3600.081->3600.081  
результати цікаві отримуємо ми якісь ->3600.081->3600.081  
цікаві отримуємо результати ми якісь ->3600.081->3600.081  
отримуємо цікаві ми результати якісь ->3600.081->3600.081  
якісь ми результати цікаві отримуємо ->3600.081->3600.081  
ми результати цікаві якісь отримуємо ->3600.081->3600.081  
ми цікаві результати якісь отримуємо ->3600.081->3600.081  
ми якісь результати отримуємо цікаві ->3600.081->3600.081  
ми якісь отримуємо результати цікаві ->3600.081->3600.081  
результати якісь отримуємо цікаві ми ->3600.081->3600.081  
результати отримуємо якісь цікаві ми ->3600.081->3600.081  
отримуємо якісь результати цікаві ми ->3600.081->3600.081  
отримуємо якісь цікаві результати ми ->3600.081->3600.081  
результати цікаві якісь отримуємо ми ->3600.081->3600.081  
отримуємо цікаві ми якісь результати ->3600.036->3600.036  
ми якісь отримуємо цікаві результати ->3600.036->3600.036  
ми результати отримуємо цікаві якісь ->3600.036->3600.036  
ми отримуємо результати цікаві якісь ->3600.036->3600.036  
результати якісь ми цікаві отримуємо ->3600.036->3600.036  
результати ми цікаві якісь отримуємо ->3600.036->3600.036  
ми цікаві якісь результати отримуємо ->3600.036->3600.036  
результати ми якісь отримуємо цікаві ->3600.036->3600.036  
якісь результати отримуємо цікаві ми ->3600.036->3600.036  
якісь отримуємо результати цікаві ми ->3600.036->3600.036  
якісь отримуємо цікаві результати ми ->3600.036->3600.036  
отримуємо результати якісь цікаві ми ->3600.036->3600.036  
якісь цікаві результати отримуємо ми ->3600.036->3600.036

цікаві якісь результати отримуємо ми ->3600.036->3600.036  
цікаві якісь отримуємо результати ми ->3600.036->3600.036  
якісь отримуємо цікаві ми результати ->625.081->625.081  
цікаві результати отримуємо ми якісь ->625.081->625.081  
ми цікаві якісь отримуємо результати ->625.036->625.036  
якісь ми цікаві отримуємо результати ->625.036->625.036  
отримуємо якісь цікаві ми результати ->625.036->625.036  
отримуємо ми цікаві результати якісь ->400.144->400.144  
цікаві результати ми отримуємо якісь ->400.144->400.144  
цікаві ми якісь отримуємо результати ->400.081->400.081  
отримуємо ми результати цікаві якісь ->400.081->400.081  
результати цікаві ми отримуємо якісь ->400.081->400.081  
цікаві ми отримуємо результати якісь ->400.081->400.081  
результати якісь цікаві ми отримуємо ->400.081->400.081  
якісь результати цікаві ми отримуємо ->400.081->400.081  
якісь цікаві результати ми отримуємо ->400.081->400.081  
якісь цікаві ми результати отримуємо ->400.081->400.081  
цікаві якісь результати ми отримуємо ->400.081->400.081  
цікаві ми результати якісь отримуємо ->400.081->400.081  
результати ми якісь цікаві отримуємо ->400.081->400.081  
ми якісь результати цікаві отримуємо ->400.081->400.081  
отримуємо якісь ми результати цікаві ->400.081->400.081  
результати отримуємо ми якісь цікаві ->400.081->400.081  
отримуємо результати ми якісь цікаві ->400.081->400.081  
результати якісь ми отримуємо цікаві ->400.081->400.081  
результати цікаві отримуємо якісь ми ->400.081->400.081  
цікаві якісь ми отримуємо результати ->400.036->400.036  
отримуємо якісь ми цікаві результати ->400.036->400.036  
якісь ми отримуємо цікаві результати ->400.036->400.036  
отримуємо ми якісь цікаві результати ->400.036->400.036  
результати отримуємо ми цікаві якісь ->400.036->400.036  
отримуємо результати ми цікаві якісь ->400.036->400.036  
цікаві ми результати отримуємо якісь ->400.036->400.036  
результати отримуємо якісь ми цікаві ->400.036->400.036  
отримуємо результати якісь ми цікаві ->400.036->400.036

отримуємо ми якісь результати цікаві ->400.036->400.036  
 якісь ми результати отримуємо цікаві ->400.036->400.036  
 якісь ми отримуємо результати цікаві ->400.036->400.036  
 цікаві отримуємо результати якісь ми ->400.036->400.036  
 ми результати якісь цікаві отримуємо ->225.144->225.144  
 ми якісь цікаві отримуємо результати ->225.081->225.081  
 результати ми цікаві отримуємо якісь ->225.081->225.081  
 ми результати цікаві отримуємо якісь ->225.081->225.081  
 цікаві результати якісь ми отримуємо ->225.081->225.081  
 якісь результати ми цікаві отримуємо ->225.081->225.081  
 ми цікаві отримуємо результати якісь ->225.036->225.036  
 результати цікаві якісь ми отримуємо ->225.036->225.036  
 цікаві якісь ми результати отримуємо ->225.036->225.036  
 якісь ми цікаві результати отримуємо ->225.036->225.036  
 результати отримуємо цікаві якісь ми ->225.036->225.036  
 отримуємо результати цікаві якісь ми ->225.036->225.036  
 отримуємо цікаві результати якісь ми ->225.036->225.036  
 цікаві результати отримуємо якісь ми ->225.036->225.036  
 ми цікаві отримуємо якісь результати ->156.331->156.331  
 цікаві отримуємо ми результати якісь ->25.144->25.144  
 результати якісь отримуємо ми цікаві ->25.144->25.144  
 отримуємо ми результати якісь цікаві ->25.144->25.144  
 результати ми отримуємо якісь цікаві ->25.144->25.144  
 результати якісь цікаві отримуємо ми ->25.144->25.144  
 ми отримуємо цікаві якісь результати ->25.081->25.081  
 цікаві ми отримуємо якісь результати ->25.081->25.081  
 цікаві отримуємо якісь ми результати ->25.081->25.081  
 цікаві якісь отримуємо ми результати ->25.081->25.081  
 якісь отримуємо ми цікаві результати ->25.081->25.081  
 ми отримуємо якісь цікаві результати ->25.081->25.081  
 результати отримуємо цікаві ми якісь ->25.081->25.081  
 отримуємо результати цікаві ми якісь ->25.081->25.081  
 результати цікаві ми якісь отримуємо ->25.081->25.081  
 цікаві ми якісь результати отримуємо ->25.081->25.081  
 ми якісь цікаві результати отримуємо ->25.081->25.081

якісь отримуємо ми результати цікаві ->25.081->25.081  
 отримуємо якісь результати ми цікаві ->25.081->25.081  
 якісь результати ми отримуємо цікаві ->25.081->25.081  
 ми результати якісь отримуємо цікаві ->25.081->25.081  
 ми отримуємо результати якісь цікаві ->25.081->25.081  
 ми отримуємо якісь результати цікаві ->25.081->25.081  
 цікаві отримуємо якісь результати ми ->25.081->25.081  
 якісь цікаві ми отримуємо результати ->25.036->25.036  
 отримуємо ми цікаві якісь результати ->25.036->25.036  
 цікаві отримуємо ми якісь результати ->25.036->25.036  
 отримуємо цікаві результати ми якісь ->25.036->25.036  
 цікаві результати ми якісь отримуємо ->25.036->25.036  
 якісь результати отримуємо ми цікаві ->25.036->25.036  
 якісь отримуємо результати ми цікаві ->25.036->25.036  
 ми результати отримуємо якісь цікаві ->25.036->25.036  
 якісь результати цікаві отримуємо ми ->25.036->25.036  
 якісь цікаві отримуємо результати ми ->25.036->25.036  
 якісь цікаві отримуємо ми результати ->6.394->6.394

ми отримуємо результати йакьісь цыікавыі  
 ми отримуємо цыікавыі результати йакьісь  
 ми результати отримуємо цыікавыі йакьісь  
 ми результати отримуємо йакьісь цыікавыі  
 ми цыікавыі отримуємо результати йакьісь  
 ми цыікавыі отримуємо йакьісь результати  
 ми результати йакьісь отримуємо цыікавыі  
 ми йакьісь результати отримуємо цыікавыі  
 ми результати йакьісь цыікавыі отримуємо  
 ми цыікавыі результати йакьісь отримуємо  
 ми йакьісь результати цыікавыі отримуємо  
 отримуємо ми йакьісь цыікавыі результати  
 результати ми отримуємо цыікавыі йакьісь  
 цыікавыі ми отримуємо результати йакьісь  
 йакьісь ми отримуємо цыікавыі результати  
 результати ми йакьісь отримуємо цыікавыі



цікаві ми результати отримуємо йакьісь  
 результати ми йакьісь цікаві отримуємо  
 отримуємо результати ми цікаві йакьісь  
 отримуємо цікаві ми результати йакьісь  
 отримуємо йакьісь ми результати цікаві  
 отримуємо цікаві ми йакьісь результати  
 отримуємо йакьісь ми цікаві результати  
 йакьісь отримуємо ми цікаві результати  
 результати цікаві ми отримуємо йакьісь  
 результати йакьісь ми отримуємо цікаві  
 йакьісь цікаві ми отримуємо результати  
 результати цікаві ми йакьісь отримуємо  
 цікаві результати ми йакьісь отримуємо  
 йакьісь результати ми цікаві отримуємо  
 отримуємо йакьісь результати ми цікаві  
 отримуємо йакьісь цікаві ми результати  
 цікаві отримуємо йакьісь ми результати  
 йакьісь отримуємо цікаві ми результати  
 результати цікаві отримуємо ми йакьісь  
 цікаві результати отримуємо ми йакьісь  
 цікаві йакьісь отримуємо ми результати  
 йакьісь цікаві отримуємо ми результати  
 цікаві результати йакьісь ми отримуємо  
 отримуємо результати йакьісь цікаві ми  
 отримуємо йакьісь результати цікаві ми  
 цікаві отримуємо йакьісь результати ми  
 результати цікаві отримуємо йакьісь ми  
 йакьісь цікаві результати отримуємо ми

Варто зазначити, що в силу особливостей обчислення властивостей, якщо надавати складам однакові значення підряд, то менше буде значень з максимальними значеннями. Питання, чи варто занадто старатися надавати різні значення послідовним складам, залишається відкритим.

Також зазначу, що такий деталізований вивід надається за умови вказання як аргументів "+v 3". Він включає "деталізований" вивід, про який згодом. Він працює сає у режимі кількох властивостей.

Цікавим є інший приклад, який показує, як можна спробувати знайти, де кращим буде логічний наголос:

lineVariantsG3 10.0_0.0 +m a01x 5 a02x 5 a03x 5 -m +s 3 +v 3 хруці над вишнями гудуть

3

? хру 6  
 ? шчыі 8  
 ? на  
 ? дви 6  
 ? шня  
 ? ми 3  
 ? гу  
 ? дуть 6  
 ? хру  
 ? шчыі 6  
 ? на 6  
 ? дви 8  
 ? шня 6  
 ? ми  
 ? гу  
 ? дуть 6  
 ? хру  
 ? шчыі 6  
 ? на  
 ? дви 6  
 ? шня  
 ? ми 3  
 ? гу 6  
 ? дуть 8

хруці надвишнями гудуть ->14.2400000000000002->14.2400000000000002

надвишнями гудуть хруці ->14.2400000000000002->14.2400000000000002

гудуть хруці надвишнями ->11.2400000000000002->11.2400000000000002

хруці гудуть надвишнями ->11.2400000000000002->11.2400000000000002

надвишнями хруці гудуть ->4.4444444444444445->4.4444444444444445

гудуть надвишнями хруці ->4.4444444444444445->4.4444444444444445

хруці надвишнями гудуть ->14.2400000000000002->14.2400000000000002

надвишнями гудуть хрущі ->14.240000000000002->14.240000000000002  
 гудуть хрущі надвишнями ->11.240000000000002->11.240000000000002  
 хрущі гудуть надвишнями ->11.240000000000002->11.240000000000002  
 надвишнями хрущі гудуть ->4.444444444444445->4.444444444444445  
 гудуть надвишнями хрущі ->4.444444444444445->4.444444444444445

хрущі надвишнями гудуть ->14.240000000000002->14.240000000000002  
 надвишнями гудуть хрущі ->14.240000000000002->14.240000000000002  
 гудуть хрущі надвишнями ->11.240000000000002->11.240000000000002  
 хрущі гудуть надвишнями ->11.240000000000002->11.240000000000002  
 надвишнями хрущі гудуть ->4.444444444444445->4.444444444444445  
 гудуть надвишнями хрущі ->4.444444444444445->4.444444444444445

гудуть надвишньами хрущчі  
 гудуть хрущчі надвишньами  
 надвишньами гудуть хрущчі  
 хрущчі гудуть надвишньами  
 надвишньами хрущчі гудуть  
 хрущчі надвишньами гудуть

Як бачимо, результати обчислень для усіх трьох випадків не відрізняються, але загалом могли би бути іншими, що може вказувати на більшу доцільність того чи іншого логічного наголосу (який заданий зсувом вправо на одну позицію значень для шкали "2-3-4-6-8-11-15").

## Рекурсивний режим при SaaW

Рекурсивний режим характеризується двома суттєвими особливостями:

1. оскільки рядок записується у "фонетичній транскрипції", то наступний виклик обчислень буде здійснений на основі нового рядка, в якому можуть відбутися незворотні фонетичні зміни (найскладнішим є випадок ставання глухих приголосних дзвінками, характерний для української мови, для інших мов можуть бути також інші характерні зміни);
2. кожен наступний етап обчислень у режимі SaaW буде викликати введення даних для складів вручну – це продовжує загальну тривалість обчислень, збільшує роботу людини, але водночас дозволяє змінивши дані отримати видозмінені результати.

Перше ускладнення може не дуже сильно впливати на результати роботи (оскільки для 2-4 змін у довгому рядку та враховуючи, що на відмінності між глухими та відповідними дзвінками приголосними не більше 11%, то це переважно не надто сильно відображається на кінцевих результатах, хоча і впливом не можна цілком знехтувати).

Наприклад:

```
lineVariantsG3 +r 3 as01 +s 1 +u малювати щось гарне це цікаве заняття
```

```
0
? ма 3
? льу
? ва 6
? ти
? щозь 6
? гар 6
? не
? це
? цьі
? ка 6
? ве
? за
? нья 6
? тьтя
1 гарне щось цецьікаве малювати заньатьтя
2 цецьікаве щозь гарне малювати заньатьтя
3 щозь гарне цецьікаве малювати заньатьтя
4 заньатьтя щозь гарне цецьікаве малювати
5 щозь гарне заньатьтя цецьікаве малювати
6 щозь гарне цецьікаве заньатьтя малювати
7 заньатьтя гарне щось цецьікаве малювати
8 гарне заньатьтя щось цецьікаве малювати
9 щось цецьікаве гарне заньатьтя малювати
10 заньатьтя щозь гарне малювати цецьікаве
11 щозь гарне заньатьтя малювати цецьікаве
12 заньатьтя гарне щось малювати цецьікаве
13 гарне заньатьтя щось малювати цецьікаве
14 малювати щозь гарне цецьікаве заньатьтя
15 гарне щось малювати цецьікаве заньатьтя
```

```

16  шчозь гарне малъувати цецьікаве занъатьтъа
17  шчось малъувати гарне занъатьтъа цецьікаве
18  шчозь гарне малъувати занъатьтъа цецьікаве
19  малъувати цецьікаве занъатьтъа гарне шчось
20  цецьікаве малъувати занъатьтъа гарне шчось
21  малъувати цецьікаве занъатьтъа шчозь гарне
22  цецьікаве малъувати занъатьтъа шчозь гарне

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

2

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

22

```

?  це
?  цьі
?  ка    6
?  ве
?  шчо    6
?  зьгар    6
?  не
?  ма    3
?  льу
?  ва    6
?  ти
?  за
?  нъа    6
?  тьтъа

```

```

1  цецьікаве шчозьгарне малъувати занъатьтъа
2  шчозьгарне цецьікаве малъувати занъатьтъа
3  занъатьтъа шчозьгарне малъувати цецьікаве
4  шчозьгарне занъатьтъа малъувати цецьікаве
5  занъатьтъа шчозьгарне цецьікаве малъувати
6  шчозьгарне занъатьтъа цецьікаве малъувати
7  шчозьгарне цецьікаве занъатьтъа малъувати

```

```

8      мальувати щчозьгарне цецьікаве заньятьтьа
9      щчозьгарне мальувати цецьікаве заньятьтьа
10     щчозьгарне мальувати заньятьтьа цецьікаве
11     цецьікаве мальувати заньятьтьа щчозьгарне
12     мальувати цецьікаве заньятьтьа щчозьгарне

```

Будь ласка, вкажіть варіант (який Ви бажаєте, щоб він став результуючим рядком) за його номером.

8

Якщо бажаєте запустити програму (функцію) рекурсивно, змінюючи сполучення слів та букв, введіть тут закодований рядок інтерпретатора. Якщо бажаєте не використовувати програму (функцію) рекурсивно, просто натисніть Enter.

мальувати щчозьгарне цецьікаве заньятьтьа

Зверніть увагу, подовження приголосних кодується і відображається як їх повторення, у тому числі для м'яких чи пом'якшених приголосних повторюються групи з м'яким знаком ("заньятьтьа"). Також зверніть увагу, що при перестановці слів відбулися фонетичні зміни — "щось гарне" стало "щчозь гарне". Якщо б ми взяли і з'єднали у цьому варіанті перше з двох слів з іншим, де не мало би бути фонетичної зміни, то програма пропустила б це і це внесло б фонетичну похибку в результат, але завдяки співвідношенням тривалостей, похибка між дзвінками та глухими відповідними (або приблизно відповідними, як у випадку з г-х) приголосними складає не більше 11%. Кілька потенційних таких на 1 рядок не надто змінює ритмічний малюнок, а на властивості з 'а' в позначенні взагалі не впливає.

## Ідеї, які можуть допомогти

Якщо Ви бажаєте провести певне дослідження рядка, то можна застосовувати різні схеми для аналізу. Програми надають багато можливостей, тому потрібно проявити доцільність. Ось кілька порад:

1. Почати аналіз можна задавши, наприклад, послідовно різнохарактерні властивості, наприклад, для двох складів (використовуючи перший аргумент у вигляді N_0.0), трьох складів (перший аргумент у вигляді 0_N), w / x, згодом 5- чи 6-складові варіанти властивостей, можна також свої користувацькі.
2. Застосування властивостей 'а' типу потребує більше часу, зате дозволяє задати більше характеристик, а також дає переваги у силовотонічному віршуванні.
3. Цікаву інформацію можна також отримати, використовуючи деталізацію (див. нижче).
4. І також багато інших, більш творчих варіантів, зрештою з використанням має розвиватися власне відчуття мови у людини і вона поступово навчатиметься створенню гарних текстів.

## Деталізацію у режимі кількох властивостей

Якщо Ви використовуєте режим кількох властивостей (+m ...-m), то для отримання додаткової інформації можна застосувати деталізацію. Для цього потрібно вказати як аргументи командного рядка поза групами +m ...-m та +a ...-a аргументи +v число 1, 2 або 3. При програма перед виводом на екран спільних варіантів для усіх вказаних властивостей та параметрів виведе додатково ще відсортовані за String (у випадку 1), за проміжним значенням (у випадку 2), за кінцевим значенням Double (за останнім і відбувається сама робота алгоритма відбору, це у випадку варіанта 3).

Наприклад:

```
lineVariantsG3 +m as01 3 w02 10 -m +s 1 +v 1 +u малювати щось гарне це цікаве заняття
```

```
1
```

```
? ма 3
```

```
? льу
```

```
? ва 6
```

```
? ти
```

```
? щчозь 6
```

```
? гар 6
```

```
? не
```

```
? це
```

```
? цьі
```

```
? ка 6
```

```
? ве
```

```
? за
```

```
? нья 6
```

```
? тьтя
```

```
гарне заняття щось малювати цецікаве ->578.0->578.0
```

```
гарне заняття щось цецікаве малювати ->578.0->578.0
```

```
гарне щось малювати цецікаве заняття ->560.0->560.0
```

```
гарне щось цецікаве малювати заняття ->578.0->578.0
```

```
заняття гарне щось малювати цецікаве ->578.0->578.0
```

```
заняття гарне щось цецікаве малювати ->578.0->578.0
```

```
заняття щось гарне малювати цецікаве ->578.0->578.0
```

```
заняття щось гарне цецікаве малювати ->578.0->578.0
```

```
малювати цецікаве заняття гарне щось ->544.0->544.0
```

```
малювати цецікаве заняття щось гарне ->544.0->544.0
```

малювати щось гарне цецікаве заняття ->560.0->560.0  
 цецікаве малювати заняття гарне щось ->544.0->544.0  
 цецікаве малювати заняття щось гарне ->544.0->544.0  
 цецікаве щось гарне малювати заняття ->578.0->578.0  
 щось гарне заняття малювати цецікаве ->578.0->578.0  
 щось гарне заняття цецікаве малювати ->578.0->578.0  
 щось гарне малювати заняття цецікаве ->560.0->560.0  
 щось гарне малювати цецікаве заняття ->560.0->560.0  
 щось гарне цецікаве заняття малювати ->578.0->578.0  
 щось гарне цецікаве малювати заняття ->578.0->578.0  
 щось малювати гарне заняття цецікаве ->560.0->560.0  
 щось цецікаве гарне заняття малювати ->578.0->578.0

гарне малювати щось заняття цецікаве ->0.6328125->0.6328125  
 гарне малювати щось цецікаве заняття ->3.65625->3.65625  
 заняття малювати щось цецікаве гарне ->0.4921875->0.4921875  
 заняття щось цецікаве гарне малювати ->0.421875->0.421875  
 малювати гарне заняття цецікаве щось ->3.0625->3.0625  
 малювати гарне цецікаве щось заняття ->0.6640625->0.6640625  
 малювати щось гарне заняття цецікаве ->0.53125->0.53125  
 малювати щось заняття гарне цецікаве ->3.0625->3.0625  
 малювати щось заняття цецікаве гарне ->0.4375->0.4375  
 малювати щось цецікаве гарне заняття ->0.53125->0.53125  
 цецікаве гарне заняття малювати щось ->0.4375->0.4375  
 цецікаве гарне малювати щось заняття ->0.6328125->0.6328125  
 цецікаве щось гарне малювати заняття ->0.4375->0.4375  
 цецікаве щось заняття малювати гарне ->0.4375->0.4375  
 щось гарне малювати заняття цецікаве ->0.4375->0.4375  
 щось заняття малювати гарне цецікаве ->3.71875->3.71875  
 щось цецікаве гарне заняття малювати ->0.40625->0.40625

щчозь гарне малъувати занъатьтъа цецьікаве  
 щчось цецьікаве гарне занъатьтъа малъувати  
 цецьікаве щчозь гарне малъувати занъатьтъа



При цьому стрілки показують у будь-якому разі хід обчислень значень властивостей, а порядок групування відповідає порядку, в якому задані в командному рядку властивості. Тут сортування за самим рядком (+v 1), рядки відсортовані за зростанням, а числа (у випадках +v 2 / +v 3) — за спаданням.

## Робота з програмою unconcatUkr

При виконанні програм lineVariantsG3, rewritePoemG3 утворюються конкатенації слів для збереження мінімальної граматики та кращого розуміння тексту. Деякі з них є типовими і їх можна легко відрізнити від звичайних (неконкатенованих) слів. Тому для прискорення редагування текстів після використання програм можна застосувати ще одну програму — unconcatUkr.

```
unconcatUkr 1 <шлях до файлу з конкатенованими сполученнями слів> [<шлях до нового файлу>]  
unconcatUkr 2 <шлях до файлу з конкатенованими сполученнями слів> [<шлях до нового файлу>]  
unconcatUkr -i
```

це три принципові способи використання цієї утиліти.

Перший найбільш безпечний, застосовує лише ті розділення сполучень, які майже напевно не призведуть до неправильних слів. Другий більш ризикований, але має більший ефект, тим не менше, застосовуйте з обереженістю (деякі не настільки широко вживані слова будуть неправильно розділені при цьому).

За замовчуванням, якщо цифра не вказана, то вона вважається 1.

# 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> [<WX аргумент> <чи “вирощувати рядки”>] <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 sadok.txt s 1 0 03z  
4  
8 8 8 1.0029 1.0029 1.0000 1.00143062 2 4  
Тарас ШЕВЧЕНКО  
Вказематі  
108 113 244 1.0425 2.2559 2.1640 0.64036546 3 1  
Садок вишневий колохати  
13 29 29 2.3234 2.3234 1.0000 1.39819820 3 4  
Хрущі надвишнями гудуть  
5 12 32 2.2321 6.0223 2.6980 0.63572791 3 1  
Плугатарі зплугами йдуть  
108 108 110 1.0000 1.0182 1.0182 0.99097861 3 1  
Співають ідучи дівчата  
12 13 49 1.0984 4.0438 3.6814 0.43556382 3 1
```

Аматері вечерять ждуть

27 29 110 1.0685 4.0428 3.7835 0.42378641 3 1

Сем'я вечери колохати

12 12 27 1.0000 2.2500 2.2500 0.61538462 3 1

Вечірня зіронька встає

3 14 14 4.3608 4.3608 1.0000 1.62692308 3 4

Дочка вечерять подає

248 508 508 2.0437 2.0437 1.0000 1.34289977 3 4

Амати хоче научати

1 13 13 24.6349 24.6349 1.0000 1.92198142 3 4

Так соловейко не дає

108 112 113 1.0346 1.0438 1.0089 1.01242119 3 4

Поклала мати колохати

1 13 13 25.1071 25.1071 1.0000 1.92339261 3 4

Маленьких діточок своїх

3 4 4 1.1995 1.3353 1.1132 1.02728205 3 3

Сама заснула колойіх

108 108 248 1.0000 2.2937 2.2937 0.60721063 4 1

Затихло все тільки дівчата

3 13 13 4.0000 4.0000 1.0000 1.60000000 2 4

Тасоловейко незатих

Міжітравня

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 крапки; 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 cabal встановив виконувані файли програм, була доступна для пошуку у змінній середовища PATH. Далі введіть у терміналі команду:

```
propertiesTextG3 <перший аргумент> [<WX аргумент> <чи “вирощувати рядки”>] <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 порядкові
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> [<WX аргумент> <чи
“вирощувати рядки”>] <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 порядкові дані>"
```

Ви побачите щось схоже на:

```
propertiesTextG3 2.1_3.0 sadok.txt s 1 0 02y | distributionTextG s 1 +W
```

	4							
3	3	5	1.0000	1.7000	1.7000	0.74074074	2	1

## Вказематі

108	108	142	1.0000	1.3111	1.3111	0.86538462	3	1
Садок вишневий колохати								
4	34	34	8.5180	8.5180	1.0000	1.78987107	3	4
Хрущі надвишнями гудуть								
3	25	25	8.1680	8.1680	1.0000	1.78184991	3	4
Плугатарі зплугами йдуть								
108	108	142	1.0000	1.3111	1.3111	0.86538462	3	1
Співають ідучи дівчата								
3	4	25	1.3111	8.1680	6.2298	0.28601900	3	1
Аматері вечерять ждуть								
108	245	245	2.2694	2.2694	1.0000	1.38827528	3	4
Сем'я вечеря колохати								
3	34	34	11.1680	11.1680	1.0000	1.83563445	3	4
Вечірня зіронька встає								
3	34	34	11.1680	11.1680	1.0000	1.83563445	3	4
Дочка вечерять подає								
108	277	277	2.5611	2.5611	1.0000	1.43837754	3	4
Амати хоче научати								
3	3	13	1.0000	4.3111	4.3111	0.37656904	3	1
Так соловейко недає								
108	142	245	1.3111	2.2694	1.7309	0.80203908	3	1
Поклала мати колохати								
3	3	25	1.0000	8.1680	8.1680	0.21815009	3	1
Маленьких діточок своїх								
3	12	25	4.0000	8.1680	2.0420	0.87260035	3	2
Сама заснула колойіх								
27	29	242	1.0778	8.9778	8.3299	0.21603563	4	1
Затихло все тільки дівчата								
4	4	12	1.0000	3.0508	3.0508	0.49372385	2	1
Тасоловейко незатих								
Міжітравня								
С-Петербург								

1	2	3	4			
9	1	0	6			
56.25%		6.25%	0.00%	37.50%		
0.9879+-0.6062		0	16			
2	3	4	5	6	7	
2	13	1	0	0	0	
*****						
2	.	.	.			
6	1	.	6			
1	.	.	.			
.	.	.	.			
.	.	.	.			
.	.	.	.			
~~~~~						
2	0	0	0			
6	1	0	6			
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.

1. Червоний колір – Десятковий дріб з похибкою – це 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.

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!”,

де “^” означає символ табуляції.

Цей напис означає, що вказаний текст не підходить для аналізу програмою, а також, що 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.

2. Синій колір – Натуральні числа – 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 замінює 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 (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.

3. Оранжевий колір – Цілі невід’ємні числа – 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).
4. Зелений колір – Відсотки – 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 побудувати гістограму.

5. Жовтий колір – Натуральні числа – Номери інтервалів. Починається відлік з 1. Під ними відповідні кількості рядків, значення властивості для яких згідно даних потрапляє у відповідний за номером інтервал.
6. Коричневий колір – Натуральні числа – Кількість слів у рядках. Лежить в межах від 2 до 7 включно (якщо слів менше, тоді рядок отримує значення рівні 1.0 і вилючається 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 yy3 -m
```

2	4	2	1	2	4	4	1	Тарас ШЕВЧЕНКО
							4	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	1	1	1	Співають ідучи дівчата
3	2	2	1	1	1	1	1	Аматері вечерять ждуть
3	1	3	4	1	3	1	2	Сем'я вечеря колохати
3	3	4	4	4	4	4	4	Вечірня зіронька встає
3	4	2	4	4	4	4	4	Дочка вечерять подає
3	4	4	4	1	4	2	1	Амати хоче научати
3	4	1	1	4	1	4	3	Так соловейко недає
3	4	4	2	1	2	4	1	Поклала мати колохати
3	1	4	1	4	1	2	4	Маленьких діточок своїх
3	1	4	2	4	1	2	4	Сама заснула колойіх
4	3	1	1	2	1	2	2	Затихло все тільки дівчата
2	4	4	1	1	1	1	1	Тасоловейко незатих
								Міжітравня
								С-Петербург

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 статистики за кількома властивостями для distributionTextG

У такому випадку також можна використати програму distributionTextG, додавши до рядка параметр +m:
propertiesTextG3 sadok.txt s 1 0 +m y0 0y 02y 03y y2 y3 yy3 -m | distributionTextG s 1 +W +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 1 1 1 Співають ідучи дівчата
3 2 2 1 1 1 1 1 Аматері вечерять ждуть
3 1 3 4 1 3 1 2 Сем'я вечеря колохати
3 3 4 4 4 4 4 4 Вечірня зіронька встає
3 4 2 4 4 4 4 4 Дочка вечерять подає
3 4 4 4 1 4 2 1 Амати хоче научати
3 4 1 1 4 1 4 3 Так соловейко недає
3 4 4 2 1 2 4 1 Поклала мати колохати
3 1 4 1 4 1 2 4 Маленьких діточок своїх
3 1 4 2 4 1 2 4 Сама заснула колойіх
4 3 1 1 2 1 2 2 Затихло все тільки дівчата
2 4 4 1 1 1 1 1 Тасоловейко незатих
Міжітравня
С-Петербург
-----
1 2 3 4
3 2 2 9
18.75% 12.50% 12.50% 56.25%
0 16
2 3 4 5 6 7
2 13 1 0 0 0
*****
. . . 2
3 2 1 7
. . 1 .
```

. . . .


~~~~~  
 0 0 0 2  
 3 2 1 7  
 0 0 1 0  
 0 0 0 0  
 0 0 0 0  
 0 0 0 0

---

1 2 3 4  
 3 3 1 9  
 18.75% 18.75% 6.25% 56.25%  
 0 16  
 2 3 4 5 6 7  
 2 13 1 0 0 0

\*\*\*\*\*  
 . 1 . 1  
 2 2 1 8  
 1 . . .  
 . . . .  
 . . . .  
 . . . .

~~~~~  
 0 1 0 1
 2 2 1 8
 1 0 0 0
 0 0 0 0
 0 0 0 0
 0 0 0 0

1 2 3 4

8 2 0 6
50.00% 12.50% 0.00% 37.50%
0 16
2 3 4 5 6 7
2 13 1 0 0 0

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

~~~~~  
2 0 0 0  
5 2 0 6  
1 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

=====

---

1 2 3 4  
7 3 0 6  
43.75% 18.75% 0.00% 37.50%  
0 16  
2 3 4 5 6 7  
2 13 1 0 0 0

\*\*\*\*\*  
1 1 . .  
6 1 . 6  
. 1 . .  
. . . .  
. . . .  
. . . .

~~~~~

```

1 1 0 0
6 1 0 6
0 1 0 0
0 0 0 0
0 0 0 0
0 0 0 0

```

```

1 2 3 4
7 1 1 7
43.75% 6.25% 6.25% 43.75%
0 16
2 3 4 5 6 7
2 13 1 0 0 0

```

```

*****

```

```

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

```

```

~~~~~
1 0 0 1
5 1 1 6
1 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

```

```

1 2 3 4
5 5 0 6
31.25% 31.25% 0.00% 37.50%
0 16
2 3 4 5 6 7

```

2 13 1 0 0 0

1 . . 1
4 4 . 5
. 1 . .
. . . .
. . . .
. . . .
~~~~~  
1 0 0 1  
4 4 0 5  
0 1 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0  
=====

-----  
1 2 3 4  
8 2 1 5  
50.00% 12.50% 6.25% 31.25%  
0 16  
2 3 4 5 6 7  
2 13 1 0 0 0  
\*\*\*\*\*  
2 . . .  
6 1 1 5  
. 1 . .  
. . . .  
. . . .  
. . . .  
~~~~~  
2 0 0 0
6 1 1 5
0 1 0 0
0 0 0 0

0 0 0 0
0 0 0 0

У такому випадку програма видасть можливу статистику за кожною властивістю, починаючи від першої зліва направо. Тут було 7 властивостей, тому і сім виведених розподілів.

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 sadok.txt s 1 0 03y +b | distributionTextG s 1 +W
```

```

4
Тарас ШЕВЧЕНКО
Вказематі
52      52      81      1.0000      1.5577      1.5577      0.78195489      3      1
Садок вишневий колохати
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 вечерея 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	11	11	11.2400	11.2400	1.0000	1.83660131	3	4
Маленьких діточок своїх								
1	4	4	4.4444	4.4444	1.0000	1.63265306	3	4
Сама заснула колоїїх								
9	52	100	5.7778	11.1111	1.9231	0.95412844	4	2
Затихло все тільки дівчата								
Тасоловейко незатих								
Міжітравня								
С-Петербург								

1	2	3	4												
4	3	0	7												
28.57%		21.43%	0.00%	50.00%											
1.1949+-0.4206		0	14												
2	3	4	5	6	7										
0	13	1	0	0	0										

.	.	.	.												
4	2	.	7												
.	1	.	.												
.	.	.	.												
.	.	.	.												
.	.	.	.												
~~~~~															
0	0	0	0												
4	2	0	7												
0	1	0	0												
0	0	0	0												
0	0	0	0												



0                      0                      0                      0

---

“Білі” рядки тут показані як відступи без статистики там, де інакше вона була б.

## Режим зменшеної множини перестановок

У режимі зменшеної множини перестановок можуть працювати обидві програми `propertiesTextG3` та `distributionTextG`. Тоді потрібно за-  
давати в першій з них як аргумент командного рядка «+р 1 чи 2», а в другій достатньо лише «+р». У такому випадку статистика виводиться  
так, наче кількість слів може бути від 2 до 10 включно.

## 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 все-  
редині інтервалів.

## Контроль кількості інтервалів

Можливі три випадки:

- “s” – кількість інтервалів буде визначена за відомим правилом Стерджеса, де кількість випробувань буде рівною результуючій кількості рядків;
- “l” – 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.

## Чи використовувати “вирощування рядків”

Якщо серед аргументів командного рядка задати “+g ab”, де a, b – деякі цифри, крім 0, то буде використаний режим “вирощування рядків”. Це означає, що текст буде перетворений таким чином, щоб спочатку максимальна кількість слів у рядках була не більша за другу цифру, а потім рядки згруповані так, щоб кількість слів у рядку була близькою до першої цифри, якщо цього можна досягнути, об’єднуючи рядки в один послідовно. Іншими словами відбудеться перегруповання тексту за рядками так, щоб зробити кількість слів у кожному з них ближчою до першої цифри і не більшою за 7 (для останнього використовується остача від ділення на 7). У такому випадку контроль розбивки тексту на рядки не грає великої ролі, але може перегрупувати рядки, якщо він рівний 1. Наприклад, “+g 73” як аргумент командного рядка означатиме, що після застоування розбивки тексту на рядки усі рядки буде поділено так, щоб у кожному було не більше 3 слів чи їх поєднань, а потім об’єднано рядки так, щоб у кожному було число слів близьке до 7 (не більше).

Зверніть увагу, що, починаючи з версії 0.16.0.0 між +g та двоцифровим числом стоїть пробіл. Раніше його не було.

## 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 порядкову, 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.

Програма rewritePoemG3, починаючи з версії пакету 0.12.0.0, може працювати у режимі кількох властивостей, або в режимі однієї властивості, або в порівняльному режимі.

## Режим кількох властивостей (+m ...-m)

Якщо серед аргументів командного рядка задати групу аргументів, виділену розділювачами +m та -m, всередині якої вказати різні властивості в їх кодуванні (див. Типи властивостей), відділяючи кожен пробілом від попередньої та наступної, а також не вказувати серед аргументів командного рядка "+s", то програма працюватиме в режимі кількох (можливий випадок також і однієї, якщо вказати лише її одну) властивостей.

Синтаксис роботи програми у такому режимі:

```
rewritePoemG3 <перший аргумент> <файл з українським текстом> [<чи використовувати "виращування рядків">] +m <типи властивостей> -m <числові аргументи>
```

Після успішного завершення виконання програми (не має бути жодних повідомлень) у тій же папці (каталозі), де і файл з текстом, що переписується, мають бути файли з додатковим закінченням .new.txt та префіксами, відділеними крапкою, кожний з яких відповідає введеній властивості серед вказаних. Саме у кожному з цих файлів записаний (доданий, якщо бути точними) перетворений текст (наприклад, вірш) згідно введених даних.

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

## Режим однієї властивості

Якщо серед аргументів командного рядка не задавати групу аргументів, виділену розділювачами +m та -m, а також не вказувати серед аргументів командного рядка "+s", то програма працюватиме в режимі однієї властивості (він можливий також, якщо в режимі кількох

властивостей вказати всередині групи лише одну властивість).

Синтаксис роботи програми у такому режимі:

```
rewritePoemG3 <перший аргумент> <файл з українським текстом> [<чи використовувати “вирощування рядків”>] <тип властивості> <числові аргументи>
```

та натисніть Enter.

Ви побачите щось на зразок наступного:

```
rewritePoemG3 10.0_1.2 sadok.txt yy 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 та префіксом, відділеним крапкою, який відповідає введеній властивості. 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

У режимі кількох властивостей та режимі однієї властивості числові аргументи мають багато в чому схоже значення, як і для програми lineVariantsG3.

Якщо вони задані, то 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 (+с)

Програму rewritePoemG3 можна запустити також у так званому “порівняльному” режимі, коли вона пропонує рядки (один за одним) з кількох (не більше 7) заданих файлів і записує вибраний (або порожній рядок, якщо жодний не обрано) у кінцевий файл. Так з кількох файлів шляхом їх порядкового порівняння можна створити новий. Це також дозволяє, запустивши програму двічі чи кілька разів з різними параметрами у режимі кількох властивостей, 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, чи навіть рекурсивний режим кількох джерел тієї ж програми.

Для роботи у порівняльному режимі (comparative mode):

```
rewritePoemG3 +с <файли, відділені пробілами, з яких будуть зчитуватися рядки> <кінцевий файл>
```

# 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, наприклад), 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, в якому кожне нове слово пропонує все нові звуки, (імовірно) у залежності від середньої кількості звуків у “періоді унікальності” його може бути легше чи важче промовляти; меншому значенню – навпаки – текст, в якому більше повернень у новому слові до звуків попереднього чи попередніх, таким чином, може бути більше повторів, акцентування в певних місцях на скупченнях певних звуків, що імовірно більш характерно для інтонаційно виділених і / чи поетичних текстів з закликами чи підсиленними емоціями. При використанні цієї властивості 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 `mmsynb ukr-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`. Можна створити інші варіанти тривалостей звуків, використовуючи можливості пакету `r-glpkphonetic-languages-ukrainian-durations` або іншим способом.
- “03y” – подібна до “02y” властивість, яка використовує інші тривалості звуків, синтезовані завдяки пакету `r-glpk-phoneticlanguages-ukrainian-durations`. Можна створити інші варіанти тривалостей звуків, використовуючи можливості пакету `r-glpkphonetic-languages-ukrainian-durations` або іншим способом.

- “04у” – подібна до “02у” властивість, яка використовує інші тривалості звуків, синтезовані завдяки пакету r-glpk-phoneticlanguages-ukrainian-durations. Ці тривалості звуків отримані з інших даних, ніж 0у, 02у та 03у, тому обережно змішуйте їх з ними в режимі кількох метрик.
- “у” – властивість, яка обчислює властивості “у0” та “0у” у більш ефективний спосіб, ніж кожен з них поодиночку, а потім перемножує отримані дані. Дає більші значення для рядків з більш віддаленими одне від одного словами з однаковими звуками та такі, які більше ритмізовані (з точки зору властивості “0у”). Використання <першого коефіцієнта> внутрішньо впливає лише на підвластивість “0у”.
- “у2” – властивість, подібна до у, але використовує замість другої підвластивості (ритмічності) варіант з “02у”.
- “у3” – властивість, подібна до у, але використовує замість другої підвластивості (ритмічності) варіант з “03у”.
- “у4” – властивість, подібна до у, але використовує замість другої підвластивості (ритмічності) варіант з “04у”.
- “уу” – властивість, яка використовує властивість “у0” та “0у”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “0у”) та групуванням однакових звуків у групі ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “0у”.
- “уу2” – властивість, яка використовує властивість “у0” та “02у”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “02у”) та групуванням однакових звуків у групі ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “02у”.
- “уу3” – властивість, яка використовує властивість “у0” та “03у”, при цьому замість їх перемноження, ділить результат другої на результат першої. Максимізується для текстів з високою ритмічністю (з точки зору властивості “03у”) та групуванням однакових звуків у групі ближче один до одного. Використання <першого коефіцієнта> впливає лише на підвластивість “03у”.
- будь-які інші варіанти даного аргумента – аналогічна до “уу” з тією відмінністю, що замість “0у” використовується “04у”. “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`.

Також при роботі з наступними властивостями використовуються ідеї поліритмічності як джерела ритму.

- “w01” – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає одне максимальне значення, менше – ще одне максимальне (але менше значення). Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD`;
- “w02” – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає одне максимальне значення, менше – ще одне максимальне (але менше значення). Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD2`;
- “w03” – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає одне максимальне значення, менше – ще одне максимальне (але менше значення). Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD3`;
- “w04” – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає одне максимальне значення, менше – ще одне максимальне (але менше значення). Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD4`;
- “w11” – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає два максимальних значення, менше – ще одне мінімальне. Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD`;

- [illegible]

- "w34" – більш складна ритмічна структура з іншим способом визначення значення властивості, розглядає рядок як текст з ритмічними групами по 4 склади кожна, де найбільше на ритмічність впливає одне мінімальне значення, менше – ще одне мінімальне. Як варіант обчислення тривалості складів використовується бібліотечна функція `syllableDurationsD4`;
- "x01" – аналогічно до "w01", але з більш комплексною залежністю для менш значимої тривалості та, можливо, менш передбачуваними результатами;
- "x02" – аналогічно до "w02", але з більш комплексною залежністю для менш значимої тривалості та, можливо, менш передбачуваними результатами;
- "x03" – аналогічно до "w03", але з більш комплексною залежністю для менш значимої тривалості та, можливо, менш передбачуваними результатами;
- "x04" – аналогічно до "w04", але з більш комплексною залежністю для менш значимої тривалості та, можливо, менш передбачуваними результатами;

Наступні значення аналогічні до відповідних "w" з більш комплексною залежністю (як щойно описані вище).

До них належать:

- "x11"
- "x12"
- "x13"
- "x14"
- "x21"
- "x22"
- "x23"
- "x24"
- "x31"
- "x32"
- "x33"

- "x34"

Якщо цей аргумент має наступний вигляд, то застосовуються поліритмічний аналіз рядка. Шукаються та перевіряються більш комплексні властивості тексту, з застосуванням складніших за структурою властивостей. Це дослідницький напрямок програм. Також є можливість задати власні налаштування, використовуючи режим "с", "С", "N" властивостей. Отже, наступні значення задають таке:

- "u01" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;
- "u02" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "u03" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "u04" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;
- "u11" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;
- "u12" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "u13" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "u14" -> Шукається поліритм з найбільш значимим 1 максимумом, потім менш значимими двома, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;

- [illegible]

- [illegible]

- "u71" -> Шукається поліритм з найбільш значимими 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;
- "u72" -> Шукається поліритм з найбільш значимими 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "u73" -> Шукається поліритм з найбільш значимими 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "u74" -> Шукається поліритм з найбільш значимими 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 5 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;
- "v01" та інші "v" – аналогічні до відповідних "u" ліній, з тією відмінністю, що використовують лише зростаючі варіанти функцій для визначення ритмічності. Це робить їх більш прямолінійними.
- "s01" -> Шукається поліритм з найбільш значимим 1 максимумумом, потім менш значимими 2 максимумумами, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;
- "s02" -> Шукається поліритм з найбільш значимим 1 максимумумом, потім менш значимими 2 максимумумами, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "s03" -> Шукається поліритм з найбільш значимим 1 максимумумом, потім менш значимими 2 максимумумами, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "s04" -> Шукається поліритм з найбільш значимим 1 максимумумом, потім менш значимими 2 максимумумами, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;
- "s11" -> Шукається поліритм з найбільш значимим 1 максимумумом, потім менш значимими 2 максимумумами, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;

- [illegible]



- [illegible]

- "s62" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "s63" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "s64" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 максимумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;
- "s71" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD;
- "s72" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD2;
- "s73" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD3;
- "s74" -> Шукається поліритм з найбільш значимим 2 максимумумами, потім менш значимим 1 мінімумом, і нарешті з ще менш значимим 1 мінімумом, при цьому розглядаються групи по 6 складів. Як функція для визначення тривалостей складів використовується syllableDurationsD4;
- "t01" та інші з "t" лінії – аналогічні до відповідних з "s" лінії, з тією відмінністю, що використовуються лише збільшуючі функції для властивостей. → Це робить ці властивості більш прямолінійними.
- "S" лінія відповідає властивостям "s" лінії, з тією відмінністю, що використовується "зважена" функція  
rhythmicityPolyWeightedF2;
- "T" лінія відповідає властивостям "t" лінії, з тією відмінністю, що використовується "зважена" функція  
rhythmicityPolyWeightedF20;

- “U” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF2`;
- “V” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF20`;
- “W” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF3`;
- “X” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF30`;
- “Y” лінія відповідає властивостям “s” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF3`;
- “Z” лінія відповідає властивостям “t” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedF30`; Наступні лінії властивостей намагаються збільшити значимість закінчення рядка і зменшити значення його початку.
- “I” лінія відповідає властивостям “W” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF2`;
- “J” лінія відповідає властивостям “X” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF20`;
- “K” лінія відповідає властивостям “Y” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF2`;
- “L” лінія відповідає властивостям “Z” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF20`;
- “O” лінія відповідає властивостям “U” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF3`;
- “P” лінія відповідає властивостям “V” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF30`;
- “Q” лінія відповідає властивостям “S” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF3`;

- “R” лінія відповідає властивостям “T” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedEF30`; Починаючи з версії 0.10.0.0 введені також наступні властивості:
- “o” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF2`;
- “p” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF20`;
- “q” лінія відповідає властивостям “s” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF2`;
- “r” лінія відповідає властивостям “t” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF20`;
- “k” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF3`;
- “l” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF30`;
- “m” лінія відповідає властивостям “s” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF3`;
- “n” лінія відповідає властивостям “t” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLF30`; Наступні лінії властивостей намагаються збільшити значимість закінчення рядка і зменшити значення його початку.
- “g” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF2`;
- “h” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF20`;
- “i” лінія відповідає властивостям “s” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF2`;
- “j” лінія відповідає властивостям “t” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF20`;

- “b” лінія відповідає властивостям “u” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF3`;
- “d” лінія відповідає властивостям “v” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF30`;
- “e” лінія відповідає властивостям “s” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF3`;
- “f” лінія відповідає властивостям “t” лінії, з тією відмінністю, що використовується “зважена” функція `rhythmicityPolyWeightedLEF30`;

## Режим власних налаштувань

Якщо тип починається з “c”, “A”, “B”, “C”, “D”, “E”, “F”, “M”, “N”, то програма намагається пропарсити цю властивість як закодовану конфігурацію поліритмічності. Режим для розробників та дослідників, можуть використовуватися ще складніші поліритмічні структури. Наприклад, властивість “c114+112=2” повертає як структуру поліритмічності P1 (Ch 1 1 4) (Rhythm 1 1 2) 2, що у свою чергу означає, що будуть шукатися 1 найбільш значимий максимум, також ще один менш значимий максимум у групах з 4 складів, як функція для визначення тривалостей складів буде використано `syllableDurationsD2`; “ctttff7+112111=7*3” повертає як структуру поліритмічності P2 (PolyCh [True,True,True,False,False] 7) (PolyRhythm [1,1,2,1,1]) 7 3, що у свою чергу означає, що будуть шукатися 1 найбільш значимий максимум, також ще один менш значимий максимум, також 2 ще менш значимі максимуми, також один ще менш значимий мінімум, а також ще один найменш значимий мінімум у групах з 7 складів, як функція для визначення тривалостей складів буде використано `syllableDurationsD3` тощо.

При використанні нових властивостей типів “A”, “B”, “C”, “D”, “E”, “F”, “M” та “N” будуть використані “зважені” функції, які намагаються врахувати також значимість розміщення частини рядка, зокрема внутрішньо будуть використані такі функції з модуля `Rhythmicity.PolyRhythm` як `rhythmicityPolyWeightedLEF2`, `rhythmicityPolyWeightedEF2`, `rhythmicityPolyWeightedF2`, `rhythmicityPolyWeightedLF2`, `rhythmicityPolyWeightedLEF3`, `rhythmicityPolyWeightedLF3`, `rhythmicityPolyWeightedEF3` та `rhythmicityPolyWeightedF3` відповідно.

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.

Для більшої наочності та зручності далі наводиться певна схематична часткова класифікація властивостей.

Властивості	Групи по N, N	Увага до	Лінійність	Чи є Infinity	Особливості	Використовується	Нова колонка 2
y0		0	0	0	Періоди унікальності	1-3	
0{M}y, 0{M}z	2   3	0	0	0	Перший аргумент	1-3	
y{M}, z{M}	2   3	0	0	0	Перший аргумент	1-3	
yy{M}, zz{M}	2   3	0	0	0	Перший аргумент	1-3	
w, x	4	0	0	1	wx-аргумент	1-3	
u, v,	5	0	0	1		1-3	
s, t,	6	0	0	1		1-3	
S, T,	6	1	0	1		1-3	
U, V,	5	1	0	1		1-3	
W, X,	5	1	0	1		1-3	
Y, Z,	6	1	0	1		1-3	
I, J,	5	-1	0	1		1-3	
K, L,	6	-1	0	1		1-3	
O, P,	5	-1	0	1		1-3	
Q, R,	6	-1	0	1		1-3	
o, p,	5	1	1	0		1-3	
q, r,	6	1	1	0		1-3	
k, l,	5	1	1	0		1-3	
m, n,	6	1	1	0		1-3	
g, h,	5	-1	1	0		1-3	
i, j,	6	-1	1	0		1-3	
b, d,	5	-1	1	0		1-3	
e, f,	6	-1	1	0		1-3	
c	Користувацька				Власна структура	1-3	
A	Користувацька	-1	1	0	Власна структура	1-3	
B	Користувацька	1	0	1	Власна структура	1-3	
C	Користувацька	1	0	1	Власна структура	1-3	
D	Користувацька	1	1	0	Власна структура	1-3	
E	Користувацька	-1	1	0	Власна структура	1-3	
F	Користувацька	1	1	0	Власна структура	1-3	
M	Користувацька	-1	0	1	Власна структура	1-3	
N	Користувацька	1	0	1	Власна структура	1-3	
H...					Тривалості	1-3	
G...					Схема обчислень	1	
a...					Склад-як-одне-ціле (SaaW)	1	

У таблиці у стовпчику «Увага до» значення 0 означає, що властивість однаково впливає на опрацювання даних, які стоять в усіх повних групах (якщо такі є) або в усьому рядку, якщо таких немає; значення 1 означає, що програма більш чутлива до початку рядка, значення -1 означає, що програма більш чутлива до останніх груп (якщо такі є), або просто до закінчення рядка (якщо груп немає). Аргументи 1 і 0 відповідають "так" і "ні" у наступних двох стовпчиках. Таким чином, наприклад, рядок o, p, 5 1 1 0 1-3 означає, що для властивостей з першими літерами в позначенні o або p має місце наступне: вони групують склади в групи по 5 складів, починаючи від початку рядка, усі склади, які не можуть бути згруповані (в кінці, певний "надлишок"), не рахуються, властивості більш чутливі до початку рядка, є "лінійними" (такою є функція, яка утворює ядро зростання чи спадання, відносно руху вздовж рядка вперед чи назад), і відповідно не може набувати значення Infinity для більш-менш "реальних" рядків; властивості можуть бути використані в усіх трьох програмах:

1. lineVariantsG3
2. propertiesTextG3
3. rewritePoemG3

Якщо в колонці "Використовується" є якесь число, тоді відповідна програма може використовувати цю властивість.

# 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 +а та -а) у вигляді число1_число2, де число1 та число2 – десяткові додатні дробили 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 підкреслити, скомбінувати чи знівелювати вплив ритмічності для двоскладових чи трискладових стоп.

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

Потрібно сказати, що цей аргумент може суттєво вплинути на результати і може змінити їх на майже протилежні. Розглянемо такий приклад.

propertiesTextG3 20000_1 Shchaslyve_derevo.txt s 1 0 +d 56.csv H02z   distributionTextG s 1 +W							
5							
2226	2226	51204	1.0000	23.0004	23.0004	0.08333195	4
1	Яке воно щасливе дерево						
0	1	204801	Infinity	Infinity	204801.0000	0.00000977	5
1	Ти можеш бачити цвіте воно						
225	2447	20144	10.8765	89.5289	8.2314	0.24028889	3
1	Закриєш очі плодоносить						
11251	20009	20009	1.7784	1.7784	1.0000	1.28016635	3
5	Радітимеш побачивши його						
881	1025	20081	1.1635	22.7934	19.5912	0.09779601	3
1	іквітним іводночас плідним						





9	25	720009	2.7778	80001.0000	28800.3600	0.00006944	4
1	свята Різдва радість огортає						
1	16	460804	16.0000	460804.0000	28800.2500	0.00006944	5
1	Для дерева затиш той теплий рух						
1	3674	35560	3674.4694	35559.5556	9.6775	0.20665984	5
1	Тасили додає йому той зріст						
36	36	51236	1.0000	1423.2222	1423.2222	0.00140428	3
1	ігнучкість імічне здоров'я						
0	45009	460809	Infinity	Infinity	10.2382	0.19534775	6
1	Цепори року йдерево просте Ами люди						
2226	51236	51236	23.0148	23.0148	1.0000	1.91671793	3
5	Багатші ірухливі іповсюди						
5016	204804	720004	40.8301	143.5415	3.5156	0.56496097	3
2	Ущасті ми активні						
1254	1254	8893	1.0000	7.0916	7.0916	0.24716936	2
1	айакнійдемо донього						
	поступово						
	людяно						

1	2	3	4	5			
22	5	1	0	3			
70.97%		16.13%	3.23%		0.00%	9.68%	
0.3905+-0.4959		0	31				
2	3	4	5	6	7		
1	14	10	5	1	0		
*****							
1	.	.	.	.			
7	3	1	.	3			
9	1	.	.	.			
4	1	.	.	.			
1	.	.	.	.			
.	.	.	.	.			
~~~~~							
1	0	0	0	0			
7	3	1	0	3			

9	1	0	0	0
4	1	0	0	0
1	0	0	0	0
0	0	0	0	0

propertiesTextG3 1_20000 Shchaslyve_derevo.txt s 1 0 +d 56.csv H02z | distributionTextG s 1 +W
5

80000	80000	80003	1.0000	1.0000	1.0000	0.99998469	4
1	Яке воно щасливе дерево						
0	20000	80003	Infinity	Infinity	4.0001	0.49998400	5
2	Ти можеш бачити цвіте воно						
180001	4500000	4500000	24.9999	24.9999	1.0000	1.92307669	3
5	Закриєш очі плодоносить						
20001	180001	180001	8.9998	8.9998	1.0000	1.79999594	3
5	Радітимеш побачивши його						
720000	4500000	4500001	6.2500	6.2500	1.0000	1.72413762	3
5	іквітним іводночас плідним						
720001	4500001	6480002	6.2500	9.0000	1.4400	1.24999987	4
4	Анавесні колищеквіту там нема						
20000	80003	80003	4.0001	4.0001	1.0000	1.60001024	3
5	Чийе щасливим дерево						
80000	320004	320004	4.0000	4.0000	1.0000	1.60000400	4
5	Напевне Унадії втій порі						
80000	320000	320000	4.0000	4.0000	1.0000	1.60000025	3
5	щосонце гріє Івранці						
0	20001	320001	55558.3333	888891.6667	15.9993	0.12500572	4
1	сяють теплі йпрохолодні роси						
0	0	80000	4.0000	2000009.0000	500002.2500	0.00000400	4
1	Приходиш ти донього навесні						
720000	720000	720003	1.0000	1.0000	1.0000	0.99999822	3
1	ібачиш дерево щасливим						
20003	80001	80001	3.9995	3.9995	1.0000	1.59996304	3
5	Прийдеш улітку восени						
80000	320000	320064	4.0000	4.0008	1.0002	1.59974371	4

поступово
людяно

1	2	3	4	5		
10	7	0	1	13		
32.26%		22.58%	0.00%	3.23%	41.94%	
1.0681+0.6072		0	31			
2	3	4	5	6	7	
1	14	10	5	1	0	

1		
5	1	.	.	8		
3	2	.	1	4		
1	3	.	.	1		
.	1	.	.	.		
.		
~~~~~						
1	0	0	0	0		
5	1	0	0	8		
3	2	0	1	4		
1	3	0	0	1		
0	1	0	0	0		
0	0	0	0	0		

У першому випадку перший аргумент суттєво підсилює аналіз за 2-складовим компонентом, у другому — за 3-складовим. Якщо Ви порівняєте виділені червоним кольором статистики (середнє значення та стандартне вибіркве квадратичне відхилення) у обох випадках, то побачите, що на рівні тексту підсилення 2-складових компонентів дає значно менший за 1 результат, що означає, що дані сильно зсунуті вліво, до менших значень, а в другому випадку дає навіть більший за 1 результат, що говорить, що дані більше зміщені вправо, до більших значень. Але це не означає, що текст ближчий до віршів з трискладовим віршовим розміром. Багато рядків є класичними рядками з двоскладовим розміром (наприклад, ямбом). зокрема: "свѣта Рїздвяна радїсть огортає" (виділена синім кольором), але з точки зору тривалостей складів вона значно більше підходить для трискладової стопи. Це також є прикладом різниці у силабічному та тонічному віршуваннях.

# 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 визначенням порядком деяких слів тощо), що дозволяє фактично розширити можливості програми. Ці обмеження кодуються як аргументи командного рядка між двома спеціальними позначеннями +а та -а. 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 далі.

- Обмеження E – Без вводу додаткових цифрових символів – Відповідає відсутності додаткового обмеження, в ході роботи програми не впливає на кінцевий результат.
- Обмеження Q – 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.
- Обмеження T – 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.

- Обмеження F – 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.

- Обмеження A – 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 ПРАВИШЕ від елемента з індексом, рівним першій цифрі.
- Обмеження B – 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.
- Обмеження P (fixed Point) (введене, починаючи з версії 0.18.0.0) – 1 чи ще кілька попарно нерівних цифр (усі між собою нерівні) в межах від 0 до кількості слів чи їх сполучень мінус 1 – кожна з них (крім 0) означає порядковий номер слова, яке при перестановках буде залишатися на своєму місці (відлік розпочинається з 1 зліва направо), цифра 0 вживається лише для випадку неповних множин перестановок і відповідає ситуації, коли останнє можливе слово, 10-е, має залишитися на своєму місці. У випадку повної множини перестановок: щоб залишити останнє слово на своєму місці, використовуйте, будь ласка, +b (+bl) параметр командного рядка при виклику програм.

# 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 [1].

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

## Подяки

Автор хоче висловити подяку авторам роботи *Provably Correct, Asymptotically Efficient, Higher-Order Reverse-Mode Automatic Differentiation* Faustyna Krawiec, Simon Peyton-Jones, Neel Krishnaswami, Tom Ellis, Richard A. Eisenberg та Andrew Fitzgibbon за ідею щодо оптимізації, а також Mikołaj Konarski за те, що звернув увагу автора на цю статтю. Також автор висловлює подяку своїм друзям, яких він хотів порадувати цим дослідженням. Також автор висловлює подяку кожній людині, яка щиро сприяла у тій чи іншій формі проведенню дослідження.



# Бібліографія

- [1] Glasgow haskell compiler user's guide. 7.5. using smp parallelism. [Електронний ресурс]. Режим доступу: [https://downloads.haskell.org/ghc/latest/docs/users_guide.pdf](https://downloads.haskell.org/ghc/latest/docs/users_guide.pdf). Перевірено 10 листопада 2020 р.
- [2] Shlomo Alexander and Raymond Orbach. Density of states on fractals:«fractons». *Journal de Physique Lettres*, 43(17):625–631, 1982.
- [3] Jan Andres, Jiří Fišer, Grzegorz Gabor, and Krzysztof Leśniak. Multivalued fractals. *Chaos, Solitons & Fractals*, 24(3):665–700, 2005.
- [4] Jan Andres and Miposlav Rypka. Multivalued fractals and hyperfractals. *International Journal of Bifurcation and Chaos*, 22(01):1250009, 2012.
- [5] Corine Astésano. *Rythme et accentuation en français: invariance et variabilité stylistique*. Editions L'Harmattan, 2001.
- [6] David Avnir, Ofer Biham, Daniel Lidar, and Ofer Malcai. Is the geometry of nature fractal? *Science*, 279(5347):39–40, 1998.
- [7] Michael F Barnsley, John E Hutchinson, and Örjan Stenflo. V-variable fractals: fractals with partial self similarity. *Advances in Mathematics*, 218(6):2051–2088, 2008.
- [8] Christopher J Bishop and Yuval Peres. *Fractals in probability and analysis*, volume 162. Cambridge University Press, 2017.
- [9] Marilyn G Boltz. Tempo discrimination of musical patterns: Effects due to pitch and rhythmic structure. *Perception & Psychophysics*, 60(8):1357–1373, 1998.
- [10] Nia Cason, Corine Astésano, and Daniele Schön. Bridging music and speech rhythm: Rhythmic priming and audio–motor training affect speech perception. *Acta psychologica*, 155:43–50, 2015.
- [11] Andrés E Coca, Gerard O Tost, and Zhao Liang. Controlling chaotic melodies. *Proc. Encuentro Nacional de Investigación en Posgrados (ENIP)*, 2009.
- [12] G.W. Cooper, G. Cooper, L.B.A. MEYER, and L.B. Meyer. *The Rhythmic Structure of Music*. Phoenix books. University of Chicago Press, 1963.

- [13] Joseph R Daniele and Aniruddh D Patel. An empirical study of historical patterns in musical rhythm: Analysis of german & italian classical music using the npvi equation. *Music Perception: An Interdisciplinary Journal*, 31(1):10–18, 2013.
- [14] Norberto Degara, Matthew EP Davies, Antonio Pena, and Mark D Plumbley. Onset event decoding exploiting the rhythmic structure of polyphonic music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1228–1239, 2011.
- [15] Norberto Degara, Antonio Pena, Matthew EP Davies, and Mark D Plumbley. Note onset detection using rhythmic structure. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5526–5529. IEEE, 2010.
- [16] Carolyn Drake and Caroline Palmer. Accent structures in music performance. *Music perception*, 10(3):343–378, 1993.
- [17] Antonio Galves, Jesus Garcia, Denise Duarte, and Charlotte Galves. Sonority as a basis for rhythmic class discrimination. In *Speech Prosody 2002, International Conference*, 2002.
- [18] EJ Garba. Music programming–rule-based randomization of melodic patterns. 2008.
- [19] Alex Goldsmith. Synthesising music: exploiting self-similarity using modular forms.
- [20] H. S. Gross. "prosody." *Encyclopedia Britannica*. <https://www.britannica.com/art/prosody>.
- [21] Erin E Hannon. Perceiving speech rhythm in music: Listeners classify instrumental songs according to language of origin. *Cognition*, 111(3):403–409, 2009.
- [22] Andre Holzapfel and Yannis Stylianou. Rhythmic similarity of music based on dynamic periodicity warping. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2217–2220. IEEE, 2008.
- [23] David Huron and Matthew Royal. What is melodic accent? converging evidence from musical practice. *Music Perception*, 13(4):489–516, 1996.
- [24] John E Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981.
- [25] John R Iversen, Aniruddh D Patel, and Kengo Ohgushi. Perception of rhythmic grouping depends on auditory experience. *The Journal of the Acoustical Society of America*, 124(4):2263–2271, 2008.
- [26] Vijay Iyer, Jeff Bilmes, Matt Wright, and David Wessel. A novel representation for rhythmic structure. In *Proceedings of the 23rd International Computer Music Conference*, pages 97–100. Citeseer, 1997.
- [27] Jun Kigami. *Analysis on fractals*. Number 143. Cambridge University Press, 2001.
- [28] Cyrille Magne, Mitsuko Aramaki, Corine Astesano, Reyna Leigh Gordon, Sølvi Ystad, Snorre Farner, Richard Kronland-Martinet, and Mireille Besson. Comparison of rhythmic processing in language and music: An interdisciplinary approach. *Journal of Music and Meaning*, 3, 2005.

- [29] Benoit B Mandelbrot and Benoit B Mandelbrot. *The fractal geometry of nature*, volume 1. WH freeman New York, 1982.
- [30] Kevin Merges. *Fractals and art*. 2005.
- [31] Ihor Nabytovych. ФРАКТАЛИ ТА ФРАКТАЛЬНІ СТРУКТУРИ У ХУДОЖНЬОМУ ТЕКСТІ (на прикладі прози Л. Керрола, КС Льюїса та ХЛ Борхеса). *ВІСНИК ЛЬВІВСЬКОГО УНІВЕРСИТЕТУ. Серія іноземні мови*, (18).
- [32] Katie Overy. Dyslexia and music: From timing deficits to musical intervention. *Annals of the New York academy of sciences*, 999(1):497–505, 2003.
- [33] Narayan Partap and Renu Chugh. Fixed point iterative techniques—an application to fractals. *International Journal of Research in Mathematics & Computation*, 4(1):1–7, 2016.
- [34] Aniruddh D Patel. Rhythm in language and music: parallels and differences. *Annals of the New York Academy of Sciences*, 999(1):140–143, 2003.
- [35] Aniruddh D Patel and Joseph R Daniele. An empirical comparison of rhythm in language and music. *Cognition*, 87(1):B35–B45, 2003.
- [36] Aniruddh D Patel and Joseph R Daniele. Stress-timed vs. syllable-timed music? a comment on huron and ollen (2003). *Music Perception*, 21(2):273–276, 2003.
- [37] Rohit Sunkam Ramanujam and Bill Lin. Randomized partially-minimal routing on three-dimensional mesh networks. *IEEE Computer Architecture Letters*, 7(2):37–40, 2008.
- [38] Franck Ramus. *Acoustic correlates of linguistic rhythm: Perspectives*. 2002.
- [39] Dietmar Saupe. Algorithms for random fractals. In *The science of fractal images*, pages 71–136. Springer, 1988.
- [40] Xi Shao, Changsheng Xu, and Mohan S Kankanhalli. Unsupervised classification of music genre using hidden markov model. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 3, pages 2023–2026. IEEE, 2004.
- [41] Joyce L Shields, Astrid McHugh, and James G Martin. Reaction time to phoneme targets as a function of rhythmic cues in continuous speech. *Journal of Experimental Psychology*, 102(2):250, 1974.
- [42] Robert S Strichartz. Fractals in the large. *Canadian Journal of Mathematics*, 50(3):638–657, 1998.
- [43] Neil P McAngus Todd. Segmentation and stress in the rhythmic structure of music and speech: A wavelet model. *The Journal of the Acoustical Society of America*, 93(4):2363–2363, 1993.

- [44] Wiebke Trost and Patrik Vuilleumier. Rhythmic entrainment as a mechanism for emotion induction by music: a neurophysiological perspective. *The emotional power of music: Multidisciplinary perspectives on musical arousal, expression, and social control*, pages 213–225, 2013.
- [45] Giuseppe Vitiello. The brain is like an orchestra. better yet, it is like a jazz combo, which doesn't need a conductor. *Chaos*, 11(1):2017, 2017.
- [46] Peter Vuust, Leif Ostergaard, Karen Johanne Pallesen, Christopher Bailey, and Andreas Roepstorff. Predictive coding of music–brain responses to rhythmic incongruity. *cortex*, 45(1):80–92, 2009.
- [47] Huang Wan-Li. The extremity laws of hydro-thermodynamics. *Applied Mathematics and Mechanics*, 4(4):501–510, 1983.
- [48] Kerri Welch. *A fractal topology of time: Implications for consciousness and cosmology*. California Institute of Integral Studies, 2010.
- [49] Wolfgang Wildgen. Chaos, fractals and dissipative structures in language. or the end of linguistic structuralism. *Gabriel Altmann und Walter A. Koch (Hg.), Systems. New Paradigms for the Human Sciences, de Gruyter, Berlin*, pages 596–620, 1998.
- [50] Marcel Zentner and Tuomas Eerola. Rhythmic engagement with music in infancy. *Proceedings of the National Academy of Sciences*, 107(13):5768–5773, 2010.
- [51] Oleksandr Zhabenko. dobutokO-poetry. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/dobutokO-poetry-0.8.1.0>. Перевірено 09 листопада 2020 р.
- [52] Oleksandr Zhabenko. phonetic-languages-rhythmicity. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/phonetic-languages-rhythmicity>. Перевірено 24 серпня 2020 р.
- [53] Oleksandr Zhabenko. phonetic-languages-simplified-generalized-examples-array. [Електронний ресурс]. Режим доступу: <https://hackage.haskell.org/package/phonetic-languages-simplified-generalized-examples-array>. Перевірено 23 жовтня 2021 р.
- [54] П Івончак. Український силабо-тонічний вірш 50-х років XIX століття. *Науковий вісник Чернівецького національного університету. Слов'янська філологія*, (585-586):80–84, 2012.
- [55] Смаглій Г. А. Теорія музики : Підруч. для навч. закл. освіти, культури і мистецтв., 2013.
- [56] Під ред. Анатолія Волкова. Адекватний переклад // Лексикон загального та порівняльного літературознавства., 2001.
- [57] Під ред. Анатолія Волкова. Айрен // Лексикон загального та порівняльного літературознавства., 2001.
- [58] Під ред. Анатолія Волкова. Александрійський вірш // Лексикон загального та порівняльного літературознавства., 2001.
- [59] Під ред. Анатолія Волкова. Алкеєва строфа // Лексикон загального та порівняльного літературознавства., 2001.

- [60] Під ред. Анатолія Волкова. Алкманова або Архілохова строфа // Лексикон загального та порівняльного літературознавства., 2001.
- [61] Під ред. Анатолія Волкова. Античні розміри // Лексикон загального та порівняльного літературознавства., 2001.
- [62] Під ред. Анатолія Волкова. Античні строфи // Лексикон загального та порівняльного літературознавства., 2001.
- [63] Під ред. Анатолія Волкова. Аруз, або Аруд // Лексикон загального та порівняльного літературознавства., 2001.
- [64] Під ред. Анатолія Волкова. Асклепіадова строфа // Лексикон загального та порівняльного літературознавства., 2001.
- [65] Під ред. Анатолія Волкова. Баяті // Лексикон загального та порівняльного літературознавства., 2001.
- [66] Під ред. Анатолія Волкова. Буриме // Лексикон загального та порівняльного літературознавства., 2001.
- [67] Під ред. Анатолія Волкова. Варіативність // Лексикон загального та порівняльного літературознавства., 2001.
- [68] Під ред. Анатолія Волкова. Верлібр // Лексикон загального та порівняльного літературознавства., 2001.
- [69] МИКОЛА ВАСИЛЬОВИЧ Гуцуляк. *Українське віршування 30-80-х рр. XVII ст.* PhD thesis, «Теорія літератури». Чернівці: ЧНУ ім. Юрія Федьковича, 2017, 2017.
- [70] Наталія Костенко. ЕЛЕМЕНТИ ТОНІЧНОГО І СИЛАБІЧНОГО ВІРШУВАННЯ В ДУМОВОМУ ВІРШІ ТГ ШЕВЧЕНКА. *НАШ УКРАЇНСЬКИЙ ДІМ*, page 38.
- [71] АГ Кошовий and ГІ Кошовий. Одновимірні самоподібні фрактали та їх використання у моделюванні. 2011.
- [72] Оксана Валентинівна Кудряшова. *Functional poetics*. 2016.
- [73] О. В. Лазер-Паньків та ін. Л. Л. Звонська, Н. В. Корольова. Ямбічна строфа // *Енциклопедичний словник класичних мов*, 2017.
- [74] ОВ Любімова. Відтворення античних розмірів в українській поезії 80-х–90-х років XIX століття. *Науковий вісник Чернівецького університету. Романо-слов'янський дискурс*, (565):190–193, 2011.
- [75] К Паладян. Початки силабо-тонічної версифікації в румунській літературі. *Питання літературознавства*, (81):164–172, 2010.
- [76] ВД Шарко. Актуальні проблеми природничо-математичної освіти в середній і вищій школі.
- [77] Тарас Шевченко. *Садок вишневий коло хати*. Strelbytskyy Multimedia Publishing, 2018.